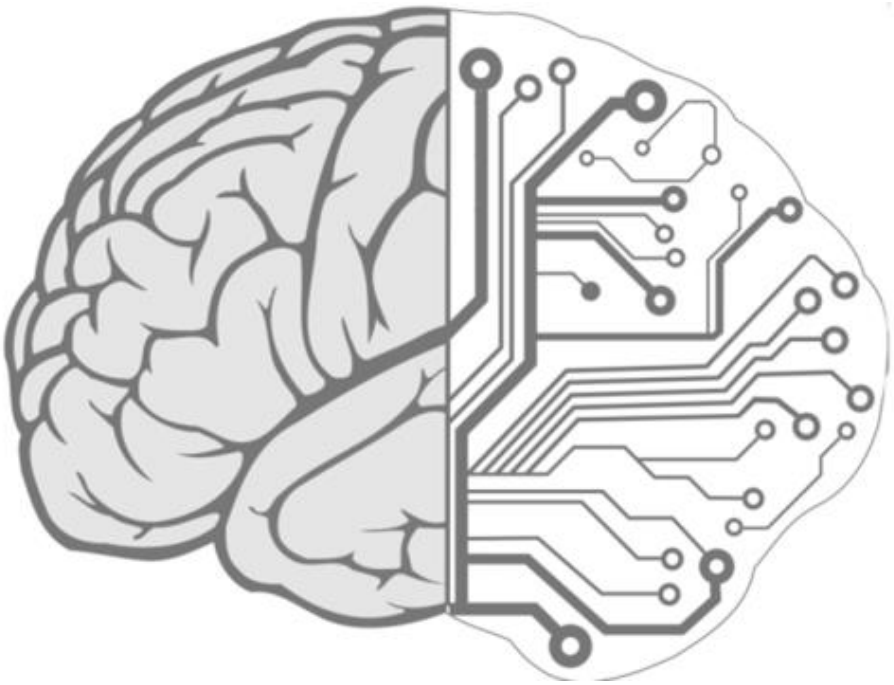
Journal of Intelligent Systems



IlmuKomputer.Com
ikatlah ilmu dengan menuliskannya

Copyright © 2015 IlmuKomputer.Com
All rights reserved. Printed in Indonesia

Copyright © 2015 IlmuKomputer.Com

Editorial Board

Editor-in-Chief: Romi Satria Wahono, M.Eng, Ph.D

Editor:

Mansyur, S.Kom

Mulyana, S.Kom

Reviewer:

Prof. Budi Santosa, Ph.D (Institut Teknologi Sepuluh Nopember)

Dr. Eng. Anto Satriyo Nugroho (Badan Pengkajian dan Penerapan Teknologi)

Fahmi Arief, Ph.D (Universiti Teknikal Malaysia)

Purwanto, Ph.D (Universitas Dian Nuswantoro)

Prof. Dr. Anton Satria Prabuwono (King Abdulaziz University)

Dr. Eng. Son Kuswadi (Politeknik Elektronika Negeri Surabaya)

Dr. Eng. Arief Budi Witarto (Lembaga Ilmu Pengetahuan Indonesia)

Iko Pramudiono, Ph.D (Mitsui Indonesia)

Romi Satria Wahono, Ph.D (Universitas Dian Nuswantoro)

Copyright © 2015 IlmuKomputer.Com

Contents

REGULAR PAPERS

Copyright © 2015 IlmuKomputer.Com

Copyright © 2015 IlmuKomputer.Com

Integrasi Kromosom Buatan Dinamis untuk Memecahkan Masalah Konvergensi Prematur pada Algoritma Genetika untuk Traveling Salesman Problem

Muhammad Rikzam Kamal, Romi Satria Wahono dan Abdul Syukur
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
rikzamrx@gmail.com, romi@brainmatics.com, abah.syukur@yahoo.com

Abstract: Algoritma genetika (Genetic Algorithm (GA)) adalah metode adaptif yang digunakan untuk memecahkan masalah pencarian dan optimasi. *Travelling Salesman Problem* (TSP) merupakan salah satu persoalan optimasi yang dipecahkan dengan GA, di mana rute terpendek merupakan solusi yang paling optimal. GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks dalam hal nonlinear, tetapi GA mempunyai masalah yaitu konvergensi prematur. Konvergensi prematur merupakan suatu kondisi yang terjadi ketika populasi algoritma genetika mencapai keadaan suboptimal di mana operator genetika tidak dapat lagi menghasilkan keturunan dengan kinerja yang lebih baik dari *parents*. Untuk mengatasi masalah konvergensi prematur, maka pada penelitian ini diusulkan dynamic artificial chromosomes yang diintegrasikan ke dalam genetic algorithm yang disebut GA-DAC. Dynamic Artificial Chromosomes (DAC) digunakan untuk mengkontrol keragaman populasi dan juga seleksi kromosom terbaik untuk memilih individu atau kromosom terbaik. Beberapa eksperimen dilakukan dengan GA-DAC, dimana *threshold* terbaik adalah 0,5, kemudian juga mendapatkan hasil perbaikan pada jarak terpendek yang dibandingkan dengan GA standar. Hasil pengujian untuk dataset KroA100 sebesar 12,60%, KroA150 sebesar 13,92% dan KroA200 sebesar 12,92%. Untuk keragaman populasi mendapatkan hasil pada KroA100 sebesar 24,97%, KroA150 sebesar 50,84% dan KroA200 sebesar 49,08%. Maka dapat disimpulkan bahwa GA-DAC bisa mendapatkan hasil lebih baik dibandingkan dengan GA standar, sehingga membuat GA dapat keluar dari konvergensi prematur.

Keywords: algoritma genetika, konvergensi prematur, dynamic artificial chromosomes, genetic algorithm dynamic artificial chromosomes, seleksi kromosom terbaik, *travelling salesman problem*.

1 PENDAHULUAN

Algoritma genetika (Genetic Algorithm (GA)) adalah bagian dari komputasi evolusioner yang berkembang pesat dalam bidang kecerdasan buatan (Siva Sathya & Radhika, 2013). GA adalah metode adaptif yang digunakan untuk memecahkan masalah pencarian dan optimasi. GA didasarkan pada proses genetik organisme biologis. Dengan meniru prinsip evolusi alam, yaitu "s*urvival of the fittest*", GA mampu mengembangkan solusi untuk masalah dunia nyata (De Giovanni & Pezzella, 2010). Sebelum GA dapat diterapkan, representasi atau pengkodean dari masalah harus dibuat terlebih dahulu. Inti dari GA adalah untuk mengkodekan satu set parameter (dikenal sebagai gen) dan gabungan dari gen-gen yang membentuk nilai tertentu dan menyatakan solusi yang mungkin dari suatu permasalahan yang disebut sebagai kromosom (Y.-H. Chang, 2010). Fungsi *fitness* juga diperlukan untuk memberikan nilai yang diperoleh dari setiap solusi. Setiap individu tergantung pada kromosom dan dievaluasi oleh fungsi *fitness* (Pavez-Lazo & Soto-Cartes, 2011). Selama proses berjalan, orang tua harus dipilih untuk proses reproduksi dan digabungkan untuk menghasilkan keturunan. Orang tua secara acak dipilih dari populasi menggunakan skema yang menguntungkan individu. Setelah memilih orang tua, kemudian kromosom digabungkan, menggunakan mekanisme *crossover* dan mutasi. Solusi akan diperoleh ketika orang tua menghasilkan keturunan yang lebih baik. Proses iterasi ini terus berjalan sampai kriteria yang ditentukan telah tercapai.

GA sebagai metode pencarian dan optimisasi masalah sering digunakan dalam berbagai macam kasus seperti *job-shop scheduling problem* (De Giovanni & Pezzella, 2010), *timetabling* (Yang & Jat, 2011), *unit commitment problem* (Pavez-Lazo & Soto-Cartes, 2011), dan selain itu juga digunakan untuk menyelesaikan *travelling salesman problem* (TSP) (Liu & Zeng, 2009)(P.-C. Chang, Huang, & Ting, 2010). TSP merupakan salah satu masalah optimasi kombinatorial mendasar yang memiliki banyak aplikasi dalam penelitian operasional (Zhang, Tong, Xu, & Lin, 2015). Selain itu TSP juga termasuk dalam kategori masalah klasik, yaitu untuk menemukan rute terpendek melalui serangkaian poin dan kembali ke awal (Çavdar & Sokol, 2014).

GA juga dikombinasikan dengan berbagai metode lain untuk mengatasi masalah konvergensi prematur. Seperti Triangular Crossover (TC) (Elfeky, Sarker, & Essam, 2008), Unimodal Distribution Crossover (UNDX) (Ono, Kita, & Kobayashi, 2003), dan deterministic annular crossover(Pavez-Lazo & Soto-Cartes, 2011). Deterministic annular crossover menggunakan annular selection untuk menyeleksi individu atau orang tua dalam populasi yang akan mengalami proses deterministic crossover, dimana individu yang dipilih adalah individu dengan nilai *fitness* tertinggi yang dipasangkan dengan individu dengan nilai *fitness* terendah. UNDX menggunakan beberapa orang tua (*parents)* untuk menciptakan solusi keturunan (*offspring*) disekitar pusat massa dari orang tua, sementara probabilitas dengan nilai kecil ditugaskan untuk solusi terjauh dari pusat massa. Meskipun telah menunjukkan kinerja yang sangat baik untuk masalah yang sangat epistasis (ketika efek dari satu gen tergantung pada kehadiran satu atau lebih pengubah gen) (Ono et al., 2003). Tetapi UNDX tidak dapat menghasilkan keturunan dalam beberapa kasus seperti ketika ukuran populasi yang relatif terlalu kecil. UNDX juga memiliki kesulitan dalam menemukan solusi optimal pada ruang pencarian terdekat. TC menggunakan tiga orang tua untuk *constrained problems* (masalah yang dibatasi), satu orang tua tidak layak dan dua orang tua harus layak. Hal ini digunakan agar dapat menghasilkan satu dari tiga keturunan.

Copyright @ 2015 IlmuKomputer.Com

Kemudian dari setiap keturunan yang dihasilkan sebagai kombinasi linear dari tiga orang tua.

GA banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah-masalah selain optimasi. Algoritma genetika terinspirasi oleh proses evolusi, yang diamati dari alam (Chen & Chien, 2011). Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (S.N Sivanandam, 2008). GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks dalam hal nonlinear, tetapi GA juga mempunyai masalah yaitu konvergensi prematur (P.-C. Chang et al., 2010) (Pandey, Chaudhary, & Mehrotra, 2014). Konvergensi prematur terjadi ketika populasi algoritma genetika mencapai keadaan suboptimal dimana operator genetika tidak dapat lagi menghasilkan keturunan dengan kinerja yang lebih baik dari orang tua (P.-C. Chang et al., 2010).

Beberapa peneliti telah mencoba melakukan uji coba menggunakan beberapa algoritma untuk menyelesaikan masalah konvergensi prematur di dalam GA, diantaranya yaitu dengan Parent Centric Crossover (PCX) (Elsayed, Sarker, & Essam, 2014), deterministic annular crossover (Pavez-Lazo & Soto-Cartes, 2011), dan Multi Parents Crossover (MPC) (Elsayed et al., 2014)(Elfeky et al., 2008). PCX memungkinkan menciptakan solusi terdekat setiap orang tua, bukan didekat pusat orang tua. Setiap keturunan satu orang tua dipilih dan dihitung perbedaan vektor antara orang tua dan orang tua yang terpilih. PCX menerapkan pendekatan adaptif diri dimana vektor solusi baru terus bergerak menuju optimum. Ketika PCX diterapkan dengan GA, dibutuhkan waktu yang lebih lama dibandingkan dengan operator *crossover* yang lain, dan menemukan kesulitan dalam memecahkan masalah multimodal. Deterministic annular crossover digunakan untuk memperkaya hasil keturunan (*offspring)* dari proses *crossover*, dengan operator seleksi deterministik. Keragaman yang lebih besar antara individu-individu dari populasi dapat diperoleh melalui informasi genetik dari individu terburuk dengan probabilitas yang sama. MPC menggunakan tiga orang tua dalam proses *crossover* untuk menghindari keturunan (*offspring*) yang kurang beragam dari orang tuanya (Elsayed et al., 2014).

Pada penelitian ini, keragaman didalam populasi dikontrol dengan keragaman operator agar lebih beragam dengan cara meningkatkan keragaman populasi tersebut ketika nilai keragramannya kurang dari *threshold* atau kurang beragam (P.-C. Chang et al., 2010). Keseimbangan yang tepat antara eksplorasi dan eksploitasi pencarian dapat dipertahankan dengan mengendalikan tingkat keragaman populasi. Mekanisme kontrol dapat dibangun ke dalam GA menggunakan Dynamic Artificial Chromosomes (DAC) yang dimasukkan ke dalam sistem sampai ukuran keragaman mencapai tingkat tertentu kemudian berhenti. Selain itu juga akan digunakan operator untuk memilih individu atau kromosom terbaik yang akan dipasangkan dalam proses *crossover* sehingga proses eksplorasi dan eksploitasi dalam mutasi juga akan lebih maksimal. Dengan menerapkan DAC pada GA diharapkan dapat meningkatkan tingkat keragaman rata-rata sehingga proses dapat keluar dari konvergensi prematur dan proses iterasi dapat lebih maksimal.

2 PENELITIAN TERKAIT

GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks dalam hal nonlinear, tetapi GA juga mempunyai masalah yaitu konvergensi prematur (P.-C. Chang et al., 2010) (Pandey et al., 2014). Konvergensi prematur terjadi ketika populasi algoritma genetika mencapai keadaan suboptimal dimana operator genetika tidak dapat lagi menghasilkan keturunan dengan kinerja yang lebih baik dari orang tua (P.-C. Chang et al., 2010).

Beberapa peneliti telah melakukan penelitian untuk memecahkan masalah konvergensi prematur pada GA, diantaranya yaitu GA dengan Multi-Parent Crossover yang disebut GAMPC, serta diusulkan juga *diversity operator* untuk lebih lanjut membuat variasi pada pembangkitan *offspring* (keturunan) (Elsayed et al., 2014) agar individu didalam populasi menjadi lebih beragam. Pada penelitian lain juga diusulkan Deterministic Annular Crossover Genetic Algorithm yang disebut DACGA (Pavez-Lazo & Soto-Cartes, 2011), dengan menggunakan dua buah metode, yaitu seleksi *deterministic* dan annular crossover. Seleksi *deterministic* digunakan untuk mencari nilai *fitness* (kecocokan) individu dengan *fitness* yang lebih tinggi, sedangkan annular crossover digunakan untuk melakukan proses pertukaran informasi genetik antara dua individu dengan operator *crossover* yang direpresentasikan dalam bentuk cincin. Chang *et al.* mengusulkan dynamic diversity control di dalam GA atau yang disebut sebagai DDC-GA untuk *mining unsearched solution space* di TSP (P.-C. Chang et al., 2010).

Pada penelitian ini digunakan pengkontrol keragaman pada populasi dengan menggunakan dynamic diversity control (P.-C. Chang et al., 2010), yang bekerja ketika tingkat keragaman pada sebuah populasi turun pada batas tertentu atau dibawah *threshold* yang sudah ditentukan. Sedangkan untuk meningkatkan keragaman populasi menggunakan Dynamic Artificial Chromosome (DAC) dan juga menggunakan seleksi kromosom terbaik untuk memilih kromosom yang terbaik yang akan diproses pada *crossover* sehingga ini bisa membuat GA keluar dari konvergensi prematur.

3 METODE YANG DIUSULKAN

Pada penelitian ini diusulkan metode Dynamic Artificial Chromosome yang diintegrasikan kedalam Genetic Algorithm yang disebut GA-DAC dan juga seleksi kromosom terbaik seperti pada Gambar 1. Pada bagian kolom GA adalah struktur proses seperti pada GA standar pada umumnya, sedangkan pada bagian kolom GA-DAC adalah ketika pada proses evaluasi *fitness* dan evaluasi keragaman populasi (*population diversity)* diukur dengan menggunakan rumus *linear scale measure* (P.-C. Chang et al., 2010):

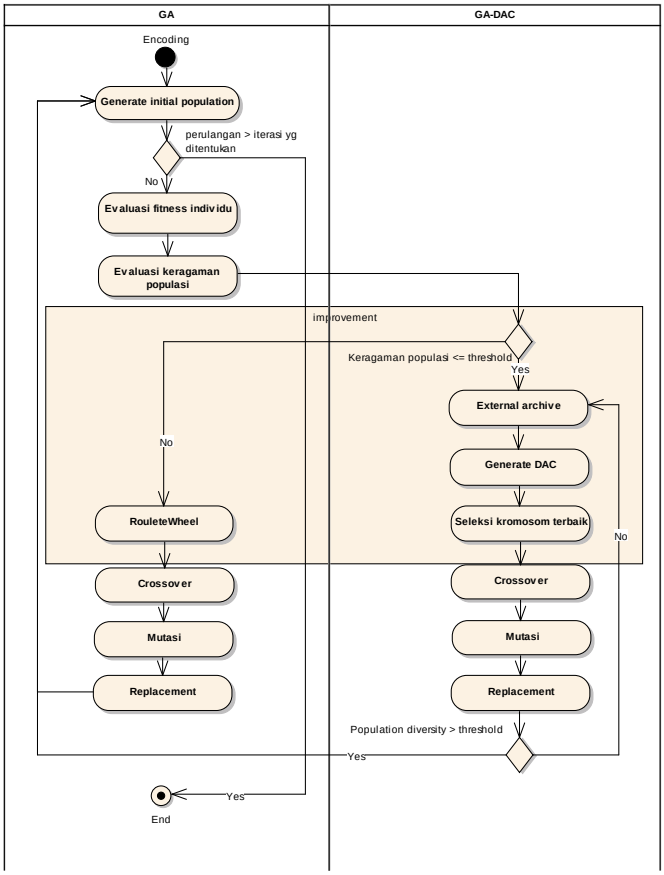
$$PD = \frac{\bar{d} - d_{min}}{d_{max} - d_{min}}$$

PD	: keragaman individu atau kromosom
$\bar{d}$	: rata-rata keragaman
d_{max}	: nilai maksimal dari keragaman
d_{min}	: nilai minimal keragaman

Jika ternyata nilai keragamannya turun ke bawah, kurang dari atau sama dengan nilai dari *threshold*, maka kromosom buatan dinamis akan bekerja. Cara kerjanya dengan mengambil kromosom baru dari *external archive* dengan nilai *fitness* dan keragaman yang lebih baik. Fungsinya untuk menggantikan kromosom dengan nilai *fitness* paling terendah di dalam populasi, yang diharapkan dapat membuat populasi dengan nilai *fitness* dan keragaman terbaik yang akan diproses pada

Copyright @ 2015 IlmuKomputer.Com

tahap selanjutnya. Kemudian pada proses ini juga ditambahkan seleksi kromosom terbaik untuk memilih kromosom atau orang tua dengan nilai *fitness* paling terbaik yang akan mengalami proses *crossover*. Langkah ini akan terus dilakukan sampai dengan nilai keragaman pada populasi lebih dari nilai *threshold*. Sehingga diharapkan dengan metode ini dapat mengatasi masalah pada GA yaitu optimum lokal atau yang disebut juga sebagai konvergensi prematur.



Gambar 1. Metode yang Diusulkan GA-DAC

4 HASIL PENELITIAN

Pengujian hasi penelitian dilakukan menggunakan komputer dengan spesifikasi CPU Pentium (R) Dual-Core CPU 2.70 GHz, RAM 2GB, dan sistem operasi Microsoft Windows 7 Professional 64-bit. Aplikasi yang digunakan adalah NetBeans IDE 8.0.2. Data penelitian ini menggunakan TSP KroA100, KroA150, dan KroA200 (Wang, 2014) yang diperoleh dari situs http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp.

Program GA dibuat sesuai dengan proses yang ada di dalam algoritma GA tersebut. GA-DAC dibuat berdasarkan program GA yang kemudian dikembangkan berdasarkan metode yang diusulkan seperti pada Gambar 1, yaitu untuk penentuan nilai keragaman populasi dan pengambilan kromosom baru (external archive) serta *generate* DAC.

Dalam penelitian ini GA-DAC menggunakan *threshold* dengan nilai yang terbaik sesuai dengan yang dilakukan Chang *et al*. (P.-C. Chang et al., 2010) yaitu 0,5, 0,6, dan 0,7.

Hasil dari uji pencarian rute terpendek GA dan juga GA-DAC ditunjukan pada Tabel 1, 2, dan 3. Pada GA-DAC, *threshold* diatur ke nilai nilai 0,5 dan menghasilkan nilai perbaikan rute terpendek terbaik untuk KroA100 sebesar 12,60%, KroA150 sebesar 13,92%, dan KroA200 sebesar 12,92%.

Tabel 1. Hasil Pengujian Rute Terpendek GA dengan GA-DAC Menggunakan KroA100

Algoritma	Threshold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	139511,32	135746,05	3005,16	12,60
	0,6	147663,40	144981,55	1449,32	7,49
	0,7	155821,22	139145,86	6482,43	2,38
GA	-	159627,14	140623,90	9890,13	0

Tabel 2. Hasil Pengujian Rute Terpendek GA dengan GA-DAC Menggunakan KroA150

Algoritma	Threshold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	217836,81	209753,12	4350,82	13,92
	0,6	225226,64	217199	3633,59	11,00
	0,7	238986,27	235508,65	2475,26	5,56
GA	-	253068,79	246154,47	6657,12	0

Tabel 3. Hasil Pengujian Rute Terpendek GA dengan GA-DAC Menggunakan KroA200

Algo Ritma	Threshold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	293804,89	288810,17	2893,44	12,92
	0,6	302040,42	297152,60	2921,81	10,48
	0,7	314607,09	308784,72	3304,86	6,75
GA	-	337406,73	326428,22	9334,52	0

Pada pengujian keragaman populasi yang dilakukan pada GA dan GA-DAC menggunakan dataset KroA100, KroA150 dan KroA200 dengan 60.000 iterasi, 10 kali *running* serta *threshold* 0,5, 0,6, dan 0,7.

Hasil dari uji keragaman populasi dengan KroA100 ini pada Tabel 4 menunjukkan bahwa GA-DAC pada *threshold* 0,6 adalah nilai yang bisa mendapatkan keragaman terbaik yaitu sebesar 24,97%. Sedangkan untuk hasil keragaman pada setiap kali *running* terlihat pada Tabel 5 dan juga Gambar 2 yang menunjukan bahwa dihampir setiap kali *running* GA-DAC mampu mendapatkan keragaman yang lebih baik dari GA.

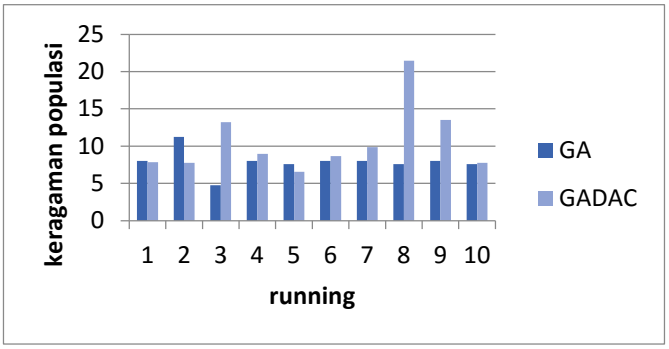
Tabel 4. Hasil Pengujian Keragaman Populasi GA dengan GA-DAC Menggunakan KroA100

Algo Ritma	Threshold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	9,2165	15,2805	2,5606	14,05
	0,6	10,55715	21,4500	4,4738	24,97
	0,7	8,7871	9,4575	0,6696	9,85
GA	-	7,9212	11,2506	1,6412	-

Copyright @ 2015 IlmuKomputer.Com

Tabel 5. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA100

Run	GA-DAC	GA
1	7,033948	8,034521
2	15,28053	11,25059
3	8,761778	4,735462
4	9,498025	8,034521
5	7,834657	7,566016
6	7,837013	8,034521
7	8,411818	8,034521
8	9,300358	7,566016
9	6,583897	8,034521
10	11,6233	7,566016



Gambar 2. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA100

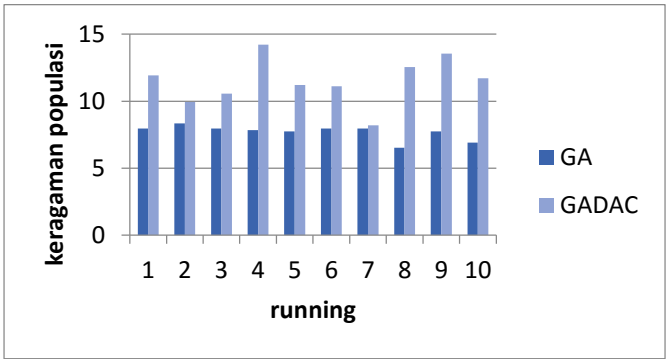
Hasil dari uji keragaman populasi dengan KroA150 ini pada Tabel 6 menunjukan bahwa GA-DAC pada *threshold* 0,6 adalah nilai yang bisa mendapatkan keragaman terbaik yaitu sebesar 50,84%. Sedangkan untuk hasil keragaman pada setiap kali *running* terlihat pada Tabel 7 dan juga Gambar 3 yang menunjukan bahwa setiap kali *running* GA-DAC mampu mendapatkan keragaman yang lebih baik dari GA.

Tabel 6. Hasil Pengujian Keragaman Populasi GA dengan GA-DAC Menggunakan KroA150

Algo Ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	11,4977	14,2078	1,7401	49,43
	0,6	11,6063	17,7734	2,7588	50,84
	0,7	11,2672	15,4335	1,6739	46,43
GA	-	7,6945	8,3345	0,54405	-

Tabel 7. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA150

Run	GA-DAC	GA
1	11,92446	7,958242
2	9,943509	8334489
3	10,55336	7,958242
4	14,20788	7,845096
5	11,21325	7,740081
6	11,11417	7,958242
7	8,208898	7,958242
8	12,55096	6,533787
9	13,55199	7,740081
10	11,70824	6,918578



Gambar 3. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA150

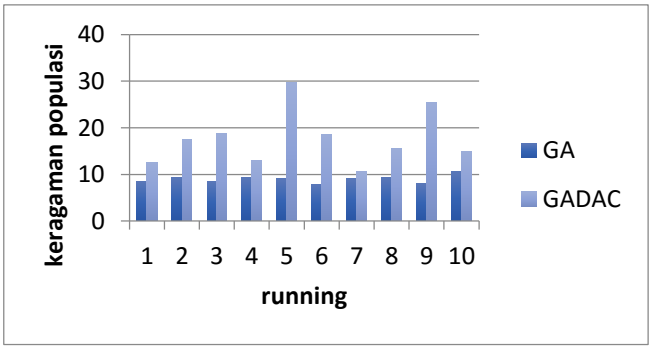
Hasil dari uji keragaman populasi dengan KroA200 ini pada Tabel 8 menunjukan bahwa GA-DAC pada *threshold* 0,5 adalah nilai yang bisa mendapatkan keragaman terbaik yaitu sebesar 49,08%. Sedangkan untuk hasil keragaman pada setiap kali *running* terlihat pada Tabel 9 dan juga Gambar 4 yang menunjukan bahwa setiap kali *running* GA-DAC mampu mendapatkan keragaman yang lebih baik dari GA

Tabel 8. Hasil Pengujian Keragaman Populasi GA dengan GA-DAC Menggunakan KroA200

Algo Ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	17,6395	29,6196	5,91173	49,08
	0,6	15,5960	26,2403	5,1432	42,41
	0,7	12,7379	15,6146	1,5683	29,48
GA	-	8,981565	10,5045	0,7489	-

Tabel 9. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA200

Run	GA	GADAC
1	8,5427707	12,601482
2	9,3446126	17,525871
3	8,5427707	18,729451
4	9,3446126	12,924827
5	9,0895829	29,619646
6	7,9008942	18,535542
7	9,0895829	10,511024
8	9,3446126	15,621517
9	8,1117358	25,348371
10	10,504478	14,977611



Gambar 4. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA200

Selain membandingkan hasil penelitian ini dengan GA standar, penelitian ini juga dibandingkan dengan penelitian lain

Copyright @ 2015 IlmuKomputer.Com

yang sejenis, yang membahas tentang permasalahan yang sama, tentang konvergensi prematur dengan dataset TSP KroA100, KroA150 dan KroA200 yaitu pada metode DDCGA (P.-C. Chang et al., 2010). Perbandingan ini membandingkan hasil rute terpendek yang dihasilkan oleh GA-DAC dan DDC-GA, yang hasilnya bisa dilihat pada Tabel 10 yang menerangkan bahwa yang tercetak tebal menandakan hasil yang lebih unggul atau yang lebih baik. Ini membuktikan bahwa GA-DAC lebih unggul di hampir semua dataset yang digunakan dibandingkan dengan DDC-GA, kecuali untuk dataset KroA200 pada metode DDC-GA mendapatkan nilai lebih baik daripada GA-DAC.

Tabel 10. Perbandingan Hasil Rute Terbaik GA-DAC dengan DDCGA

Dataset	Algoritma	Hasil
KroA100	GA-DAC	**12,60%**
	DDC-GA	5,08%
KroA150	GA-DAC	**13,92%**
	DDC-GA	2,27%
KroA200	GA-DAC	12,92%
	DDC-GA	**16,12%**

5 KESIMPULAN

Dalam penelitian ini diusulkan metode kromosom buatan dinamis dan seleksi kromosom terbaik untuk mengatasi masalah konvergensi prematur didalam GA. Pada penelitian ini dilakukan beberapa pengujian untuk mencapai hasil perbaikan tertinggi rute terpendek dalam dataset KroA100 sebesar 12,60%, KroA150 sebesar 13,92% dan KroA200 sebesar 12,92%. Pada keragaman populasi GA-DAC dapat mencapai nilai perbaikan lebih baik dalam dataset KroA100 sebesar 24,97%, KroA150 sebesar 50,84% dan KroA200 sebesar 49,08% dibandingkan dengan GA.

Pada perbandingan hasil rute terbaik yang telah dilakukan GA-DAC dengan DDC-GA didapatkan hasil bahwa GA-DAC lebih unggul di beberapa dataset yaitu KroA100 dan KroA150 dibandingkan dengan DDCGA, tetapi pada dataset KroA200 DDC-GA lebih unggul dibandingkan dengan GA-DAC.

Dari hasil pengujian diatas maka bisa disimpulkan bahwa dengan menggunakan metode GA-DAC dan seleksi kromosom terbaik bisa menemukan rute terpendek dan membuat tingkat keragaman populasi menjadi lebih beragam, sehingga ini bisa membuat GA keluar dari optimum lokal (konvergensi prematur).

REFERENSI

Çavdar, B., & Sokol, J. (2014). TSP Race: Minimizing completion time in time-sensitive applications. *European Journal of Operational Research*, *000*, 1–8. doi:10.1016/j.ejor.2014.12.022

Chang, P.-C., Huang, W.-H., & Ting, C.-J. (2010). Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. *Expert Systems with Applications*, *37*(3), 1863–1878. doi:10.1016/j.eswa.2009.07.066

Chang, Y.-H. (2010). Adopting co-evolution and constraint-satisfaction concept on genetic algorithms to solve supply chain network design problems. *Expert Systems with Applications*, *37*(10), 6919–6930. doi:10.1016/j.eswa.2010.03.030

Chen, S.-M., & Chien, C.-Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. *Expert Systems with Applications*, *38*(12), 14439–14450. doi:10.1016/j.eswa.2011.04.163

De Giovanni, L., & Pezzella, F. (2010). An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem. *European Journal of Operational Research*, *200*(2), 395–408. doi:10.1016/j.ejor.2009.01.008

Elfeky, E., Sarker, R., & Essam, D. (2008). Analyzing the simple ranking and selection process for constrained evolutionary optimization. *Journal of Computer Science and …*, *23*(1), 19–34. doi:10.1007/s11390-008-9109-z

Elsayed, S. M., Sarker, R. a., & Essam, D. L. (2014). A new genetic algorithm for solving optimization problems. *Engineering Applications of Artificial Intelligence*, *27*, 57–69. doi:10.1016/j.engappai.2013.09.013

Liu, F., & Zeng, G. (2009). Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, *36*(3), 6995–7001. doi:10.1016/j.eswa.2008.08.026

Ono, I., Kita, H., & Kobayashi, S. (2003). A real-coded genetic algorithm using the unimodal normal distribution crossover. *Advances in Evolutionary Computing*. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-18965-4_8

Pandey, H. M., Chaudhary, A., & Mehrotra, D. (2014). A comparative review of approaches to prevent premature convergence in GA. *Applied Soft Computing*, *24*, 1047–1077. doi:10.1016/j.asoc.2014.08.025

Pavez-Lazo, B., & Soto-Cartes, J. (2011). A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem. *Expert Systems with Applications*, *38*(6), 6523–6529. doi:10.1016/j.eswa.2010.11.089

S.N Sivanandam, S. N. D. (2008). *Introduction to Genetic Algorithms*. (I. Integra Software Services Pvt. Ltd., Ed.)*Vasa* (p. 462). Berlin Heidelberg: Springer. doi:10.1007/978-3-540-73190-0_2

Siva Sathya, S., & Radhika, M. V. (2013). Convergence of nomadic genetic algorithm on benchmark mathematical functions. *Applied Soft Computing*, *13*(5), 2759–2766. doi:10.1016/j.asoc.2012.11.011

Wang, Y. (2014). The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem q. *COMPUTERS & INDUSTRIAL ENGINEERING*, *70*, 124–133. doi:10.1016/j.cie.2014.01.015

Yang, S., & Jat, S. N. (2011). Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *41*(1), 93–106. doi:10.1109/TSMCC.2010.2049200

Zhang, H., Tong, W., Xu, Y., & Lin, G. (2015). The Steiner Traveling Salesman Problem with online edge blockages. *European Journal of Operational Research*, *243*(1), 30–40. doi:10.1016/j.ejor.2014.11.013

BIOGRAFI PENULIS

Muhammad Rikzam Kamal. Menyelesaikan pendidikan S1 Teknik Informatika di STMIK Widya Pratama Pekalongan, S2 Magister Teknik Informatika di Universitas Dian Nuswantoro Semarang. Saat ini menjadi Staf dan dosen STAI Ki Ageng Pekalongan. Minat penelitian saat ini adalah softcomputing.

Copyright @ 2015 IlmuKomputer.Com

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiai ilmiah ACM, PMI dan IEEE Computer Society.

Abdul Syukur. Menerima gelar sarjana di bidang Matematika dari Universitas Diponegoro Semarang, gelar master di bidang manajemen dari Universitas Atma Jaya Yogyakarta, dan gelar doktor di bidang ekonomi dari Universitas Merdeka Malang. Dia adalah dosen dan dekan di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia. Minat penelitiannya saat ini meliputi decision support systems dan information management systems.

Copyright @ 2015 IlmuKomputer.Com

Penerapan Exponential Smoothing untuk Transformasi Data dalam Meningkatkan Akurasi Neural Network pada Prediksi Harga Emas

Indah Suryani
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
ihy.indah@gmail.com

Romi Satria Wahono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
romi@brainmatics.com

Abstrak: Emas menjadi salah satu logam mulia yang paling banyak diminati baik untuk investasi maupun untuk dijadikan perhiasan. Memprediksi harga emas telah menjadi signifikan dan sangat penting bagi investor karena emas merupakan alat yang penting untuk perlindungan nilai resiko serta sebagai jalan investasi. Metode Neural Network merupakan salah satu model yang paling banyak digunakan dalam berbagai bidang penelitian. Neural Network memiliki banyak fitur yang diinginkan yang sangat cocok untuk aplikasi peramalan. Namun sebagai sistem *black box*, pemodelan Neural Network sepenuhnya tergantung pada input dan output data sehingga kualitas dan distribusi set sampel pembelajaran penting bagi kemampuan generalisasi jaringan. Maka pada penelitian ini, metode Exponential Smoothing digunakan untuk melakukan transformasi data guna meningkatkan kualitas data sehingga dapat meningkatkan akurasi prediksi pada Neural Network. Eksperimen yang dilakukan pada penelitian ini adalah untuk memperoleh arsitektur optimal sehingga menghasilkan prediksi harga emas yang akurat. Penelitian ini menggunakan Neural Network dan Exponential Smoothing dengan 10 kombinasi parameter pada eksperimen yang dilakukan. Kesimpulan yang didapatkan dari eksperimen yang dilakukan adalah bahwa prediksi harga emas menggunakan Neural Network dan Exponential Smoothing lebih akurat dibanding metode individual Neural Network.

Kata Kunci: emas, prediksi, neural network, exponential smoothing,

1 PENDAHULUAN

Emas merupakan barang berharga yang nilainya tak pernah lekang oleh waktu. Emas menjadi salah satu primadona logam mulia yang paling banyak diminati. Sepanjang sejarah, emas telah diperdagangkan secara aktif di pasar internasional (Zhou, Lai, & Yen, 2012) . Dari masa ke masa meskipun nilai emas selalu mengalami perubahan seiring pertambahan zaman, namun emas tetap menjadi investasi yang menarik. Emas juga menjadi barang berharga yang tidak hanya sekedar menjadi simpanan yang aman, emas juga memainkan peranan penting dalam sistem moneter uang riil (Apergis, 2014). Disamping itu juga ditemukan hubungan sistematis yang kuat antara harga emas dan nilai tukar (Apergis, 2014). Maka dari itu pantaslah jika di masa lampau maupun di masa modern ini bahwa emas memang masih memiliki nilai tinggi dan menjanjikan.

Pasar emas telah memperlihatkan peningkatan harga yang stabil selama beberapa dekade terakhir. Namun peramalan atau prediksi mengenai harga emas tetap menjadi hal yang penting karena menurut (Montgomery, 2008)**,** peramalan kejadian masa depan adalah masukan penting dalam banyak jenis perencanaan dan proses pengambilan keputusan**.** Maka dari itu, memprediksi harga emas telah menjadi signifikan dan sangat penting bagi investor (Zhou et al., 2012), karena data survey dari perkiraan harga emas dan perak menyediakan ligkungan data yang sangat kaya bagi para pembuat kebijakan dan investor untuk mempelajari perkembangan di pasar emas dan perak (Pierdzioch, Risse, & Rohloff, 2014). Dengan adanya prediksi harga emas dengan hasil yang akurat, diharapkan dapat digunakan untuk membantu para pembuat kebijakan serta membantu para investor dalam mengambil keputusan yang tepat dalam investasi emas.

Penelitian mengenai harga emas juga telah banyak dilakukan oleh para peneliti sebelumnya diantaranya penelitian mengenai pasar emas yang telah dilakukan oleh Zhou et al (2012). Penelitian lainnya yang dilakukan oleh Apergis (2014) meneliti mengenai keterkaitan antara harga emas dengan pergerakan Dolar Australia dan penelitian mengenai efisiensi pasar emas oleh Pierdzioch et al. (2014).

Dalam *data mining*, penelitian mengenai peramalan atau prediksi telah banyak berkembang. Senada dengan hal ini, maka banyak penelitian yang hadir adalah menggunakan metode gabungan dalam melakukan prediksi. Ada peneliti yang menggabungkan Genetic Algorithm dengan BP Neural Network (Yu & Xu, 2014), PSO dengan Neural Network (Pulido, Melin, & Castillo, 2014), SVR dengan RBFNN (Ko & Lee, 2013). Dan metode pengembangan metode gabungan ARIMA dengan Neural Network seperti diantaranya (Babu & Reddy, 2014).

Neural Network merupakan salah satu model yang paling banyak digunakan dalam berbagai bidang penelitian. Neural Network menyediakan alat yang menjanjikan bagi peramal, Neural Network juga memiliki banyak fitur yang diinginkan yang sangat cocok untuk aplikasi peramalan praktis (Zhang, 2004). Sebagai *approximators* dan sistem pembelajaran yang fleksibel, jaringan saraf telah menarik meningkatnya minat dalam menggunakan mereka untuk pemodelan dan peramalan runtun waktu (Ouyang & Yin, 2014)**.** Manfaat utama dari penggunaan Neural Network termasuk kemampuan mereka untuk menggeneralisasi, mengidentifikasi hubungan non-linear dan penerapan ke berbagai aplikasi (Bennett, Stewart, & Lu, 2014).

Selain memiliki banyak keunggulan, ternyata model Neural Network juga memiliki beberapa kelemahan yaitu Sebagai pembelajaran dengan jaringan saraf dapat dianggap

Copyright @ 2015 IlmuKomputer.Com

sebagai proses khusus fungsi pas atau pendekatan, dan solusi jaringan saraf terhadap masalah umumnya tidak akurat, ketepatan solusinya dan kemampuan fungsi pendekatan harus dipertimbangkan (He & Xu, 2009) dan ada juga kekurangan dalam Neural Network konvensional, seperti kecepatan pelatihan yang lambat dan menyelidiki solusi integrasi optimal yang lemah (Liao, 2014). Salah satu kekurangan dari Neural Network lainnya adalah ketidakmampuan mereka untuk mengidentifikasi variabel peramalan penting (Lu, Lee, & Lian, 2012). Sebagai sistem *black box*, pemodelan Neural Network sepenuhnya tergantung input dan output data, sehingga kualitas dan distribusi sampel set pembelajaran penting bagi kemampuan generalisasi jaringan. Seperti dalam prakteknya kita hanya bisa mendapatkan sampel data yang terbatas dengan diberi ruang lingkup dan kondisi tersebut, karena *noise pollution* dan analisis kesalahan, kualitas data sampel akan berkurang. Sehubungan dengan Itu, dalam pemilihan sampel pembelajaran, kita harus membangun data lengkap pengumpulan dan analisis mekanisme untuk meningkatkan kepercayaan dalam *sample learning* (He & Xu, 2009). Data preprocessing adalah masalah lain yang sering direkomendasikan untuk menyorot hubungan penting atau untuk membuat data yang lebih seragam untuk memfasilitasi pembelajaran Neural Network, memenuhi persyaratan algoritma dan menghindari masalah perhitungan (Zhang, 2004).

Data harga emas merupakan salah satu data yang termasuk ke dalam data runtun waktu. Berbagai studi runtun waktu, terutama peramalan runtun waktu statistik telah menjadi teknik yang paling populer untuk skala waktu yang singkat. Analisis runtun waktu linier seperti Random Walk (RW), Autoregressive (AR), Moving Average (MA), Simple Exponential Smoothing (SES) dan metode Autoregressive Integrated Moving Average (ARIMA) yang banyak digunakan untuk pemodelan dan prediksi data radiasi matahari (Dong, Yang, Reindl, & Walsh, 2013). Pemulusan data runtun waktu adalah tugas yang terjadi pada banyak aplikasi dan digunakan pervasiv sebagai alat untuk prediksi atau peramalan dan belajar dalam sistem berkembang. Di antara metode yang paling populer digunakan untuk melaksanakan proses ini adalah Moving Average dan Exponential Smoothing (Yager, 2013). Banyak penulis telah bekerja untuk mengembangkan Exponential Smoothing dalam kerangka statistik (Dong et al., 2013), selain itu Exponential Smoothing memiliki berbagai kelebihan diantaranya adalah metode Exponential Smoothing adalah kelas metode yang menghasilkan perkiraan dengan rumus sederhana, dengan mempertimbangkan tren dan efek musiman data (Tratar, 2015), selain itu model Exponential Smoothing merupakan alat prediksi yang penting baik dalam bisnis dan ekonomi makro (Sbrana & Silvestrini, 2014) dan metode Exponential Smoothing yang sangat sukses, mengalahkan banyak metode yang lebih canggih lainnya (Beaumont, 2014).

Persiapan data merupakan langkah penting dalam membangun sebuah model Neural Network yang sukses. Tanpa kumpulan data yang baik, memadai dan representatif, tidak mungkin untuk mengembangkan prediksi Model Neural Network yang berguna. Dengan demikian, keandalan model Neural Network tergantung pada sejauh seberapa besar kualitas data (Zhang, 2004). Maka pada penelitian ini metode Exponential Smoothing digunakan untuk memperbaiki kualitas data yang akan digunakan pada prediksi harga emas menggunakan metode Neural Network.

2 PENELITIAN TERKAIT

Anbazhagan & Kumarappan (2014) menyatakan bahwa model Neural Network telah menunjukkan peningkatan dalam akurasi peramalan yang terhubung dengan model yang ditentukan dengan baik lainnya. Pada penelitiannya, (Anbazhagan & Kumarappan, 2014) mengangkat masalah mengenai perlunya proses pra pengolahan data untuk mengekstrak informasi berlebihan dari sinyal asli. Metode yang digunakan dalam penelitian ini adalah metode Neural Network. Untuk dapat meningkatkan efisiensi pembelajaran pada Feed Forward Neural Network (FFNN), maka dilakukan proses pra pengolahan data dengan melakukan transformasi data menggunakan *Discrete Cosine Transform* (DCT), Model DCT-FFNN ini diramalkan dapat mendekati *state of the art* dengan pencapaian waktu komputasi yang lebih rendah. Adapun dataset yang digunakan adalah data harga listrik di Spanyol dan NewYork. Dengan ini maka pendekatan yang diusulkan tanpa melakukan hibridisasi terhadap model *hard* dan *soft computing*. Evaluasi terhadap *performance* dilakukan dengan membandingkan nilai MAPE, *Sum Squared Error* (SSE) dan Standard *Defiation of Error* (SDE). Hasil penelitian menunjukkan bahwa Model DCT-FFNN menyajikan kompleksitas pemodelan yang lebih rendah yang sangat cocok untuk *real-time* pasar listrik yang kompetitif. Selain itu, model DCT-FFNN jugan menunjukkan waktu komputasi yang lebih rendah jika dibandingkan dengan 17 model lainnya.

Pada penelitian lainnya yang dilakukan oleh Jammazi & Aloui (2012) menggunakan dataset harga minyak mentah dunia dari IEA pada tahun 2011. Data harga minyak mentah dunia memiliki volatilitas yang tinggi dan non stasioner. Dalam penelitian ini, harga minyak mentah WTI bulanan digunakan untuk menilai *the A Haar Trous Wavelet Transforms* dalam mendapatkan pemulusan komponen tanpa kehilangan sifat yang mendasari dari sinyal yang bersangkutan. *Filter wavelet* yang digunakan untuk dekomposisi adalah *the discrete low filter*. Setelah dilakukan dekomposisi wavelet, selanjutnya pemodelan dilakukan menggunakan metode Neural Network yang diterapkan pada dataset yang dibagi menjadi *in-sample data* dan *out-of-sample data*. Berdasarkan simulasi yang dilakukan, dilakukan evaluasi performance dengan membandingkan tingkat *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE) yang dihasilkan.

Dalam penelitian yang dilakukan oleh (Beaumont, 2014), mengangkat permasalahan mengenai arti penting transformasi data dalam melakukan peramalan. Dataset yang digunakan dalam penelitian ini berupa data kompetisi M3 yang diterapkan untuk 645 seri tahunan, 756 seri triwulanan dan 1428 seri bulanan. *Log transform* dan *Box-Cox transform* diterapkan pada pengamatan seri, dan sisanya yaitu *Johnson Error Trend Seasonal (JETS) Transform, Heteroscedastic state space transform* dan *t transform* diterapkan dalam transformasi kesalahan. Evaluasi hasil dilakukan dengan membandingkan tingkat *Mean Absolutly Squared Error* (MASE), MAPE dan *Range Probability Scored* (RPS) terendah serta membandingkan tingkat *Minus Log Prediction Likelihood* (MLPL) dengan tingkat tertinggi.

Dari hasil penelitian sebelumnya tersebut, maka dapat disimpulkan bahwa kualitas input data dapat membuktikan ketidakpuasan untuk ketidaklengkapan, kebisingan dan ketidakkonsistenan data (Vercellis, 2009). Atas berbagai kelebihan yang dimiliki oleh Exponential Smoothing maka diharapkan dengan adanya penerapan Exponential Smoothing dalam transformasi data dapat meningkatkan akurasi prediksi pada Neural Network, karena menurut Beaumont (2014), salah satu keterbatasan metode penelitian adalah bahwa mereka

Copyright @ 2015 IlmuKomputer.Com

mengabaikan potensi transformasi untuk meningkatkan perkiraan.

3 PENGUMPULAN DATA

Data yang dikumpulkan dalam penelitian ini merupakan data sekunder berupa data harga emas yang terdiri dari 1301 *record* yang menampilkan tanggal dan harga penutupan emas harian. Data harga emas harian tersebut dapat dilihat pada Tabel 1.

TabeL 1. Data Harga Emas Harian NYSE

Date	Close
June 30, 2014	1,322.00
June 27, 2014	1,320.00
June 26, 2014	1,316.10
June 25, 2014	1,322.20
June 24, 2014	1,320.90
June 23, 2014	1,318.00
June 20, 2014	1,316.20
June 19, 2014	1,313.70
June 18, 2014	1,272.40
June 17, 2014	1,271.70
June 16, 2014	1,274.90
June 13, 2014	1,273.70
June 12, 2014	1,273.60
June 11, 2014	1,260.80
June 10, 2014	1,259.80
June 9, 2014	1,253.50
June 6, 2014	1,252.10
June 5, 2014	1,253.00
June 4, 2014	1,244.00
June 3, 2014	1,244.30
June 2, 2014	1,243.70

4 PENGOLAHAN DATA AWAL

Pengolahan data awal yang dilakukan dalam penelitian ini terdiri dari:

1. *Replace Missing Values*
   Dalam dataset yang digunakan dalam penelitian ini, masih ditemukan mengandung *missing values*, sehingga untuk mengatasi sebagian data yang tidak lengkap tersebut dapat mengadopsi teknik substitusi. Yaitu menggantikan nilai yang hilang dengan mengubahnya menjadi nilai yang diinginkan yaitu rata-rata atribut dihitung untuk pengamatan yang tersisa. Teknik ini dapat diterapkan untuk atribut yang bernilai numerik.
2. *Set Role*
   Set role operator digunakan dalam penelitian ini untuk mengubah peran suatu atribut yaitu atribut *date* dari atribut regular menjadi atribut spesial yaitu sebagai *id*.
3. *Normalize*
   Normalisasi data dilakukan sesuai dengan fungsi aktivasi yang digunakan, dalam penelitian ini menggunakan dua fungsi aktivasi yaitu fungsi *binary sigmoid* dan *bipolar sigmoid*. Fungsi *binary sigmoid*, menormalisasikan data dalam range 0 sampai 1 (Yu & Xu, 2014). Adapun rumus untuk fungsi *binary sigmoid* (*Logsig*) adalah:

$$y' = \frac{x - xmin}{xmax - xmin} \times (0.9 - 0.1) + 0.1 \quad (1)$$

   Sedangkan rumus untuk *bipolar sigmoid* (*Tansig*) adalah:

$$y' = \frac{x - xmin}{xmax - xmin} \times 2 - 1 \quad (2)$$

   Keterangan:
   y' = Hasil transformasi data
   x = Nilai asli
   $xmin$ = Nilai minimal
   $xmax$ = Nilai maksimal

4. *Windowing*
   Windowing merupakan salah satu teknik dalam menentukan data *input* dan data *output* dalam prediksi data runtun waktu dengan tipe univariat. Data univariat adalah distribusi data dengan melibatkan satu atribut atau variabel (Han et al., 2012). Dengan teknik *windowing* tersebut data univariat yang berupa harga penutupan yang diwakili oleh atribut *Close* pada data, selanjutnya akan dipecah menjadi 5 data *input* dan 1 data *output*. Data *input* merupakan data 5 hari sebelumnya dan data *output* adalah data 1 hari berikutnya
5. Transformasi Data dengan Exponential Smoothing
   Exponential Smoothing pada penelitian ini digunakan dalam melakukan transformasi data guna memperbaiki kualitas data yang diharapkan dapat meningkatkan akurasi Neural Network. dengan contoh perhitungan sebagai berikut:

$$y't + 1 = y't + \alpha(yt - y't) \quad (3)$$

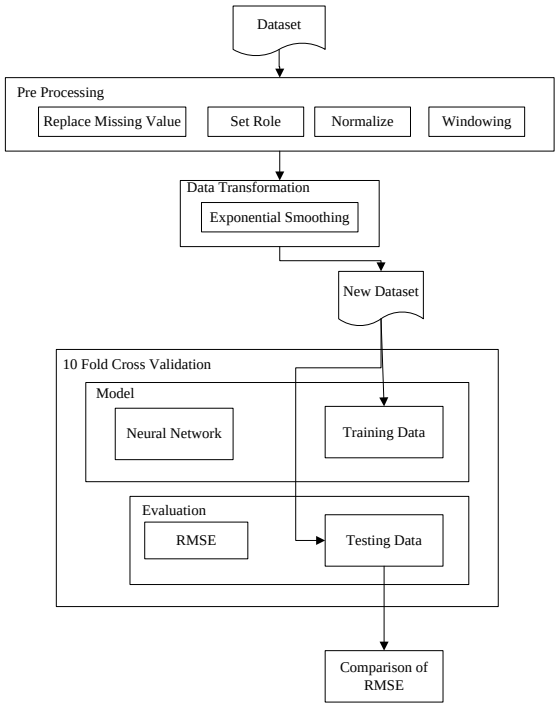
Berdasarkan salah satu *sample* nilai *output* (*class*) dataset harga emas setelah dilakukan normalisasi dengan *binary sigmoid* diketahui:

$y't + 1$ = Nilai peramalan periode berikutnya
$y't$ = Nilai pemulusan yang lama atau rata-rata yang dimuluskan hingga periode t-1
yt = Data baru atau nilai y yang sebenarnya di periode t
α = Konstanta pemulusan ($0 < \alpha < 1$)

5 METODE YANG DIUSULKAN

Metode yang diusulkan pada penelitian ini yaitu penerapan metode Exponential Smoothing untuk transformasi data yang diterapkan pada metode Neural Network. Metode Exponential Smoothing yang digunakan adalah Simple Exponential Smoothing dan untuk metode Neural Network yang digunakan adalah Backpropagation Neural Network. Adapun metode yang diusulkan dapat dilihat pada Gambar 1.

Copyright @ 2015 IlmuKomputer.Com



Gambar 1. Metode yang Diusulkan

Alur penelitian yang digunakan dalam penelitiaan ini dimulai dengan penginputan data untuk selanjutnya dilakukan proses pra pengolahan data berupa *replace missing value*, *set role*, *normalize* dan *windowing*.

Replace missing value yaitu salah satu operator yang terdapat di *data cleansing* pada *RapidMiner* yang membantu menangani nilai null yang mungkin ada dalam data, yang dapat digunakan untuk menemukan nilai-nilai yang hilang dalam atribut atau serangkaian atribut dan merubahnya, dari nilai yang hilang ke nilai yang diinginkan (Hofmann, 2009). Langkah berikutnya adalah untuk mengatur peran pada atribut yang akan digunakan, dalam hal ini untuk operator yang digunakan pada RapidMiner adalah operator *set role*. Penetapan peran atribut ini sangat penting untuk menetapkan peran yang tepat untuk atribut dalam dataset. Kebanyakan operator klasifikasi tidak akan bekerja jika tidak ada atribut dengan peran label dalam *dataset* (atau bahkan jika ada beberapa atribut dengan peran label). Perlu memastikan bahwa hanya satu atribut (dan yang benar) Memiliki peran label. Dalam *setup* klasifikasi sangat dasar semua atribut lainnya akan memiliki peran reguler. Jika atribut unik mengidentifikasi contoh dapat diberikan peran *id* (Hofmann, 2009). Setelah dilakukan *set role*, kemudian dilakukan normalisasi dataset menggunakan operator *normalize* pada RapidMiner. Normalisasi diterapkan pada semua atribut, dan semua nilai atribut diubah menjadi kisaran 0-1. Langkah ini sangat penting karena atribut dalam *dataset* mungkin berbeda skala, yang mempengaruhi perhitungan jarak (Hofmann, 2009). Selanjutnya dilakukan proses *windowing*, *windowing* biasanya digunakan untuk mengubah data *time series* menjadi contoh set yang berisi contoh dengan beberapa atribut yang sesuai dengan poin berurutan. Ini contoh set kemudian dapat digunakan untuk pembentukan model, klasifikasi, atau analisis prediktif. Jendela juga dapat digunakan untuk memvisualisasikan data (Chisholm, 2013). Selanjutnya diterapkan metode Exponential Smoothing untuk melakukan transformasi data untuk kemudian dilakukan *training* dan *testing* menggunakan metode Neural network.

6 HASIL DAN PEMBAHASAN

Tahapan eksperimen yang dilakukan dalam penelitian ini adalah:

1. Menyiapkan dataset untuk penelitian
2. Melalukan pra pengolahan data
3. Merancang arsitektur Neural Network dengan memasukkan nilai parameter Neural Network yang terdiri dari *training cycle*, *learning rate*, *momentum* dan *hidden layer*
4. Melakukan *training* dan *testing* terhadap model Neural Network, kemudian mencatat hasil RMSE yang didapat
5. Merancang arsitektur Neural Network dengan memasukkan parameter Neural Network berupa *training cycle*, *learning rate*, *momentum* dan *hidden layer* dan parameter Exponential Smoothing yang berupa *alpha* (α)
6. Melakukan *training* dan *testing* terhadap model usulan berupa pengembangan Neural Network dengan Exponential Smoothing, kemudian mencatat hasil RMSE yang didapat

Melakukan perbandingan hasil RMSE pada kedua model dengan uji beda menggunakan t-Test. Setelah dilakukan pengujian model menggunakan *tools* Rapidminer 5.3, selanjutnya dilakukan evaluasi pebandingan hasil RMSE seluruh eksperimen dengan *10-fold cross validation* . Eksperimen dilakukan dengan metode Neural Network, kemudian dibandingkan dengan hasil eksperimen menggunakan metode Neural Network yang dikembangkan dengan Exponential Smoothing.

Pada eksperimen pertama ini percobaan dilakukan dengan melakukan inisialisasi parameter Neural Network yang terdiri dari *training cycle*, *learning rate*, *momentum* dan *hidden layer* dan dengan dilakukan normalisasi data terlebih dahulu menggunakan fungsi aktivasi *binary sigmoid* untuk kemudian diuji coba menggunakan sistem *random* dan *error* sehingga dihasilkan model terbaik yang ditandai dengan perolehan hasil RMSE dengan nilai terkecil seperti pada Tabel 2.

Tabel 2. Hasil Eksperimen Metode Neural Network (Fungsi Aktivasi *Binary Sigmoid*)

Hidden Layer	Hidden Layers Size	Training Cycle	Learning Rate	Momentum	Horizon	RMSE
1	1	500	0.3	0.2	1	0.015
1	1	500	0.6	0.3	1	0.015
1	**3**	**1000**	**0.6**	**0.3**	**1**	**0.014**
1	**3**	**1000**	**0.9**	**0.6**	**1**	**0.014**
1	**3**	**500**	**0.9**	**0.6**	**1**	**0.014**
1	1	300	0.5	0.5	1	0.015
1	1	300	0.1	0.3	1	0.019
1	3	500	0.3	0.2	1	0.015
2	2,2	500	0.6	0.3	1	0.015
2	**3,3**	**500**	**0.9**	**0.6**	**1**	**0.014**

Pada eksperimen selanjutnya dengan melakukan inisialisasi parameter Neural Network yang terdiri dari *training cycle*, *learning rate*, *momentum* dan *hidden layer* dan dengan dilakukan normalisasi data terlebih dahulu menggunakan fungsi aktivasi *bipolar sigmoid* untuk kemudian diuji coba menggunakan sistem *random* dan *error* sehingga dihasilkan model terbaik yang ditandai dengan perolehan hasil RMSE dengan nilai terkecil seperti pada Tabel 3.

Tabel 3. Hasil Eksperimen Metode Neural Network (Fungsi Aktivasi *Bipolar Sigmoid*)

Copyright @ 2015 IlmuKomputer.Com

Hidden Layer	Hidden Layers Size	Training Cycle	Learning Rate	Momentum	Horizon	RMSE
1	1	500	0.3	0.2	1	0.039
1	1	500	0.6	0.3	1	0.037
1	**3**	**1000**	**0.6**	**0.3**	**1**	**0.035**
1	**3**	**1000**	**0.9**	**0.6**	**1**	**0.035**
1	**3**	**500**	**0.9**	**0.6**	**1**	**0.035**
1	1	300	0.5	0.5	1	0.037
1	1	300	0.1	0.3	1	0.047
1	3	500	0.3	0.2	1	0.037
2	2,2	500	0.6	0.3	1	0.037
2	3,3	500	0.9	0.6	1	0.036

Pada eksperimen ketiga dan keempat, percobaan dilakukan dengan melakukan inisialisasi parameter Neural Network yang terdiri dari *training cycle*, *learning rate*, *momentum* dan *hidden layer* dan dengan dilakukan normalisasi data terlebih dahulu menggunakan fungsi aktivasi *binary sigmoid* dan *bipolar sigmoid* untuk kemudian dikembangkan dengan metode Exponential Smoothing dan diuji coba menggunakan sistem *random* dan *error* sehingga dihasilkan model terbaik yang ditandai dengan perolehan hasil RMSE dengan nilai terkecil seperti pada Tabel 4 dan 5.

Tabel 6. Hasil Eksperimen Metode Neural Network dengan Fungsi Aktivasi *Binary Sigmoid* + Exponential Smoothing

Hidden Layer	Hidden Layers Size	Training Cycle	Learning Rate	Momentum	Horizon	Alpha	RMSE
1	1	500	0.3	0.2	1	0.5	0.010
1	1	500	0.6	0.3	1	0.7	0.006
1	3	1000	0.6	0.3	1	0.2	0.013
1	**3**	**500**	**0.9**	**0.6**	**1**	**0.9**	**0.003**
1	**3**	**1000**	**0.9**	**0.6**	**1**	**0.9**	**0.003**
1	1	300	0.5	0.5	1	0.6	0.007
1	1	300	0.1	0.3	1	0.9	0.015
1	3	500	0.3	0.2	1	0.3	0.013
2	2,2	500	0.6	0.3	1	0.9	0.007
2	3,3	500	0.9	0.6	1	0.9	0.004

Tabel 5. Hasil Eksperimen Metode Neural Network dengan Fungsi Aktivasi *Bipolar Sigmoid* + Exponential Smoothing

Hidden Layer	Hidden Layers Size	Training Cycle	Learning Rate	Momentum	Horizon	Alpha	RMSE
1	1	500	0.3	0.2	1	0.5	0.027
1	1	500	0.6	0.3	1	0.7	0.015
1	3	1000	0.6	0.3	1	0.2	0.033
1	**3**	**1000**	**0.9**	**0.6**	**1**	**0.9**	**0.007**
1	**3**	**500**	**0.9**	**0.6**	**1**	**0.9**	**0.007**
1	1	300	0.5	0.5	1	0.6	0.015
1	1	300	0.1	0.3	1	0.9	0.035
1	3	500	0.3	0.2	1	0.3	0.032
2	2,2	500	0.9	0.3	1	0.9	0.013
2	3,3	500	0.9	0.6	1	0.9	0.010

Pada eksperimen kelima dan keenam, percobaan dilakukan dengan melakukan inisialisasi parameter Neural Network yang terdiri dari *training cycle*, *learning rate*, *momentum* dan *hidden layer* dan dengan dilakukan normalisasi data terlebih dahulu menggunakan fungsi aktivasi *binary sigmoid* dan *bipolar sigmoid* dengan transformasi menggunakan Discrete Cosine Transfom seperti yang dilakukan pada penelitian sebelumnya oleh Anbazhagan & Kumarappan (2014) dan selanjutnya diuji coba menggunakan sistem *random* dan *error* sehingga dihasilkan model terbaik yang ditandai dengan perolehan hasil RMSE dengan nilai terkecil seperti pada Tabel 6 dan Tabel 7.

Tabel 6. Hasil Eksperimen Metode Neural Network dengan Fungsi Aktivasi *Binary Sigmoid* + Discrete Cosine Transform

Neural Network dengan Fungsi Aktivasi Binary Sigmoid + Discrete Cosine Transform						
Hidden Layer	**Hidden Layers Size**	**Training Cycle**	**Learning Rate**	**Momentum**	**Horizon**	**RMSE**
1	1	500	0.3	0.2	1	0.016
1	1	500	0.6	0.3	1	0.015
1	3	1000	0.6	0.3	1	0.014
1	**3**	**500**	**0.9**	**0.6**	**1**	**0.014**
1	**3**	**1000**	**0.9**	**0.6**	**1**	**0.014**
1	1	300	0.5	0.5	1	0.015
1	1	300	0.1	0.3	1	0.017
1	3	500	0.3	0.2	1	0.015
2	2,2	500	0.6	0.3	1	0.015
2	3,3	500	0.9	0.6	1	0.014

Tabel 7. Hasil Eksperimen Metode Neural Network dengan Fungsi Aktivasi *Bipolar Sigmoid* + Discrete Cosine Transform

Neural Network dengan Fungsi Aktivasi Bipolar Sigmoid + Discrete Cosine Transform						
Hidden Layer	**Hidden Layers Size**	**Training Cycle**	**Learning Rate**	**Momentum**	**Horizon**	**RMSE**
1	1	500	0.3	0.2	1	0.039
1	1	500	0.6	0.3	1	0.037
1	3	1000	0.6	0.3	1	0.036
1	3	1000	0.9	0.6	1	**0.035**
1	3	500	0.9	0.6	1	**0.035**
1	1	300	0.5	0.5	1	0.037
1	1	300	0.1	0.3	1	0.040
1	3	500	0.3	0.2	1	0.036
2	2,2	500	0.9	0.3	1	0.037
2	3,3	500	0.9	0.6	1	0.036

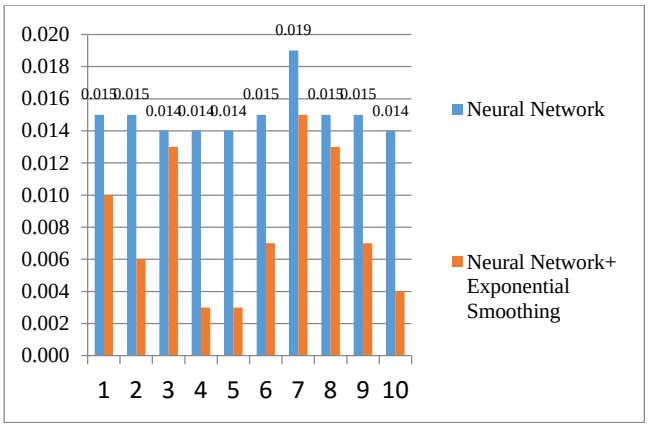
Berdasarkan hasil eksperimen yang telah dilakukan, diperoleh perbedaan nilai rata-rata RMSE pada pengujian model Neural Network sebelum dan sesudah dilakukan transformasi data menggunakan Exponential Smoothing pada tingkat akurasi prediksi harga emas. Tingkat perbandingan rata-rata nilai RMSE yang dihasilkan tersebut dapat dilihat pada Tabel 8.

Tabel 8. Perbandingan Nilai RMSE Neural Network dengan Neural Network + Exponential Smoothing (Fungsi Aktivasi *Binary Sigmoid*)

Perbandingan RMSE	
Neural Network	**Neural Network+ Exponential Smoothing**
0.015	0.010
0.015	0.006
0.014	0.013
0.014	**0.003**
0.014	**0.003**
0.015	0.007
0.019	0.015
0.015	0.013
0.015	0.007
0.014	0.004

Berdasarkan data yang diperoleh dari Tabel 8 maka dapat ditampilkan grafik pada Gambar 2.

Copyright @ 2015 IlmuKomputer.Com



Gambar 2. Grafik Perbandingan RMSE Neural Network dan Neural Network + Exponential Smoothing (Fungsi Aktivasi *Binary Sigmoid*)

Untuk mengetahui ada tidaknya perbedaan antara dua model, maka dibutuhkan suatu pengujian, oleh karena itu dilakukan uji beda menggunakan t-Test untuk menguji hipotesa. Hipotesa nol (H_0) akan dihasilkan jika tidak terdapat perbedaan antara model Neural Network dengan model Neural Network dengan transformasi data Exponential Smoothing. Hipotesa alternatif (H_1) dihasilkan jika terdapat perbedaan antara model Neural Network dengan model Neural Network dengan transformasi data Exponential Smoothing. Adapun hasil dari uji sampel berpasangan untuk RMSE yang dihasilkan pada model Neural Network dengan model Neural Network dengan transformasi data Exponential Smoothing dengan fungsi aktivasi menggunakan binary sigmoid dapat dilihat pada Tabel 9.

Tabel 9. *Paired Two Sample T-Test* dengan metode Neural Network dan Neural Network + Exponential Smoothing (*Fungsi Aktivasi Binary Sigmoid*)

	Neural Network	Neural Network +Exponential Smoothing
Mean	0.015	0.0081
Variance	2.22222E-06	1.94333E-05
Observations	10	10
Pearson Correlation	0.625592915	
Hypothesized Mean Difference	0	
Df	9	
t Stat	5.953292143	
P(T<=t) one-tail	0.000107234	
t Critical one-tail	1.833112923	
P(T<=t) two-tail	0.000214468	
t Critical two-tail	2.262157158	

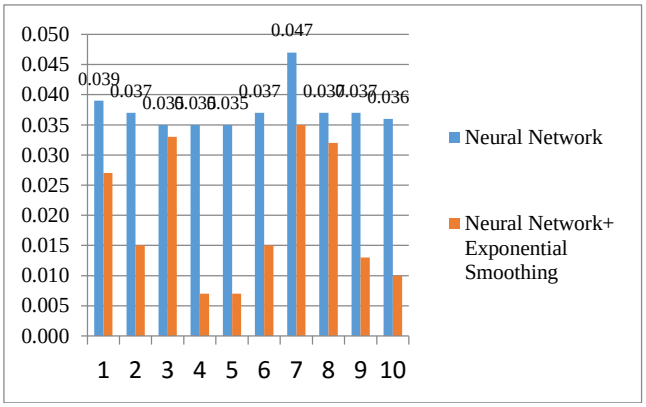
Berdasarkan hasil uji t dua sampel berpasangan pada Tabel 9, diketahui bahwa t hitung yang diwakili t *stat* dengan nilai sebesar 5.953292143 dan nilai t tabel yang diwakili oleh *t critical two tail* sebesar 2.262157158, maka nilai t itung > dari nilai t tabel dan dapat disimpulkan bahwa H_0 ditolak dan H_1 diterima. Sedangkan untuk nilai probabilitas yang dihasilkan sebesar 0.000214468, artinya terdapat perbedaan signifikan antara RMSE model Neural Network dan model Neural Network + Exponential Smoothing dengan fungsi aktivasi menggunakan *binary sigmoid*.

Berdasarkan hasil eksperimen yang telah dilakukan, diperoleh perbedaan nilai rata-rata RMSE pada pengujian model Neural Network sebelum dan sesudah dilakukan transformasi data menggunakan Exponential Smoothing pada tingkat akurasi prediksi harga emas. Tingkat perbandingan rata-rata nilai RMSE yang dihasilkan tersebut dapat dilihat pada Tabel 10.

Tabel 10. Perbandingan Nilai RMSE Neural Network dengan Neural Network + Exponential Smoothing (Fungsi Aktivasi *Binary Sigmoid*)

Perbandingan RMSE	
Neural Network	**Neural Network+ Exponential Smoothing**
0.039	0.027
0.037	0.015
0.035	0.033
0.035	0.007
0.035	0.007
0.037	0.015
0.047	0.035
0.037	0.032
0.037	0.013
0.036	0.010

Berdasarkan data yang diperoleh dari Tabel 10 maka dapat ditampilkan grafik pada Gambar 3.



Gambar 3. Grafik Perbandingan RMSE Neural Network dan Neural Network +Exponential Smoothing (Fungsi Aktivasi *Bipolar Sigmoid*)

Untuk mengetahui ada tidaknya perbedaan antara hasil dari uji sampel berpasangan untuk RMSE yang dihasilkan pada model Neural Network dengan model Neural Network dengan transformasi data Exponential Smoothing dengan fungsi aktivasi menggunakan *bipolar sigmoid* dapat dilihat pada Tabel 11.

Tabel 11. *Paired Two Sample T-Test* dengan metode Neural Network dan Neural Network + Exponential Smoothing (*Fungsi Aktivasi Bipolar Sigmoid*)

	Neural Network	Neural Network + Exponential Smoothing
Mean	0.0375	0.0194
Variance	1.27222E-05	0.000124489
Observations	10	10
Pearson Correlation	0.566769857	
Hypothesized Mean Difference	0	
Df	9	
t Stat	5.964152777	
P(T<=t) one-tail	0.000105809	
t Critical one-tail	1.833112923	
P(T<=t) two-tail	0.000211618	
t Critical two-tail	2.262157158	

Berdasarkan hasil uji t dua sampel berpasangan pada Tabel 11, diketahui bahwa t hitung yang diwakili t stat dengan nilai

Copyright @ 2015 IlmuKomputer.Com

sebesar 5.964152777 dan nilai t tabel yang diwakili oleh *t critical two tail* sebesar 2.262157158, maka nilai t itung > dari nilai t tabel dan dapat disimpulkan bahwa H_0 ditolak dan H_1 diterima. Sedangkan untuk nilai propabilitas yang dihasilkan sebesar 0.000211618, artinya terdapat perbedaan signifikan antara RMSE model Neural Network dan model Neural Network + Exponential Smoothing dengan fungsi aktivasi menggunakan *binary sigmoid*.

Keseluruhan eksperimen yang dilakukan menghasilkan 6 variasi model. Keenam variasi model terdiri dari metode individual Neural Network dengan fungsi aktivasi *binary sigmoid*, metode individual Neural Network dengan fungsi aktivasi *bipolar sigmoid*, metode Neural Network dan Exponential Smoothing dengan fungsi aktivasi *binary sigmoid*, metode Neural Network dan Exponential Smoothing dengan fungsi aktivasi *bipolar sigmoid*, metode Neural Network dan Discrete Cosine Transform dengan fungsi aktivasi *binary sigmoid* dan metode Neural Network dan Discrete Cosine Transform dengan fungsi aktivasi *bipolar sigmoid* dengan ringkasan yang tertera pada Tabel 12.

Tabel 12. Hasil Eksperimen Keseluruhan Metode

Perbandingan Nilai RMSE					
NN (Logsig)	NN (Tansig)	NN+ES (Logsig)	NN+ES (Tansig)	NN+DCT (Logsig)	NN+DCT (Tansig)
0.015	0.039	0.010	0.027	0.016	0.039
0.015	0.037	0.006	0.015	0.015	0.037
0.014	0.035	0.013	0.033	0.014	0.036
0.014	0.035	**0.003**	0.007	0.014	0.035
0.014	0.035	**0.003**	0.007	0.014	0.035
0.015	0.037	0.007	0.015	0.015	0.037
0.019	0.047	0.015	0.035	0.017	0.040
0.015	0.037	0.013	0.032	0.015	0.036
0.015	0.037	0.007	0.013	0.015	0.037
0.014	0.036	0.004	0.010	0.014	0.036

Untuk membandingkan keakuratan model yang berbeda, dapat dilakukan uji Friedman. Hal ini dilakukan untuk mencegah kemungkinan penolakan terhadap hipotesis homogenitas terhadap keakuratan model yang dibandingkan tersebut. Uji Friedman seringkali digunakan dengan sukses dengan praktek dalam variaebl diskrit yang mengambil banyak nilai. Uji Friedman ini dilakukan dengan menggunakan *Software* XLSTAT. Hasil perbandingan akurasi keenam model di atas yang dilakukan menggunaka uji Friedman dapat dilihat pada Tabel 13.

Tabel 13. Hasil Uji Friedman

Q (Observed value)	**45.608**
Q (Critical value)	11.070
DF	5
p-value (Two-tailed)	< 0.0001
Alpha	0.05

Dari hasil pengujian menggunakan Friedman *test*, dihasilkan nilai *p-value* < 0.0001 yang artinya lebih kecil dari nilai alpha 0.05, maka dengan demikian H_1 atau hipotesa alternatif diterima atau dengan kata lain, hipotesis nol ditolak. Hal ini mengindikasikan perbedaan signifikan antara keenam model yang dihasilkan. Ketika hipotesis nol ditolak, maka perlu dilakukan post-hoc test untuk mengidentifikasi pasangan tertentu atau pasang faktor dengan perbedaan dalam peringkat jumlah yang signifikan secara statistik, dan yang mungkin telah menyebabkan penolakan hipotesis nol.

Tabel 14. Hasil Uji Nemenyi

Sample	Frequency	Sum of ranks	Mean of ranks	Groups	
NN+ES (Logsig)	10	10.000	1.000	A	
NN (Logsig)	10	30.000	3.000	A	
NN+ES (Tansig)	10	30.000	3.000	A	
NN+DCT (Logsig)	10	30.000	3.000	A	
NN+DCT (Tansig)	10	54.500	5.450		B
NN (Tansig)	10	55.500	5.550		B

Selanjutnya pada Tabel 4. 14 dilakukan perbandingan hasil berpasangan dengan meringkas perbandingan berpasangan menggunakan analisis *post-hoc* yang dalam hal ini menggunakan uji Nemenyi. Hal ini dilakukan karena uji Friedman hanya menunjukkan adanya perbedaan model tetapi tidak menyediakan model yang berbeda. *Mean rank* pada Tabel 14 diperoleh dari perbandingan antar model. Semakin tinggi peringkat, semakin tinggi titik dan kemudian dibagi dengan jumlah sampel data.

Tabel 15. Hasil Uji Perbedaan Kinerja

	NN (Logsig)	NN (Tansig)	NN+ES (Logsig)	NN+ES (Tansig)	NN+DCT (Logsig)	NN+DCT (Tansig)
NN (Logsig)	0	-2.550	2.000	0.000	0.000	-2.450
NN (Tansig)	2.550	0	4.550	2.550	2.550	0.100
NN+ES (Logsig)	-2.000	-4.550	0	-2.000	-2.000	-4.450
NN+ES (Tansig)	0.000	-2.550	2.000	0	0.000	-2.450
NN+DCT (Logsig)	0.000	-2.550	2.000	0.000	0	-2.450
NN+DCT (Tansig)	2.450	-0.100	4.450	2.450	2.450	0
Critical difference: 2.3842						

Uji nemenyi menghitung semua perbandingan berpasangan antara model yang berbeda dan memeriksa kinerja mana yang berbeda dengan nilai *critical difference* (cd) 2.3842 seperti yang ditunjukkan pada Tabel 4.15.

Tabel 16. Nilai *P-Value* Hasil Uji Nemenyi

	NN (Logsig)	NN (Tansig)	NN+ES (Logsig)	NN+ES (Tansig)	NN+DCT (Logsig)	NN+DCT (Tansig)
NN (Logsig)	1	**0,028**	0,159	1,000	1,000	**0,040**
NN (Tansig)	**0,028**	1	**< 0.0001**	**0,028**	**0,028**	1,000
NN+ES (Logsig)	0,159	**< 0.0001**	1	0,159	0,159	**< 0.0001**
NN+ES (Tansig)	1,000	**0,028**	0,159	1	1,000	**0,040**
NN+DCT (Logsig)	1,000	**0,028**	0,159	1,000	1	**0,040**
NN+DCT (Tansig)	**0,040**	1,000	**< 0.0001**	**0,040**	**0,040**	1

Dapat dilihat pada Tabel 16, nilai *p-value* yang dicetak tebal merupakan nilai-nilai *p-value* yang memiliki nilai terkecil. Nilai p-value yang terkecil juga didapat pada angka 0,0001 dari hasil pengujian Model Neural Network dan Exponential Smoothing dengan fungsi aktivasi logsig atau *binary sigmoid* yang artinya angka tersebut kurang dari nilai α =0,05. Dengan demikian maka hipotesis nol ditolak yang berarti bahwa terdapat tingkat perbedaan yang signifikan secara statistik.

Copyright @ 2015 IlmuKomputer.Com

Tabel 17. Hasil Uji Signifikan Keseluruhan Model

	NN (Logsig)	NN (Tansig)	NN+ES (Logsig)	NN+ES (Tansig)	NN+DCT (Logsig)	NN+DCT (Tansig)
NN (Logsig)	No	**Yes**	No	No	No	**Yes**
NN (Tansig)	**Yes**	No	**Yes**	**Yes**	**Yes**	No
NN+ES (Logsig)	No	**Yes**	No	No	No	**Yes**
NN+ES (Tansig)	No	**Yes**	No	No	No	**Yes**
NN+DCT (Logsig)	No	**Yes**	No	No	No	**Yes**
NN+DCT (Tansig)	**Yes**	No	**Yes**	**Yes**	**Yes**	No

Hasil yang ditunjukan pada Tabel 16 tersebut sesuai dengan hasil yang ditunjukan pada Tabel 17, yang berarti bahwa model yang memiliki nilai *p-value* kurang dari nilai alpha α=0.05 maka akan menghasilkan nilai *Yes* pada Tabel 17. Pada Tabel 17 di atas, dapat dilihat bahwa model yang memiliki perbedaan signifikan ditunjukkan pada kolom dan baris yang bernilai *Yes*. Sedangkan untuk perbedaan yang tidak signifikan ditandai dengan nilai *No*. Dari hasil uji Frideman dan Nemenyi post-hoc di atas menunjukkan bahwa model Neural Network dengan Discrete Cosine Transform juga menunjukan akurasi yang tinggi dengan perbedaan yang signifikan (Anbazhagan & Kumarappan, 2014). Selain itu model Neural Network dengan Exponential Smoothing pada fungsi aktivasi *binary sigmoid* menunjukkan hasil yang lebih tinggi dan menunjukan hasil perbedaan yang signifikan.

7 KESIMPULAN

Pada penelitian ini dilakukan penerapan metode usulan berupa pengembangan metode Neural Network menggunakan metode Exponential Smoothing untuk transformasi data yang kemudian berdasarkan hasil eksperimen yang dilakukan terbukti meningkatkan hasil prediksi harga emas dengan membandingkan nilai RMSE yang dihasilkan. Nilai RMSE terkecil yang didapatkan dari penerapan metode Neural Network dengan fungsi aktivasi *binary sigmoid* adalah 0,014 dan RMSE terkecil yang dihasilkan dengan penerapan Neural Network dan Exponential Smoothing dengan fungsi aktivasi *binary sigmoid* adalah 0,003 dan penerapan Neural Network dan Exponential Smoothing dengan fungsi aktivasi *bipolar sigmoid* dengan nilai 0,007.

Melalui hasil t-Test dan Friedman Test menunjukkan bahwa adanya perbedaan atau pengaruh yang signifikan dari penerapan metode Neural Network yang dibandingkan dengan penerapan metode Neural Network dan Exponential Smoothing. Dengan demikian maka dapat disimpulkan bahwa penerapan tranformasi data menggunakan Exponential Smoothing dapat memperbaiki kualitas data yang digunakan pada penerapan Neural Network sehingga mampu meningkatkan akurasi prediksi harga emas.

REFERENSI

Anbazhagan, S., & Kumarappan, N. (2014). Day-ahead deregulated electricity market price forecasting using neural network input featured by DCT. *Energy Conversion and Management*, *78*, 711–719.

Apergis, N. (2014). Can gold prices forecast the Australian dollar movements? *International Review of Economics & Finance*, *29*, 75–82.

Babu, C. N., & Reddy, B. E. (2014). A moving-average filter based hybrid ARIMA – ANN model for forecasting time series data. *Applied Soft Computing Journal*, *23*, 27–38.

Beaumont, A. N. (2014). Data transforms with exponential smoothing methods of forecasting. *International Journal of Forecasting*, *30*(4), 918–927.

Bennett, C. J., Stewart, R. a., & Lu, J. W. (2014). Forecasting low voltage distribution network demand profiles using a pattern recognition based expert system. *Energy*, *67*, 200–212.

Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2008). *Thesis Projects*.

Chatfield, C. (2000). *Time Series Forecasting*.

Chisholm, A. (2013). *Exploring Data with RapidMiner*.

Dong, Z., Yang, D., Reindl, T., & Walsh, W. M. (2013). Short-term solar irradiance forecasting using exponential smoothing state space model. *Energy*, *55*, 1104–1113.

Eisler, R. (2004). *Biogeochemical, Health, and Ecotoxicological Perspectives on Gold and Gold Mining*.

Gorunescu. (2011). *Data Mining Concept Model Technique*.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*.

He & Xu, shaohua. (2009). *Process Neural Network*.

Hofmann, M. (2009). *Data Mining and Knowledge Discovery Series*.

Hyndman, Koehlr, Ord, S. (2008). *Springer Series in Statistics Forecasting with Exponential Smoothing*.

Jammazi, R., & Aloui, C. (2012). Crude oil price forecasting: Experimental evidence from wavelet decomposition and neural network modeling. *Energy Economics*, *34*(3), 828–841.

Kirchgässner, G., & Wolters, J. (2007). *Introduction to Modern Time Series Analysis*.

Ko, C.-N., & Lee, C.-M. (2013). Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter. *Energy*, *49*, 413–422.

Larose, D. T. (2006). *Data Mining Methods and Models*. *Data Mining Methods and Models*.

Lavrac, N., & Zupan, B. (2006). Data mining in medicine. In *Data Mining and Knowledge Discovery Handbook* (pp. 21–36).

Liao, G. (2014). Electrical Power and Energy Systems Hybrid Improved Differential Evolution and Wavelet Neural Network with load forecasting problem of air conditioning. *IInternational Journal of Electrical Power and Energy Systems*, *61*, 673–682.

Lu, C.-J., Lee, T.-S., & Lian, C.-M. (2012). Sales forecasting for computer wholesalers: A comparison of multivariate adaptive regression splines and artificial neural networks. *Decision Support Systems*, *54*(1), 584–596.

Montgomery, D. C. (2008). *Introduction to Time Series Analysis and Forecasting*.

Ouyang, Y., & Yin, H. (2014). A neural gas mixture autoregressive network for modelling and forecasting FX time series. *Neurocomputing*, *135*, 171–179.

Pierdzioch, C., Risse, M., & Rohloff, S. (2014). On the efficiency of the gold market: Results of a real-time forecasting approach. *International Review of Financial Analysis*, *32*, 95–108.

Pulido, M., Melin, P., & Castillo, O. (2014). Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. *Information Sciences*, *280*, 188–204.

Sbrana, G., & Silvestrini, A. (2014). Int . J . Production Economics Random switching exponential smoothing and inventory forecasting. *Intern. Journal of Production Economics*, *156*, 283–294.

Shumway, R. H., Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2006). *Time Series Analysis and Its Applications*. *Design*.

Tratar, L. F. (2015). Int . J . Production Economics. *Intern. Journal of Production Economics*, *161*, 64–73.

Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*.

Yager, R. R. (2013). Exponential smoothing with credibility weighted observations. *Information Sciences*, *252*, 96–105.

Yu, F., & Xu, X. (2014). A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network. *Applied Energy*, *134*, 102–113.

Zhang, G. P. (2004). *Neural Networks in Business Forecasting*. (G. P. Zhang, Ed.)*Review of Economic Sciences* (Vol. 6). IGI Global.

Copyright @ 2015 IlmuKomputer.Com

Zhou, S., Lai, K. K., & Yen, J. (2012). A dynamic meta-learning rate-based model for gold market forecasting. *Expert Systems with Applications*, *39*(6), 6168–6173.

BIOGRAFI PENULIS

Indah Suryani. Menempuh pendidikan Strata 1 Sistem Informasi dan Strata 2 Magister Ilmu Komputer di Pasca Sarjana STMIK Nusa Mandiri Jakarta. Saat ini menjadi salah satu staf pengajar di salah satu perguruan tinggi swasta di Indonesia. Minat penelitian pada bidang data mining.

Romi Satria Wahono. Memperoleh gelar B.Eng adan M.Eng pada bidang Ilmu Komputer di Universitas Saitama, Jepang, dan Ph.D pada bidang Software Engineering di Universiti Teknikal Malaysia, Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT. Brainmatics, sebuah perusahaan yang bergerak dibidang pengembangan software. Minat penelitian pada bidang Software Engineering dan Machine Learning. Tergabung sebagai anggota profesional dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Integrasi Metode *Sample Bootstrapping* dan *Weighted* Principal Component Analysis untuk Meningkatkan Performa k Nearest Neighbor pada Dataset Besar

Tri Agus Setiawan, Romi Satria Wahono dan Abdul Syukur
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
tri.triagus.setiawan45@gmail.com, romi@romisatriawahono.net, abah.syukur01@gmail.com

Abstract: Algoritma k Nearest Neighbor (kNN) merupakan metode untuk melakukan klasifikasi terhadap objek baru berdasarkan k tetangga terdekatnya. Algoritma kNN memiliki kelebihan karena sederhana, efektif dan telah banyak digunakan pada banyak masalah klasifikasi. Namun algoritma kNN memiliki kelemahan jika digunakan pada dataset yang besar karena membutuhkan waktu komputasi cukup tinggi. Pada penelitian ini integrasi metode *Sample Bootstrapping* dan *Weighted* Principal Component Analysis (PCA) diusulkan untuk meningkatkan akurasi dan waktu komputasi yang optimal pada algoritma kNN. Metode *Sample Bootstrapping* digunakan untuk mengurangi jumlah data training yang akan diproses. Metode *Weighted* PCA digunakan dalam mengurangi atribut. Dalam penelitian ini menggunakan dataset yang memiliki dataset *training* yang besar yaitu Landsat Satellite sebesar 4435 data dan Tyroid sebesar 3772 data. Dari hasil penelitian, integrasi algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA pada dataset Landsat Satellite akurasinya meningkat 0.77% (91.40%-90.63%) dengan selisih waktu 9 (1-10) detik dibandingkan algoritma kNN standar. Integrasi algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA pada dataset Thyroid akurasinya meningkat 3.10% (89.31%-86.21%) dengan selisih waktu 11 (1-12) detik dibandingkan algoritma kNN standar. Dari hasil penelitian yang dilakukan, dapat disimpulkan bahwa integrasi algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA menghasilkan akurasi dan waktu komputasi yang lebih baik daripada algoritma kNN standar.

Keywords: algoritma kNN, *Sample Bootstrapping*, *Weighted* PCA

1 PENDAHULUAN

Data mining merupakan suatu proses untuk mengidentifikasi pola yang memiliki potensi dan berguna untuk mengelola dataset yang besar (Witten, I. H., Frank, E., & Hall, 2011). Dalam data mining ada 10 algoritma teratas yang paling berpengaruh yang dipilih oleh peneliti dalam komunitas data mining, dimana 6 (enam) diantaranya adalah algoritma klasifikasi yaitu C4.5, Support Vector Machines (SVM), AdaBoost, k Nearest Neighbor (kNN), Naïve Bayes dan CART (Fayed & Atiya, 2009).

Salah satu algoritma yang banyak diteliti adalah algoritma klasifikasi kNN (Wan, Lee, Rajkumar, & Isa, 2012). Algortima kNN merupakan sebuah metode untuk melakukan klasifikasi terhadap objek baru berdasarkan (k) tetangga terdekatnya (Witten, I. H., Frank, E., & Hall, 2011)(Amores, 2006)(Morimune & Hoshino, 2008). Tujuan dari algoritma kNN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan training sample (Morimune & Hoshino, 2008)(Han, J., & Kamber, 2012), dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada kNN.

Algoritma kNN memiliki kelebihan karena sederhana, efektif dan telah banyak digunakan pada banyak masalah klasifikasi (Wu, Xindong & Kumar, 2009). Namun algoritma kNN memiliki kelemahan jika digunakan pada database yang besar karena membutuhkan waktu komputasi cukup tinggi (Fayed & Atiya, 2009)(Wan et al., 2012)(Neo & Ventura, 2012). Adapun dataset yang besar berupa volume yang banyak, label data yang banyak, kecepatan tinggi, dana / atau aset informasi yang membutuhkan bentuk-bentuk baru dari pengolahan untuk pengambilan keputusan, penemuan wawasan dan optimasi proses (O'Reilly, 2012)(Zikopoulos, Eaton, & DeRoos, 2012).

Beberapa peneliti telah melakukan penelitian tentang pengurangan jumlah data dan waktu komputasi. Penelitian Fayed (Fayed & Atiya, 2009) menggunakan pendekatan *Novel Template Reduction* yang digunakan untuk membuang nilai yang jauh dari batasan *threshold* dan memiliki sedikit pengaruh pada klasifikasi kNN. Penelitian Wan (Wan et al., 2012) menggunakan Support Vector Machines-Nearest Neighbor (SVM-NN) dengan pendekatan klasifikasi *hybrid* dengan tujuan bahwa untuk meminimalkan dampak dari akurasi klasifikasi. Penelitian Koon (Neo & Ventura, 2012) menggunakan algoritma *Direct Boosting* untuk meningkatkan akurasi klasifikasi kNN dengan modifikasi pembobotan jarak terhadap data latih.

Oleh karena itu perlu adanya metode untuk mengurangi jumlah *data training* untuk diproses dan mengurangi atribut sehingga mampu meningkatkan akurasi dan meminimalkan waktu komputasi.

Metode *Sample Bootstrapping* digunakan untuk mengurangi jumlah data *training* yang akan diproses (Dudani, 1976)(Amores, 2006). Untuk dapat mengatasi dataset yang besar maka perlu perlu adanya sampel data (*sampling*) secara acak agar data yang akan diproses menjadi lebih kecil (Liaw, Wu, & Leou, 2010)(Morimune & Hoshino, 2008), sedangkan untuk mengukur jarak tetangga terdekat digunakan *euclidian distance* (Han, J., & Kamber, 2012) dalam proses klasifikasi.

Dalam menentukan waktu komputasi dalam proses klasifikasi kNN yang akan dicari adalah nilai mayoritas sehingga dapat dihitung nilai *query instance*, pada tahapan ini semakin banyak nilai mayoritas data yang tidak dekat dan tidak relevan maka akan mengakibatkan proses klasifikasi kategori *nearest neighbor* semakin lama dan proses komputasi tidak dapat optimal (Larose, 2005). Untuk mengatasi masalah tersebut maka data yang tidak penting ataupun relevan harus dieleminasi sehingga waktu komputasi dan *error* dapat dikurangi (Han, J., & Kamber, 2012). Adapun untuk mengurangi atribut dalam mengolah data yang besar maka dapat menggunakan metode Principal Component Analysis (PCA) (Neo & Ventura, 2012)(Han, J., & Kamber, 2012).

Copyright @ 2015 IlmuKomputer.Com

Namun PCA memiliki kekurangan dalam kemampuan memilih fitur yang tidak relevan dari dataset (Kim & Rattakorn, 2011), karena bisa saja fitur yang dibuang ternyata adalah fitur yang berpengaruh. Untuk mengatasi masalah tersebut maka dapat dilakukan seleksi fitur dengan melakukan pembobotan atribut yaitu *Weighted* PCA (Kim & Rattakorn, 2011)(Liu & Wang, 2012) berdasarkan nilai *threshold*, dimana fitur yang nilainya kurang dari batas *treshold* akan dieliminasi. Dengan menggunakan metode Weighted PCA dapat mengurangi waktu komputasi (Kim & Rattakorn, 2011) sehingga efisien untuk menangani dataset yang memiliki dimensi yang tinggi.

Dari penelitian yang sudah dilakukan belum ditemukan model yang menggunakan kombinasi pengurangan jumlah data *training* dan pengurangan atribut dalam proses klasifikasi kategori *nearest neighbor*. Oleh karena itu, akan dilakukan integrasi metode *Sample Bootstrapping* dengan *Weighted* PCA sehingga mampu meningkatkan akurasi dan waktu komputasi yang optimal pada algoritma kNN.

Dalam penulisan ini dibagi menjadi beberapa bagian. Pada bagian 2, menjelaskan tentang peneltian terkait.Pada bagian 3, menjelaskan metode yang diusulkan. Hasil penelitian dan dan pembahasan mengenai komparasi metode yang diusulkan dijelaskan dalam bagian 4. Penutup, pada bagian ini akan menjelaskan tentang kesimpulan dan saran dari penelitian.

2 PENELITIAN TERKAIT

Dalam penelitian yang dilakukan oleh Fayed et al. (Fayed & Atiya, 2009) menggunakan pendekatan *Novel Template Reduction* yang digunakan untuk membuang nilai yang jauh dari batasan *threshold* dan memiliki sedikit pengaruh pada klasifikasi kNN. Adapun untuk pengujian waktu proses klasifikasi menggunakan metode *condensed set* dengan melakukan pengurangan terhadap data yang tidak terpakai sehingga dapat meningkatkan akurasi

Adapun penelitian yang dilakukan oleh Wan et al. menyajikan pendekatan klasifikasi *hybrid* dengan menggabungkan algoritma Support Vector Machine (SVM) dan algoritma kNN pada ketergantungan parameter yang rendah (Wan et al., 2012), untuk mendapatkan akurasi terbaik dengan menggunakan training dataset yang besar. Dalam model *hybrid* SVM-kNN, SVM digunakan untuk mengurangi data training ke Support Vectors (SVs) dari masing-masing kategori, dan algoritma *nearest neighbor*, kemudian digunakan untuk menghitung jarak rata-rata antara pengujian titik data ke set SVs dari kategori yang berbeda. Langkah selanjutnya menentukan kategori data baru yang tidak berlabel berdasarkan jarak rata-rata terpendek antara SVs kategori dan titik data baru, kemudian menghitung jarak rata-rata untuk masing-masing kategori dengan menggunakan rumus *euclidean distance*.

Pada penelitian yang dilakukan Konn at el (Neo & Ventura, 2012) menyajikan pendekatan menggunakan algoritma *Direct Boosting* untuk meningkatkan akurasi klasifikasi kNN dengan modifikasi pembobotan jarak terhadap data latih dengan *local warping of distance matric*. Metode *local warping of the distance matric* digunakan untuk merubah bobot jarak setiap data latih, kemudian memodifikasi klasifikasi kNN dengan memberi bobot jarak 1/d untuk mengklasifikasikan setiap data latih menggunakan sisanya setiap iterasi. Dalam melakukan validasi sehingga menghasilkan akurasi terbaik menggunakan metode 10-*fold cross validation* untuk setiap melakukan iterasi melakukan validasi sehingga menghasilkan akurasi terbaik menggunakan metode 10-*fold cross validation* untuk setiap melakukan iterasi.

Dalam penelitian ini kita akan melakukan perbaikan metode dengan melakukan integrasi metode *Sample Bootstrapping* dan *Weighted* Principal Component Analysis (PCA) diusulkan untuk meningkatkan akurasi dan waktu komputasi yang optimal pada algoritma kNN. Metode *Sample Bootstrapping* digunakan untuk mengurangi jumlah data *training* yang akan diproses. Metode *Weighted* Principal Component Analysis (PCA) digunakan untuk mengurangi atribut. Untuk pengujian akurasi hasil klasikasi dilakukan menggunakan metode *confusion matrix* (Witten, I. H., Frank, E., & Hall, 2011)(Maimon Oded, 2010) dan uji efisiensi (lamanya waktu proses klasikasi) dinyatakan dalam waktu (detik).

3 METODE YANG DIUSULKAN

Untuk melakukan penelitian ini menggunakan spesifikasi komputer Intel Core i5-2557M 1.7GHz, RAM 2 GB, operating system Microsoft Windows 7 Home Premium.Untuk pengembangan sistem menggunakan Rapid Miner 5.3.015.

Proses eksperimen dan pengujian model menggunakan bagian dari dataset yang ada. Data yang digunakan dalam penelitian ini menggunakan dataset Landsat Satellite dan Thyroid, hal ini berdasarkan penelitian-penelitian sebelumnya (Fayed & Atiya, 2009)(Wan et al., 2012)(Neo & Ventura, 2012) tentang kNN menggunakan datataset tersebut seperti pada Tabel.1.

Tabel 1 Dataset yang Digunakan di Penelitian

No	Name	Type	Record	Dimension	Class
1	Landsat Satellite	Classification	6435	36	6
2	Thyroid	Classification	7200	21	3

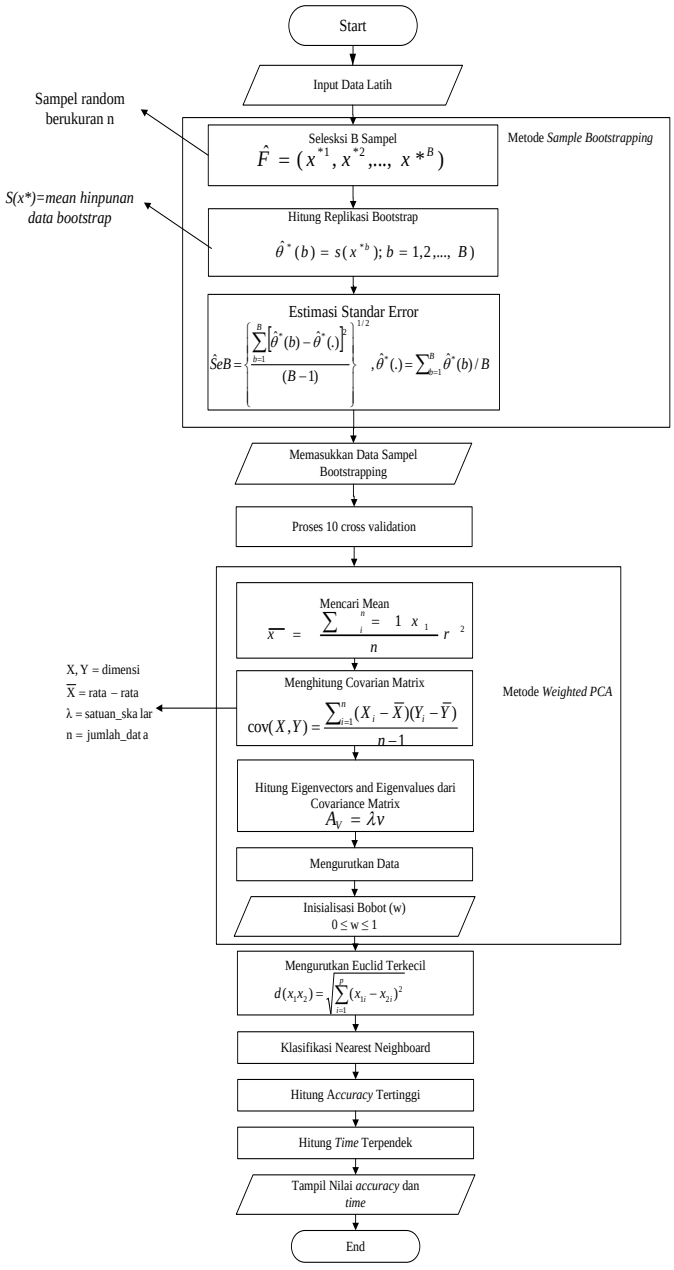
Metode yang diusulkan dalam penelitian ini yaitu dengan melakukan integrasi *Sample Bootstrapping* dan *Weighted* PCA dalam meningkatkan akurasi dan menentukan waktu komputasi pada algoritma kNN. Metode *Sample Bootstrapping* digunakan untuk mengurangi jumlah data *training* yang akan diproses (Dudani, 1976)(Amores, 2006), sedangkan untuk mengurangi jumlah atribut dalam mengolah data yang besar maka dapat menggunakan metode *Weighted* Principal Component Analysis (PCA) (Neo & Ventura, 2012)(Han, J., & Kamber, 2012) karena mampu mereduksi untuk data yang memiliki dimensi tinggi. Adapun tahapan eksperimen pada penelitian ini adalah:

1. Menyiapkan dua dataset untuk eksperimen yang diambil dari University of California, Irvine (UCI)
2. Melakukan pengujian menggunakan algoritma kNN menggunakan dataset Landset Satellite dan Thyroid kemudian dari hasil pengujian tersebut dicatat hasil yang diperoleh
3. Melakukan pengujian menggunakan algoritma kNN dengan *Sample Bootstrapping* menggunakan dataset Landsat Satellite dan Thyroid kemudian dari hasil pengujian tersebut dicatat hasil yang diperoleh
4. Melakukan pengujian menggunakan algoritma kNN dengan Weighted PCA menggunakan dataset Landsat Satellite dan Thyroid kemudian dari hasil pengujian tersebut dicatat hasil yang diperoleh
5. Melakukan pengujian menggunakan algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA menggunakan dataset Landsat Satellite dan Thyroid

Copyright @ 2015 IlmuKomputer.Com

kemudian dari hasil pengujian tersebut dicatat hasil yang diperoleh
6. Membandingkan hasil akurasi terbaik dan waktu komputasi minimal dan mengambil hasil terbaik
7. Mengintegrasikan hasil algoritma klasifikasi terbaik.

Adapun algoritma yang diusulkan dalam penelitian ini seperti pada Gambar 1, diawali dengan memasukkan dataset baik *data training* maupun *data testing*, kemudian melakukan transformasi dimana metode *Sample Bootstrapping* digunakan untuk mengurangi jumlah data *training* yang akan diproses kemudian menghitung validitas data *training*, setelah itu menghitung kuadrat jarak *euclidian* (*euclidean distance*) masing-masing objek terhadap data sampel yang diberikan. Kemudian menghitung nilai *distance weighted* yang didapat dari memasukkan nilai validitas dan nilai *eucledian*, setelah melakukan pembobotan atribut dan diperoleh klasifkasi *nearest neighbor*. Metode *Weighted* Principal Component Analysis (PCA) digunakan untuk mengurangi atribut.



Gambar 1 Algoritma *Sample Bootstrapping* Weighted PCA

4 HASIL PENELITIAN

Dalam penelitian ini akan dilakukan komparasi antara algoritma kNN dengan algoritma kNN dan *Sample Bootstrapping* dan *Weighted* PCA. Metode *Sample Bootstrapping* digunakan untuk mengurangi jumlah data *training* yang akan diproses dan metode *Weighted* Principal Component Analysis (PCA) digunakan untuk mengurangi atribut serta mengitegrasikan metode *Sample Bootstrapping* dan *Weighted* PCA diusulkan untuk meningkatkan akurasi dan waktu komputasi yang optimal pada algoritma *kNN* pada dataset Thyroid dan Landsat Satellite.

Pada eksperimen pertama akan melakukan perhitungan menggunakan algoritma kNN dengan dataset Thyroid dan Thyroid.Adapun proses perhitungan kNN sebagai berikut:
1. Menyiapkan dataset Thyroid dan Landsat Satellite, kita lakukan validasi dengan *cross validation* dimana dataset kita bagi menjadi data *training* dan data *testing*
2. Menentukan nilai*k*, pada penentuan *k* dilakukan input antara 1…7200 (dataset Thyroid) dan 1…6435 (dataset Lansat Satellite)
3. Menghitung kuadrat jarak euclid (*query instance*) masing-masing objek terhadap sampel data yang diberikan dengan menggunakan *euclidian distance* denga parameter *numeri*c dengan rumus:

$$d_i = \sqrt{\sum_{1=i}^{p}(x_{1i} - x_{2i})^2}$$

4. Mengurutkan objek-objek termasuk ke dalam kelompok yang mempunyai jarak euclid terkecil
5. Menghitung Akurasi
   Untuk menghitung nilai akurasi digunakan *confusion matrix* dengan rumus:

$$\text{akurasi} = \frac{Jumlah\ Data\ Benar}{Jumlah\ Data} \text{X } 100\%$$

Hasil perhitungan yang dilakukan untuk algoritma kNN mendapatkan nilai akurasi terbaik pada k=1 dengan waktu komputasi minimal adalah 1 detik baik untuk dataset Thyroid maupun Landsat Satellite seperti pada Tabel 2.

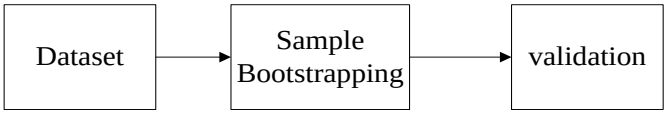
Tabel 2 Akurasi Algoritma kNN Dengan Dataset Thyroid dan Landsat Satellite

Dataset	Nilai Akurasi (%)	Waktu (detik)
Thyroid	86.21	10
Landsat Satellite	90.63	12

Pada eksperimen kedua akan melakukan perhitungan menggunakan algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA pada dataset Thyroid dan Landsat Satellite. Adapun proses yang dilakukan adalah:

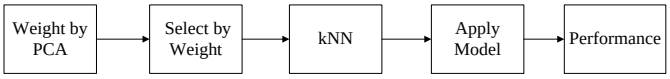
1. Melakukan preprocessing menggunakan metode *sampling*. Algoritma yang dipakai yaitu *Sample Bootstrapping*, kemudian memilih parameter *sample* yaitu *relative* (sampel dibuat sebagai sebagian kecil dari jumlah total contoh dalam sampel data) dan nilai sampel rasio yang diinput antara 0-1. Setelah dilakukan *sampling* maka data *bootstrap* tersebut divalidasi dengan *cross validation* sebagaimana ditunjukkan dalam Gambar 2.

Copyright @ 2015 IlmuKomputer.Com



Gambar 2 Pengujian Performa Algoritma kNN dengan *Sample Bootstrapping* untuk Dataset Thyroid

2. Langkah berikutnya yaitu melakukan normalisasi terhadap *attribute class* pada dataset dan melakukan pembobotan terhadap *attribute class* dengan *weighted relation*. *Weighted relation* mencerminkan relevansi bobot atribut dengan nilai attribut class 0 sampai 1.0, pada penelitian ini bobot atribut diisi, kemudian menentukan k. Adapun proses Weighted PCA kNN seperti pada Gambar 3.



Gambar 3 Pengujian Performa Algoritma kNN dengan Weighted PCA untuk dataset Thyroid

Dalam hal ini bobot atribut dan nilai k sangat berperan dalam memdapatkan akurasi dan waktu yang baik. Nilai akurasi dan waktu yang optimal dari *confusion matrix* tersebut seperti dalam Tabel 3 dengan rumus:

$$\text{akurasi} = \frac{Jumlah\ Data\ Benar}{Jumlah\ Data} \text{X } 100\%$$

Tabel 3 Hasil Akurasi dan Waktu KomputasiAlgoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA Pada Dataset Thyroid dan Landsat Satellite

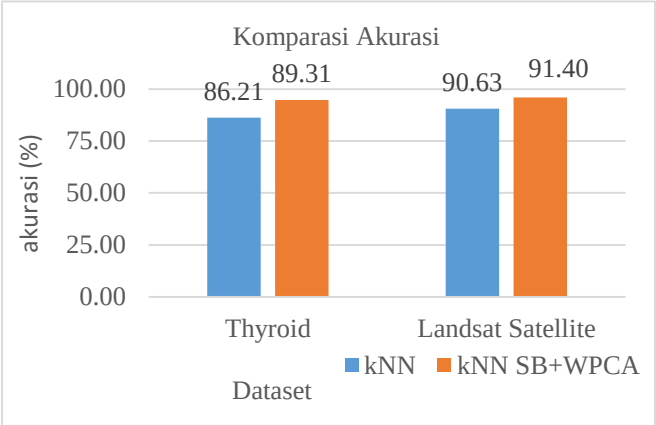
Dataset	Nilai Akurasi (%)	Waktu (detik)
Thyroid	89.31	1
Landsat Satellite	91.40	1

Dari hasil eksperimen tentang nilai akurasi yang dilakukan antara algoritma kNN dengan kNN dengan *Sample Bootstrapping* dan *Weighting* PCA dapat meiningkatkan akurasi (Champagne, Mcnairn, Daneshfar, & Shang, 2014)(Ghaderyan, Abbasi, & Hossein, 2014)(Polat & Kara, 2008) untuk dataset Thyroid dan Lansat Satellite. Pada algoritma kNN data yang digunakan sejumlah data secara keseluruhan tidak ada proses *filtering* maupun sampel data yang digunakan sehingga membutuhkan waktu yang lama sehingga tingkat akurasi menjadi rendah, sedangkan pada algoritma kNN dan *Sample Boostrapping* dan *Weighted* PCA data yang digunakan tidak keseluruhan tetapi dilakukan menggunakan data *sampling* (Witten, I. H., Frank, E., & Hall, 2011) untuk melakukan *filtering* agar mengurangi jumlah data sampel (Champagne et al., 2014)(McRoberts, Magnussen, Tomppo, & Chirici, 2011)(Chen & Samson, 2015).

Dalam metode *Sample Bootstrapping* terdapat rasio parameter *sample* yang berfungsi memberikan nilai jumlah data sample yang digunakan dari seluruh data yang ada dengan nilai 0-1. Dengan metode ini jumlah data yang diproses tidak secara keseluruhan melainkan beberapa data tetapi tidak mengurangi jumlah data yang ada karena setelah data tersebut digunakan maka akan dikembalikan lagi (Tian, Song, Li, & Wilde, 2014). Dari hasil perhitungan dapat dilihat perbandingan berdasarkan akurasi pada Tabel 4 dan Gambar 4. Pada tabel dan gambar ini didapat hasil dimana integrasi algoritma kNN dengan *Sample Bootstrapping* dan *Weighting* PCA mempunyai nilai akurasi yang lebih baik yaitu 3.10% untuk dataset Thyroid dan 0.77% untuk dataset Landsat Satellite

Tabel 4 Komparasi Akurasi Algoritma kNN Dengan *Sample Bootstrapping* dan Weighted PCA (kNN SB+WPCA) Pada Dataset Thyroid dan Landsat Satellite

Dataset	Nilai Akurasi (dalam %)		Kenaikan Akurasi (dalam %)
	kNN	*(kNN SB+WPCA)*	
Thyroid	86.21	89.31	3.10
Landsat Satellite	90.63	91.40	0.77



Gambar 4 Komparasi Akurasi Algoritma kNN Dengan *Sample Bootstrapping* dan Weighted PCA Pada Dataset Thyroid dan Landsat Satellite

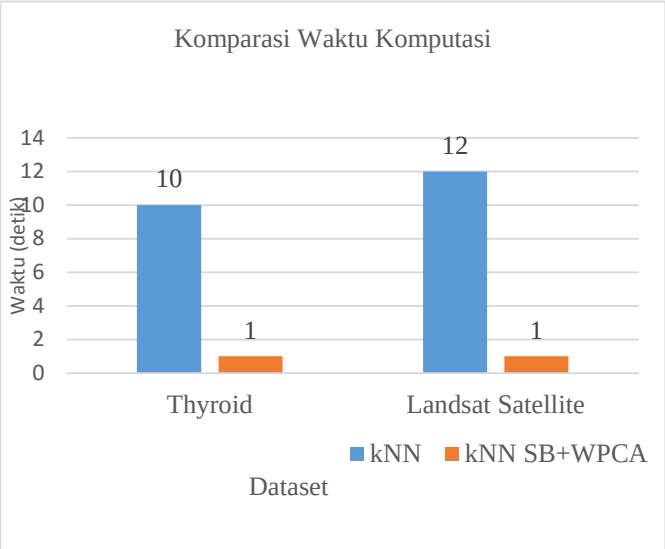
Dari hasil eksperimen tentang waktu komputasi yang dilakukan antara algoritma kNN dengan kNN dan *Sample Bootstrapping* dan *Weighting* PCA untuk dataset Thyroid keduanya sama-sama menggunakan metode *confusion matrix* untuk pengujian akurasi hasil klasikasi (Witten, I. H., Frank, E., & Hall, 2011)(Maimon Oded, 2010) dan uji efisiensi (lamanya waktu proses klasikasi) dinyatakan dalam *waktu* (detik).

Adapun untuk mengurangi jumlah waktu dan memori yang dibutuhkan maka digunakan metode Principal Componen Analysis (PCA) (Amores, 2006)(Morimune & Hoshino, 2008)(Ghaderyan et al., 2014) karena mampu mengurangi atribut pada data yang besar (Han, J., & Kamber, 2012)(Polat & Kara, 2008). Pada algoritma *Sample Bootstrapping* dan *Weighted* PCA diberikan bobot terhadap atribut *class* dengan nilai 0-1, ketika bobot atribut kurang dari nilai *threshold* maka akan dibuang sehingga dapat meningkatkan waktu komputasi. Dalam penelitian ini bobot diinputkan dengan nilai 0-1 dan nilai k. Pada Tabel 5 dan Gambar 5 didapat hasil komparasi antara algoritma kNN dengan algoritma kNN dan *Sample Bootstrapping* dan *Weighted* PCA.

Tabel 5 Komparasi Waktu Komputasi Algoritma kNN Dengan *Sample Bootstrapping* dan Weighted PCA (kNN SB+WPCA) Pada Dataset Thyroid dan Landsat Satellite

Dataset	Waktu Komputasi (detik)		Selisih Waktu Komputasi (detik)
	kNN	*(kNN SB+WPCA)*	
Thyroid	10	1	9
Landsat Satellite	12	1	11

Copyright @ 2015 IlmuKomputer.Com



Gambar 5 Komparasi Waktu Komputasi Algoritma kNN Dengan *Sample Bootstrapping* dan *Weighted* PCA Pada Dataset Thyroid dan Landsat Satellite

5 KESIMPULAN

Integrasi algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA untuk dataset Thyroid ada kenaikan akurasi sebesar 3.10% (89.31-86.21%) dengan menggunakan sampel *data training* sebesar 2160 dan selisih waktu komputasi 9 (1-10) detik dengan pengurangan atribut sebanyak 1 atribut, sedangkan untuk dataset Landsat Satellite ada kenaikan akurasi sebesar 0.77% (91.40-90.63%) dengan menggunakan sampel *data training* sebesar 1931 dan selisih waktu komputasi 11 (1-12) detik dengan pengurangan atribut sebanyak 27 atribut. Dari hasil penelitian tersebut dapat disimpulkan bahwa integrasi algoritma kNN dengan *Sample Bootstrapping* dan *Weighted* PCA dapat meningkatkan akurasi dan mengurangi waktu komputasi dibandingkan dengan algoritma kNN standar.

REFERENCES

Amores, J. (2006). Boosting the distance estimation Application to the K -Nearest Neighbor Classifier. *Pattern Recognition Letters*, *27*(d), 201–209. doi:10.1016/j.patrec.2005.08.019

Champagne, C., Mcnairn, H., Daneshfar, B., & Shang, J. (2014). A bootstrap method for assessing classification accuracy and confidence for agricultural land use mapping in Canada. *International Journal of Applied Earth Observations and Geoinformation*, *29*, 44–52. doi:10.1016/j.jag.2013.12.016

Chen, X., & Samson, E. (2015). Environmental assessment of trout farming in France by life cycle assessment : using bootstrapped principal component analysis to better de fi ne system classification. *Journal of Cleaner Production*, *87*, 87–95. doi:10.1016/j.jclepro.2014.09.021

Dudani, S. a. (1976). The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-6*(4), 325–327. doi:10.1109/TSMC.1976.5408784

Fayed, H. A., & Atiya, A. F. (2009). A Novel Template Reduction Approach for the -Nearest Neighbor Method. *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, *20*(5), 890–896.

Ghaderyan, P., Abbasi, A., & Hossein, M. (2014). An efficient seizure prediction method using KNN-based undersampling and linear frequency measures. *Journal of Neuroscience Methods*, *232*, 134–142. doi:10.1016/j.jneumeth.2014.05.019

Han, J., & Kamber, M. (2012). *Data Mining Concepts and Techniques*. (M. Han, J., & Kamber, Ed.) (Third Edit.). USA: Morgan Kaufmann Publishers.

Kim, S. B., & Rattakorn, P. (2011). Unsupervised feature selection using weighted principal components. *Expert Systems with Applications*, *38*(5), 5704–5710. doi:10.1016/j.eswa.2010.10.063

Larose, D. T. (2005). *Discovering Knowledge In Data*. USA: John Wiley & Sons, Inc. New York, NY, USA.

Liaw, Y.-C., Wu, C.-M., & Leou, M.-L. (2010). Fast k-nearest neighbors search using modified principal axis search tree. *Digital Signal Processing*, *20*(5), 1494–1501. doi:10.1016/j.dsp.2010.01.009

Liu, N., & Wang, H. (2012). Weighted principal component extraction with genetic algorithms. *Applied Soft Computing Journal*, *12*(2), 961–974. doi:10.1016/j.asoc.2011.08.030

Maimon Oded, R. L. (2010). *Data Mining And Knowledge Discovery Handbook*. (R. L. Maimon Oded, Ed.) (Second Edi.). Israel: Springer.

McRoberts, R. E., Magnussen, S., Tomppo, E. O., & Chirici, G. (2011). Parametric, bootstrap, and jackknife variance estimators for the k-Nearest Neighbors technique with illustrations using forest inventory and satellite image data. *Remote Sensing of Environment*, *115*(12), 3165–3174. doi:10.1016/j.rse.2011.07.002

Morimune, K., & Hoshino, Y. (2008). Testing homogeneity of a large data set by bootstrapping. *Mathematics And Computers In Simulation*, *78*, 292–302. doi:10.1016/j.matcom.2008.01.021

Neo, T. K. C., & Ventura, D. (2012). A direct boosting algorithm for the k-nearest neighbor classifier via local warping of the distance metric. *Pattern Recognition Letters*, *33*(1), 92–102. doi:10.1016/j.patrec.2011.09.028

O'Reilly. (2012). *Big Data Now: 2012 Edition* (First Edit.). O'Reilly Media, Inc.

Polat, K., & Kara, S. (2008). Medical diagnosis of atherosclerosis from Carotid Artery Doppler Signals using principal component analysis (PCA), k -NN based weighting pre-processing and Artificial Immune Recognition System (AIRS). *Elsevier Inc.*, *41*, 15–23. doi:10.1016/j.jbi.2007.04.001

Tian, W., Song, J., Li, Z., & Wilde, P. De. (2014). Bootstrap techniques for sensitivity analysis and model selection in building thermal performance analysis. *Applied Energy*, *135*, 320–328. doi:10.1016/j.apenergy.2014.08.110

Wan, C. H., Lee, L. H., Rajkumar, R., & Isa, D. (2012). A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Systems with Applications*, *39*(15), 11880–11888. doi:10.1016/j.eswa.2012.02.068

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining*. (M. A. Witten, I. H., Frank, E., & Hall, Ed.) (Third Edit.). USA: Morgan Kaufmann Publishers.

Wu, Xindong & Kumar, V. (2009). *The Top Ten Algorithms in Data Mining*. (V. Wu, Xindong & Kumar, Ed.). USA: Taylor & Francis Group.

Zikopoulos, P., Eaton, C., & DeRoos, D. (2012). *Understanding big data*. *New York et al: McGraw …*. Mc Graw Hill. doi:1 0 9 8 7 6 5 4 3 2 1

BIOGRAFI PENULIS

Tri Agus Setiawan. Menyelesaikan pendidikan S1 Sistem Informasi di Universitas Dian Nuswantoro Semarang, S2 Magister Teknik Informatika di Universitas Dian Nuswantoro Semarang. Saat ini menjadi dosen Politeknik Pusmanu Pekalongan. Minat penelitian saat ini adalah softcomputing.

Copyright @ 2015 IlmuKomputer.Com

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiai ilmiah ACM, PMI dan IEEE Computer Society.

Abdul Syukur. Menerima gelar sarjana di bidang Matematika dari Universitas Diponegoro Semarang, gelar master di bidang manajemen dari Universitas Atma Jaya Yogyakarta, dan gelar doktor di bidang ekonomi dari Universitas Merdeka Malang. Dia adalah dosen dan dekan di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia. Minat penelitiannya saat ini meliputi decision support systems dan information management systems.

Copyright @ 2015 IlmuKomputer.Com

Optimasi Parameter pada Support Vector Machine Berbasis Algoritma Genetika untuk Estimasi Kebakaran Hutan

Hani Harafani
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
haniharafani@gmail.com

Romi Satria Wahono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
romi@romisatriawahono.net

Abstract: Kebakaran hutan merupakan salah satu masalah lingkungan yang mengancam hutan, menimbulkan dampak negatif pada lingkungan, menciptakan masalah ekonomi, dan kerusakan ekologis, serta menyebabkan kerugian penting di seluruh dunia setiap tahunnya. Estimasi area yang terbakar penting dilakukan, karena area yang terbakar dapat mencerminkan berapa kuat radiasi api pada vegetasi disekitarnya. SVM dapat mengatasi masalah klasifikasi dan regresi linier ataupun nonlinier kernel yang dapat menjadi satu kemampuan algoritma pembelajaran untuk klasifikasi serta regresi. Namun, SVM juga memiliki kelemahan yaitu sulitnya menentukan nilai parameter yang optimal. Untuk menyelesaikan permasalahan tersebut algoritma genetika diusulkan untuk diterapkan sebagai algoritma pencarian nilai parameter yang efisien pada SVM. Beberapa eksperimen dilakukan untuk menghasilkan estimasi yang akurat. Awalnya percobaan dilakukan pada kernel –kernel SVM (dot, RBF, polynomial) untuk menentukan kernel mana yang akan digunakan, kemudian model SVM+GA juga dibandingkan dengan model regresi lainnya seperti Linear Regression, k-NN, dan Neural Network. Berdasarkan eksperimen dengan 10 kombinasi parameter pada metode SVM dan SVM+GA dengan kernel dot, RMSE terkecil dihasilkan oleh model SVM+GA sebesar 1.379, sementara pada percobaan SVM dan SVM+GA dengan kernel polynomial RMSE terkecil diperoleh model SVM+GA sebesar 1.379, sedangkan pada percobaan SVM dan SVM+GA dengan kernel RBF diperoleh RMSE terkecil pada model SVM+GA sebesar 1.379.Selanjutnya berdasarkan perbandingan rata-rata RMSE, kernel RBF unggul dengan nilai RMSE terkecil yaitu 1.432 pada SVM, dan 1.418 pada SVM+GA. Pada perbandingan nilai rata-rata RMSE antara SVM(RBF)+GA dengan model lainnya, RMSE terkecil dihasilkan oleh SVM(RBF)+GA yaitu sebesar 1.418, disusul dengan model SVM(RBF) sebesar 1.432, keudian Linear Regression sebesar 1.459, dilanjutkan oleh model k-NN sebesar 1.526 dan yang terakhir adalah NN dengan nilai RMSE sebesar 1.559. maka dapat disimpulkan bahwa optimasi parameter yang dilakukan GA pada model SVM terbukti dapat mengurangi tingkat error pada model SVM tanpa optimasi parameter pada dataset *forestfire*, selain model SVM(RBF)+GA pada penelitian ini juga terbukti lebih baik dari model regresi lainnya.

Keywords: Estimasi, Kebakaran Hutan, Support Vector Machine, Algoritma Genetika, Optimasi Parameter.

1 PENDAHULUAN

Kebakaran hutan merupakan salah satu masalah lingkungan yang mengancam hutan, menimbulkan dampak negatif pada lingkungan, menciptakan masalah ekonomi, dan kerusakan ekologis (Özbayoğlu & Bozer, 2012), serta menyebabkan kerugian penting di seluruh dunia setiap tahunnya(Brun, Margalef, & Cortés, 2013). Kebakaran hutan terjadi karena beberapa hal diantaranya: pembakaran hutan yang disengaja (Denham, Wendt, Bianchini, Cortés, & Margalef, 2012), petir (Cortez & Morais, 2007), dan perubahan cuaca yang ekstrim (Eastaugh & Hasenauer, 2014), serta beberapa penyebab lainnya.

Estimasi area yang terbakar penting dilakukan, karena area yang terbakar dapat mencerminkan berapa kuat radiasi api pada vegetasi disekitarnya (Quintano, Fernández-Manso, Stein, & Bijker, 2011), sehingga dapat memberikan informasi mengenai kerusakan lahan yang terjadi. Namun, metode estimasi konvensional yang dilakukan oleh banyak peneliti berdasarkan *Thresholding* menghasilkan nilai estimasi yang akurat.

Ada beberapa studi yang dilakukan untuk mengestimasi lahan yang terbakar pada kebakaran hutan dengan menggunakan metode komputasi antara lain: *support vector machine* (SVM) (Cortez & Morais, 2007), dan *multi layer perceptron* (MLP) (Özbayoğlu & Bozer, 2012). Selain itu banyak juga metode regressi yang digunakan para peneliti dunia pada berbagai permasalahan estimasi seperti k-NN (Lee, Kang, & Cho, 2014), *linear regression* (LR) (Lira, Da Silva, Alves, & Veras, 2014), dan *neural network* (NN) (Tiryaki, Öz, & Y, 2014). Metode-metode tersebut sangat direkomendasikan oleh banyak peneliti di dunia.

Multilayer perceptron sebagai salah satu model yang paling populer dari *artificial neural network* (ANN) (Singh & Borah, 2014) memiliki kelebihan untuk menemukan pola dari data yang terlalu rumit untuk diketahui oleh manusia atau dengan teknik komputasi lainnya (Yilmaz & Kaynar, 2011). Selain itu MLP memiliki kekuragan yaitu sulit menemukan pola bila data berdimensi tinggi atau sering disebut dengan "kutukan dimensionalitas" (Pan, Iplikci, Warwick, & Aziz, 2012), dan *overfitting* (Rynkiewicz, 2012).

Support vector machine (SVM) memiliki keunggulan dibandingkan metode MLP yaitu: SVM dapat mengatasi masalah klasifikasi dan regresi dengan linier ataupun nonlinier kernel yang dapat menjadi satu kemampuan algoritma pembelajaran untuk klasifikasi serta regresi (Maimon & Rokach, 2010), dan baik untuk mengatasi kutukan dimensionalitas (Wang, Wen, Zhang, & Wang, 2014). SVM juga memiliki akurasi tinggi dan tingkat kesalahan yang relative kecil, kemampuan untuk mengatasi *overfitting* tidak membutuhkan data yang terlalu besar dan dapat digunakan untuk melakukan prediksi. Berdasarkan beberapa kelebihan SVM pada ulasan yang telah disebutkan, maka SVM cocok diterapkan untuk memprediksi kebakaran hutan. Selain SVM

Copyright @ 2015 IlmuKomputer.Com

memiliki banyak kemampuan, SVM juga memiliki kelemahan yaitu sulitnya menentukan nilai parameter yang optimal (Ilhan & Tezel, 2013; Raghavendra. N & Deka, 2014; M. Zhao, Fu, Ji, Tang, & Zhou, 2011).

Beberapa algoritma pun banyak direkomendasikan oleh peneliti dunia untuk mengoptimasi parameter pada machine learning, seperti: *particle swarm optimization* (PSO)(Wang et al., 2014), *simulated annealing* (SA) (Z.-Y. Jia, Ma, Wang, & Liu, 2010), dan *genetic algorithm* (GA) (Guo, Li, & Zhang, 2012).

Simmulated Annealing (SA) efektif pada pemuatan masalah optimasi pola, namun SA memiliki kecenderungan untuk terjebak dalam minimum lokal ketika suhu anil rendah (tingkat anil cepat) dan semakin tidak konvergen ketika suhu anil tinggi (tingkat anil lambat) (Zameer, Mirza, & Mirza, 2014), selain itu PSO juga sulit mendapatkan nilai yang optimum dalam mengoptimasi lebih dari sepuluh parameter. *Genetic algorithm* atau algoritma genetika dapat mengatasi masalah yang nonlinier dengan diskontinuitas dan minima lokal secara efisien, serta GA juga lebih efisien dalam mengoptimasi lebih dari sepuluh parameter (Machairas, Tsangrassoulis, & Axarli, 2014).

Dalam penelitian ini kami mengusulkan algoritma genetika (GA) untuk melakukan optimasi parameter pada *support vector machine* untuk meningkatkan akurasi dalam mengestimasi kebakaran hutan.

Paper ini disusun sebagai berikut: pada bagian 2 paper-paper terkait dijelaskan. Pada bagian 3, metode yang diusulkan disajikan. Hasil percobaan perbandingan antara metode yang diusulkan dengan metode lainnya disajikan pada bagian 4. Akhirnya, kesimpulan dari penelitian kami disajikan pada bagian terakhir.

2 PENELITIAN TERKAIT

Support Vector Machine (SVM) secara konseptual adalah mesin linier yang dilengkapi dengan fitur special (Gorunescu, 2011), dan didasarkan pada metode minimalisasi resiko struktural (Dua, 2011), serta teori pembelajaran statistik. Dua sifat khusus dari SVM yaitu (1) mencapai generalisasi yang tinggi dengan memaksimalkan margin, dan (2) mendukung pembelajaran yang efisien dari fungsi nonlinier pada trik kernel sehingga membuat kinerja generalisasinya baik dalam menyelesaikan masalah pengenalan pola (Gorunescu, 2011).

Untuk permasalahan klasifikasi SVM mencoba untuk mencari garis pemisah yang optimal yang diekspresikan sebagai kombinasi linier dari subset data pelatihan dengan menyelesaikan masalah keterbatasan linier pemrograman quadrat (QP) dengan margin maksimum antara dua kelas. Sementara untuk permasalahan regresi, Vapnik juga memperkenalkan fungsi $\varepsilon - insensitive\ loss$ yang disebut sebagai SVM untuk regresi.

Support Vector Regression (SVR) adalah metode untuk mengestimasi sebuah fungsi yang dipetakan dari objek input ke jumlah riil berdasarkan data pelatihan. Serupa dengan pengklasifikasian SVM, SVR memiliki properti yang sama tentang memaksimalkan margin dan trik kernel untuk memetakan data yang nonlinier. Secara singkat sekumpulan data training untuk regresi SVM digambarkan sebagai dataset $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots (\boldsymbol{x}_m, y_m)\}$ Dimana $\boldsymbol{x}_i$ adalah vector n-dimensional, sedangkan *y* adalah jumlah real untuk setiap $\boldsymbol{x}_i$.

Tugas dari regresi adalah untuk menemukan fungsi antara $\boldsymbol{x}_i$ dengan $\boldsymbol{y}_i$ (W. Zhao, Tao, & Zio, 2015)yang dalam kasus linier dapat ditulis:

$$y_i = f(x) = w.x + b \quad (1)$$

Dimana w adalah vector beban dan b adalah bias. Kedua parameter ini adalah parameter yang perlu ditentukan nilainya agar dapat memberikan fungsi yang terbaik untuk memetakan data input ke data output.

Pada kasus nonlinier, pemetaan nonlinier : $R^1 \rightarrow F$, dimana F merupakan ruang fitur dari ϕ yang diperkenalkan untuk menerjemahkan kerumitan masalah regresi nonlinier pada R^1 untuk sebuah masalah sederhana regresi linier pada F. Fungsi regresi setelah transformasi menjadi seperti berikut:

$$y_i = f(x) = w * \phi(x) + b \quad (2)$$

Untuk mengevaluasi seberapa baik fungsi regresi, fungsi $\varepsilon - insensitive\ loss$ digunakan:

$$L_\varepsilon\big(y, f(x)\big) = \begin{cases} 0\ untuk\ |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon\ dan\ sebaliknya \end{cases} \quad (3)$$

Fungsi $\varepsilon - insensitive\ loss$ digunakan untuk mengukur resiko empiris, resiko empiris diukur berdasarkan persamaan (3), selisih output/target dengan hasil estimasi. Oleh karenanya parameter ε harus diatur. Kemudian, prosedur diatur untuk meminimalisir resiko empiric dengan memperkenalkan variable slack ξ, ξ^* yang menggambarkan simpangan dari data pelatihan diluar zona $\varepsilon - insensitive$.

Disamping meminimalisir kesalahan empiris dengan fungsi $\varepsilon - insensitive\ loss$, kita juga harus meminimalisir norma Euclidean dari beban yang linier $\|\boldsymbol{w}\|$ yang mana berhubungan dengan kemampuan generalisasi dari model SVR yang dilatih (W. Zhao et al., 2015). Tujuannya adalah untuk memperlebar (*maximize*) margin sehingga kelandaian kurva beserta kompleksitas model dapat dipastikan (Suganyadevi & Babulal, 2014). Sehingga permasalahan regresi dapat dinyatakan seperti masalah optimasi quadratik berikut ini:

$$L(w, \xi) = \frac{1}{2}||w||^2 + c\ \textstyle\sum_i \big(\xi_{2i}, \xi'_{2i}\big), c > 0$$
$$subject\ to \begin{cases} y_i - w * \phi(\boldsymbol{x}_i) - b \le\ \varepsilon + \xi_i \\ w * \boldsymbol{\phi}(\boldsymbol{x}_i) + b - y_i \le\ \varepsilon +\ \xi^*_i \\ \Xi_i, \xi'_i \ge 0 \end{cases} \quad (4)$$

Dimana C menyatakan koefisien penalti yang mendeterminasikan *trade-off* antara keempirisan dengan kesalahan generalisasi yang mana nilai C tersebut perlu diatur (W. Zhao et al., 2015). Untuk menyelesaikan permasalahan pada optimasi quadratik pada persamaan (4), kita dapat menggunakan dual Lagrangian:

$$f(x_i) = (\boldsymbol{w}.\boldsymbol{\phi}(\boldsymbol{x}_i) + b) = \textstyle\sum_{j=1}^{n} \alpha_i K\big(x_i, x_j\big) + b \quad (5)$$

Dimana $K\big(x_i, x_j\big)$ merupakan fungsi kernel yang memuaskan kondisi Mercer. Fungsi kernel yang digunakan pada penelitian ini adalah kernel RBF dengan parameter γ yang juga perlu diatur(W. Zhao et al., 2015).

$$K\big(x_i, x_j\big) = \exp(-\gamma\|x_i - x_j\|^2) \quad (6)$$

Berdasarkan uraian mengenai SVR dapat dilihat bahwa terdapat tiga parameter bebas $C, \varepsilon, dan\ \gamma$ yang penting bagi performa metode SVR, kebanyakan peneliti masih mengikuti prosedur yang standar dalam pemilihan parameter (*trial and error*), yaitu dengan membangun model SVR dengan parameter yang berbeda-beda, kemudian mengujinya pada set validasi untuk menghasilkan parameter yang optimal. Namun

Copyright @ 2015 IlmuKomputer.Com

prosedur ini sangat memakan waktu (Chen, 2007) dan tergantung factor keberuntungan. Oleh karenanya parameter-parameter tersebut perlu di atur nilainya. Beberapa penelitian telah dilakukan untuk mengatur nilai parameter $C, \varepsilon, dan\ \gamma$ pada SVM dengan beberapa metode metaheuristik.

Penelitian yang dilakukan oleh (Wang et al., 2014) terfokus pada peningkatan akurasi prediksi SVM untuk mengatasi kinerja SVM yang terpengaruh akibat pemilihan parameter yang tidak tepat untuk memprediksi harga *real estate* di China. Pada penelitiannya (Wang et al., 2014) menerapkan *Particle swarm optimization* (PSO) untuk menentukan nilai $\sigma, \varepsilon, dan\ c$ pada SVM. Selain itu, sebagai perbandingan (Wang et al., 2014) juga membandingkan hasil prosentase MAPE SVM, dan SVM+PSO dengan MAPE BPNN juga. Hasil penelitian dapat disimpulkan bahwa akurasi prediksi PSO+SVM lebih tinggi dibandingkan dengan model SVM dan BPNN.

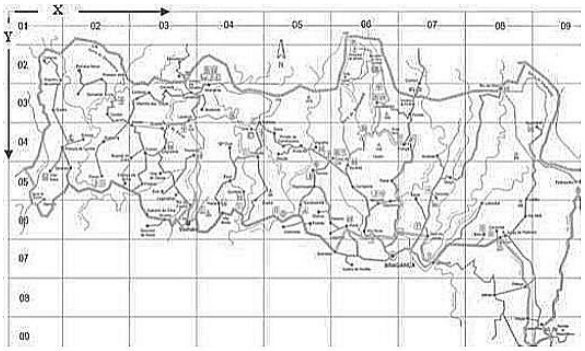
Sedangkan (Z. Jia, Ma, Wang, & Liu, 2011) menerapkan *Simmulated annealing* (SA) untuk mencari nilai parameter $\sigma, \varepsilon, dan\ c$ yang optimal pada SVM. Percobaan yang dilakukan oleh (Z. Jia et al., 2011) yaitu dengan mengatur batasan minimal dan batasan maksimal pada masing-masing parameter seperti parameter C dengan batasan minimal 100 dan maksimal 1000, parameter ε dengan nilai minimal 0,001 dan maksimal 0,01, dan parameter σ dengan nilai minimal 0,5 dan maksimal 1,5. Kemudian Jia melakukan hal yang sama dengan Wang, yaitu membandingkan hasil RMSE dan MAPE metode yang diusulkan dengan metode yang lainnya yaitu (Z. Jia et al., 2011) membandingkan hasil RMSE dan MAPE antara SA+SVM, G-ANFIS, dan ANN. Berdasarkan hasil penelitian dapat disimpulkan bahwa nilai RMSE SA+SVM hanya sedikit lebih tinggi dibandingkan dengan hasil RMSE dan MAPE G-ANFIS, namun SA+SVM berhasil membuat jarak akurasi yang jauh jika dibandingkan dengan ANN.

Sementara untuk mengatasi masalah pencarian nilai parameter pada SVM, (Gu, Zhu, & Jiang, 2011) menerapkan *Genetic Algorithm* (GA) dan juga membandingkan hasil prediksi dengan metode *Grey model* (GM). Berdaarkan hasil penelitian dapat disimpulkan bahwa GA+SVM menghasilkan nilai MAPE yang lebih superior dari pada GM.

Pada penelitian ini kami menggunakan GA sebagai algoritma untuk mengatur nilai parameter pada SVR dengan kombinasi kernel yaitu (RBF, dot, polynomial). Kemudian untuk membuktikan kehandalan metode yang telah diusulkan, dengan metode-metode regresi lainnya untuk memprediksi kebakaran hutan. Pemilihan metode-metode regresi tersebut sesuai dengan beberapa penelitian terbaru yang telah dilakukan terhadap permasalahan regresi seperti penelitian yang telah dilakukan oleh (Tiryaki et al., 2014) yang mencari metode regresi yang tepat untuk dapat mendeteksi efek perlakuan pada pada ikatan kayu. Tiryaki membandingkan dua metode diantaranya ANN dan MLR. Berdasarkan percobaan, dapat disimpulkan ANN telah terbukti menjadi metode yang sukses dan *sufficient* untuk memodelkan karakteristik kekuatan ikatan jenis kayu. Selain ANN & MLR ada juga *Linear* regression (LR) (Lira et al., 2014), dan k-NN (Lee et al., 2014).

3 METODE YANG DIUSULKAN

Pada penelitian ini, data yang digunakan adalah dataset *forestfire* yang diambil dari laman https://archieve.ics.edu/ml/datasets/Forest+Fires. Data *forest fire* merupakan data kebakaran hutan sejak Januari tahun 2000 sampai dengan Desember tahun 2003. Data ini berasal dari taman alam Mountesinho, Portugal yang terdiri dari 13 attribut (12 atribut/*input* dan 1 target estimasi/*output*), dan 517 record yang dapat dilihat pada Tabel 1.



Gambar 1. Peta Taman Alam Mountesinho

Pada Gambar 1 sumbu x dan y merupakan koordinat terjadinya kebakaran hutan, kemudian hari dan bulan terjadinya kebakaran hutan, data meteorologi yang terdiri dari: temperature, kelembaban relatif (RH), angin, dan hujan, selanjutnya terdapat empat komponen dari *Forest Fire Weather Index* (FWI) yaitu: *Fine Fuel Moisture Code* (FFMC) yang merupakan indikator untuk bahan yang mudah terbakar dan mudah membentuk pengapian di dalam hutan, *Duff Moisture Code* (DMC) yang merupakan indikasi dari konsumsi bahan bakar pada kayu, dan *Drought Code* (DC) yang merupakan indicator dari efek kemarau pada bahan bakar hutan, *Initial Spread Index* (ISI) yaitu indicator tingkat penyebaran api dan indikator kesulitan pengendalian api. Berdasarkan data yang tersedia, telah ditentukan area yang terbakar sebagai *output* dan 12 atribut lainnya merupakan *input*. Dataset *forest fire* dapat dilihat pada Tabel 1.

Tabel 1. Dataset Forest Fire

X	Y	Month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
7	5	Mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
7	5	Mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
7	4	Oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0
7	4	Oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
8	6	Mar	fri	91.7	33.3	77.5	9	8.3	97	4	0.2	0
8	6	Mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
8	6	Aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0
8	6	Aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
8	6	Aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0
8	6	Sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0
7	5	Sep	sat	92.5	88	698.6	7.1	22.8	40	4	0	0
7	5	Sep	sat	92.5	88	698.6	7.1	17.8	51	7.2	0	0
7	5	Sep	sat	92.8	73.2	713	22.6	19.3	38	4	0	0
6	5	Aug	fri	63.5	70.8	665.3	0.8	17	72	6.7	0	0
6	5	Sep	mon	90.9	126.5	686.5	7	21.3	42	2.2	0	0
6	5	Sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0	0
6	5	Sep	fri	93.3	141.2	713.9	13.9	22.9	44	5.4	0	0
5	5	Mar	sat	91.7	35.8	80.8	7.8	15.1	27	5.4	0	0
8	5	Oct	mon	84.9	32.8	664.2	3	16.7	47	4.9	0	0
6	4	Mar	wed	89.2	27.9	70.8	6.3	15.9	35	4	0	0
6	4	Apr	sat	86.3	27.4	97.1	5.1	9.3	44	4.5	0	0
6	4	Sep	tue	91	129.5	692.6	7	18.3	40	2.7	0	0
5	4	Sep	mon	91.8	78.5	724.3	9.2	19.1	38	2.7	0	0
7	4	Jun	sun	94.3	96.3	200	56.1	21	44	4.5	0	0
7	4	Aug	sat	90.2	110.9	537.4	6.2	19.5	43	5.8	0	0
7	4	Aug	sat	93.5	139.4	594.2	20.3	23.7	32	5.8	0	0
7	4	Aug	sun	91.4	142.4	601.4	10.6	16.3	60	5.4	0	0
7	4	Sep	fri	92.4	117.9	668	12.2	19	34	5.8	0	0
7	4	Sep	mon	90.9	126.5	686.5	7	19.4	48	1.3	0	0
6	3	Sep	sat	93.4	145.4	721.4	8.1	30.2	24	2.7	0	0
6	3	Sep	sun	93.5	149.3	728.6	8.1	22.8	39	3.6	0	0
6	3	Sep	fri	94.3	85.1	692.3	15.9	25.4	24	3.6	0	0
6	3	Sep	mon	88.6	91.8	709.9	7.1	11.2	78	7.6	0	0
6	3	Sep	fri	88.6	69.7	706.8	5.8	20.6	37	1.8	0	0
6	3	Sep	sun	91.7	75.6	718.3	7.8	17.7	39	3.6	0	0
6	3	Sep	mon	91.8	78.5	724.3	9.2	21.2	32	2.7	0	0

Copyright @ 2015 IlmuKomputer.Com

Pada tahap awal pengolahan data (*preprocessing*), kami melakukan penghapusan beberapa atribut yang tidak diperlukan seperti dalam (Cortez & Morais, 2007) sekaligus menghapus *record* yang berisi data ganda, sehingga dari ke 13 atribut hanya tersisa 9 atribut saja (8 estimator dan 1 target estimasi), dan dari 517 record tersisa 513 data yang unik. Kemudian kami melakukan transformasi pada target estimasi (atribut area) atau label dengan rumus $y = \ln(x + 1)$ untuk menghilangkan kecenderungan positif pada dataset (Cortez & Morais, 2007), hasil dari tahapan *preprocessing* menghasilkan data baru yang dapat dilihat pada Tabel 2.

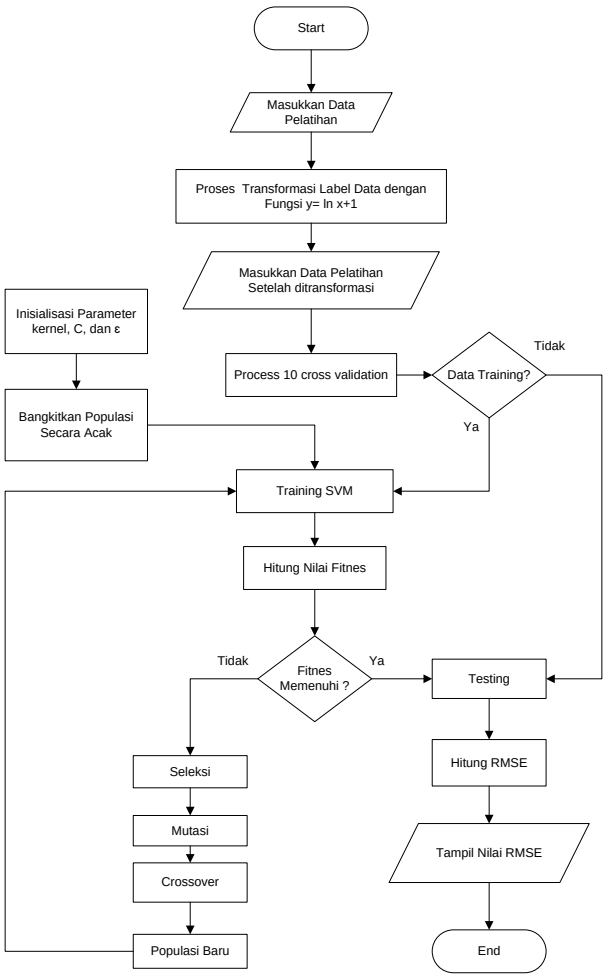
Tabel 2. Dataset Forest Fire Setelah Dilakukan *Preprocessing*

FFMC	DMC	DC	ISI	temp	RH	wind	rain	ln(area+1)
86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
90.6	35.4	669.1	6.7	18	33	0.9	0	0
90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
91.7	33.3	77.5	9	8.3	97	4	0.2	0
89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
92.3	85.3	488	14.7	22.2	29	5.4	0	0
92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
91.5	145.4	608.2	10.7	8	86	2.2	0	0
91	129.5	692.6	7	13.1	63	5.4	0	0
92.5	88	698.6	7.1	22.8	40	4	0	0
92.5	88	698.6	7.1	17.8	51	7.2	0	0
92.8	73.2	713	22.6	19.3	38	4	0	0
63.5	70.8	665.3	0.8	17	72	6.7	0	0
90.9	126.5	686.5	7	21.3	42	2.2	0	0
92.9	133.3	699.6	9.2	26.4	21	4.5	0	0
93.3	141.2	713.9	13.9	22.9	44	5.4	0	0
91.7	35.8	80.8	7.8	15.1	27	5.4	0	0
84.9	32.8	664.2	3	16.7	47	4.9	0	0
89.2	27.9	70.8	6.3	15.9	35	4	0	0
86.3	27.4	97.1	5.1	9.3	44	4.5	0	0
91	129.5	692.6	7	18.3	40	2.7	0	0
91.8	78.5	724.3	9.2	19.1	38	2.7	0	0
94.3	96.3	200	56.1	21	44	4.5	0	0
90.2	110.9	537.4	6.2	19.5	43	5.8	0	0
93.5	139.4	594.2	20.3	23.7	32	5.8	0	0
91.4	142.4	601.4	10.6	16.3	60	5.4	0	0
92.4	117.9	668	12.2	19	34	5.8	0	0
90.9	126.5	686.5	7	19.4	48	1.3	0	0
93.4	145.4	721.4	8.1	30.2	24	2.7	0	0
93.5	149.3	728.6	8.1	22.8	39	3.6	0	0

Selanjutnya kami mengusulkan metode yang disebut SVM+GA yang mana GA digunakan untuk mengoptimasi parameter pada SVM untuk mendapatkan hasil estimasi area yang terbakar akibat kebakaran hutan yang lebih akurat seperti yang dapat dilihat pada Gambar 2.

Pada Gambar 2. Dataset asli *forest fire* akan melalui tahapan *preprocessing data* yang mana pada tahapan ini terdapat dua proes yaitu *deletion attribute* dan *label transformation*, kemudian setelah tahap *preprocessing* akan terbentuk dataset yang baru. Sebelum dataset baru dilatih (*training*) dan diuji (*testing*), dataset akan dipecah terlebih dahulu dengan menerapkan 10-*fold cross validation* untuk membagi data menjadi dua yaitu 90% data *training* dan 10% data *testing.* Kemudian data dilatih dan diuji dengan metode SVM yang mana nilai parameter- parameter ($c, \varepsilon, dan\ \gamma$) dari kernel-kernel SVM (dot, polynomial, RBF) telah diatur oleh GA sebelumnya. Selanjutnya kernel dengan nilai RMSE yang terkecil akan dipergunakan pada model SVM+GA untuk dibandingkan dengan model regresi lainnya.

Algoritma genetika (GA) merupakan algoritma evolusioner yang paling populer (Yang, 2014) yang mana algoritma ini menggunakan prinsip dasar dari seleksi alam yang diperkenalkan oleh Charles Darwin. Algoritma genetika diterapkan sebagai pendekatan untuk mengidentifikasi pencarian nilai dan solusi bagi berbagai permasalahan optimasi(Gorunescu, 2011).



Gambar 2. Metode Penelitian yang Diusulkan

Algoritma genetika memiliki tiga operator genetik utama yaitu *crossover* (proses penukaran kromosom), mutasi (proses penggantian salah satu solusi untuk meningkatkan keragaman populasi), seleksi (penggunaan solusi dengan nilai fitness yang tinggi untuk lulus ke generasi berikutnya). Terdapat langkah-langkah yang sering dilakukan untuk menyelesaikan permasalahan-permasalahan dalam optimasi:

1. Inisialisasi populasi
2. Evaluasi populasi
3. Seleksi populasi
4. Proses penyilangan kromosom (*crossover*)
5. Evaluasi populasi baru
6. Selama syarat belum terpenuhi ulangi dari langkah 3.

Terdapat beberapa kelebihan dari algoritma genetika dibandingkan algoritma optimasi tradisional lainnya, dua diantaranya yaitu kemampuan untuk menangani permasalahan kompleks dan parallel. Algoritma genetika dapat menangani berbagai macam optimasi tergantung pada fungsi objektifnya (*fitness*) apakah seimbang ataukan tidak seimbang, linier atau tidak linier, berkesinambungan atau tak berkesinambungan, atau dengan *random noise.* Fungsi fitness (Zhang, Liu, Wang, & Deng, 2011) ditunjukkan pada persamaan (7).

$$fitness = \sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (7)$$

Copyright @ 2015 IlmuKomputer.Com

Dimana $\hat{y}_i$ merupakan nilai prediksi, $dan\ y_i$ merupakan nilai asli dan dari sampel dataset. N adalah jumlah sampel total.

Akurasi prediksi keseluruhan percobaan pada penelitian ini ditunjukkan pada Gambar 6. Metrik yang kami gunakan untuk mengukur akurasi predikai adalah nilai *root mean square error* (RMSE) yang didefinisikan pada persamaan 8.

$$RMSE = \sqrt{\frac{\sum_{j=i}^{n}(y'-y)^2}{n}} \quad (8)$$

RMSE sangat populer untuk menilai algoritma mesin pembelajaran, termasuk algoritma yang jauh lebih canggih dari regresi linier (Conway & White, 2012). Nilai RMSE digunakan untuk membedakan kinerja model dalam periode kalibrasi dengan periode validasi serta untuk membandingkan kinerja model individual dengan model prediksi lainnya (Hosseini, Javaherian, & Movahed, 2014).

4 HASIL EKSPERIMEN

Eksperimen dilakukan menggunakan komputer personal Intel Core i3, 4GB RAM, 500GB HDD, sistem operasi Windows 8.1, dan Rapidminer 5.0.

Penelitian ini dilakukan dalam tiga tahapan. Tahap pertama untuk mendapatkan hasil estimasi kebakaran hutan yang lebih akurat, kami membandingkan hasil eksperimen antara data yang diproses menggunakan metode SVM yang parameternya diatur secara manual dengan data yang diproses menggunakan metode SVM yang parameternya telah diatur oleh GA pada masing-masing kernel SVM. Kemudian kami melakukan uji beda untuk mengukur signifikansi keakuratan estimasinya.

Tahap kedua, untuk mengidentifikasi kernel terbaik pada SVM, kami melakukan perbandingan rata-rata RMSE dari 10 kali percobaan antara metode SVM dengan SVM+GA pada masing-masing kernel.

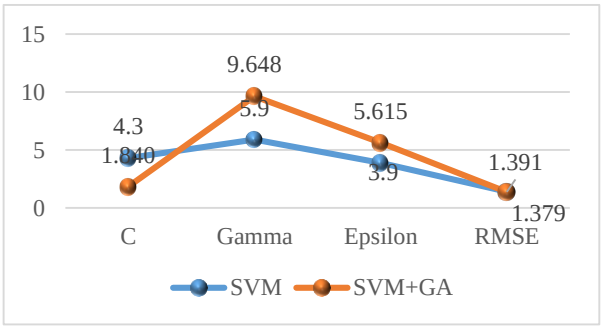
Tahap ketiga untuk mengetahui kehandalan model SVM+GA, kami melakukan perbandingan akurasi antara metode SVM+GA dengan metode regresi lainnya seperti k-NN, LR, dan NN.

Pada tahapan pertama, eksperimen pada masing-masing kernel SVM dilakukan sebanyak 10 kali. Pada kernel dot, pada tahapan inisialisasi populasi, kami memasukkan 10 kombinasi jangkauan nilai input parameter seperti yang dapat dilihat pada Tabel 3.

Tabel 3. Hasil Eksperimen dengan Kernel Dot

γ			C			ε		
MIN	MAX	OPTIMAL	MIN	MAX	OPTIMAL	MIN	MAX	OPTIMAL
0	0.5	0.308	-1	-0.5	-0.890	0	0.5	0.263
0	1	0.939	-1	1	-0.888	0	1	0.685
0	5	4.663	-1	5	-0.618	0	5	3.443
0	7	4.363	-1	7	5.906	0	7	3.952
0	10	1.840	-1	10	9.648	0	10	5.615
0	100	18.261	-1	100	96.812	0	100	56.201
0.001	0.1	0.001	-1	0.1	0.002	0.001	0.1	0.057
0.01	0.1	0.035	-1	0.1	0.014	0.01	0.1	0.100
0.5	10	1.143	-0.5	10	1.821	0.5	10	5.089
0.5	100	18.670	-0.5	100	96.828	0.5	100	56.420

Nilai RMSE terkecil yaitu 1.379 diperoleh dari percobaan SVM (kernel dot) dan GA dengan pencarian nilai parameter $c = 1.840,\ \varepsilon = 5.615, dan\ \gamma = 9.648$ seperti yang dapat dilihat pada Gambar 3.



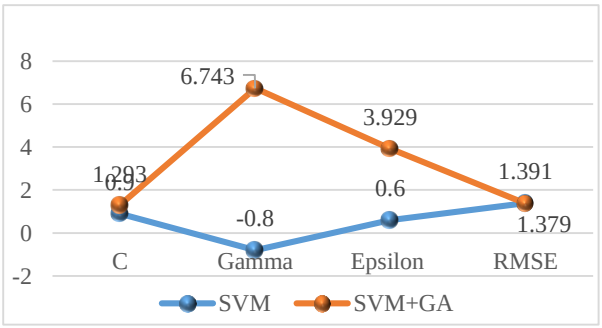
Gambar 3. Perbandingan Estimasi SVM dengan SVM+GA pada Kernel Dot

Pada kernel polynomial Pada kernel dot, pada tahapan inisialisasi populasi, kami memasukkan 10 kombinasi jangkauan nilai input parameter seperti yang dapat dilihat pada Tabel 4.

Tabel 4. Hasil Eksperimen dengan Kernel Polynomial

γ			C			ε		
MIN	MAX	OPTIMAL	MIN	MAX	OPTIMAL	MIN	MAX	OPTIMAL
0	0.5	0.311	-1	-0.5	-0.513	0	0.5	0.292
0	1	0.953	-1	1	-0.881	0	1	0.687
0	5	0.369	-1	5	0.316	0	5	2.405
0	7	1.293	-1	7	6.743	0	7	3.929
0	10	1.840	-1	10	9.648	0	10	5.615
0	100	18.261	-1	100	96.812	0	100	56.201
0.001	0.1	0.089	-1	0.1	-0.940	0.001	0.1	0.082
0.01	0.1	0.090	-1	0.1	-0.940	0.01	0.1	0.085
0.5	10	2.249	-0.5	10	9.664	0.5	10	5.834
0.5	100	18.670	-0.5	100	96.828	0.5	100	56.420

Nilai RMSE terkecil yaitu 1.379 diperoleh dari percobaan SVM+GA dengan hasil pencarian nilai parameter $c = 1.293,\ \varepsilon = 6.743, dan\ \gamma = 3.929$ seperti yang dapat dilihat pada Gambar 4.



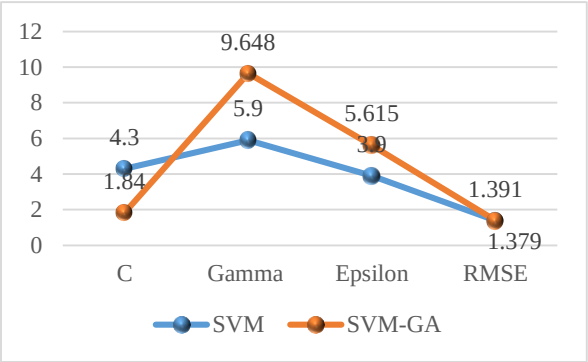
Gambar 4. Perbandingan Estimasi SVM dengan SVM+GA pada Kernel Polynomial

Pada kernel RBF Pada kernel dot, pada tahapan inisialisasi populasi, kami memasukkan 10 kombinasi jangkauan nilai input parameter seperti yang dapat dilihat pada Tabel 5. Hasil terbaik juga diperoleh dari percobaan SVM+GA dengan nilai RMSE=1.379 dengan hasil pencarian parameter $c = 1.840,\ \varepsilon = 9.648, dan\ \gamma = 5.615$ seperti yang dapat dilihat pada Gambar 5.

Copyright @ 2015 IlmuKomputer.Com

Tabel 5. Hasil Eksperimen dengan Kernel RBF

γ			C			ε		
MIN	MAX	OPTIMAL	MIN	MAX	OPTIMAL	MIN	MAX	OPTIMAL
0	0.5	0.091	-1	-0.5	-0.503	0	0.5	0.297
0	1	0.952	-1	1	-0.885	0	1	0.679
0	5	1.216	-1	5	1.410	0	5	0.939
0	7	4.370	-1	7	5.914	0	7	3.944
0	10	1.840	-1	10	9.648	0	10	5.615
0	100	18.261	-1	100	96.812	0	100	56.201
0.001	0.1	0.027	-1	0.1	-0.735	0.001	0.1	0.053
0.01	0.1	0.031	-1	0.1	-0.558	0.01	0.1	0.026
0.5	10	2.249	-0.5	10	9.664	0.5	10	5.834
0.5	100	18.670	-0.5	100	96.828	0.5	100	56.420



Gambar 5. Perbandingan Estimasi SVM dengan SVM+GA pada Kernel RBF

Untuk membuktikan signifikansi antara metode SVM dengan SVM+GA pada masing-masing kernel, kami melakukan uji t sampel berpasangan dengan membandingkan nilai rata-rata masing-masing RMSE dari 10 kali percobaan. Uji beda dilakukan untuk menguji hipotesa:

H_0: Tidak ada perbedaan nilai rata-rata RMSE antara model SVM yang dioptimasi secara manual dengan model SVM yang telah dioptimasi dengan GA.

H_1: Terdapat perbedaan nilai rata-rata RMSE antara model SVM yang dioptimasi secara manual dengan model SVM yang telah dioptimasi dengan GA

Berdasarkan hasil uji t sampel berpasangan yang telah dilakukan pada ketiga kernel, kernel dot menunjukan bahwa tidak ada perbedaan yang signifikan antara metode SVM dengan metode SVM+GA yang dapat dilihat pada Tabel 3, namun kernel polynomial dan kernel RBF menunjukkan bahwa terdapat perbedaan yang signifikan antara metode SVM dengan SVM+GA yang dapat dilihat pada Tabel 4, dan 5.

Berdasarkan Tabel 3. Diketahui nilai t hitung yang diwakili oleh t stat sebesar 1.710268, sedangkan nilai t tabel diwakili oleh t *critical two tail* sebesar 2.262157 , maka dapat dipastikan nilai t hitung<t tabel yang artinya H_0 diterima dan H_1 ditolak, sedangkan nilai probabilitas yang ditunjukan oleh nilai P(T<=t)two tail sebesar 0.121382 lebih besar dari 0.05 yang artinya tidak terdapat perbedaan yang signifikan dari rata-rata RMSE model SVM dengan SVM+GA menggunakan kernel dot.

Pada Tabel 4. Diketahui nilai t hitung yang diwakili oleh t stat sebesar 2.429748, dan nilai t tabel yang diwakili oleh nilai t *critical two tail* sebesar 2.262157 maka dapat dipastikan nilai t hitung>t tabel yang artinya H_0 ditolak dan H_1 diterima, sedangkan diketahui nilai probabilitas sebesar 0.037998 yang mana nilai probabilitas < 0.05 yang artinya terdapat perbedaan yang signifikan dari rata-rata RMSE model SVM dengan SVM+GA menggunakan kernel polynomial.

Tabel 3. Hasil Uji Beda Statistik SVM dan SVM+GA Menggunakan Kernel Dot

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	1.9109	1.8428
Variance	0.643116	0.481873
Observations	10	10
Pearson Correlation	0.996192	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.710268	
P(T<=t) one-tail	0.060691	
t Critical one-tail	1.833113	
P(T<=t) two-tail	0.121382	
t Critical two-tail	2.262157	

Tabel 4. Hasil Uji Beda Statistik SVM dengan SVM+GA Menggunakan Kernel Polynomial

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	1.6293	1.4762
Variance	0.106602	0.018734
Observations	10	10
Pearson Correlation	0.958101	
Hypothesized Mean Difference	0	
df	9	
t Stat	2.429748	
P(T<=t) one-tail	0.018999	
t Critical one-tail	1.833113	
P(T<=t) two-tail	0.037998	
t Critical two-tail	2.262157	

Berdasarkan Tabel 5. Diketahui nilai t hitung yang diwakili oleh t stat sebesar 2.537358, dan nilai t tabel yang diwakili oleh nilai t *critical two tail* sebesar 2.262157. Sehingga dapat dipastikan nilai t hitung> t tabel yang artinya H_0 ditolak dan H_1 diterima, sedangkan diketahui nilai probabilitasnya sebesar 0.031849 yang mana nilai probabilitas ini lebih kecil dari pada 0.05 yang artinya terdapat perbedaan yang signifikan dari rata-rata RMSE model SVM dengan SVM+GA menggunakan kernel RBF.

Pada tahapan kedua kami mengambil nilai rata-rata RMSE dari 10 kali percobaan. Berdasarkan hasil perhitungan rata-rata kernel RBF terbukti memiliki nilai rata-rata RMSE yang terkecil baik pada metode SVM maupun pada metode SVM+GA. Sementara kernel dot memiliki nilai rata-rata RMSE yang paling besar diantara yang lainnya pada metode SVM dan SVM+GA. Perolehan nilai rata-rata RMSE dapat dilihat pada Gambar 6.

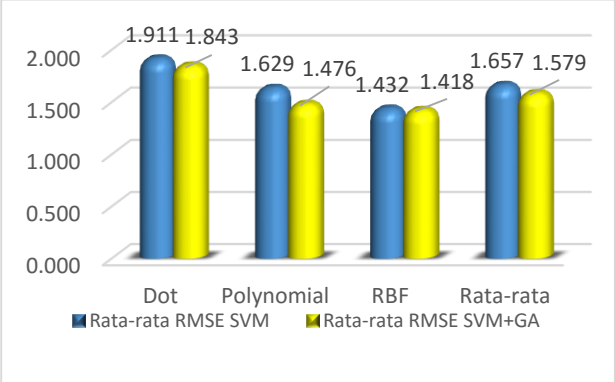
Pada Gambar 6 dapat disimpulkan bahwa nilai rata-rata RMSE pada model SVM+GA dengan kernel (dot, polynomial, dan RBF) lebih kecil dibandingkan dengan nilai rata-rata RMSE pada model SVM dengan kernel (dot, polynomial, RBF) tanpa optimasi parameter.

Copyright @ 2015 IlmuKomputer.Com

Tabel 5. Hasil Uji Beda Statistik SVM dengan SVM+GA Menggunakan Kernel RBF

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	1.4318	1.4179
Variance	0.002526	0.002326
Observations	10	10
Pearson Correlation	0.938956	
Hypothesized Mean Difference	0	
df	9	
t Stat	2.537358	
P(T<=t) one-tail	0.015924	
t Critical one-tail	1.833113	
P(T<=t) two-tail	0.031849	
t Critical two-tail	2.262157	

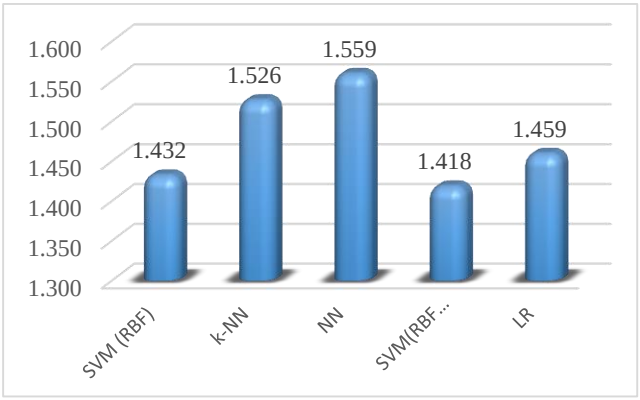


Gambar 6. Perbandingan Nilai Rata-rata RMSE SVM dengan SVM+GA

Pada tahapan ketiga, dataset *forest fire* juga dipercobakan pada metode regresi lainnya. hal ini dilakukan untuk mengetahui kehandalan model SVM+GA jika dibandingkan dengan model-model regresi lainnya. Metode yang akan dibandingkan dengan model SVM+GA pada tahap ketiga ini antara lain SVM, k-NN, LR, dan NN dengan 6 kali percobaan. Hasil perbandingan metode-metode regresi akan ditunjukkan pada Gambar 7. Pada metode k-NN parameter k di atur secara manual yaitu dengan memasukkan nilai parameter 15, 10, 50, 100, dan 150 pada 6 kali percobaan. Sementara pada metode LR nilai *forward alpha* dan *backward alpha* juga diinput secara manual. Begitu pula pada NN, kami memasukkan variasi nilai parameter untuk *learning rate* dan *momentum* secara manual.

k-NN memiliki algoritma yang sederhana dan kinerja prediksi yang tinggi terhadap berbagai aplikasi, karena kelebihannya k-NN dianggap sebanding dengan model yang lebih kompleks seperti ANN atau SVM (Lee et al., 2014), Namun Gambar 6 menjelaskan bahwa perbandingan nilai rata-rata RMSE k-NN jauh lebih besar dibandingkan metode SVM. Oleh karena itu hasil penelitian ini secara otomatis bertolak belakang dengan pendapat (Lee et al., 2014) mengingat nilai akurasi k-NN yang buruk terhadap estimasi kebakaran hutan.

Penggunaan NN menjadi semakin popular dibanyak model prediksi (Kaytez, Taplamacioglu, Cam, & Hardalac, 2015). Pada penelitian (Tiryaki et al., 2014) ANN juga terbukti dapat menyelesaikan permasalahan estimasi melebihi MLR.



Gambar 7. Perbandingan Rata-rata RMSE Kelima Metode Regresi

Namun Gambar 6 pada penelitian ini menunjukkan NN menghasilkan akurasi yang paling buruk diantara metode lainnya dalam mengestimasi kebakaran hutan. Sementara metode LR seperti dalam (Lira et al., 2014) menghasilkan nilai akurasi yang baik ketika diterapkan pada dataset *forest fire*. Hal ini dapat dilihat berdasarkan perbandingan nilai rata-rata RMSE yaitu nilai rata-rata RMSE LR jauh bebih kecil jika dibandingkan dengan nilai rata-rata RMSE NN.

Perbandingan nilai RMSE NN dengan SVM(RBF)+GA juga sangat timpang, nilai rata-rata RMSE NN sangat jauh lebih besar dari pada nilai rata-rata RMSE SVM(RBF)+GA. Selain itu nilai rata-rata RMSE LR hamper sebanding dengan nilai rata-rata SVM(RBF) dan SVM(RBF)+GA dengan selisih nilai RMSE hanya sebesar 0.027 dan 0.041 saja. Oleh karena itu penelitian ini secara otomatis mendukung penelitian yang dilakukan oleh (Lira et al., 2014) sekaligus bertolak belakang dengan hasil penelitian (Tiryaki et al., 2014).

Berdasarkan hasil percobaan yang telah dilakukan, metode yang telah diusulkan jauh lebih unggul jika dibandingkan dengan hasil percobaan (Cortez & Morais, 2007). Pada penelitian sebelumnya, penerapan *sequential minimize optimization algorithm* (SMO) untuk mengoptimasi parameter $C, \varepsilon, dan\ \gamma$ pada SVM berhasil melebihi hasil prediksi metode regresi lainnya yaitu *naïve predictor* (NP), *multiple regression* (MR), *descicion tree* (DT), *random forest* (RF), dan *neural network* (NN) dengan nilai RMSE SVM(RBF)+SMO sebesar 12.71. oleh karena itu Gambar 6 pada penelitian ini sesuai dengan pernyataan (Cortez & Morais, 2007) bahwa SVM dengan optimasi parameter dapat mengungguli metode-metode regresi lainnya.

5 KESIMPULAN

SVM dapat mengatasi masalah klasifikasi dan regresi dengan kernel linier ataupun kernel nonlinier nonlinier yang dapat menjadi satu kemampuan algoritma pembelajaran untuk klasifikasi serta regresi. Namun, dibalik keunggulannya SVM juga memiliki kelemahan yaitu sulitnya menentukan nilai parameter yang optimal. Pada penelitian ini SVM digunakan untuk memprediksi area yang terbakar pada dataset *forest fire* dengan fungsi regresinya. Penerapan algoritma genetika (GA) pada metode SVM diusulkan untuk mengoptimasi nilai parameter $C, \varepsilon, dan\ \gamma$ pada kernel-kernel SVM (dot, polynomial, RBF) untuk mendapatkan akurasi yang terbaik dan untuk mengidentifikasi kernel yang terbaik pula.

Copyright @ 2015 IlmuKomputer.Com

Dikarenakan permasalahan pada data *forest fire* merupakan tugas regresi, Beberapa metode regresi pun diusulkan untuk membuktikan kehandalan metode yang telah diusulkan. Hasil eksperimen membuktikan bahwa metode yang diusulkan SVM (RBF)+GA memiliki nilai akurasi estimasi yang lebih baik dari pada metode regresi lainnya. Untuk penelitian dimasa mendatang, kami percaya kombinasi SVM dengan algoritma metaheuristik lainnya serta penambahan metode spatial sebagai pendeteksian *outlier* dapat meningkatkan akurasi lebih signifikan.

REFERENSI

Brun, C., Margalef, T., & Cortés, A. (2013). Coupling Diagnostic and Prognostic Models to a Dynamic Data Driven Forest Fire Spread Prediction System. *Procedia Computer Science*, *18*, 1851–1860. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S1877050913004973

Chen, K.-Y. (2007). Forecasting systems reliability based on support vector regression with genetic algorithms. *Reliability Engineering & System Safety*, *92*(4), 423–432.

Conway, D., & White, J. M. (2012). *Machine Learning for Hackers*. (J. Steele, Ed.).

Cortez, P., & Morais, A. (2007). A Data Mining Approach to Predict Forest Fires using Meteorological Data.

Denham, M., Wendt, K., Bianchini, G., Cortés, A., & Margalef, T. (2012). Dynamic Data-Driven Genetic Algorithm for forest fire spread prediction. *Journal of Computational Science*, *3*(5), 398–404. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S1877750312000658

Dua. (2011). *Data Mining and Machine Learning in Cybersecurity*. (Dua, Ed.).

Eastaugh, C. S., & Hasenauer, H. (2014). Deriving forest fire ignition risk with biogeochemical process modelling. *Environmental Modelling & Software*, *55*, 132–142. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S1364815214000280

Gorunescu, F. (2011). *Intelligent Systems Reference Library*. (Gorunescu, Ed.).

Gu, J., Zhu, M., & Jiang, L. (2011). Housing price forecasting based on genetic algorithm and support vector machine. *Expert Systems with Applications*, *38*(4), 3383–3386. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0957417410009310

Guo, X., Li, D., & Zhang, A. (2012). Improved Support Vector Machine Oil Price Forecast Model Based on Genetic Algorithm Optimization Parameters. *AASRI Procedia*, *1*, 525–530. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S2212671612000832

Hosseini, M., Javaherian, A., & Movahed, B. (2014). Determination of permeability index using Stoneley slowness analysis, NMR models, and formation evaluations: a case study from a gas reservoir, south of Iran. *Journal of Applied Geophysics*, *109*, 80–87.

Ilhan, I., & Tezel, G. (2013). A genetic algorithm-support vector machine method with parameter optimization for selecting the tag SNPs. *Journal of Biomedical Informatics*, *46*(2), 328–40. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/23262450

Jia, Z., Ma, J., Wang, F., & Liu, W. (2011). Hybrid of simulated annealing and SVM for hydraulic valve characteristics prediction. *Expert Systems with Applications*, *38*(7), 8030–8036.

Jia, Z.-Y., Ma, J.-W., Wang, F.-J., & Liu, W. (2010). Characteristics forecasting of hydraulic valve based on grey correlation and ANFIS. *Expert Systems with Applications*, *37*(2), 1250–1255. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0957417409005624

Kaytez, F., Taplamacioglu, M. C., Cam, E., & Hardalac, F. (2015). Electrical Power and Energy Systems Forecasting electricity consumption : A comparison of regression analysis , neural networks and least squares support vector machines. *International Journal of Electrical Power and Energy Systems*, *67*, 431–438. Retrieved from

Lee, S., Kang, P., & Cho, S. (2014). Neurocomputing Probabilistic local reconstruction for k -NN regression and its application to virtual metrology in semiconductor manufacturing. *Neurocomputing*, *131*, 427–439. Retrieved from

Lira, M. A. T., Da Silva, E. M., Alves, J. M. B., & Veras, G. V. O. (2014). Estimation of wind resources in the coast of Ceará, Brazil, using the linear regression theory. *Renewable and Sustainable Energy Reviews*, *39*, 509–529. Retrieved from

Machairas, V., Tsangrassoulis, A., & Axarli, K. (2014). Algorithms for optimization of building design: A review. *Renewable and Sustainable Energy Reviews*, *31*(1364), 101–112. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S1364032113007855

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*.

Özbayoğlu, a. M., & Bozer, R. (2012). Estimation of the Burned Area in Forest Fires Using Computational Intelligence Techniques. *Procedia Computer Science*, *12*, 282–287.

Pan, S., Iplikci, S., Warwick, K., & Aziz, T. Z. (2012). Parkinson's Disease tremor classification – A comparison between Support Vector Machines and neural networks. *Expert Systems with Applications*, *39*(12), 10764–10771.

Quintano, C., Fernández-Manso, A., Stein, A., & Bijker, W. (2011). Estimation of area burned by forest fires in Mediterranean countries: A remote sensing data mining perspective. *Forest Ecology and Management*, *262*(8), 1597–1607. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0378112711004385

Raghavendra. N, S., & Deka, P. C. (2014). Support vector machine applications in the field of hydrology: A review. *Applied Soft Computing*, *19*, 372–386. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S1568494614000611

Rynkiewicz, J. (2012). General bound of overfitting for MLP regression models. *Neurocomputing*, *90*, 106–110.

Singh, P., & Borah, B. (2014). International Journal of Approximate Reasoning Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization. *International Journal of Approximate Reasoning*, *55*(3), 812–833.

Suganyadevi, M. V, & Babulal, C. K. (2014). Support Vector Regression Model for the prediction of Loadability Margin of a Power System. *Applied Soft Computing Journal*, *24*, 304–315.

Tiryaki, S., Öz, Ş., & Y, İ. (2014). International Journal of Adhesion & Adhesives Comparison of arti fi cial neural network and multiple linear regression models to predict optimum bonding strength of heat treated woods, *55*, 29–36.

Wang, X., Wen, J., Zhang, Y., & Wang, Y. (2014). Optik Real estate price forecasting based on SVM optimized by PSO. *Optik - International Journal for Light and Electron Optics*, *125*(3), 1439–1443.

Yang, X. (2014). *Nature-Inspired Optimization Algorithms*. Elsevier. doi:10.1016/B978-0-12-416743-8.00005-1

Yilmaz, I., & Kaynar, O. (2011). Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert Systems with Applications*, *38*(5), 5958–5966.

Zameer, A., Mirza, S. M., & Mirza, N. M. (2014). Core loading pattern optimization of a typical two-loop 300MWe PWR using Simulated Annealing (SA), novel crossover Genetic Algorithms (GA) and hybrid GA(SA) schemes. *Annals of Nuclear Energy*, *65*, 122–131.

Copyright @ 2015 IlmuKomputer.Com

Zhang, D., Liu, W., Wang, A., & Deng, Q. (2011). Parameter Optimization for Support Vector Regression Based on Genetic Algorithm with Simplex Crossover Operator. *Journal of Information & Computational Science*, *6*(June), 911–920. Retrieved from http://www.joics.com/publishedpapers/2011_8_6_911_920.pdf

Zhao, M., Fu, C., Ji, L., Tang, K., & Zhou, M. (2011). Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, *38*(5), 5197–5204.

Zhao, W., Tao, T., & Zio, E. (2015). System reliability prediction by support vector regression with analytic selection and genetic algorithm parameters selection. *Applied Soft Computing*, *30*, 792–802.

BIOGRAFI PENULIS

Hani Harafani. Memperoleh gelar M.Kom dari Sekolah Tinggi Manajemen Ilmu Komputer Nusa Mandiri, Jakarta. Staff pengajar di salah satu Perguruan Tinggi Swasta. Minat penelitian saat ini pada bidang data mining.

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Penerapan Metode *Average Gain*, *Threshold Pruning* dan *Cost Complexity Pruning* untuk *Split* Atribut pada Algoritma C4.5

Erna Sri Rahayu, Romi Satria Wahono dan Catur Supriyanto
Fakultas Ilmu Komputer Universitas Dian Nuswantoro
ernaesr81@gmail.com, romi@romisatriawahono.net, caturs@research.dinus.ac.id

Abstrak: C4.5 adalah algoritma klasifikasi *supervised learning* untuk membentuk pohon keputusan (*Decision Tree*) dari data. *Split* atribut merupakan proses utama dalam pembentukan pohon keputusan (*Decision Tree*) di C4.5. Proses pemilihan *split* atribut di C4.5 belum dapat mengatasi *misclassification cost* di setiap *split* sehingga berpengaruh pada kinerja pengklasifikasi. Setelah dilakukan pemilihan *split* atribut, proses selanjutnya adalah *pruning*. *Pruning* adalah proses yang dilakukan untuk memotong atau menghilangkan beberapa cabang (*branches*) yang tidak diperlukan. Cabang (*branches*) atau *node* yang tidak diperlukan dapat menyebabkan ukuran *Decision Tree* menjadi sangat besar dan hal ini disebut *over-fitting*. Untuk saat ini *over-fitting* merupakan trend riset di kalangan peneliti. Metode-metode untuk pemilihan *split* atribut diantaranya *Gini Index*, *Information Gain*, *Gain Ratio* dan *Average Gain* yang diusulkan oleh Mitchell. *Average Gain* tidak hanya mengatasi kelemahan pada *Information Gain* tetapi juga membantu untuk memecahkan permasalahan dari *Gain Ratio*. Metode *split* atribut yang diusulkan pada penelitian ini adalah menggunakan nilai *average gain* yang dikalikan dengan selisih misklasifikasi. Sedangkan teknik *pruning* dilakukan dengan mengkombinasikan *threshold pruning* dan *cost complexity pruning*. Pada penelitian ini, pengujian metode yang diusulkan akan diterapkan pada dataset kemudian hasil kinerjanya akan dibandingkan dengan hasil kinerja metode *split* atribut yang menggunakan *Gini Index*, *Information Gain* dan *Gain Ratio*. Metode pemilihan *split* atribut yang menggunakan *average gain* yang dikalikan dengan selisih misklasifikasi dapat meningkatkan kinerja pengklasifikasi C4.5. Hal ini ditunjukkan melalui uji *Friedman* bahwa metode *split* atribut yang diusulkan, ditambah dengan *threshold pruning* dan *cost complexity pruning* mempunyai hasil kinerja berada di peringkat 1. Pohon keputusan *(Decision Tree)* yang terbentuk melalui metode yang diusulkan berukuran lebih kecil.

Kata kunci: *Decision Tree*, C4.5, *split* atribut, *pruning*, *over-fitting*, *average gain*.

1 PENDAHULUAN

Decision Tree merupakan algoritma pengklasifikasian yang sering digunakan dan mempunyai struktur yang sederhana dan mudah untuk diinterpretasikan (Mantas & Abellán, 2014). Pohon yang terbentuk menyerupai pohon terbalik, dimana akar (*root*) berada di bagian paling atas dan daun (*leaf*) berada di bagian paling bawah. *Decision Tree* merupakan model klasifikasi yang berbentuk seperti pohon, dimana *Decision Tree* mudah untuk dimengerti meskipun oleh pengguna yang belum ahli sekalipun dan lebih efisien dalam menginduksi data (C. Sammut, 2011). Induksi di *Decision Tree* adalah salah satu teknik tertua dan yang paling tertua untuk model *learning discriminatory*, yang mana model tersebut telah dikembangkan secara mandiri di statistik dan di komunitas *machine learning*. Proses pembentukan *Decision Tree* dibagi menjadi 3 (T Warren Liao, 2007) yaitu, (1) pembentukan pohon (*tree*), (2) *pruning*, (3) mengekstrak aturan (*rule*) dari pohon keputusan yang terbentuk. *Decision Tree* baik digunakan untuk klasifikasi atau prediksi.

Decision Tree telah diaplikasikan di berbagai bidang contohnya di bidang pengobatan (Setsirichok et al., 2012). Salah satu contohnya adalah penerapan C4.5 *Decision Tree* yang digunakan untuk mengklasifikasikan karakteristik darah sehingga dapat mengklasifikasikan 80 *class* kelainan thalassemia yang menyebar di Thailand. Contoh lain penerapan *Decision Tree* untuk memprediksi pasien kanker payudara (Ture, Tokatli, & Kurt, 2009). Selain di bidang pengobatan, *Decision Tree* juga diterapkan di bidang bisnis (Duchessi & Lauría, 2013)(Duchessi & Lauría, 2013) dan deteksi kegagalan (Sahin, Bulkan, & Duman, 2013). Tantangan di *Decision Tree* saat ini adalah sehubungan dengan performa tingkat akurasi, skalabilitas, perkembangan *dataset* dan aplikasi-aplikasi baru yang belum dikembangkan.

Beberapa algoritma yang telah dikembangkan berdasar *Decision Tree* adalah (1) CHAID (*Chi-squared Automatic Interaction Detection*) yang mana *split* tiap *node* berdasar pada *Chi-square test* pada masing-masing atribut, (2) CART (*Classification And Regression Tree*) membentuk *Decision Tree* dengan penghitungan *Gini Index* untuk kriteria *split*, (3) C4.5 yang merupakan variasi pengembangan dari ID3 (*Iterative Dichotomiser 3)* (Gorunescu, 2011). Jika ID3 (*Iterative Dichotomiser 3)* menggunakan *Entropy* untuk kriteria *split*, sedangkan di C4.5 menggunakan *Gain Ratio* untuk kriteria *split*nya. Atribut yang memiliki *Gain Ratio* tertinggi yang akan dipilih. Lim et al (2000) telah membandingkan tingkat akurasi, kompleksitas dan waktu training dari ketiga algoritma klasifikasi tersebut, dan hasilnya menunjukkan bahwa C4.5 mempunyai tingkat akurasi yang bagus dan mudah untuk diinterpretasikan.

C4.5 adalah algoritma klasifikasi *supervised learning* untuk membentuk pohon keputusan (*Decision Tree*) dari data (Mantas & Abellán, 2014)(Mantas & Abellán, 2014)(Quinlan, 1993). C4.5 *Decision Tree* menggunakan kriteria *split* yang telah dimodifikasi yang dinamakan *Gain Ratio* oleh Mitchael (1997) dalam proses pemilihan *split* atribut. *Split* atribut merupakan proses utama dalam pembentukan pohon keputusan (*Decision Tree*) di C4.5 (Quinlan, 1986). Tahapan dari algoritma C4.5 adalah (1) menghitung nilai *Entropy*, (2) menghitung nilai *Gain Ratio* untuk masing-masing atribut, (3) atribut yang memiliki *Gain Ratio* tertinggi dipilih menjadi akar (*root*) dan atribut yang memiliki nilai *Gain Ratio* lebih rendah dari akar (*root*) dipilih menjadi cabang (*branches*), (4) menghitung lagi nilai *Gain Ratio* tiap-tiap atribut dengan tidak mengikutsertakan atribut yang terpilih menjadi akar (*root*) di tahap sebelumnya, (5) atribut yang memiliki *Gain Ratio*

Copyright © 2015 IlmuKomputer.Com

tertinggi dipilih menjadi cabang (*branches*), (6) mengulangi langkah ke-4 dan ke-5 sampai dengan dihasilkan nilai *Gain* = 0 untuk semua atribut yang tersisa.

Setelah dilakukan pemilihan *split attribute*, proses selanjutnya adalah *pruning*. *Pruning* adalah proses yang dilakukan untuk memotong atau menghilangkan beberapa cabang (*branches*) yang tidak diperlukan (C. Sammut, 2011). *Pruning* dilakukan untuk mengembangkan kehandalan generalisasi *Decision Tree* dan akurasi prediksi *Decision Tree* dengan memindahkan *node* yang tidak diperlukan di *Decision Tree* (Otero, Freitas, & Johnson, 2012). Cabang (*branches*) atau *node* yang tidak diperlukan dapat menyebabkan ukuran *Decision Tree* menjadi sangat besar dan hal ini disebut *over-fitting* (Larose, 2006) (Larose, 2005). Untuk saat ini *over-fitting* merupakan trend riset di kalangan peneliti.

Over-fitting dapat menghasilkan model yang baik di training data tetapi secara normal tidak dapat menghasilkan model *tree* yang baik ketika diterapkan di *unseen data* (Wang, Qin, Jin, & Zhang, 2010). *Over-fitting* disebabkan oleh *noisy* data, *irrelevant feature* (Wang et al., 2010). *Noisy* data akan menyebabkan terjadinya misklasifikasi, sehingga *over-fitting* akan menyebabkan tingkat akurasi yang buruk dalam pengklasifikasian. Permasalahan lain di C4.5 adalah ketidakseimbangan data yang juga menyebabkan akurasi C4.5 buruk dalam pengklasifikasian data.

Permasalahan *over-fitting* dapat diatasi dengan melakukan teknik *pruning* (Zhang, 2012). Macam-macam teknik *pruning* untuk mengatasi *over-fitting* adalah *Laplace pruning* yang diperkenalkan oleh Bradford, yang kemudian disempurnakan oleh Provost dan Domingos. Model yang dikembangkan yaitu *Decision Tree* yang melewatkan proses *pruning* dengan melakukan *smoothing* menggunakan *Laplace correction method* (Wang, Qin, Zhang, & Zhang, 2012). Tetapi metode ini mempunyai kelemahan pada *dataset* dengan distribusi data yang tidak seimbang sehingga Zadrozny dan Elkan mengusulkan *Decision Tree* yang tidak *di-pruning* dan menempatkan skor *smoothing* dari daun (*leaf*). Metode *smoothing* yang diusulkan dinamakan *m-estimation* (Wang et al., 2010). Metode ini dilakukan untuk mendapatkan perkiraan probabilitas yang lebih baik.

Banyak strategi untuk pemilihan *split* atribut, diantaranya *Information Gain* (Quinlan, 1986) dan *GINI Index* (Gorunescu, 2011). Kedua strategi diatas digunakan untuk mengukur *impurity*, dimana atribut yang mempunyai nilai pengurangan *impurity* maksimal (*most impurity reduce*) akan terpilih untuk membangun *Decision Tree*. Metode yang lain adalah *average gain* yang diusulkan oleh Mitchell. *Average gain* tidak hanya mengatasi kelemahan pada *informasi gain* tetapi juga membantu untuk memecahkan permasalahan dari *gain ratio*.

Metode yang diusulkan pada penelitian ini untuk proses pemilihan *split* atribut adalah menggunakan nilai *average gain* yang dikalikan dengan selisih antara misklasifikasi setelah *di-split* dan sebelum *di-split*. Sedangkan permasalahan *over-fitting* akan diatasi dengan menerapkan metode *threshold pruning* sebagai proses *pre-pruning*. *Threshold pruning* dilakukan dengan menghitung *misclassification cost* untuk masing-masing potensial *split* atribut. Sedangkan untuk *post pruning* dipilih metode *cost complexity pruning* yang merupakan salah satu jenis *pessimistic error pruning*.

Paper ini disusun sebagai berikut: pada bagian 2 paper terkait dijelaskan. Pada bagian 3, metode yang diusulkan dijelaskan. Hasil percobaan perbandingan antara metode yang diusulkan dengan metode lainnya disajikan pada bagian 4. Akhirnya, kesimpulan dari penelitian kami disajikan pada bagian terakhir.

2 PENELITIAN TERKAIT

2.1 Metode Info Gain (Quinlan, 1993)

Metode penelitian ini diperkenalkan oleh Quinlan dengan berdasar model ID3 (*Iterative Dichotomiser 3)*. Metode yang diperkenalkan Quinlan cocok untuk *dataset* dengan variabel diskret akan tetapi metode yang diperkenalkan tidak cocok untuk *dataset* dengan *missing value*. Metode penelitian Quinlan menggunakan pemilihan *split* atribut yang disebut *Gain*. Informasi yang disampaikan tergantung pada probabilitas dan dapat diukur dalam *bits* sebagai minus algoritma berbasis 2. Sebagai contoh -log2(1/8) = 3 *bits*. Untuk mendapatkan nilai yang diharapkan (*expected information*) yang berkaitan dengan *class-class* yang ada, maka Quinlan menjumlahkan seluruh *class* secara proporsional dengan frekuensi mereka di *S*, seperti di bawah ini.

$$info(S) = -\sum_{j=1}^{k} \frac{freq(Cj,S)}{|S|} \; x \; log_2(\frac{freq(C_j,S)}{|S|}) \; bits \quad (2.1)$$

Ketika diterapkan di training kasus, info (*T*) diukur dari rata-rata informasi yang dibutuhkan untuk mengidentifikasi *class* yang terdapat di kasus *T*. Hal ini disebut dengan *Entropy* (S). Sekarang bandingkan perhitungan yang mirip setelah *T* selesai dipartisi sesuai dengan *n* hasil dari tes *X*. Nilai yang diharapkan (*expected information*) dapat ditentukan melalui pembobotan jumlah dari semua *subset*.

$$info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \; x \; info(T_i) \quad (2.2)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

n = jumlah *subset*
T = atribut
T_i = *subset* dari sebuah atribut

Entropy didefinisikan sebagai nilai informasi yang diharapkan. Dan nilai *Entropy* dapat dihitung melalui rumus persamaan dibawah ini:

$$Gain\ (X) = info(S) - info_x(T) \quad (2.3)$$

Perhitungan informasi di atas didapat dari partisi *T* sesuai dengan tes *X*. Kemudian dipilih atribut yang mempunyai nilai *information gain* yang maksimal.

Pada metode yang diperkenalkan Quinlan tidak melakukan proses *pruning*.

2.2 Metode Info Gain Ratio (Quinlan, 1993)

Metode penelitian ini merupakan pengembangan dari metode *Iterative Dichotomiser 3* (ID3). Quinlan memperkenalkan metode ini dengan nama C4.5, dimana untuk pemilihan *split* atribut menggunakan metode *Info Gain Ratio (IGR)* menggantikan *Info Gain (IG).* C4.5 yang diperkenalkan dapat bekerja pada variabel kontinyu dan *missing value*.

Rumus persamaan *Info Gain Ratio (IGR)* seperti berikut:

$$gain\ ratio\ (X) = \frac{gain\ (X)}{split\ info\ (X)} \quad (2.4)$$

Dimana split info (X) mempunyai persamaan rumus sebagai berikut:

$$split\ info\ (X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \; x \; log_2(\frac{|T_i|}{|T|}) \quad (2.5)$$

Pada metode penelitian ini proses *pruning* menggunakan *posterior complex pruning*.

2.3 Metode Credal Decision Tree (Abellán, 2013)

Joaquin Abellan menggunakan *Imprecise Info Gain* (IIG) untuk pembentukan *Decision Tree*. Pohon (*tree*) yang dibentuk hanya untuk variabel diskret. Metode yang diusulkan oleh Joaquin Abellan & Andres R. Masegosa disebut *Credal Decision Tree. Credal Decision Tree* tidak dapat bekerja pada *dataset* yang mempunyai *missing values*.

Copyright © 2015 IlmuKomputer.Com

Pada proses pemilihan *split* atribut, *Credal Decision Tree* menggunakan *imprecise probability* dan *uncertainty measure* di *credal set*. Interval probabilitas didapat dari *dataset* untuk masing-masing kasus dalam sebuah variabel *class* menggunakan *Walley's Imprecise Dirichlet Model* (IDM). Didefinisikan metode yang diusulkan Abellan dan Moral sebagai *Imprecise Info Gain* (IIG) dengan persamaan rumus seperti berikut:

$$IIG\ (X,C) = S * \big(K(C)\big) - \sum_i p(x_t)\ S * (K(C|X = x_t)) \quad (2.6)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

C	=	*class* variabel
X	=	atribut
S	=	maksimum *Entropy*
$K(C)\ dan\ (K(C\|X = x_t))$	=	*credal set* yang diperoleh melalui *Imprecise Dirichlet Model* (IDM)
$C\ dan\ C\|X = x_t$	=	variabel
$P(X = x_t)$	=	probabilitas distribusi

Pada metode Credal *Decision Tree*, atribut yang terpilih adalah atribut yang mempunyai nilai *Imprecise Info Gain (IIG)* maksimal. Metode *Credal Decision Tree* melewatkan proses *post pruning*.

Dataset yang digunakan diunduh dari *UCI repository of machine learning data sets* dengan alamat *ftp://ftp.ics.uci.edu/machine-learning-databases*. Dataset tersebut antara lain; *Anneal, Audiology, Autos, Breast-cancer, Colic, Cmc, Credit-german, Diabetes-pima, Glass 2, Hepatitis, Hypothyroid, Ionosphere, Kr-vs-kp, Labor, Lymph, Mushroom, Segment, Sick, Solar-flare1, Sonar, Soybean, Sponge, Vote, Vowel, Zoo*.

2.4 Metode Credal C4.5 (Mantas & Abellán, 2014)

Metode penelitian ini diusulkan oleh Carlos J. Mantas & Joaquin Abellan. Metode penelitian yang diusulkan cocok untuk pengklasifikasian *dataset* dengan *noise*. Pemilihan *split* atribut pada metode ini menggunakan *Imprecise Info Gain Ratio* (IIGR) menggantikan *Info Gain Ratio* (IGR). *Imprecise Info Gain Ratio* (IIGR) menggunakan *imprecise probability* untuk menghitung nilai atribut dan variabel *class*. Metode penelitian yang diusulkan oleh Carlos J. Mantas & Joaquin Abellan disebut dengan *Credal C4.5*. Perhitungan *Imprecise Info Gain Ratio* (IIGR) pada *Credal C4.5* menggunakan persamaan rumus berikut ini:

$$IIGR^D\ (Class, X) = \frac{IIG^D(Class,X)}{H(X)} \quad (2.7)$$

$$IIG^D(Class, X) = H * \big(K^D(Class)\big) - \sum_i P^D(X = x_i)\ H * (K^D(Class|X = x_i)) \quad (2.8)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

$Class$	=	*class* variabel
X	=	atribut
H	=	maksimum *Entropy*
$K^D(C)\ dan\ (K^D(C\|X = x_i))$	=	*credal set* yang diperoleh melalui IDM
$C\ dan\ C\|X = x_i$	=	variabel
$P^D(X = x_i)$	=	probabilitas distribusi

Jika C4.5 klasik menggunakan maksimal *Entropy* tapi di *Credal C4.5* menggunakan prinsip maksimal *uncertainty*.

Prosedur pembentukan pohon (*tree*) *Credal C4.5*

1. *If L=Ø, then Exit*
2. *Let D be the partition associated with node* No
3. *If |D| < minimum number of instances, then Exit*
4. *Calculate* $P^D(X=x_i)$ $(i=1,...,n)$ *on the convex set* $K^D(X)$
5. *Compute the value*
   $\alpha = max\ x_j \epsilon M\{IIGR^D(C, X_j)\}$
   With $M = \{X_j \in L / IIG^D(C, X_j) > avg x_j \in L\ \{IIG^D(C, X_j)\}\}$
6. *If* $\alpha \leq 0$ *then Exit*
7. *Else*
8. *Let* X_l *be the variable for which the maximum* α *is attained*
9. *Remove* X_l *from L*
10. *Assign* X_l *to node No*
11. *For each possible value* x_i *of* X_l
12. *Add a node* No_l
13. *Make* No_l *a child of No*
14. *Call BuildCredalC4.5Tree* (No_l, L)

Untuk proses *pruning*, metode yang diusulkan *Carlos J. Mantas & Joaquin Abellan* seperti C4.5 klasik yaitu menggunakan *post pruning* yakni menggunakan *Pessimistic Error Pruning*.

Dataset yang digunakan pada model penelitian *Credal C4.5* didapat dari *UCI repository of machine learning datasets* dengan alamat http://archive.ics.uci.edu/ml. *Dataset* yang digunakan 50 *dataset* diantaranya: *Anneal, Arrhythmia,Audiology, Autos, Balance-scale, Breast-cancer, Wisconsin-breast-cancer, Car, CMC, Horse-colic, Credit-rating, German-credit, Dermatology, Pima-diabetes, Ecoli, Glass, Haberman, Cleveland-14-heart-disease, Hungarian-14-heart-disease, Heart-statlog, Hepatitis, Hypothyroid, Ionosphere, Iris, kr-vs-kp, Letter, Liver-disorder, lymphography, mfeat-pixel, Nursery, Optdigits, Page-blocks, Pendigits, Primary-tumor, Segment, Sick, Solar-flare2, Sonar, Soybean, Spambase, Spectrometer, Splice, Sponge, Tae, Vehicle, Vote, Vowel, Waveform, Wine, Zoo*.

Pada penelitian ini akan menerapkan 1) metode baru *split* atribut yaitu dengan menghitung nilai *average gain* yang dikalikan dengan nilai selisih dari misklasifikasi sebelum *di-split* dan sesudah *di-split*. 2) menerapkan *pruning* yang yang terdiri dari *threshold pruning* dan *cost complexity pruning* guna mengatasi *over-fitting*.

3 METODE YANG DIUSULKAN

Dataset yang digunakan dalam penelitian ini berasal dari yaitu (1) *Breast Cancer Wisconsin*, (2) *Vote*, (3) *Flare1, (4) Hepatitis, (5) Pima Indian Diabetes*. Kelima *dataset* ini dipilih karena kelima *dataset* tersebut populer digunakan, mempunyai angka yang yang proporsional dan mempunyai *missing value*. Dalam *dataset* ini dibagi dengan 90% sebagai data *training* dan 10% sebagai data *testing*.

Tabel 1. *Dataset* yang Digunakan dalam Eksperimen

Dataset	Jumlah *Record*	Jumlah Atribut	Jumlah Atribut Nominal	Jumlah Atribut Numerik	Missing Value	Jumlah *Class*
Breast Cancer Wisconsin	286	9	9	0	16	2
Vote	435	6	6	0	288	2
Flare1	323	12	12	0	5	2
Pima Indian Diabetes	768	8	0	8	752	2
Hepatitis	155	19	15	4	122	2

Dataset yang digunakan dalam penelitian ini mempunyai *missing value* yang harus diperlakukan secara khusus. Adapun penanganan *missing value* menurut Han dan Kamber (Han, Jiawei; Kamber, Micheline; Pei, 2012) adalah:

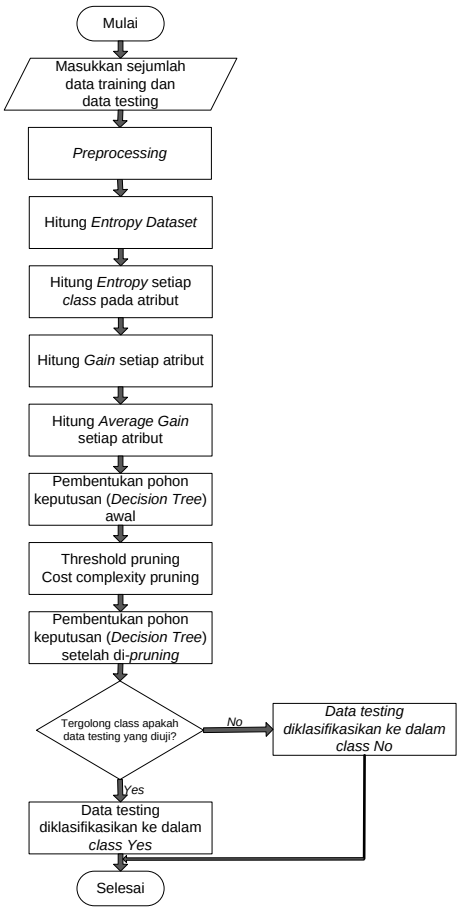
1. Mengabaikan *tuple* yang berisi *missing value*.

Copyright © 2015 IlmuKomputer.Com

2. Mengganti *missing value* secara manual.
3. Mengganti *missing value* dengan konstanta global (misal "*Unknown"* atau ∞).
4. Mengganti *missing value* dengan nilai *mean* atau *median* dari atribut.
5. Mengganti *missing value* dengan nilai *mean* atau *median* dari semua sampel.
6. Mengganti *missing value* dengan nilai kemungkinan terbanyak dari *dataset*.

Pada penelitian ini, *missing value* pada *dataset nominal* akan digantikan dengan nilai yang mempunyai frekuensi terbanyak pada *dataset*. Sedangkan pada *dataset numerik* maka *missing value* digantikan dengan nilai *median* dari atribut.

Selanjutnya kami mengusulkan metode AG, dimana AG adalah metode *split* atribut menggunakan *average gain* yang dikalikan dengan selisih misklasifikasi. Setelah proses *split* atribut dilanjutkan dengan teknik pruning. Teknik *pruning* yang digunakan yaitu *threshold pruning* dan *cost complexity pruning*. Metode AG yang diintegrasikan dengan *threshold pruning* dan *cost complexity pruning* selanjutnya dalam penelitian ini disebut AG_*Pruning*. Metode split atribut yang kami usulkan ditunjukkan pada Gambar 1.



Gambar 1. Metode Split Atribut yang Diusulkan

Berikut alur *pseudocode* dari metode yang diusulkan:

1. Masukkan *dataset*
2. Normalisasi *dataset*
3. Hitung *Entropy dataset*

$$info(S) = -\sum_{j=1}^{k} \frac{freq(Cj,S)}{|S|} \; x \; log_2(\frac{freq(C_j,S)}{|S|}) \; bits$$

4. Hitung *Entropy* setiap *class* pada atribut

$$info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \; x \; info(T_i)$$

5. Hitung *Gain* setiap atribut

$$Gain\ (X) = info(S) - \; info_x(T)$$

6. Hitung *Average Gain* setiap atribut

$$Split\ Atribut = \frac{\left(2^{Averagegain\left(A_i,T\right)}-1\right) x\ Redu\_Mc(A_i)}{TC(A_i)+1}$$

7. Pembentukan pohon keputusan *(Decision Tree)* awal
8. Melakukan *pruning* daun keputusan *(leaf decision)* menggunakan rumus *threshold pruning*

$$Threshold = \frac{(\propto^2+1)x\ Nm\ x\ (FP+FN)}{\propto^2\ x\ Nm + (FP+FN)}$$

9. Melakukan *pruning subtree* menggunakan rumus *cost complexity pruning*

$$\alpha = \frac{\varepsilon(pruned(T,t),S)-\varepsilon(T,S)}{|leaves(T)|-|leaves(pruned(T,t)}$$

10. Pembentukan pohon keputusan *(Decision Tree)* setelah di-*pruning*
11. Pengklasifikasi akhir, dimana data *testing* diklasifikasikan menjadi *class Yes* atau *No*.

Metode *split* atribut yang diusulkan didesain untuk mencegah bias yang muncul dari atribut. Persamaan *split* atribut yang digunakan ditunjukkan melalui Persamaan 3.1.

$$Split\ Atribut = \frac{\left(2^{Averagegain\left(A_i,T\right)}-1\right) x\ Redu\_Mc(A_i)}{TC(A_i)+1} \quad (3.1)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

$TC(A_i)$ = *test cost* atribut A_i
Redu_Mc(A_i) = selisih dari *misclassification cost*

Redu_Mc(A_i) didapat dari rumus dibawah ini:

$$Redu_{Mc(A_i)} = Mc - \sum_{i=0}^{n} Mc(A_i) \quad (3.2)$$

Berdasarkan persamaan diatas, berikut merupakan keterangannya:

Mc = *misclassification cost* atribut A_i sebelum tes
$\sum_{i=0}^{n} Mc(A_i)$ = jumlah total *misclassification cost* atribut A_i setelah *di-split*

Nilai *test cost* dipertimbangkan karena tujuan dari metode yang diusulkan adalah untuk mengurangi atau meminimalkan *misclassification cost.*

Node yang terpilih adalah yang memenuhi syarat dibawah ini:

1. Atribut yang mempunyai nilai *split* atribut *average gain* tertinggi.
2. *Threshold*

Jika satu atau dua kondisi tersebut di atas tidak sesuai maka algoritma yang diusulkan adalah dengan memilih *node* yang mempunyai nilai *average gain* urutan ke-2 kemudian dilanjutkan dengan menguji *node* tersebut dengan 2 kondisi tersebut di atas. Jika ditemukan nilai dari sebuah atribut mempunyai nilai yang sama maka dipilih atribut yang mempunyai nilai *Redu_Mc* yang lebih besar.

Metode *pruning* yang diusulkan dalam penelitian ini dengan mengkombinasikan *threshold pruning* dan *cost complexity pruning.*

1. *Threshold pruning*

Threshold pruning memperhitungkan *misclassification costs* untuk membuat *cost reduction* pada masing-masing *split* lebih signifikan (Zhang, 2012).

Persamaan *threshold pruning* ditunjukkan pada Persamaan 3.3.

$$Threshold = \frac{(\propto^2+1)x\ Nm\ x\ (FP+FN)}{\propto^2 x\ Nm+(FP+FN)} \quad (3.3)$$

Copyright © 2015 IlmuKomputer.Com

Berdasarkan persamaan di atas, berikut merupakan keterangannya:

α = parameter
Nm = jumlah minimal sampel
FP = *False Positive*
FN = *False Negative*

Parameter diatas didapat dari rentang antara jumlah minimal pada sampel yang didapat dari *10-fold cross validation* sampai dengan jumlah *misclassification reduction* (*FP* + *FN*). Dalam penelitian ini, peneliti menentukan nilai parameter α =1 dengan melalui *trial and error*.

Nilai *threshold* yang dihasilkan digunakan untuk menentukan apakah sebuah atribut perlu dipangkas atau tidak. Jika dipangkas maka atribut tersebut akan digantikan dengan daun keputusan *(decision leaf).*

2. *Cost complexity pruning*

Teknik *pruning* ini mempertimbangkan *cost complexity* dari pohon (*tree*) yaitu jumlah daun-daun *(leaves)* dalam pohon *(tree)* dan *error rate* dalam pohon *(tree)* (Rokach & Maimon, 2005). *Cost complexity pruning* terbagi menjadi 2 proses yaitu urutan dari pohon (*tree*) $T_0, T_1,\ldots,T_k$ dimana T_0 merupakan pohon (*tree*) asli sebelum *di-pruning* dan T_k adalah akar pohon (*root tree*). Tahap selanjutnya, salah satu dari pohon (*tree*) tersebut *di-pruning* berdasar perhitungan Persamaan 2.17.

$$\alpha = \frac{\varepsilon(pruned(T,t),S) - \varepsilon(T,S)}{|leaves(T)| - |leaves(pruned(T,t)} \quad (3.4)$$

Jika *subtree* menghasilkan *cost complexity* lebih rendah maka *subtree* akan di-*pruning*.

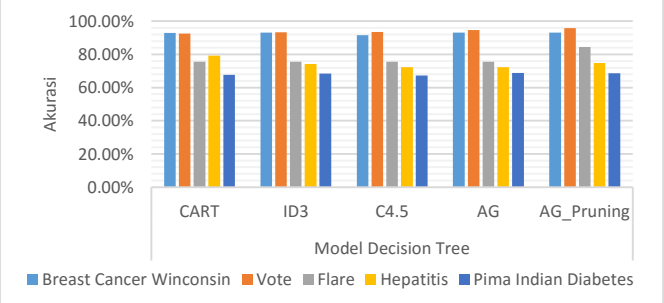
4 HASIL EKSPERIMEN

Eksperimen dilakukan menggunakan komputer personal Intel Core i3, 4 GB RAM, sistem operasi Windows 7 dan Rapid Miner 5.2.003.

Pengukuran model dilakukan dengan mengujinya menggunakan 5 *dataset UCI Repository* (*Breast Cancer Wisconsin, Vote, Flare1, Hepatitis dan Pima Indian Diabetes*). Model yang diuji adalah model *Decision Tree Classification And Regression Tree* (CART), *Iterative Dichotomiser 3* (ID3), C4.5 dan metode yang diusulkan, yaitu *Average Gain* (AG) dan *Average Gain* yang *di-pruning (*AG_*Pruning).*

Tabel 2. Rekap Pengukuran Akurasi Model Decision Tree

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	92,85%	93,13%	91,56%	93,16%	93,21%
Vote	92,64%	93,33%	93,56%	94,71%	95,86%
Flare1	75,54%	75,54%	75,54%	75,54%	84,52%
Hepatitis	79,25%	79,25%	79,25%	79,25%	79,25%
Pima Indian Diabetes	67,71%	67,71%	67,71%	67,71%	67,71%

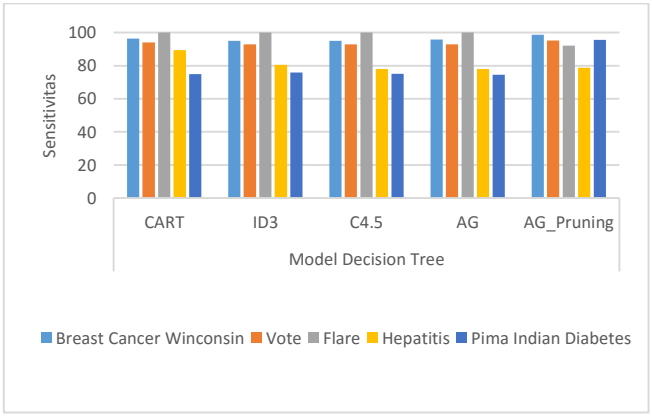


Gambar 2. Diagram Perbandingan Akurasi

Hasil pengukuran model *Decision Tree* untuk pengukuran akurasi ditunjukkan pada Tabel 2. Gambar 2 menunjukkan bahwa akurasi dari AG_*Pruning* meningkat pada 3 *dataset* yaitu *Breast Cancer Wisconsin, Vote* dan *Flare1*.

Tabel 3. Rekap Pengukuran Sensitivitas Model *Decision Tree*

Dataset	Model Decision Tree				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	96,51%	94,98%	94,98%	95,85%	98,69%
Vote	94,05%	92,86%	92,86%	92,86%	95,24%
Flare1	100%	100%	100%	100%	92,21%
Hepatitis	89,43%	89,43%	89,43%	89,43%	89,43%
Pima Indian Diabetes	75,00%	75,00%	75,00%	75,00%	75,00%

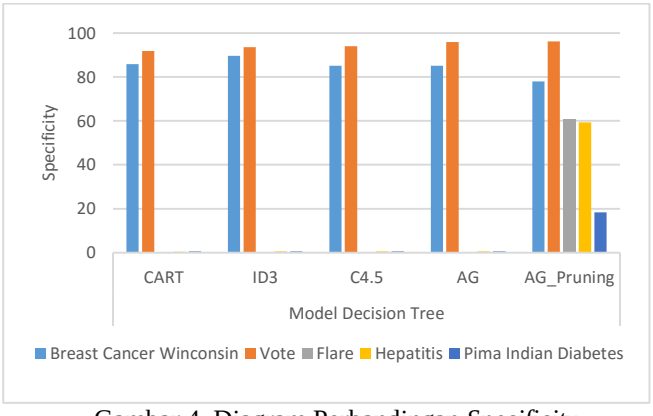


Gambar 3. Diagram Perbandingan Sensititivitas

Hasil pengukuran model *Decision Tree* untuk pengukuran sensitivitas ditunjukkan pada Tabel 3. Gambar 3 menunjukkan bahwa sensitivitas dari AG_*Pruning* meningkat hanya pada 2 *dataset* yaitu *Breast Cancer Wisconsin* dan *Vote*. Hanya pada *dataset Flare1* mengalami penurunan sensitivitas.

Tabel 4. Rekap Pengukuran *Specificity* Model *Decision Tree*

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	85,89%	89,63%	85,06%	85,06%	78,01%
Vote	91,76%	93,63%	94,01%	95,88%	96,25%
Flare1	0	0	0	0	60,76%
Hepatitis	0,41%	0,41%	0,41%	0,41%	0,41%
Pima Indian Diabetes	0,54%	0,54%	0,54%	0,54%	0,54%



Gambar 4. Diagram Perbandingan *Specificity*

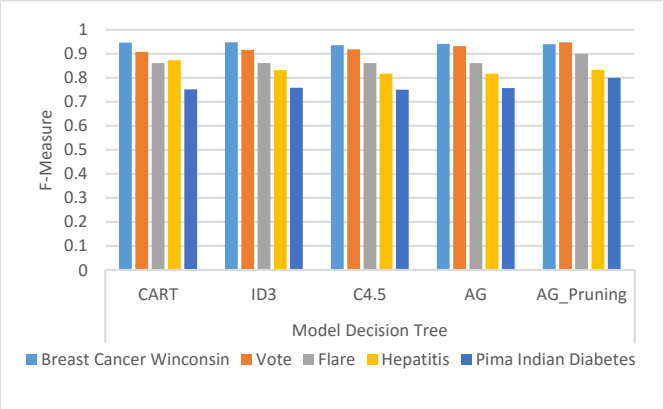
Hasil pengukuran model *Decision Tree* untuk pengukuran *specificity* ditunjukkan pada Tabel 4. Gambar 4 menunjukkan bahwa *specificity* dari AG_*Pruning* meningkat hanya pada

Copyright © 2015 IlmuKomputer.Com

dataset Vote dan *Flare1*. Sedangkan pada *dataset Breast Cancer Wisconsin* mengalami penurunan *specificity*.

Tabel 5. Rekap Pengukuran *F-Measure* Model *Decision Tree*

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	0,946	0,948	0,936	0,941	0,939
Vote	0,908	0,915	0,918	0,931	0,947
Flare1	0,861	0,861	0,861	0,861	0,900
Hepatitis	0,873	0,873	0,873	0,873	0,873
Pima Indian Diabetes	0,752	0,752	0,752	0,752	0,752

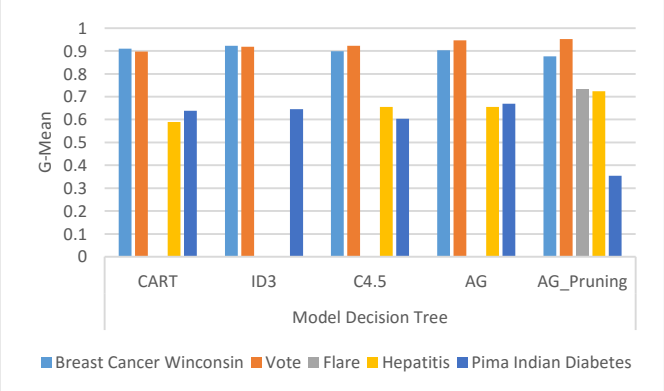


Gambar 5. Diagram Perbandingan *F-Measure*

Hasil pengukuran model *Decision Tree* untuk pengukuran *F-Measure* ditunjukkan pada Tabel 5. Gambar 5 menunjukkan bahwa *F-Measure* dari AG_*Pruning* meningkat pada dataset *Vote* dan *Flare1*. Sedangkan pada *dataset Breast Cancer Wisconsin* mengalami penurunan *F-Measure*.

Tabel 6. Rekap Pengukuran *G-Mean* Model *Decision Tree*

Dataset	Model *Decision Tree*				
	CART	*ID3*	*C4.5*	*AG*	*AG_Pruning*
Breast Cancer Wisconsin	0,910	0,923	0,899	0,903	0,877
Vote	0,897	0,919	0,923	0,946	0,952
Flare1	0	0	0	0	0,731
Hepatitis	0,589	0,589	0,589	0,589	0,589
Pima Indian Diabetes	0,638	0,638	0,638	0,638	0,638



Gambar 6. Diagram Perbandingan *G-Mean*

Hasil pengukuran model *Decision Tree* untuk pengukuran *G-Mean* ditunjukkan pada Tabel 6. Gambar 6 menunjukkan bahwa *G-Mean* dari AG_*Pruning* meningkat pada *dataset Vote* dan *Flare1*. Sedangkan penurunan *G-Mean* terjadi pada *dataset Breast Cancer Wisconsin*.

Untuk mengetahui rangking peningkatan kinerja maka dilakukan uji statistik. Uji statistik yang digunakan adalah uji *Friedman*. Tabel 7 menunjukkan peringkat pengukuran kinerja *Average Gain* (AG) jika dibandingkan dengan *Classification And Regression Tree* (CART), *Iterative Dichotomiser 3* (ID3), C4.5.

Tabel 7. Peringkat Pengukuran Kinerja Pada CART, ID3, C4.5 dan AG

Kinerja	Mean Rank			
	CART	ID3	C4.5	AG
Akurasi	2,30	2,70	1,80	3,20
Sensitivitas	3,30	2,60	2,10	2,00
Specificity	1,90	2,90	2,20	3,00
F-Measure	2,50	3,10	1,80	2,60
G-Mean	1,90	3,10	2,00	3,00

Berdasarkan hasil uji *Friedman*, akurasi dan *specificity* AG berada pada peringkat 1.

Tabel 8 menunjukkan peringkat pengukuran kinerja *Average Gain* yang di-*pruning* (AG_*Pruning*) jika dibandingkan dengan *Classification And Regression Tree* (CART), *Iterative Dichotomiser 3* (ID3), C4.5 dan *Average Gain* (AG).

Tabel 8. Peringkat Pengukuran Kinerja Pada CART, ID3, C4.5, AG dan AG_*Pruning*

Kinerja	Mean Rank				
	CART	ID3	C4.5	AG	AG_*Pruning*
Akurasi	2,50	2,70	1,80	3,40	4,60
Sensitivitas	3,70	3,00	2,30	2,20	3,80
Specificity	2,10	3,10	2,40	3,20	4,20
F-Measure	2,90	3,30	1,80	2,80	4,20
G-Mean	2,30	3,50	2,40	3,40	3,40

Berdasarkan hasil uji *Friedman*, kinerja AG_*Pruning* yang meliputi akurasi, sensitivitas, *specificity*, *F-Measure* dan *G-Mean* berada pada peringkat 1.

5 KESIMPULAN

Pada pengukuran akurasi dan sensitivitas AG dapat meningkatkan kinerja algoritma C4.5 dan melalui uji *Friedman* AG berada di peringkat 1. Pada penelitian ini, pengukuran akurasi, sensitivitas, *specificity*, *F-Measure* dari model AG_*Pruning* menunjukkan bahwa AG_*Pruning* dapat meningkatkan kinerja algoritma C4.5. Berdasarkan hasil uji *Friedman* model AG_*Pruning* menunjukkan peningkatan kinerja dan berada di peringkat 1 dibanding CART, ID3, C4.5 dan AG. Model AG_*Pruning* juga menghasilkan pohon keputusan (*Decision Tree*) yang lebih kecil dibanding CART, ID3, C4.5 dan AG. Hasil penelitian ini menunjukkan bahwa *threshold pruning* dan *cost complexity pruning* dapat mengatasi permasalahan *over-fitting*.

REFERENSI

Abellán, J. (2013). Ensembles of decision trees based on imprecise probabilities and uncertainty measures, *14*, 423–430.

C. Sammut, G. W. (2011). *Encyclopedia of Machine Learning*. (C. Sammut & G. I. Webb, Eds.). Boston, MA: Springer US. doi:10.1007/978-0-387-30164-8

Duchessi, P., & Lauría, E. J. M. (2013). Decision tree models for profiling ski resorts' promotional and advertising strategies and

Copyright © 2015 IlmuKomputer.Com

the impact on sales. *Expert Systems with Applications*, *40*(15), 5822–5829. doi:10.1016/j.eswa.2013.05.017

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques*. (Springer, Ed.) (12th ed., Vol. 12). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-19721-5

Han, Jiawei; Kamber, Micheline; Pei, J. (2012). *Data Mining Concepts and Techniques*. *Morgan Kaufmann* (Third Edit., Vol. 40, p. 9823). Morgan Kaufmann Publishers. doi:10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

Larose, D. T. (2005). *Discovering Knowledge in Data*. United States of America: John Wiley & Sons, Inc.

Larose, D. T. (2006). *Data Mining Methods And Models*. New Jersey: A John Wiley & Sons, Inc Publication.

Mantas, C. J., & Abellán, J. (2014). Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, *41*(10), 4625–4637. doi:10.1016/j.eswa.2014.01.017

Otero, F. E. B., Freitas, A. A., & Johnson, C. G. (2012). Inducing decision trees with an ant colony optimization algorithm. *Applied Soft Computing*, *12*(11), 3615–3626. doi:10.1016/j.asoc.2012.05.028

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. The Morgan Kaufmann Publishers.

Rokach, L., & Maimon, O. (2005). Decision Tree. *Data Mining and Knowledge Discovery Handbook*, pp 165–192. doi:10.1007/978-0-387-09823-4_9

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, *40*(15), 5916–5923. doi:10.1016/j.eswa.2013.05.021

Setsirichok, D., Piroonratana, T., Wongseree, W., Usavanarong, T., Paulkhaolarn, N., Kanjanakorn, C., … Chaiyaratana, N. (2012). Classification of complete blood count and haemoglobin typing data by a C4.5 decision tree, a naïve Bayes classifier and a multilayer perceptron for thalassaemia screening. *Biomedical Signal Processing and Control*, *7*(2), 202–212. doi:10.1016/j.bspc.2011.03.007

T Warren Liao, E. T. (2007). *Recent Advances in Data Mining of Enterprise Data : Algorithms and Applications* (Vol.6 ed.). World Scientific Publishing Co.

Ture, M., Tokatli, F., & Kurt, I. (2009). Using Kaplan–Meier analysis together with decision tree methods (C&RT, CHAID, QUEST, C4.5 and ID3) in determining recurrence-free survival of breast cancer patients. *Expert Systems with Applications*, *36*(2), 2017–2026. doi:10.1016/j.eswa.2007.12.002

Wang, T., Qin, Z., Jin, Z., & Zhang, S. (2010). Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. *Journal of Systems and Software*, *83*(7), 1137–1147. doi:10.1016/j.jss.2010.01.002

Wang, T., Qin, Z., Zhang, S., & Zhang, C. (2012). Cost-sensitive classification with inadequate labeled data, *37*, 508–516. doi:10.1016/j.is.2011.10.009

Zhang, S. (2012). Decision tree classifiers sensitive to heterogeneous costs. *Journal of Systems and Software*, *85*(4), 771–779. doi:10.1016/j.jss.2011.10.007

BIOGRAFI PENULIS

Erna Sri Rahayu. Memperoleh gelar M.Kom dari Universitas Dian Nuswantoro, Semarang. Menjadi pendidik di SMP Negeri 1 Pabelan dengan mata pelajaran yang diampu Teknologi Informasi dan Komunikasi (TIK). Minat penelitian pada saat ini di bidang data mining.

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Catur Supriyanto. Memperoleh gelar Master dari University Teknikal Malaysia Melaka (UTEM), Malaysia. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Minat penelitiannya pada bidang information retrieval, machine learning, soft computing dan intelligent system.

Copyright © 2015 IlmuKomputer.Com

Penerapan Bootstrapping untuk Ketidakseimbangan Kelas dan *Weighted* Information Gain untuk *Feature Selection* pada Algoritma Support Vector Machine untuk Prediksi Loyalitas Pelanggan

Abdul Razak Naufal, Romi Satria Wahono dan Abdul Syukur
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
abdul.razak.naufal@gmail.com, romi@romisatriawahono.net, abdul_s@dosen.dinus.ac.id

Abstrak: Prediksi loyalitas pelanggan merupakan sebuah strategi bisnis yang penting bagi industri telekomunikasi modern untuk memenangkan persaingan global, karena untuk mendapatkan pelanggan baru biayanya lebih mahal lima sampai enam kali lipat daripada mempertahankan pelanggan yang sudah ada. Klasifikasi loyalitas pelanggan bertujuan untuk mengidentifikasi pelanggan yang cenderung beralih ke perusahaan kompetitor yang sering disebut *customer churn*. Algoritma Support Vector Machine (SVM) adalah algoritma klasifikasi yang juga berfungsi untuk memprediksi loyalitas pelanggan. Penerapan algoritma SVM dalam memprediksi loyalitas pelanggan mempunyai kelemahan yang mempengaruhi keakuratan dalam memprediksi loyalitas pelanggan yaitu sulitnya pemilihan fungsi kernel dan penentuan nilai parameternya. Dataset yang besar pada umumnya mengandung ketidakseimbangan kelas (*class imbalance*), yaitu adanya perbedaan yang signifikan antar jumlah kelas, yang mana kelas negatif lebih besar daripada kelas positif. Dalam penelitian ini diusulkan metode *resampling* bootstrapping untuk mengatasi ketidakseimbangan kelas. Selain itu dataset juga mengandung fitur yang tidak relevan sehingga dalam pemilihan fitur dalam penelitian ini digunakan metode dua fitur seleksi yaitu Forward Selection (FS) dan *Weighted* Information Gain (WIG). FS berfungsi untuk menghilangkan fitur yang paling tidak relevan serta membutuhkan waktu komputasi yang relatif pendek dibandingkan dengan backward elimination dan stepwise selection. WIG digunakan untuk memberi nilai bobot pada setiap atribut, karena WIG lebih cocok digunakan dalam memilih fitur terbaik daripada Principal Component Analysis (PCA) yang biasa digunakan untuk mereduksi data yang berdimensi tinggi. Tujuan pembobotan ini untuk merangking atribut yang memenuhi kriteria (*threshold*) yang ditentukan dipertahankan untuk digunakan oleh algoritma SVM. Sedangkan untuk pemilihan parameter algoritma SVM dengan menggunakan metode *grid search*. Metode *grid search* dapat mencari nilai parameter terbaik dengan memberi *range* nilai parameter. *Grid search* juga sangat handal jika diaplikasikan pada dataset yang mempunyai atribut sedikit daripada menggunakan *random search*. Hasil eksperimen dari beberapa kombinasi parameter dapat disimpulkan bahwa prediksi loyalitas pelanggan dengan menggunakan sampel bootstrapping, FS-WIG serta *grid search* lebih akurat dibanding dengan metode individual SVM.

Kata Kunci: loyalitas pelanggan, boostrapping, *weighted* information gain, support vector machine

1 PENDAHULUAN

Industri telekomunikasi merupakan salah satu industri teknologi tinggi yang berkembang paling cepat diantara industri disektor lainnya. Terbukanya persaingan bebas diperusahaan jasa telekomunikasi juga merupakan salah satu tantangan serius yang diharus dihadapi oleh industri telekomunikasi (Huang, Kechadi, & Buckley, 2012), dimana dengan banyaknya kompetitor dibidang telekomunikasi, memaksa perusahaan telekomunikasi harus lebih meningkatkan pelayanan terhadap pelanggan agar tidak pindah ke operator lain (Jadhav & Pawar, 2011), karena biaya untuk mendapatkan pelanggan baru lebih mahal daripada mempertahankan pelanggan yang sudah ada. Dengan *market share* yang besar maka perlu dilakukan usaha-usaha untuk mempertahankan pelanggan agar *market share* yang telah diraih tidak menurun, sehingga perlu menerapkan suatu strategi yang handal dengan mengeluarkan sedikit biaya terhadap memprediksi loyalitas pelanggan namun hasilnya yang besar bisa diraih. Karena untuk mendapatkan pelanggan baru biayanya lebih mahal lima sampai enam kali lipat daripada mempertahankan pelanggan yang sudah ada (Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012). Pelanggan yang meninggalkan atau berhenti terhadap layanan yang telah diberikan oleh perusahaan telekomunikasi dan menjadi pelanggan perusahaan komunikasi kompetitor disebut dengan perilaku *customer churn* (Yu, Guo, Guo, & Huang, 2011), perilaku ini menjadi salah satu kerugian pendapatan perusahaan. Hal ini juga telah menjadi isu penting dan merupakan salah satu tantangan utama perusahaan yang harus dihadapi di era global ini.

Tiap bulan rata-rata pelanggan yang pindah ke operator lain diperusahaan telekomunikasi di Eropa mencapai 8%-12% dan biaya perpindahan pelanggan sangat besar yaitu mencapai sekitar 500 EURO (Richeldi & Perrucci, 2002). Sementara di Amerika Serikat, per bulan *churn rate* pada tahun 1998 adalah 2% sampai 3%. Biaya yang harus dikeluarkan untuk memperoleh seorang pelanggan baru rata-rata mencapai USD 400 dan biaya perpindahan pelanggan pada industri telekomunikasi ini mendekati USD 6,3 juta. Total kerugian per tahun dapat meningkat menjadi USD 9,6 juta apabila kerugian pendapatan per bulan dari pelanggan juga ikut diperhitungkan, sehingga pada suatu perusahaan yang memiliki 1,5 juta pelanggan, pengurangan loyalitas pelanggan dari 2% menjadi 1% dapat meningkatkan pendapatan tahunan sedikitnya sebanyak USD 54 juta dolar (Mozer, Wolniewicz, Grimes, Johnson, & Kaushansky, 2000). Dari data dan fakta inilah bahwa mengatasi perilaku loyalitas pelanggan merupakan suatu bisnis proses yang dapat meningkatkan keuntungan dan bisa menjadi salah satu kunci sukses untuk memenangkan persaingan di dunia binis jasa telekomunikasi. Telekomunikasi merupakan salah satu industri, dimana basis pelanggan memainkan peran penting dalam mempertahankan pendapatan yang stabil, dengan demikian perhatian serius dikhususkan untuk mempertahankan pelanggan sehingga menggeser fokus utama perusahaan dari mencari *market share* menjadi mempertahankan pelanggan.

Prediksi loyalitas pelanggan merupakan sebuah strategi bisnis yang sangat penting bagi perusahaan telekomunikasi

Copyright @ 2015 IlmuKomputer.Com

modern. Pendekatan mendasar yang dilakukan untuk memprediksi loyalitas pelanggan dapat diperoleh dari kemungkinan loyalitas pelanggan dengan menggunakan metode prediksi. Prediksi loyalitas pelanggan bertujuan untuk mengidentifikasi pelanggan yang cenderung beralih ke perusahaan kompetitor (Z.-Y. Chen, Fan, & Sun, 2012). Oleh karena itu, banyak perusahaan bisnis menggunakan prediksi loyalitas pelanggan untuk mengidentifikasi pelanggan yang cenderung berpindah operator. Tindakan prediksi loyalitas pelanggan ini dapat diambil untuk membantu perusahaan dalam meningkatkan strategi intervensi untuk meyakinkan pelanggan supaya tetap berlangganan dan mencegah hilangnya bisnis perusahaan (Z.-Y. Chen et al., 2012). Hal ini memungkinkan pengelolaan pelanggan yang efisien dan alokasi yang lebih baik dari sumber daya marketing untuk kampanye dalam mempertahankan pelanggan.

Serangkaian proses untuk mendapatkan pengetahuan atau pola dari sekumpulan data disebut dengan data mining (Witten, I. H., Frank, E., & Hall, 2011). Klasifikasi merupakan topik penelitian yang penting dalam data mining, karena teknik klasifikasi dapat memecehkan masalah dengan menganalisis data dalam database yang masing-masing data dikelompokkan kedalam kelas tertentu. Ukuran data yang sangat besar pada perusahaan telekomunikasi (Farvaresh & Sepehri, 2011) menjadi rintangan utama dalam mencapai kinerja yang diinginkan untuk memprediksi loyalitas pelanggan.

Ada beberapa teknik data mining untuk memprediksi loyalitas pelanggan yang telah dilakukan oleh para peneliti yaitu (Huang et al., 2012) dengan menggunakan algoritma Neural Network (NN) (Tsai & Lu, 2009), Decision Tree (DT) (Han, Lu, & Leung, 2012), Logistic Regression (LR) (Nie, Rowe, Zhang, Tian, & Shi, 2011), Support Vector Machine (SVM) (Xia & Jin, 2008). Dari beberapa penelitian yang telah dilakukan tersebut dapat disimpulkan bahwa neural network mempunyai kemampuan dalam memprediksi hasil keputusan diagnostik dibandingkan dengan algoritma logistic regression, NN juga mempunyai kemampuan dalam mendekteksi hubungan kompleks yang bersifat *nonlinear* antara faktor prediksi dan hasil prediksi (H. Chen, Zhang, Xu, Chen, & Zhang, 2012), tetapi NN mempunyai kelemahan pada perlunya data training yang besar, sulit mengenali pola apalabila menggunakan data yang berdimensi tinggi sehingga memerlukan waktu komputasi yang lama (Pan, Iplikci, Warwick, & Aziz, 2012) dan sering mengalami *over-fitting* (Rynkiewicz, 2012). Algoritma SVM bisa mengatasi kelemahan pada NN yaitu masalah tidak membutuhkan data training yang besar (Vapnik, 1998) serta memiliki kemampuan generalisasi yang baik ketika diterapkan pada data noise (Farvaresh & Sepehri, 2011) yang secara teori lebih memuaskan daripada metode neural network (Nugroho, 2008). Selain itu algoritma klasifikasi SVM juga mempunyai resiko yang kecil terhadap masalah *over-fitting* dibanding dengan metode lainnya (Han, J., & Kamber, 2012), tetapi algoritma support vector machine mempunyai kelemahan pada sulitnya pemilihan parameter dan fungsi kernel yang optimal untuk mendapatkan pendekatan yang lebih baik yang secara signifikan dapat mempengaruhi akurasinya (Wu, Xindong & Kumar, 2009) (Coussement & Van den Poel, 2008) (Xia & Jin, 2008) (Z. Chen & Fan, 2013) (Wu, 2011). Jadi pemilihan parameter dan fungsi kernel yang tepat sangat mempengaruhi hasil prediksi dalam memprediksi loyalitas pelanggan.

Algoritma Support Vector Machine (SVM) adalah algoritma klasifikasi dalam data mining yang berfungsi untuk memprediksi loyalitas pelanggan, tetapi penerapan algoritma support vector machine dalam memprediksi loyalitas pelanggan mempunyai kelemahan yang mempengaruhi keakuratan dalam memprediksi loyalitas pelanggan yaitu sulitnya pemilihan fungsi kernel dan penentuan nilai parameter yang tepat. Dataset yang besar pada umumnya mengandung ketidakseimbangan kelas (*class imbalance*), yaitu adanya perbedaan yang signifikan antar jumlah kelas, yang mana kelas negatif lebih besar daripada kelas positif. Dalam penelitian ini diusulkan metode *resampling* bootstrapping untuk mengatasi ketidakseimbangan kelas. Selain itu dataset juga mengandung fitur yang tidak relevan sehingga dalam pemilihan fitur dalam penelitian ini digunakan metode dua fitur seleksi yaitu Forward Selection (FS) dan *Weighted* Information Gain (WIG). FS berfungsi untuk menghilangkan fitur yang paling tidak relevan serta membutuhkan waktu komputasi yang relatif. WIG digunakan untuk memberi nilai bobot pada setiap atribut. Tujuan pembobotan ini untuk merangking atribut yang memenuhi kriteria (*threshold*) yang ditentukan dipertahankan untuk digunakan oleh algoritma SVM. Sedangkan untuk pemilihan parameter algoritma SVM dengan menggunakan metode *grid search*. Metode *grid search* dapat mencari nilai parameter terbaik dengan memberi *range* nilai parameter. Sedangkan fungsi kernel yang akan digunakan dalam penelitian ini yaitu dengan menggunakan fungsi kernel Radial Basis Function (RBF).

2 LANDASAN TEORI

2.1 PENELITIAN TERKAIT

Beberapa peneliti menggunakan metode untuk mengatasi sulitnya pemilihan parameter dan fungsi kernel pada algoritma support vector machine diantaranya yang dilakukan oleh Kristof Coussement dan Dirk Van den Poel (Coussement & Van den Poel, 2008), untuk mengatasi sulitnya pemilihan parameter dan fungsi kernel menggunakan fungsi kernel Radial Basis Function (RBF) dengan dua parameter yang bermanfaat mengurangi sulitnya membaca data numerik karena nilai kernel terletak antara nol dan satu. Penelitian yang dilakukan oleh Qi Wu (Wu, 2011), diusulkan metode Adaptive and Couchy Mutation Particle Swarm Optimization (ACPSO) sebagai optimasi parameter SVM, karena ACPSO dianggap sebagai teknik yang baik untuk memecahkan masalah kombinatorial, selain itu juga dapat menentukan *hyperparameter* secara bersamaan dan mempunyai kemampuan pencarial global yang kuat, tetapi ACPSO diyakini memiliki ketergantungan yang sensitif pada parameter dan cenderung terjebak dalam local optimum. Selanjutnya penelitian yang dilakukan oleh Chen dan Fan (Z. Chen & Fan, 2013). menggunakan Multi Kernel Support Vector Regression (MK-SVR) yaitu dengan mengoptimalkan kernel dengan mengkombinasikan beberapa kernel dasar yang masing-masing memiliki *hyperparameter* yang identik, multi kernel ini diformulasikan dengan menerapkan strategi dua iterasi secara berulang-ulang yang bermanfaat untuk mempelajari *lagrange multipliers* oleh *Quadratic Programming* (QP) dan pembobotan koefisien dari dua kernel oleh *linear program* sehingga bisa mengontrol variabel independen. Hasil penelitian terkait ini menunjukkan bahwa penggunaan fungsi kernel dan parameter yang tepat dapat meningkatkan performa algoritma support vector machine.

Ada beberapa metode untuk mengatasi masalah ketidakseimbangan kelas (*class imbalance*) salah satunya adalah dengan teknik *resampling* (Wahono, Herman, & Ahmad, 2014) (Yap et al., 2014) (Farvaresh & Sepehri, 2011) (Burez & Van den Poel, 2009) yang dikolaborasikan dengan seleksi fitur (Khoshgoftaar & Gao, 2009) (Lin, Ravitz, Shyu,

Copyright @ 2015 IlmuKomputer.Com

& Chen, 2008), oleh karena itu untuk mengatasi masalah *class imbalance* akan digunakan metode sampel boostrapping. Setelah mendapatkan data *sampling* hasil bootstrapping, kemudian di filter kembali menggunakan metode forward selection. Forward selection sebagai teknik seleksi fitur yang juga berperan dalam mengangani data yang berdimensi tinggi serta mengandung ketidakseimbangan kelas (Maldonado, Weber, & Famili, 2014) (Idris, Rizwan, & Khan, 2012) dengan memilih subset yang tepat dari set fitur asli, karena tidak semua fitur relevan dengan masalah, bahkan ada beberapa dari fitur tersebut malah menjadi penghalang yang dapat mengurangi akurasi. Pemilihan pembobotan atribut pada penelitian ini dengan menggunakan *Weighted* Information Gain (WIG), karena dengan menggunakan WIG setiap atribut dapat diketahui nilainya dan dapat dipilih yang terbaik (Charu C. Aggarwal, 2008), selain itu WIG merupakan algoritma yang sangat cocok digunakan dalam memilih fitur yang terbaik khususnya dalam merangking data(Novakovic, 2010). Dan untuk pemilihan parameter pada algoritma SVM dengan menggunakan metode *grid search*. Metode *grid search* digunakan karena sangat handal jika diaplikasikan pada dataset yang mempunyai atribut sedikit (Bergstra & Bengio, 2012) daripada metode *random search*.

Pemilihan fungsi kernel yang tepat adalah hal yang sangat penting, karena fungsi kernel ini akan menentukan *feature space* dimana fungsi klasifier akan dicari. Dalam penelitian ini akan diusulkan metode bootstrapping, FS-WIG untuk seleksi fitur dan *grid search* sebagai pemilihan parameter pada algoritma support vector machine. Dalam penelitian ini digunakan fungsi kernel Radial Basis Function (RBF), karena kernel RBF menunjukkan *tradeoff* parameter C dalam algoritma support vector machine yang sangat mempengaruhi hasilnya (Zhou, Liu, & Ye, 2009), beberapa eksperimen juga menunjukkan bahwa kernel RBF menghasilkan tingkat kesalahan klasifikasi yang kecil serta mempercepat perhitungan komputasinya (Xu et al., 2014), hal ini sangat cocok dengan sifat dataset yang besar sebagaimana dataset yang digunakan dalam penelitian ini.

2.2 TINJAUAN PUSTAKA

2.2.1 VARIABEL PREDIKSI LOYALITAS PELANGGAN

Loyalitas pelanggan banyak diteliti dan diterapkan oleh perusahaan, karena dengan mengetahui loyalitas pelanggan perusahaan dapat lebih meningkatkan relasi terhadap pelanggan yang loyal maupun yang tidak loyal, selain itu perusahaan juga dapat menerapkan strategi bisnis untuk memenangkan persaingan dengan perusahaan kompetitor. Ada beberapa faktor dalam menilai loyalitas pelanggan, termasuk menganalisa klasifikasi terhadap *lifecycle* pelanggan. Pelanggan dibagi menjadi 2 jenis, yaitu pelanggan jenis pegawai dan individu. Pelanggan jenis pegawai terdiri dari pegawai asuransi, lembaga pemerintah, angkatan bersenjata dari pemerintah, transportasi dan logistik, departemen energi, pendidikan, pariwisata, hotel dan restoran, bar internet, rumah sakit, bank, operator telekomunikasi dan ISP, agen partai dan organisasi sosial, industri manufaktur, perusahaan besar, menengah dan usaha kecil). Sedangkan pelanggan jenis individu terdiri dari pelanggan pribadi dan pelanggan yang tidak jelas seperti pelanggan yang tinggal dikota dan pelanggan yang tinggal disuatu Negara dalam jangka waktu tertentu (Han et al., 2012). Selain itu faktor dari segi waktu lamanya menjadi pelanggan dibagi menjadi 3 kelompok. Kelompok pertama yang menjadi pelanggan lebih dari 36 bulan tetapi tidak lebih dari 60 bulan, kelompok kedua yang menjadi pelanggan 18 bulan tetapi tidak lebih dari 36 bulan, kelompok ketiga yang menjadi pelanggan kurang dari 18 bulan (Han et al., 2012). Jadi dalam penelitian ini beberapa faktor utama yang menjadi penilaian atas loyalitasnya pelanggan terdiri dari (Huang et al., 2012):

a. *Demographic profiles*
   Menggambarkan pengelompokan demografis atau segmen pasar dan informasi demografis mengandung kemungkinan perilaku pelanggan. Biasanya, informasi ini meliputi usia, pekerjaan, jenis kelamin, dan lain-lain.
b. *Information of grants*
   Beberapa pelanggan telah memperoleh beberapa hibah khusus sehingga tagihan mereka dibayar penuh atau sebagian oleh pihak ketiga. Sebagai contoh, pelanggan yang cacat atau usia lebih dari 80 tahun yang ingin melanjutkan layanan.
c. *Customer account information*
   Informasi ini berisi jenis paket layanan, indikator pengontrol kredit, indikator junk mail, tanggal pertama menggunakan layanan, tanggal pembuatan, frekuensi tagihan, saldo rekening, sewa peralatan, jenis pembayaran dan atribut durasi panggilan, jumlah panggilan dan harga standar dan biaya.
d. *The historical information of bills and payments*
   Informasi ini menyangkut penagihan biaya setiap pelanggan dan jumlah layanan langganan pelanggan dalam setahun.
e. *Complaint Information*
   Keluhan pelanggan merupakan masalah yang terjadi yang merugikan pelanggan kemudian disampaikan kepada perusahaan. Pada perusahaan yang diteliti keluhan yang dicatat adalah keluhan yang disampaikan secara langsung oleh perusahaan ataupun keluhan yang datang saat pelanggan dikunjungi oleh marketing.
f. *Call details*
   Pada perusahaan jasa telekomunikasi rincian panggilan mengacu pada durasi panggilan, harga dan jenis panggilan, misalnya seberapa sering pengguna telepon lokal, interlokal, internasional, atau apakah pelanggan berlangganan juga terhadap produk internet atau yang telah ditawarkan lainnya, dan pelanggan yang jumlah pemakaian teleponnya sedikit, layanan telepon sedikit seperti sms mereka inilah yang masuk ke kelompok pelanggan yang *churn*.
g. *Incoming calls details*
   Rincian panggilan yang diterima termasuk durasi panggilan dan jumlah panggilan yang diterima, jumlah perubahan panggilan, perubahan durasi panggilan yang diterima.

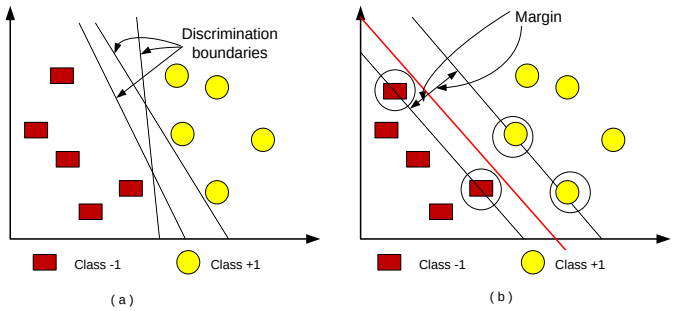
Data seperti detail demografi, tarif pemakaian perorangan tidak digunakan, sehingga hasil yang diperoleh untuk menggelompokkan pelanggan yang berpotensi menjadi tidak loyal dapat dideteksi secara efisien menggunakan *framework* ini sebelum kelompok tersebut masuk ke kelompok yang berpeluang menjadi pelanggan yang loyal. Hasil dari ekperimen terhadap dataset yang ada menunjukkan bahwa pelanggan yang berpotensi untuk *churn* adalah pelanggan yang jumlah pemakaian teleponnya sedikit dan layanan teleponnya juga sedikit seperti sms, mereka inilah yang masuk ke kelompok pelanggan yang tidak loyal (Richter, Yom-Tov, & Slonim, 2010). Loyalitas pelanggan yang akan diukur dalam penelitian ini adalah seberapa banyak pelanggan yang tidak loyal dan benar-benar tidak dapat dipertahankan. Ada beberapa metode untuk mengatasi pelanggan yang tidak loyal, seperti melakukan retensi untuk meningkatkan loyalitas, serta

Copyright @ 2015 IlmuKomputer.Com

perusahaan harus memiliki program yang dapat secara akurat membedakan pelanggan yang berisiko tidak loyal dengan pelanggan yang loyal, agar biaya retensi tidak membesar.

2.2.2 SUPPORT VECTOR MACHINE

Support vector machine (SVM) dikembangkan oleh Boser, Guyon, Vapnik dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*. Gambar 4.1-a memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* : +1 dan –1. *Pattern* yang tergabung pada class –1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* pada *class* +1, disimbolkan dengan warna kuning (lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut (Cortes & Vapnik, 1995). Garis pemisah (*discrimination boundaries*) ditunjukkan pada Gambar 4.1-a merupakan salah satu alternatif garis pemisah yang memisahkan kedua class.



Gambar 4.1*Hyperplane* Terbaik yang Memisahkan Kedua *Class* –1 dan +1 (Nugroho, 2008)

Hyperplane pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai support vektor. Garis solid pada Gambar 4.1-b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada support vector machine (Nugroho, 2008). Karena dengan menemukan *hyperplane* terbaik tersebut bisa meningkatkan keakuratan dalam memprediksi loyalitas pelanggan.

Data yang tersedia dinotasikan sebagai $\overrightarrow{X_i} \in R^d$ sedangkan label masing-masing dinotasikan $y_i \in \{-1 + 1\}$ untuk $i = 1,2,\dots,l$, yang mana l adalah banyaknya data. Diasumsikan kedua *class* –1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi d, yang didefinisikan (Nugroho, 2008):

$$\overrightarrow{W}.\overrightarrow{X} + b = 0 \quad \text{.....} \quad (1)$$

Pattern $\overrightarrow{X_i}$ yang termasuk *class* -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan

$$\overrightarrow{W}.\overrightarrow{X_i} + b \leq -1 \quad \text{.....} \quad (2)$$

Sedangkan patter $\overrightarrow{X_i}$ yang termasuk *class* +1 (sampel positif)

$$\overrightarrow{W}.\overrightarrow{X_i} + b \geq +1 \quad \text{.....} \quad (3)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu $1/\|\vec{w}\|$. Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) problem, yaitu mencari titik minimal persamaan, dengan memperhatikan constraint persamaan.

$$\min_{\vec{w}} \tau(w) = \frac{1}{2}\|\vec{w}\|^2 \quad \text{.....} \quad (4)$$

$$y_i(\overrightarrow{x_i}.\vec{w} + b) - 1 \geq 0, \forall i \quad \text{.....} \quad (5)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya Lagrange Multiplier

$$L(\vec{w}, b, a) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{l} a_i \left(y_i\left((\vec{x}_i.\vec{w} + b) - 1\right)\right) (i = 1,2,\dots,l) \quad \text{.....} \quad (6)$$

a_i adalah *Lagrange multipliers*, yang bernilai nol atau positif $(a_i \geq 0)$. Nilai optimal dari persamaan dapat dihitung meminimalkan L terhadap $\overrightarrow{w_i}$ dan b , dan dengan memaksimalkan L terhadap a_i. Dengan memperhatikan sifat bahwa pada titik optimal gradient $L = 0$, persamaan dapat dimodifikasi sebagai maksimalisasi *problem* yang hanya mengandung a_i saja, sebagaimana persamaan α di bawah.
Maximize:

$$\sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i\, a_j\, y_i\, y_j \overrightarrow{x_i}.\overrightarrow{x_j} \quad \text{.....} \quad (7)$$

Subject to:

$$a_i \geq 0 (i = 1,2,\dots,l)\ \sum_{i=1}^{l} a_i y_i = 0 \quad \text{.....} \quad (8)$$

Dari hasil perhitungan ini diperoleh i yang a_i kebanyakan bernilai positif. Data yang berkorelasi dengan a_i yang positif inilah yang disebut sebagai *support vector*(Cortes & Vapnik, 1995).

2.2.3 KERNEL TRIKS

Pada umumnya permasalahan dalam domain dunia nyata (*real world problem*) datanya sangat sedikit sekali yang bersifat *linear*, kebanyakan data bersifat *non-linear*. Untuk menyelesaikan kasus *non-linear*, perhitungan SVM dimodifikasi menjadi dua tahap, dimana didalamnya memanfaatkan konsep yang disebut *Kernel trick* (Nugroho, 2008). Ide yang mendasarinya adalah mengubah data bersifat *non-linear* dan dimensi dari *feature space* cukup tinggi, maka data pada *input space* dapat dipetakan ke *feature space* yang baru, dimana pola-pola tersebut pada probabilitas tinggi dapat dipisahkan secara *linear*. Untuk menyelesaikan masalah *non-linear*, support vector machine dimodifikasi dengan memasukkan *kernel trick* yang mengubah data *non-linear* ke data *linear* (Hamel, 2009). *Kernel trick* dapat dirumuskan dengan:

$$K(\overrightarrow{X_i}, \overrightarrow{X_j}) = \Phi(\overrightarrow{X_i}).\Phi(\overrightarrow{X_j}) \quad \text{.....} \quad (9)$$

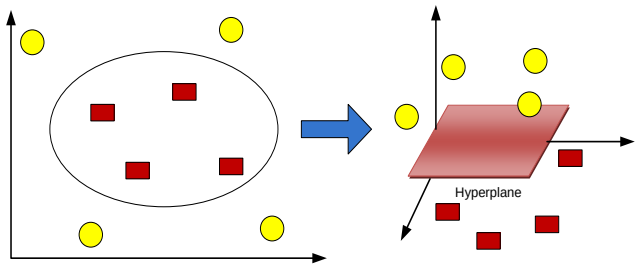
Ilustrasi dari konsep ini dapat dilihat pada gambar 2.6. Pada Gambar 2.6-a diperlihatkan data pada *class* kuning dan data pada *class* merah yang berada pada *input space* berdimensi dua tidak dapat dipisahkan secara *linear*. Selanjutnya Gambar 2.6-b menunjukkan bahwa fungsi Φ memetakan tiap data pada

Copyright @ 2015 IlmuKomputer.Com

input space tersebut ke ruang vector baru yang berdimensi lebih tinggi (dimensi 3), dimana kedua *class* dapat dipisahkan secara *linear* oleh sebuah *hyperplane*. Notasi matematika dari *mapping* ini adalah sebagai berikut:

$\Phi : R^d \rightarrow R^q \; d < q$ ..... (10)

Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*, sebaliknya dua data yang berjarak jauh pada *input space* akan juga berjarak jauh pada *feature space*.



(a) *Input space X* (b) *High-dimensional feature space* Φ *(X)*

Gambar 2.1 Kedua *Class* Dipisahkan Secara *Linear* oleh Sebuah *Hyperplane* (Nugroho, 2008)

Selanjutnya proses pembelajaran pada SVM dalam menemukan titik-titik *support vector*, hanya bergantung pada *dot product* dari data yang sudah berdimensi lebih tinggi, yaitu $\Phi\left(\overrightarrow{X_i}\right).\Phi\left(\overrightarrow{X_j}\right)$. Karena umumnya transformasi Φ ini tidak diketahui, dan sangat sulit untuk dipahami secara mudah, maka perhitungan *dot product* tersebut dapat digantikan dengan fungsi kernel $K\left(\overrightarrow{X_i}, \overrightarrow{X_j}\right)$ yang mendefinisikan secara implisit transformasi Φ. *Kernel trick* memberikan berbagai kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support vector*, hanya cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi *non-linear* Φ. Selanjutnya hasil klasifikasi dari data $\overrightarrow{X}$ diperoleh dari persamaan berikut (Cortes & Vapnik, 1995):

$f\left(\Phi\left(\vec{X}\right)\right) = \vec{w}.\Phi\left(\overrightarrow{X}\right) + \mathrm{b}$ ..... (11)

$= \sum_{i=1,\overrightarrow{X_i} \in SV}^{n} a_i \, y_i \, \Phi\left(\vec{X}\right).\Phi\left(\overrightarrow{X_i}\right) + b$ ..... (12)

$= \sum_{i=1,\overrightarrow{X_i} \in SV}^{n} a_i \, y_i \, K\left(\overrightarrow{X}, \overrightarrow{X_i}\right) + b$ ..... (13)

Support vektor pada persamaan di atas dimaksudkan dengan subset dari training set yang terpilih sebagai support vector, dengan kata lain data $\overrightarrow{X_i}$ yang berkorespondensi pada $a_i \geq 0$. Fungsi kernel yang biasanya dipakai dalam dalam literatur SVM yaitu(Nugroho, 2008):

a. Linear : $K(x, y) = x.y$
b. Polynomial : $K(x, y) = (x.y + 1)^{\,d}$
c. Radial Basis Funtion (RBF):
$$K(x, y) = exp\left(-\frac{\|x_i \, x_j\|^2}{2\sigma^2}\right)$$
d. Sigmoid : $K(x, y) = tanh(\beta x^T x_i + \beta_1)$,dimana $\beta, \beta_1 \in \mathcal{R}$

2.2.4 BOOTSTRAPPING

Bootstrap adalah metode *resampling* yang telah diterapkan secara luas dan memungkinkan pembuatan model yang lebih realistis (Efron & Tibshirani, 1998). Sebelumnya setiap kali sampel yang diambil dari dataset untuk membentuk data *training* atau *testing* itu diambil tanpa penggantian, artinya contoh yang sama setelah dipilih tidak dapat dipilih lagi (Witten, I. H., Frank, E., & Hall, 2011), jadi dengan menggunakan bootstrapping sampel yang sudah dipilih dapat dipilih kembali, hal ini memungkinkan penilaian terhadap estimasi akurasi dengan random sampling dengan penggantian dari dataset yang asli sehingga bisa meningkatkan akurasi dan mempercepat waktu komputasinya.

Metode bootstrap dilakukan dengan mengambil sampel dari sampel yang asli dengan ukuran sama dengan aslinya dan dilakukan dengan penggantian, biasanya ukuran resampling diambil secara ribuan kali agar dapat mewakili data populasinya. Kedudukan sampel asli dalam metode bootstrap dipandang sebagai populasi. Metode penyampelan ini biasa disebut dengan *resampling bootstrap with replacement* (Tian, Song, Li, & de Wilde, 2014). Bootstrap dapat digunakan untuk mengatasi permasalahan statistik baik dalam masalah data yang sedikit, data yang menyimpang dari asumsinya maupun data yang tidak memiliki asumsi dalam distribusinya. Dalam beberapa aplikasi yang mengolah data yang besar biasanya menggunakan metode sampling (Witten, I. H., Frank, E., & Hall, 2011) untuk menyaring agar data yang diproses dapat menjadi lebih kecil.

Metode distribusi populasi didasarkan pada prinsip bootstrap yang dapat digunakan prinsip plug-in untuk memperkirakan parameter dari sampel. Untuk memperkenalkan secara singkat prinsip plug-in yang diasumsikan bahwa θ adalah parameter dan F adalah kemungkinan distribusi. Tujuannya adalah untuk menghitung θ dengan menerapkan beberapa prosedur numerik $\mathrm{t}(.)$ yaitu:

$\theta = t(F)$ ..... (14)

Oleh karena itu estimasinya dapat didefinisikan sebagai

$\hat{\theta} = t\left(\hat{F}\right)$ ..... (15)

dimana $\hat{F}$) fungsi distribusi empirik dari sampel random (Tian et al., 2014). Dalam hal ini parameter θ akan menjadi indeks sensitivitas atau kesalahan prediksi dalam menilai pelanggan yang loyal atau tidak loyal. Secara ringkas langkah-langkah bootstrap adalah sebagai berikut (Efron & Tibshirani, 1998):

1. Menentukan jumlah *B* sampel independen bootstrap *X*1, X*2, ..., X*B* di mana masing-masing sampel berisi *n* data yang diperoleh dari ***x*** (data awal).
2. Mengevaluasi replikasi yang ada pada masing-masing sampel bootstrap.
3. Mengestimasi sampel dengan menggunakan standar deviasi untuk bootstrap yang direplikasi *B* kali.

2.2.5 SELEKSI FITUR FORWARD SELECTION

Seteleh dilakukan *sampling* pada dataset, kemudian data difilter kembali dengan menggunakan metode seleksi fitur forward selection. Seleksi fitur adalah salah satu teknik terpenting dan sering digunakan dalam *pre-processing data mining* khususnya untuk *knowledge discovery*. Teknik ini mengurangi atau mereduksi jumlah fitur yang terlibat dalam menentukan suatu nilai kelas target, mengurangi fitur yang tidak relevan (Larose, 2007) dan data yang menyebabkan salah pengertian terhadap kelas target yang membuat efek bagi aplikasi.

Copyright @ 2015 IlmuKomputer.Com

2.2.6 *WEIGHTED* INFORMATION GAIN

Weighted Information Gain (WIG) sering digunakan untuk meranking atribut yang paling berpengaruh terhadap kelasnya. Nilai gain dari suatu atribut, diperoleh dari nilai entropi sebelum pemisahan dikurangi dengan nilai entropi setelah pemisahan. Tujuan pengurangan fitur pengukuran nilai informasi diterapkan sebagai tahap sebelum pengolahan awal. Hanya atribut memenuhi kriteria (*treshold*) yang ditentukan dipertahankan untuk digunakan oleh algoritma klasifikasi (Bramer, 2007). Ada 3 tahapan dalam pemilihan fitur menggunakan information gain diantaranya adalah sebagai berikut:

1. Hitung nilai gain informasi untuk setiap atribut dalam dataset asli.
2. Buang semua atribut yang tidak memenuhi kriteria yang ditentukan.
3. Dataset direvisi.

Pengukuran atribut ini dipelopori oleh Cloude Shannon pada teori informasi (Gallager, 2001) dituliskan sebagai (Han, J., & Kamber, 2012):

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i) \quad \text{..... (16)}$$

Berdasarkan persamaan di atas berikut merupakan keterangannya:
D : Himpunan Kasus
m : Jumlah partisi D
p_i : Proporsi dari D_i terhadap D
Dalam hal ini p_i adalah probabilitas sebuah *tuple* pada *D* masuk ke kelas C_i dan diestimasi dengan $|C_i, D|/|D|$. Fungsi log diambil berbasis 2 karena informasi dikodekan berbasis bit. Selanjutnya nilai entropi setelah pemisahan dengan cara sebagai berikut (Han, J., & Kamber, 2012):

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \quad \text{..... (17)}$$

Berdasarkan persamaan di atas berikut merupakan keterangannya:
D : Himpunan kasus
A : Atribut
v : Jumlah partisi atribut A
$|D_j|$: Jumlah kasus pada partisi ke j
$|D|$: Jumlah kasus dalam D
Info(D_j) : Total entropi dalam partisi

Untuk mencari nilai information gain atribut A diperoleh dengan persamaan berikut(Han, J., & Kamber, 2012):

$$Gain(A) = Info(D) - Info_A(D) \quad \text{..... (18)}$$

Berdasarkan persamaan di atas berikut merupakan keterangannya:
Gain(A) : Information atribut A
Info(D) : Total entropi
$Info_A(D)$: Entropi A

Dengan penjelasan lain, Gain(A) adalah reduksi yang diharapkan didalam entropi yang disebabkan oleh pengenalan nilai atribut dari A. Atribut yang memiliki nilai information gain terbesar dipilih sebagai uji atribut untuk himpunan S. Selanjutnya suatu simpul dibuat dan diberi label dengan label atribut tersebut, dan cabang-cabang dibuat untuk masing-masing nilai dari atribut

3 METODE PENELITIAN

3.1 DATASET

Dalam penelitian ini dikumpulkan dua dataset, dataset yang pertama dataset *churn* dari database *University of California* (UCI) dan dataset yang kedua dataset *telecom* dari *Customer Relationship Management* (*CRM*) *of Duke University* seperti pada Tabel 3.1. Dataset *churn* dapat diunduh disitus http://www.sgi.com/tech/mlc/db/ dan dataset *telecom* dapat diunduh melalui situs http://www.fuqua.duke.edu/centers/ccrm/datasets/download.html. Kedua dataset ini banyak digunakan oleh para peneliti sebelumnya (Xia & Jin, 2008) (Z.-Y. Chen et al., 2012) karena dataset *churn* dan *telecom* ini merupakan dataset publik dibidang telekomunikasi yang banyak digunakan oleh para peneliti dibidang loyalitas pelanggan.

Tabel 3.1 Dataset Loyalitas Pelanggan

No	Name	Type	Record	Dimension
1	*Churn*	*Churn*	4.974	21
2	*Telecom*	*Churn*	3.399	6

Pada dataset *telecom* karena jumlah *record* aslinya terlalu besar yaitu sebanyak 195.956, maka dataset *telecom* hanya akan diambil *record* sebanyak 3.399. Jumlah ini sudah melibihi batas minimal untuk porsi yang ideal sebagaimana yang telah diterapkan dalam ilmu statistik, dimana pengambilan sebagian dataset dapat mewakili jumlah dari populasi aslinya dapat dirumuskan dengan formula sebagai berikut (Kriyantono, 2008):

$$n = \frac{N}{N.d^2+1} \quad \text{..... (31)}$$

Di mana:
n : Ukuran sampel
N : Ukuran populasi
d : Persen kelonggaran ketidaktelitian karena kesalahan pengambilan sampel yang masih dapat ditolerir.

3.2 METODE YANG DIUSULKAN

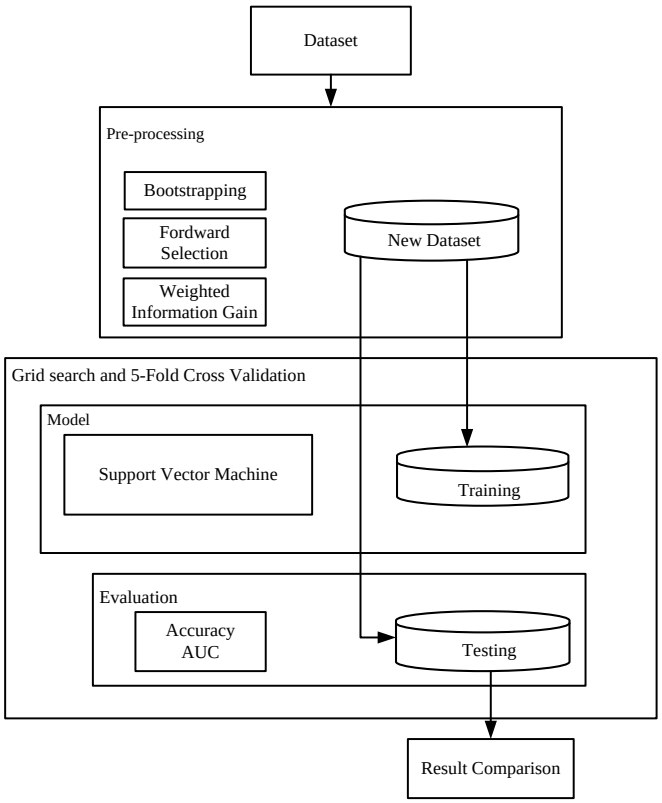
Untuk mengatasi masalah ketidakseimbangan kelas, memfilter atribut yang tidak relevan serta kelemahan algoritma SVM yaitu sulitnya menentukan nilai parameter dan pemilihan fungsi kernel yang tepat, dalam penelitan ini diusulkan dengan menggunakan metode bootstrapping, FS-WIG dan *grid search* pada algoritma SVM dengan fungsi kernel Radial Basis Function (RBF). Metode yang diusulkan sebagaimana yang ditunjukkan pada Gambar 3.1 yaitu:

1. Pada tahap pertama yaitu dengan *pre-processing* dengan metode sampel bootstrapping.
2. Kemudian data hasil sampel bootstrapping difilter kembali menggunakan seleksi fitur Forward Selection (FS) dan kemudian atribut atau fitur diberi pembobotan dengan menggunakan metode *Weighted* Information Gain (WIG). Kemudian fitur-fitur tersebut dihitung nilai gainnya, setelah semua fitur diketahui nilai gainnya langkah selanjutnya adalah menentukan nilai ambang yaitu nilai batas antara fitur terpilih dengan fitur yang tidak terpilih.
3. Kemudian metode *grid search* digunakan untuk memilih nilai parameter yang paling optimal yaitu dengan memasukkan nilai parameter C = 0,001 dan 1,5, epsilon = 0.004 – 1.5 dan gamma 0.001 – 1.5.
4. Setelah itu dataset dibagi dengan metode 5 *fold-cross validation* yaitu data *training* dan data *testing*.

Copyright @ 2015 IlmuKomputer.Com

5. Kemudian diklasifikasi dengan algoritma support vector machine (LibSVM) sebagai usaha untuk mencari *support vector* sebagai batas *margin* yang memisahkan kedua kelas.
6. Yang terakhir adalah tahapan pengukuran akurasi prediksi dengan melihat performa akurasi dan AUC.

Dalam penelitian pada umumnya pengujian nilai k pada *cross validation* dilakukan sebanyak 10 kali untuk memperkirakan akurasi estimasi, tetapi dalam penelitian ini nilai k yang digunakan berjumlah 5 atau 5-*fold cross validation*, hal ini sengaja dilakukan karena pengujian dengan 5-*fold cross validation* dapat mempercepat waktu komputasinya mengingat dataset yang digunakan cukup besar.



Gambar 3.1 Diagram Metode yang Diusulkan

4 HASIL EKSPERIMEN

4.1 HASIL EKSPERIMEN DATASET *CHURN*

Dalam melakukan penelitian ini digunakan spesifikasi komputer dengan processor Intel® Core ™ i3-2330M, 2.20GHz, memory 6 GB, hardisk 500 GB, sistem operasi Windows 7 Ultimate SP-1 64-bit dan aplikasi *RapidMiner* 5.3.015.

4.1.1 HASIL EKSPERIMEN BOOTSTRAPPING DENGAN SVM

Hasil eksperimen terbaik sampel bootstrapping dengan SVM yang telah dilakukan dengan menginputkan nilai parameter C = 0.5, gamma 0.001 dan epsilon = 0.004 yang menghasilkan nilai akurasi = 87.14% sebagaimana ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Komparasi Algoritma SVM dan SVM+Bootstraping pada Dataset *Churn*

	Akurasi	AUC
SVM	85.87%	0.504
SVM dengan Bootstrapping	**87.14%**	**0.500**

4.1.2 HASIL EKSPERIMEN BOOTSTRAPPING, FS-WIG DENGAN SVM

Eksperimen selanjutnya dengan menggunakan bootstrapping, FS-WIG pada SVM, dengan menginputkan parameter SVM dengan nilai parameter C = 0,5, gamma 0.001 dan epsilon = 0.004. Hasil eksperimen terbaik sebagaimana yang ditunjukkan oleh Tabel 4.2.

Tabel 4.2 Komparasi Algoritma SVM dan SVM+Bootstraping+FS-WIG pada Dataset *Churn*

	Akurasi	AUC
SVM	85.87%	0.504
SVM dengan Bootstrapping dan FS-WIG	**91.52%**	**0.762**

4.1.3 HASIL EKSPERIMEN BOOTSTRAPPING, FS-WIG DAN *GRID SEARCH* DENGAN SVM

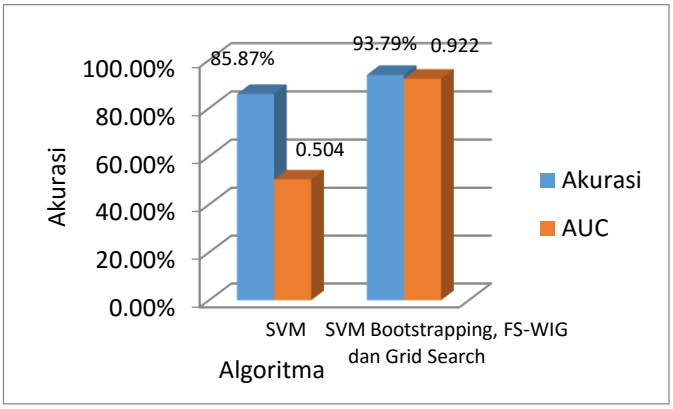
Pada pengujian dataset *churn* ini dilakukan eksperimen dengan memakai kernel Radial Basis Function (RBF), kemudian memasukkan nilai parameter C, epsilon dan gamma. Algoritma bootstrapping menggunakan sampel *relative* dan parameter sampel *ratio* 0.1, parameter ini berfungsi menginputkan dari sebagian kecil data dari jumlah total dataset yang digunakan dan memberikan bobot secara manual pada sampel *ratio*. Hasil eksperimen terbaik yang telah dilakukan untuk penentuan nilai akurasi dan AUC adalah dengan parameter C secara *logarithmic* antara 0.001 – 1.5, gamma = 0.001 – 1.5 dan epsilon = 0.004 – 1.5 yang menghasilkan nilai akurasi = 93.79% dan AUC nya adalah 0.922.

Pada Tabel 4.3 dari eksperimen yang telah dilakukan dan rata-rata keseluruhan eksperimen pada dataset *churn* secara konsisten menunjukkan peningkatan nilai akurasi dan AUC yang signifikan antara yang menggunakan SVM dengan sampel bootstrapping, FS-WIG dan *grid search* pada algoritma SVM. Gambar 4.1 merupakan grafik komparasi antara algoritma SVM dengan bootstrapping, FS-WIG dan *grid search* pada SVM

Tabel 4.3 Komparasi Algoritma SVM dengan Bootstrapping, FS-WIG dan *Grid Search* pada Dataset *Churn*

	Akurasi	AUC
SVM	85.87%	0.504
SVM dengan Bootstrapping, FS-WIG serta *Grid Search*.	**93.79%**	**0.922**

Copyright @ 2015 IlmuKomputer.Com



Gambar 4.1 Grafik Akurasi dan AUC Algoritma SVM dan SVM dengan Bootstrapping, WIG serta *Grid Search* pada Dataset *Churn*

4.2 HASIL EKSPERIMEN DATASET *TELECOM*

4.2.1 HASIL EKSPERIMEN BOOTSTRAPPING DENGAN SVM

Hasil eksperimen, setelah dataset baru dipilih sampel bootstrapping kemudian dengan menginputkan nilai parameter SVM, yaitu parameter C = 0.05, gamma = 0.001 dan epsilon = 0.004 dan kemudian diambil nilai rata-ratanya. Hasil eskperimen sebagaimana ditunjukkan oleh Tabel 4.4.

Tabel 4.4 Komparasi Algoritma SVM dan SVM dengan Bootstrapping pada Dataset *Telecom*

	Akurasi	**AUC**
SVM	97.03%	0.610
SVM dengan Bootstrapping	**98.56%**	**0.870**

4.2.2 HASIL EKSPERIMEN BOOTSTRAPPING, FS-WIG DENGAN SVM

Hasil eksperimen selanjutnya, setelah dataset baru dipilih sampel bootstrapping kemudian atribut diseleksi fitur dengan menggunakan FS-WIG, kemudian menginputkan nilai parameter SVM, yaitu parameter C = 0.05, gamma = 0.001 dan epsilon = 0.004 dan kemudian diambil nilai rata-ratanya. Hasil eksperimen sebagaimana ditunjukkan pada Tabel 4.5.

Tabel 4.5 Komparasi Algoritma SVM dan SVM dengan Bootstrapping dan FS-WIG pada Dataset *Telecom*

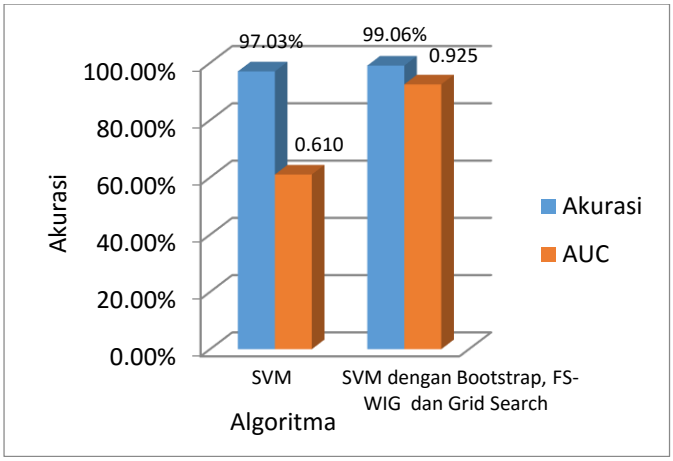
	Akurasi	**AUC**
SVM	97.03%	0.610
SVM dengan Bootstrapping dan FS-WIG	**98.56%**	**0.870**

4.2.3 HASIL EKSPERIMEN BOOTSTRAPPING, FS-WIG DAN *GRID SEARCH* DENGAN SVM

Sedangkan pengujian pada dataset *telecom* dilakukan dengan memakai kernel Radial Basis Function (RBF), kemudian memasukkan nilai parameter C dan epsilon. Pada pengujian ini dilakukan eksperimen dengan memakai kernel Radial Basis Function (RBF), kemudian memasukkan nilai parameter C, epsilon dan gamma. Algoritma bootstrapping menggunakan sampel *relative* dan parameter sampel *ratio* 0.1, parameter ini berfungsi menginputkan dari sebagian kecil data dari jumlah total dataset yang digunakan dan memberikan bobot secara manual pada sampel *ratio*. Hasil eksperimen terbaik yang telah dilakukan untuk penentuan nilai akurasi dan AUC adalah dengan parameter C secara *logarithmic* antara 0,001 – 1,5, gamma = 0,001 – 1,5 dan epsilon = 0,004 – 1,5 yang menghasilkan nilai akurasi = 99.96% dan AUC nya adalah 0.925%, sebagaimana yang telah ditunjukkan pada Tabel 4.6. Gambar 4.2 merupakan grafik komparasi antara algoritma SVM dengan bootstrapping, FS-WIG dan *grid search* pada SVM

Tabel 4.6 Komparasi Algoritma SVM dan SVM dengan Bootstrapping, FS-WIG serta *Grid Search* pada Dataset *Telecom*

	Akurasi	**AUC**
SVM	97.03%	0.610
SVM dengan Bootstrapping, FS-WIG serta *Grid Search*	**99.06%**	**0.925**



Gambar 4.2 Grafik akurasi dan AUC Algoritma SVM dan SVM dengan Bootstrapping, FS-WIG serta *Grid Search* pada Dataset *Telecom*

4.3 HASIL KOMPARASI DENGAN PENELITIAN TERDAHULU

Dari eksperimen yang telah peneliti lakukan membuktikan bahwa model yang peniliti usulkan ini lebih unggul dari pada model yang telah dilakukan oleh Xia et al (Xia & Jin, 2008), yang mana dataset *churn* yang peneliti gunakan sama dengan dataset yang digunakan oleh Xia *et al* dengan rata-rata akurasi 90.88%. Penelitian selanjutnya juga dilakukan oleh Sharma (Sharma & Panigrahi, 2011) pada tahun 2011 yang juga menggunakan dataset *churn* dengan menerapkan model *Feed-Forward* Back-Propagation Neural Network, untuk mencegah masalah *over training* pada neural netwok, Sharma mengambil data *training* secara acak tetapi setelah melewati beberapa kali jaringan, metode yang diusulkan Sharma malah kehilangan kemampuan generalisasi sehingga tetap terjebak dalam permasalahan *over training* tersebut, dan metode Sharma dalam memprediksi loyalitas pelanggan ini memperoleh rata-rata akurasi 92.35%, sementara dalam penelitian ini dataset *churn* memperoleh rata-rata akurasi 93.79%.

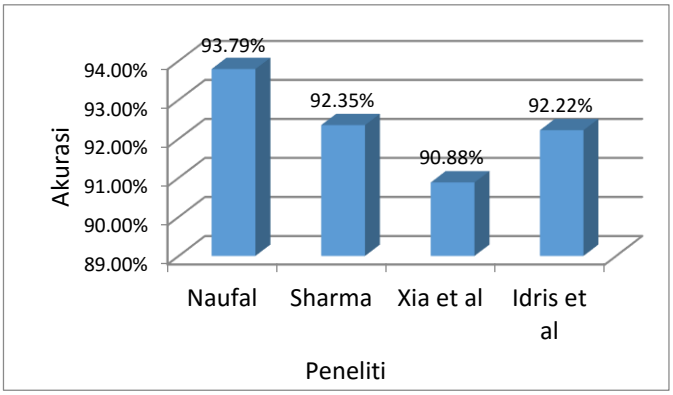
Metode yang diusulkan dalam eksperimen ini juga lebih baik dibanding dengan metode yang diusulkan oleh Idris et al (Idris et al., 2012) yang menggunakan *undersampling* Particle Swarm Optimization (PSO) dengan algoritma random forest dan fitur seleksi Principle Component Analysis (PCA) dan

Copyright @ 2015 IlmuKomputer.Com

menggunakan dataset *french telecom company* yang hasilnya memperoleh tingkat rata-rata akurasi 92.22%, dengan demikian model yang diusulkan dalam penelitian ini lebih baik dengan memperoleh rata-rata akurasi 93.79% untuk dataset *churn* dan 99.06% untuk dataset *telecom*. Tabel 4.13 menunjukkan komparasi model yang telah diusulkan dalam penelitian ini dengan beberapa model peneliti terdahulu.

Tabel 4.7 Komparasi Performansi dengan Model yang Diusulkan Peneliti Terdahulu

Peneliti	Model	Dataset	Hasil
Naufal	Sampel Bootstrapp, *FS-WIG* dan *grid search* pada SVM dengan RBF Kernel	*Churn*	93.79%
		Telecom	99.06%
Sharma	*Feed-Forward* Back-Propagation Neural Network	*Churn*	92.35%
Xia et al	SVM dengan RBF Kernel	*Churn*	90.88%
Idris et al	Undersampling PSO, Random Forest dan fitur seleksi PCA	*French Telecom Company*	92.22%



Gambar 4.1 Grafik Komparasi Performansi dengan Model yang Diusulkan Peneliti Terdahulu

5 KESIMPULAN

Beberapa eksperimen dilakukan untuk mendapatkan arsitektur yang optimal dan menghasilkan prediksi yang akurat. Berdasarkan hasil eksperimen, dapat disimpulkan sebagai berikut:

1. Hasil eksperimen pada dataset *churn* dengan SVM standar didapatkan nilai akurasi 85.87% dan AUC 0.504, sedangkan dengan menggunakan metode SVM+bootstrapping dengan nilai parameter C = 0,05, epsilon = 0,004 dan gamma = 0,001 didapatkan nilai akurasi 87.14% dan AUC 0.500. Sedangkan hasil eksperimen SVM standard pada dataset *telecom* didapatkan akurasi 97.03% dan AUC 0.610, sedangkan dengan SVM+bootstrapping dengan parameter C = 0,05 epsilon = 0,004 dan gamma = 0,001 didapatkan akurasi 98.56% dan AUC 0.870. Dengan menggunakan SVM+bootstrapping ada kenaikan akurasi dan AUC yang konsisten jika diterapkan pada dataset *churn* maupun dataset *telecom*, dimana dataset *churn* dan *telecom* merupakan dataset yang mengandung ketidakseimbangan kelas. Jadi dapat disimpulkan bahwa metode SVM dengan bootstrapping dalam eksperimen ini bisa untuk mengatasi permasalahan ketidakseimbangan kelas sehingga bisa meningkatkan performansi SVM.
2. Hasil eksperiman selanjutnya pada dataset *churn* dengan SVM standar didapatkan nilai akurasi 85.87% dan AUC 0.504, sedangkan dengan menggunakan metode SVM+bootstrapping+FS-WIG dengan nilai parameter C = 0,05, epsilon = 0,004 dan gamma = 0,001 didapatkan nilai akurasi 91.52% dan AUC 0.762. Sedangkan hasil eksperimen SVM standard pada dataset *telecom* didapatkan akurasi 97.03% dan AUC 0.610, sedangkan dengan SVM+bootstrapping+FS-WIG dengan parameter C = 0,05 epsilon = 0,004 dan gamma = 0,001 didapatkan akurasi 98.56% dan AUC 0.870. Dengan demikian SVM+bootstrapping+FS-WIG ada kenaikan akurasi dan AUC yang konsisten jika diterapkan pada dataset *churn*, tetapi pada dataset *telecom* tidak menunjukkan ada peningkatan performansi dari eksperimen SVM+bootstrapping. Hal ini dikarenakan atribut dalam dataset *telecom* hanya 6 atribut sehingga algoritma FS-WIG tidak terlalu berpengaruh jika diterapkan pada dataset yang mempunyai atribut sedikit. Sedangkan pada dataset *churn* ada peningkatan akurasi dan AUC, hal ini dikarenakan dataset *churn* mempunyai atribut yang lebih banyak daripada dataset *telecom*, yaitu 21 atribut. Dari sini dapat disimpulkan bahwa metode fitur seleksi FS-WIG tidak cocok jika diterapkan pada dataset yang mempunyai atribut sedikit, tetapi sangat baik jika diterapkan pada dataset yang berdimensi tinggi dimana dataset yang berdimensi tinggi secara umum mengandung atribut yang kurang relevan.
3. Hasil pengujian pada dataset *churn* dengan SVM standar didapatkan nilai akurasi 85.87% dan AUC 0.504, sedangkan dengan menggunakan metode SVM dengan bootstrapping, FS-WIG serta *grid search* dengan nilai parameter secara *logarithmic* C = 0,001 – 1,5, epsilon = 0,004 – 1,5 dan gamma = 0,001 – 1,5 didapatkan nilai akurasi 93.79% dan AUC 0.922. Sedangkan hasil pengujian SVM standard pada dataset *telecom* didapatkan akurasi 97.03% dan AUC 0.610, sedangkan SVM dengan bootstrapping, FS-WIG dan *grid search* dengan parameter secara *logarithmic* C = 0,001 – 1,5, epsilon = 0,004 – 1,5 dan gamma = 0,001 – 1,5 didapatkan akurasi 99.06% dan AUC 0.925. Dengan menggunakan SVM dengan sampel bootstrapping, FS-WIG dan *grid search* ada kenaikan akurasi dan AUC yang konsisten jika diterapkan pada dataset *churn* maupun dataset *telecom*. Hal ini dikarenakan pemilihan parameter SVM secara *grid search* dapat mencari nilai parameter terbaik yang nilainya antara 0,001 – 1,5 untuk parameter C, parameter epsilon 0,004 – 1,5 dan parameter gamma = 0,001 – 1,5. Jadi dapat disimpulkan bahwa metode *grid search* untuk pemilihan parameter SVM dalam eksperimen ini bisa mengatasi permasalahan sulitnya pemilihan nilai parameter pada SVM sehingga bisa memudahkan dalam menentukan nilai parameter yang tepat untuk algoritma SVM.

Dari hasil pengujian ini maka secara umum dapat dianalisa bahwa ada kenaikan akurasi sebesar 7.92% dan AUC sebesar 0.418 pada dataset *churn*, dan kenaikan akurasi sebesar 2.03% dan AUC sebesar 0.315 pada dataset *telecom* setelah diterapkan metode yang diusulkan. Dengan demikian dapat disimpulkan bahwa metode sampel bootstrapping dapat

Copyright @ 2015 IlmuKomputer.Com

mengatasi permasalahan pada ketidakseimbangan kelas (*class imbalance*) yang membantu menaikkan kinerja algoritma SVM dan seleksi fitur FS – WIG mampu memilih atribut terbaik. Sedangkan untuk menentukan parameter dengan metode *grid search* mampu memudahkan dalam pemilihan nilai parameter terbaik pada Algoritma SVM, sehingga menghasilkan kinerja atau tingkat akurasi prediksi dalam loyalitas pelanggan yang lebih baik dibanding dengan menggunakan metode individual algoritma SVM.

DAFTAR PUSTAKA

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal ofMachine Learning Research*, *13*, 281–305.

Bramer, M. (2007). *Principles of Data Mining*. Springer. Retrieved from http://link.springer.com/article/10.2165/00002018-200730070-00010

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*(3), 4626–4636. doi:10.1016/j.eswa.2008.05.027

Charu C. Aggarwal, P. S. Y. (2008). *Privacy Preserving Data Mining* (Vol. 19). Springer US. doi:10.1007/978-0-387-29489-6

Chen, H., Zhang, J., Xu, Y., Chen, B., & Zhang, K. (2012). Performance comparison of artificial neural network and logistic regression model for differentiating lung nodules on CT scans. *Expert Systems with Applications*, *39*(13), 11503–11509. doi:10.1016/j.eswa.2012.04.001

Chen, Z., & Fan, Z. (2013). Knowledge-Based Systems Dynamic customer lifetime value prediction using longitudinal data : An improved multiple kernel SVR approach. *Knowledge-Based Systems*, *43*, 123–134. doi:10.1016/j.knosys.2013.01.022

Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, *223*(2), 461–472. doi:10.1016/j.ejor.2012.06.040

Cortes, C., & Vapnik, V. (1995). Support vector machine. In *Machine learning* (pp. 1303–1308). doi:10.1007/978-0-387-73003-5_299

Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, *34*(1), 313–327. doi:10.1016/j.eswa.2006.09.038

Efron, B., & Tibshirani, R. (1998). *An introduction to the bootstrap*. New York: Chapman & Hall Book. Retrieved from http://books.google.com/books?hl=en&lr=&id=gLlpIUxRntoC&oi=fnd&pg=PR14&dq=An+Introduction+to+the+Bootstrap&ots=A8wrX6QbF7&sig=6gK8Gx-KtVcUXJM7qSFv92zi3eM

Farvaresh, H., & Sepehri, M. M. (2011). A data mining framework for detecting subscription fraud in telecommunication. *Engineering Applications of Artificial Intelligence*, *24*(1), 182–194. doi:10.1016/j.engappai.2010.05.009

Gallager, R. (2001). Claude E. Shannon: A retrospective on his life, work, and impact. *Information Theory, IEEE Transactions on*, *47*(7), 2681–2695. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=959253

Hamel, L. (2009). *Knowledge discovery with support vector machines*. John Wiley& Sons, Inc. Retrieved from http://books.google.com/books?hl=en&lr=&id=WaUnU4pEVVUC&oi=fnd&pg=PT10&dq=Knowledge+Discovery+with+Support+Vector+Machine&ots=U9cp-ZSxZ3&sig=XN99rPTt36-mZO-PpHdhwbhJ9-I

Han, S. H., Lu, S. X., & Leung, S. C. H. (2012). Segmentation of telecom customers based on customer value by decision tree model. *Expert Systems with Applications*, *39*(4), 3964–3973. doi:10.1016/j.eswa.2011.09.034

Han, J., & Kamber, M. (2012). *Data Mining : Concepts and Techniques* (3nd Editio.). Morgan Kaufmann Publishers.

Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, *39*(1), 1414–1425. doi:10.1016/j.eswa.2011.08.024

Idris, A., Rizwan, M., & Khan, A. (2012). Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. *Computers & Electrical Engineering*, *38*(6), 1808–1819. doi:10.1016/j.compeleceng.2012.09.001

Jadhav, R., & Pawar, U. (2011). Churn Prediction in Telecommunication Using Data Mining Technology. *International Journal of Advanced Computer Science and Applications*, *2*(2), 17–19. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.190.5029&rep=rep1&type=pdf#page=30

Khoshgoftaar, T. M., & Gao, K. G. K. (2009). Feature Selection with Imbalanced Data for Software Defect Prediction. *2009 International Conference on Machine Learning and Applications*. doi:10.1109/ICMLA.2009.18

Kriyantono, R. (2008). *Teknik Praktis Riset Komunikasi*. Jakarta: Kencana. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Teknik+Praktis+Riset+Komunikasi#0

Larose, D. T. (2007). *Data Mining Methods and Models*. Canada: John Wiley & Sons, Inc.

Lin, L., Ravitz, G., Shyu, M. L., & Chen, S. C. (2008). Effective feature space reduction with imbalanced data for semantic concept detection. *Proceedings - IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 262–269. doi:10.1109/SUTC.2008.66

Maldonado, S., Weber, R., & Famili, F. (2014). Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Information Sciences*, *286*, 228–246. doi:10.1016/j.ins.2014.07.015

Mozer, M. C., Wolniewicz, R., Grimes, D. B., Johnson, E., & Kaushansky, H. (2000). Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, *11*(3), 690–6. doi:10.1109/72.846740

Nie, G., Rowe, W., Zhang, L., Tian, Y., & Shi, Y. (2011). Credit card churn forecasting by logistic regression and decision tree. *Expert Systems with Applications*, *38*(12), 15273–15285. doi:10.1016/j.eswa.2011.06.028

Novakovic, J. (2010). The Impact of Feature Selection on the Accuracy of Naive Bayes Classifier. In *18th Telecommunications forum TELFOR* (Vol. 2, pp. 1113–1116).

Nugroho, A. S. (2008). Support Vector Machine: Paradigma Baru dalam Softcomputing. *Neural Networks*, 92–99.

Pan, S., Iplikci, S., Warwick, K., & Aziz, T. Z. (2012). Parkinson's Disease tremor classification – A comparison between Support Vector Machines and neural networks. *Expert Systems with Applications*, *39*(12), 10764–10771. doi:10.1016/j.eswa.2012.02.189

Richeldi, M., & Perrucci, A. (2002). *Churn analysis case study*. *Telecom Italian Lab*. Torino. Retrieved from http://www-ai.cs.uni-dortmund.de:8080/PublicPublicationFiles/richeldi_perrucci_2002b.pdf

Richter, Y., Yom-Tov, E., & Slonim, N. (2010). Predicting customer churn in mobile networks through analysis of social groups. In *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM 2010)* (pp. 732–741). doi:10.1137/1.9781611972801.64

Rynkiewicz, J. (2012). General bound of overfitting for MLP regression models. *Neurocomputing*, *90*, 106–110. doi:10.1016/j.neucom.2011.11.028

Sharma, A., & Panigrahi, P. (2011). A neural network based approach for predicting customer churn in cellular network services. *International Journal of Computer Applications (0975-8887)*, *27*(11), 26–31. doi:10.5120/3344-4605

Copyright @ 2015 IlmuKomputer.Com

Tian, W., Song, J., Li, Z., & de Wilde, P. (2014). Bootstrap techniques for sensitivity analysis and model selection in building thermal performance analysis. *Applied Energy*, *135*, 320–328. doi:10.1016/j.apenergy.2014.08.110

Tsai, C.-F., & Lu, Y.-H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, *36*(10), 12547–12553. doi:10.1016/j.eswa.2009.05.032

Vapnik, V. (1998). *The Nature of Statistical Learning Theory*. *Technometrics*. John Wiley & Sons, Inc. Retrieved from http://www.tandfonline.com/doi/pdf/10.1080/00401706.1996.10484565

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, *218*(1), 211–229. doi:10.1016/j.ejor.2011.09.031

Wahono, R. S., Herman, N. S., & Ahmad, S. (2014). Neural Network Parameter Optimization Based on Genetic Algorithm for Software Defect Prediction. *Advanced Science Letters*, *20*(10), 1951–1955. doi:10.1166/asl.2014.5641

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques* (3rd ed.). USA: Morgan Kaufmann Publishers.

Wu, Q. (2011). Hybrid forecasting model based on support vector machine and particle swarm optimization with adaptive and Cauchy mutation. *Expert Systems with Applications*, *38*(8), 9070–9075. doi:10.1016/j.eswa.2010.11.093

Wu, Xindong & Kumar, V. (2009). *The Top Ten Algorithm in Data Mining*. Boca Raton: Taylor & Francis Group, LLC.

Xia, G., & Jin, W. (2008). Model of Customer Churn Prediction on Support Vector Machine. *Systems Engineering - Theory & Practice*, *28*(1), 71–77. doi:10.1016/S1874-8651(09)60003-X

Xu, J., Tang, Y. Y., Zou, B., Xu, Z., Li, L., & Lu, Y. (2014). Generalization performance of Gaussian kernels SVMC based on Markov sampling. *Neural Networks : The Official Journal of the International Neural Network Society*, *53*, 40–51. doi:10.1016/j.neunet.2014.01.013

Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. *Proceedings of the First International Conference on Advanced and Information Engineering*, *285*, 429–436. doi:10.1007/978-981-4585-18-7_2

Yu, X., Guo, S., Guo, J., & Huang, X. (2011). An extended support vector machine forecasting framework for customer churn in e-commerce. *Expert Systems with Applications*, *38*(3), 1425–1430. doi:10.1016/j.eswa.2010.07.049

Zhou, S.-S., Liu, H.-W., & Ye, F. (2009). Variant of Gaussian kernel and parameter setting method for nonlinear SVM. *Neurocomputing*, *72*(13-15), 2931–2937. doi:10.1016/j.neucom.2008.07.016

BIOGRAFI PENULIS

Abdul Razak Naufal. Menyelesaikan pendidikan S1 di Universitas Islam Negeri (UIN) Walisongo, Semarang dan S2 Magister Teknik Informatika di Universitas Dian Nuswantoro Semarang, Indonesia. Saat ini menjadi trainer dan konsultan IT di CV. Media Hasanah. Minat penelitian saat ini adalah data mining.

Romi Satria Wahono. Menempuh pendidikan S1, S2 di departement computer science di Saitama University, Jepang, dan Ph.D di department Software Engineering di Universiti Teknikal Malaysia Melaka. Saat ini menjadi dosen pascasarjana Magister Teknik Informatika di Universitas Dian Nuswantoro, Indonesia. Founder IlmuKomputer.com dan CEO PT Brainmatics, sebuah perusahaan pembuatan software di Indonesia. Minat penelitian saat ini adalah software engineering dan machine learning. Professional member dari ACM and IEEE.

Abdul Syukur. Menerima gelar sarjana dibidang Matematika dari Universitas Diponegoro Semarang, gelar master dibidang Manajemen dari Universitas Atma Jaya Yogyakarta, dan gelar doctor dibidang ekonomi dari Universitas Merdeka Malang. Saat ini menjadi dosen dan dekan di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia. Minat penelitiannya saat ini meliputi decision support systems dan information management systems.

Copyright @ 2015 IlmuKomputer.Com

Hybrid Keyword Extraction Algorithm and Cosine Similarity for Improving Sentences Cohesion in Text Summarization

Rizki Darmawan
Informatics Engineering Graduate Program, STMIK Eresha
rizkidmw@gmail.com

Romi Satria Wahono
Faculty of Computer Science, Dian Nuswantoro University
romi@romisatriawahono.net

Abstract: As the amount of online information increases, systems that can automatically summarize text in a document become increasingly desirable. The main goal of a text summarization is to present the main ideas in a document in less space. In the create text summarization, there are two procedures i.e. extraction and abstraction procedure. One of extraction procedure is using keyword extraction algorithm which is easier and common but has problem in the lack of cohesion or correlation between sentences. The cohesion between sentences can be applied by using a cosine similarity method. In this study, a hybrid keyword extraction algorithm and cosine similarity for improving sentences cohesion in text summarization has been proposed. The proposed method is using compression 50%, 30% and 20% to create candidate of the summary. The result shows that proposed method affect significant increasing cohesion degree after evaluated in the t-Test. The result also shows that 50% compression ratio obtains the best result with Recall, Precision, and F-Measure are 0.761, 0.43 and 0.54 respectively; since summary with compression ratio 50% has higher intersection with human summary than another compression ratio.

Keywords: text summarization, keyword extraction, cosine similarity, cohesion

1 INTRODUCTION

As the amount of online information increases, systems that can automatically summarize one or more documents become increasingly desirable. Recent research has investigated types of summaries, method to create them, and methods to evaluate them (Hovy & Lin, 1999). It is necessary that the end user can access the information in summary form and without losing the most important aspects presented therein. Some of the application areas of the generation of extractive summaries from a single document are the summaries of web pages presented on the search engines (Porselvi & Gunasundari, 2013). Frequent workshop and symposia in text summarization reflect the ongoing interest of the researchers around the world.

The main goal of a summary is to present the main ideas in a document in less space. If all sentences in a text document were of equal importance, producing a summary would not be very effective, as any reduction in the size of a document would carry a proportional decrease in its informative of document (Hovy & Mckeown, 2001). Luckily, information content in document appears in bursts, and one can therefore distinguish between more and less informative segments.

The method for creating the summary can be divided into two ways: manually and automatically. Text summarization is a method to automatically summarize the text. In the create text summarization, there are two procedures i.e.: extraction and abstraction (Das, 2007). Extraction is a procedure used to create a summary by taking important sentences word by word that comes from the text, while abstraction is a procedure that is used to create a summary by information fusion, sentence compression and reformulation (Aliguliyev, 2009).

Text summarization with extraction procedure called extract summarization is easier to create than using abstraction Extractive procedure are usually performed in three step create an intermediate representation of the original text, sentence scoring and select high scores sentences to summary. There are several method that use in extractive procedure such as Keyword Extraction, Naïve-Bayes, Hidden Markov Models, Graph Method, Latent Sematic Indexing (Das, 2007).

Keyword extraction is an important technique for document retrieval, web page retrieval, document clustering, summarization, text mining, and so on (Rajman, 1998). By extracting appropriate keywords, we can easily choose which document to read to learn the relationship among documents. A popular algorithm for indexing is the TF/IDF measure, which extracts keywords that appear frequently in a document, but that don't appear frequently in the remainder of the corpus. The term "keyword extraction" is used in the context of text mining, for example (Rajman, 1998). A comparable research topic is called "automatic term recognition" in the context of computational linguistics and "automatic indexing" or "automatic keyword extraction" in information retrieval research. Recently, numerous documents have been made available electronically. Domain independent keyword extraction, which does not require a large corpus, has many (Ishizuka, 2003).

The first step creates a representation of the document. Usually, it divides the text into paragraphs, sentences, and tokens. Sometimes some preprocessing, such as stop word removal is also performed. The second step tries to determine which sentences are important to the document or to which extent it combines information about different topics, by sentence scoring (Ferreira et al., 2013). Usually, abstractive summarization requires heavy machinery for language generation and is difficult to replicate or extends to broader domain (Das, 2007).

Keyword Extraction Algorithm is easier and common in extract summarization. Yet the keyword extraction algorithm has problem in the lack of cohesion or correlation between sentences (Nandhini & Balasundaram, 2013) (Mendoza, Bonilla, Noguera, Cobos, & León, 2014) (Ishizuka, 2003). The correlation between sentences can be seen from the relationship between sentences and extent to which the ideas in the text are expressed clearly and relate to one another in a

Copyright @ 2015 IlmuKomputer.Com

systematic fashion by avoiding a confusing jumble of information (Nandhini & Balasundaram, 2013).

One way to resolve the problem of cohesion between sentences in extract summary is with determine the optimal combination between sentences (Fattah & Ren, 2009). The determination and cohesion optimization can be applied by using a cosine similarity method (Bestgen & Universit, 2006). The function for similarity measure should be easy to compute, it should implicitly capture the relatedness of the documents, and it should also be explainable (Rafi & Shaikh, 2010). The similarity between two sentences, according to the vector representation described is calculated as the cosine similarity (Manning, Raghavan, & Schlutze, 2009).

The objective of this work is to improve cohesion in text summarization by keyword extraction algorithm using cosine similarity method. Finally, our work of this paper is summarized in the last section.

2 RELATED WORKS

Many studies have been published in cohesion problem for text summarization in some approach like using optimal combination for the summarization (Mendoza, 2013; Nandhini, 2013) and another technique that concern with cohesion in text summarization.

Mendoza et al. (2013) proposed is combined the population based global search with a local search heuristic (memetic approach). The local search heuristic exploits the problem knowledge for redirect the search toward best solution. The objective function for this method is defined formed by the features like cohesion which proved effective in selecting relevant sentences from a document. The best results of MA-SingleDocSum evaluated with ROUGE-1 and ROUGE-2 is 8.59% with DUC 2001.

Nandhini et al. (2013) work to extract the optimal combination of sentences that increase readability through sentence cohesion using genetic algorithm. The results show that the summary extraction using their proposed approach performs better in *F*-measure, readability, and cohesion than the baseline approach (lead) and the corpus-based approach. In the case of 10% compression rate the F-measure is 0.284, 20% compression is 0.466 and 30% compression is 0.502. The best F-measure is 30% compression

Smith et al. (2011) work to measure cohesion is automatically through the amount of co-references in the text and how intact the text is after summarization. They compare four different types of techniques (Every3, 100First, CogSum, PrevSum) were used to create the summaries. The results proved that the summary produced by a traditional vector space-based summarizer is not less cohesive than a summary created by taking the most important sentences from the summarizer. Comparing the cohesion there are significances, for instance, for broken references the 100First is significantly better than all the other ($p < 0.001$) is 0.459.

Silber et al. (2002) present a linear time algorithm for lexical chain computation. The algorithm makes lexical chains computationally feasible candidate as an intermediate representation for automatic text summarization By using lexical chains, they can find statistically the most important concepts by looking at the structure in the document rather than the deep semantic meaning. Lexical chains appropriately represent the nouns in the summary is 79,12%.

3 PROPOSED METHOD

The proposed model using keyword extraction algorithm with compression ratio parameter and combining with cosine similarity for conducting this experiment. Cosine similarity is used to re-arrange sentence extraction from the result of keyword extraction algorithm process.

The keyword extraction algorithm using calculation based on TF/IDF, weight a given term to determine how well the term describes an individual document within a corpus. It does this by weighting the term positively for the number of times the term occurs within the specific document, while also weighting the term negatively relative to the number of documents which contain the term. Consider term t and document d, where t appears in n of N documents in D. The TF-IDF function is of the form as follows:

$$TFIDF\ (t,d,n,N) = TF\ (t,d)\ x\ IDF\ (n,N)$$

When the TF-IDF function is run against all terms in all documents in the document corpus, the words can be ranked by their scores. A higher TF-IDF score indicates that a word is both important to the document, as well as relatively uncommon across the document corpus. This is often interpreted to mean that the word is significant to the document, and could be used to accurately summarize the document. TF-IDF provides a good heuristic for determining likely candidate keywords, and it (as well as various modifications of it) has been shown to be effective after several decades of research.

Cosine similarity is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them, often used to compare documents in text mining (Satya & Murthy, 2012). Given two vectors of attributes, A and B, the cosine similarity, θ, is represented using a dot product and magnitude as:

$$Similarity = Cos\ \theta = \frac{A.B}{|A||B|}$$

The resulting similarity ranges from 0 with usually indicating independence, and 1 with usually indicating exactly the same and in between those values indicating intermediate similarity and dissimilarity. For the text matching, the attribute vector A and B are usually the term frequency vectors of the documents. In the case of information retrieval the cosine similarity of two documents will range 0 to 1, since the term frequencies (TF-IDF weights) cannot be negative. The angle between two term frequency vectors cannot be greater than 90^0.

In Figure 1 can be explained that after the data from UCI Reuters- 21578 completed prepared then the data will be tested into summarization stage.

The summarization stage consists of three component i.e. keyword extraction algorithm, compression ratio selector and cosine similarity method. These three component will summarize the text were feeding as the result final text were summarized.

The first pre-processed document is tokenized by keyword extraction algorithm and then calculates TF/IDF for each term. Then sum all of TF/IDF term for each sentence and get sum of each sentence the next process is rank all of sentence based on sum of TF/IDF. The compression ratio determine the position of sentence rank. In this study using a compression of 50% that means the sentence summary shrinkage 50% from the original text. After sentence is selected then perform calculation of their similarity with cosine similarity method. After the calculation of cosine similarity, the next process is re-arranging all of

Copyright @ 2015 IlmuKomputer.Com

sentence based on cosine similarity from the highest to the lowest similarity. This new text with new sentence arrangement will be the final summarized text.

Extractive summary can be evaluated using various characteristic such as F-measure and cohesion (Nandhini & Balasundaram, 2013b). F-Measure is measuring how far the technique is capable of predicting of correct sentence. Evaluation can be classified into intrinsic and extrinsic evaluation (Nandhini, 2013). Intrinsic evaluation judges the summary quality by its coverage between machine-generated summary and human generated summary. Extrinsic evaluation focuses mainly on the quality by its effect on other tasks. In intrinsic evaluation, Precision (P), recall (R), and F-measure (F) are used to judge the coverage between the manual and the machine generated summary:

$$P = \frac{|S \cap T|}{|S|}$$

$$R = \frac{|S \cap T|}{|T|}$$

$$\mathrm{F} = \frac{|2 * \mathrm{P} * \mathrm{R}|}{|\mathrm{R} + \mathrm{P}|}$$

Where S is the machine generated summary and T is the manual summary (Nandhini & Balasundaram, 2013b). For the cohesion evaluation, we can measure with the formula as follows:

$$CoH = \frac{Log(C_s * 9 + 1)}{Log\ (M * 9 + 1)} \qquad Ns = \frac{(o) * (o-1)}{2}$$

$$C_s = \frac{\Sigma_{\forall S_i, S_j \in Summary}\ Sim_{cos}\ (S_i, S_j)}{N_s}$$

$$M = \max Sim_{cos}\ (i,j)\ ,\ i,j \leq N$$

$$Ns = \frac{(o) * (o-1)}{2}$$

Where CoH corresponds to the cohesion of a summary, Cs is the average similarity of all sentences in the summary S, Sim_{cos}(Si,Sj) is the cosine similarity between sentences Si and Sj, Ns is the number of nonzero similarity relationships in the summary, O is the number of sentences in the summary, M corresponds to the maximum similarity of the sentences in the document and N is the number of sentences in the document. In this way, CoH tends to zero when the summary sentences are too different among them, while that CoH tends to one when these sentences are too similar among them. Thus, this feature tends to favor the summaries that contain sentences about the same topic (Mendoza et al., 2014).

The dataset used in this research is collected from UCI Dataset containing documents of Reuters-21578 that has collection appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd. (Sam Dobbins, Mike Topliss, and Steve Weinstein) and Carnegie Group, Inc. (Peggy Andersen, Monica Cellio, Phil Hayes, Laura Knecht, Irene Nirenburg) in 1987. The detail dataset can be downloaded at https://archive.ics.uci.edu/ml/datasets/Reuters21578+Text+Categorization+Collection.

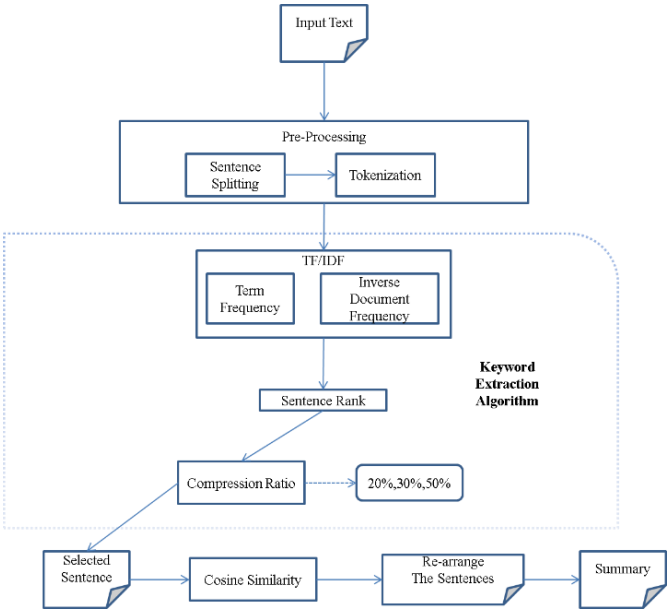


Figure 1. Block Diagram Proposed Model

4 EXPERIMENTAL RESULTS

The research using computer platform with specification based on Intel Core i3 2.30 Ghz CPU, 2 GB RAM, and Microsoft Windows 7 Ultimate 32 Byte. The software is using Java with Netbeans IDE 7.3.1.

Evaluation of the results is the calculation of recall, precision and F-measure. It can be seen that the lowest recall at dataset 6 that is equal to 0.484 and the highest recall on dataset 10 is equal to 0.909. The lowest precision is dataset 6 is equal to 0.284 and the highest precision on dataset 2 is equal to of 0.685. While the lowest F-measure at dataset 6 that equal to 0.358 and highest F-measure at dataset 2 that equal to 0.748. It is shown in Table 1.

Table 1. Recall-Precision of Summary with Compression 50%

Dataset	Recall	Precision	F-Measure
Dataset 1	0.771	0.492	0.600
Dataset 2	0.824	0.685	0.748
Dataset 3	0.908	0.478	0.626
Dataset 4	0.565	0.565	0.565
Dataset 5	0.635	0.328	0.433
Dataset 6	0.484	0.284	0.358
Dataset 7	0.888	0.381	0.532
Dataset 8	0.861	0.331	0.478
Dataset 9	0.772	0.392	0.520
Dataset 10	0.909	0.454	0.606

In compression summary 30% can be seen that the lowest recall at dataset 5 that is equal to 0.418 and the highest recall on dataset 8 is equal to 0.907. The lowest precision is dataset 5 is equal to 0.295 and the highest precision on dataset 2 is equal to of 0.666. While the lowest F-measure at dataset 5 that equal to 0.346 and highest F-measure at dataset 10 that equal to 0.690 as shown in Table 2.

Copyright @ 2015 IlmuKomputer.Com

Table 2. Recall-Precision of Summary with Compression 30%

Dataset	Recall	Precision	F-Measure
Dataset 1	0.554	0.464	0.505
Dataset 2	0.702	0.666	0.684
Dataset 3	0.653	0.444	0.528
Dataset 4	0.526	0.412	0.462
Dataset 5	0.418	0.295	0.346
Dataset 6	0.453	0.397	0.423
Dataset 7	0.688	0.428	0.525
Dataset 8	0.907	0.561	0.694
Dataset 9	0.555	0.458	0.478
Dataset 10	0.863	0.575	0.690

In compression summary 20% can be seen that the lowest recall at dataset 2 that is equal to 0.148 and the highest recall on dataset 10 is equal to 0.863. The lowest precision is dataset 2 is equal to 0.215 and the highest precision on dataset 10 is equal to of 0.647. While the lowest F-measure at dataset 5 that equal to 0.176 and highest F-measure at dataset 10 that equal to 0.740 as shown at Table 3.

Table 3 Recall-Precision of Summary with Compression 20%

Dataset	Recall	Precision	F-Measure
Dataset 1	0.253	0.538	0.344
Dataset 2	0.148	0.215	0.176
Dataset 3	0.306	0.329	0.317
Dataset 4	0.434	0.412	0.423
Dataset 5	0.459	0.459	0.459
Dataset 6	0.406	0.522	0.456
Dataset 7	0.666	0.424	0.497
Dataset 8	0.907	0.678	0.776
Dataset 9	0.469	0.584	0.521
Dataset 10	0.863	0.647	0.740

For the 50 % compression the highest recall in summary of dataset 10 and lowest recall in dataset 6, while the highest precision in summary of dataset 2 and lowest precision in summary of dataset 6. The highest F-measure of 50 % compression in summary of dataset 2 and the lowest F measure in summary of dataset 6.

For the 30 % compression the highest recall in summary of dataset 8 and lowest recall in summary of dataset 5, while the highest precision in summary of dataset 2 and lowest precision in summary of dataset 5. The highest F-measure of 50 % compression in summary of dataset 10 and the lowest F measure in summary of dataset 5. For the 20 % compression the highest recall in summary of dataset 8 and lowest recall in summary of dataset 2, while the highest precision in summary of dataset 8 and lowest precision in summary of dataset 2. The highest F-measure of 50 % compression in summary of dataset 10 and the lowest F measure in summary of dataset 5. Overall of that analysis is shown in Table 4.

Table 4. Overall Analyses of Recall, Precision And F-Measure

	Recall		Precision		F-Measure	
	Highest	Lowest	Highest	Lowest	Highest	Lowest
Compression 50 %	Dataset10	Dataset 6	Dataset 2	Dataset 6	Dataset 2	Dataset 6
Compression 30 %	Dataset 8	Dataset 5	Dataset 2	Dataset 5	Dataset 10	Dataset 5
Compression 20 %	Dataset10	Dataset 2	Dataset10	Dataset 2	Dataset10	Dataset 5

The main factor of that performance is how much the intersection against human summary because it related to the equation of recall and precision. If intersection is high, automatically make the high result, although length of word in machine and human has big influence contribution to the result. This study result also confirm some studies that intersection between human summary and machine play big influence for evaluation measurement such as recall, precision and F-measure (Conroy, 2001).The comparison of average recall, precision and F-measure is shown in Table 5 and Figure 2.

Table 5. Comparison of Average Recall, Precision and F-Measure

Compression	Recall	Precision	F-Measure
50%	0.761	0.439	0.547
30%	0.625	0.470	0.533
20%	0.484	0.481	0.471

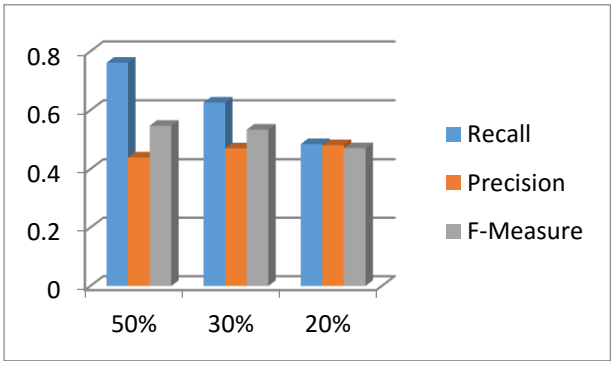


Figure 2. Average Recall, Precision and F-Measure Diagram

From the data that shown in Table 5, it's shown that the best F-measure is 50% compression that has value is 0.547. It's because they have highest intersection than the other compression that compare with human summary. The results also reflect that summary with 50% compression is the better summary than the others. Another study also reflect that higher compression has higher average of recall, precision and F-measure (Nandhini & Balasundaram, 2013b) and this result also confirmed by Ferreira et al (2014) that the best result summary is 50 % compression (Ferreira et al., 2014)

To prove whether there are differences in the degree of cohesion after using the cosine similarity method is using t-test models. A significant difference in performance is considered when the results of t-test showed that (P <= t) < alpha (0.05). T-test of the statistical test on the summary results that using the cosine similarity method and without using the cosine similarity method is shown in the Table 6.

Copyright @ 2015 IlmuKomputer.Com

Table 6. T-Test: Paired Two Sample for Means
Of Cohesion Degree

	Without Cosine Similarity	*Cosine Similarity*
Mean	31.92255942	**35.42168762**
Variance	33.59318702	40.794073
Observations	10	10
Pearson Correlation	0.968118831	
Hypothesized Mean Difference	0	
df	9	
t Stat	-6.721927271	
P(T<=t) one-tail	4.3178E-05	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	**8.63559E-05**	
t Critical two-tail	2.262157163	

From Table 6, it shows the average of cohesion degree of summary that using the cosine similarity method is higher than without using cosine similarity that has value is 35.42168762 with P value = 8.63559E-05 . The significance level is set to be 0.05. It means that cohesion degree in summary using cosine similarity and without using cosine similarity have significant differences (P value < 0.05). Therefore, it can be concluded that summary with cosine similarity method makes an improvement when compared with summary without using cosine similarity in cohesion degree.

The best average F-measure of summary in three compressions is 50% compression. According to another study that using compression ratio to get the result, also reflect that highest compression ratio has best F-measure (Nandhini & Balasundaram, 2014). One reason to explain about this phenomena is intersection human summary and machine summary is higher according to compression ratio. Intersection means that how many words in machine summary have same similarity with number of word in human summary. If intersection is high, automatically make the high result, although length of word in machine and human has big influence contribution to the result. This study result also confirm some studies that intersection between human summary and machine play big influence for evaluation measurement such as recall, precision and F-measure (Conroy, 2001).

From t-test result, summary that using cosine similarity has increased significantly in cohesion degree compared with the summary without using cosine similarity. The results of these experiments also show that the highest F-measure is compression of 50%. The result can be compared with another research like Nandhini & Balasundaram (Nandhini & Balasundaram, 2013b) and Aliguliyev (Aliguliyev, 2009) that increase of compression in order to increase of F-measure.

5 CONCLUSION

Recent research has investigated types of summaries, method to create them, and methods to evaluate them. It is necessary that the end user can access the information in summary form and without losing the most important aspects presented therein. Some of the application areas of the generation of extractive summaries from a single document are the summaries of web pages presented on the search engines. The main goal of a summary is to present the main ideas in a document in less space. If all sentences in a text document were of equal importance, producing a summary would not be very effective, as any reduction in the size of a document would carry a proportional decrease in its informative

In this research is used keyword extraction algorithm model with cosine similarity method that combined in some compression ratio. In the experiment is tested that keyword extraction algorithm using compression ratio of 20%, 30% and 50%. The best compression ratio from the extraction of keyword extraction algorithm is 50% with the F-measure is 0.761. In this research also shows there is different between summary with cosine similarity and without cosine similarity related to cohesion between sentences after tested with t-test, where summary with cosine is the best performance.

REFERENCES

Aliguliyev, R. M. (2009). Expert Systems with Applications A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems With Applications, 36(4), 7764–7772. doi:10.1016/j.eswa.2008.11.022

Bestgen, Y., & Universit, F. (2006). Improving Text Segmentation Using Latent Semantic Analysis. Association for Computational Linguistic, (2001).

Conroy, J. (2001). Matrix Decomposition 1 Introduction. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. (pp. 1–20). ACM.

Das, D. (2007). A Survey on Automatic Text Summarization Single-Document Summarization. Carnegie Mellon University, 1–31.

Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Computer Speech & Language, 23(1), 126–144. doi:10.1016/j.csl.2008.04.002

Hovy, E., & Lin, C. (1999). Automated Text Summarization in Summarist. Asscociation for Computer Linguistic.

Hovy, E., & Mckeown, K. (2001). Summarization. Asscociation for Computer Linguistic, 28.

Ishizuka, M. (2003). Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. International Journal on Artificial Intelligence Tools.

Manning, C., Raghavan, P., & Schlutze, H. (2009). Introduction to Information Retrieval (p. 581). Cambridge University.

Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., & León, E. (2014). Expert Systems with Applications Extractive single-document summarization based on genetic operators and guided local search. Expert Systems With Appilications, 41(9), 4158–4169. doi:10.1016/j.eswa.2013.12.042

Miller, G. A., Beckwith, R., Fellbaum, C., & August, R. (1993). Introduction to WordNet : An On-line Lexical Database, (August).

Nandhini, K., & Balasundaram, S. R. (2013). Improving readability through extractive summarization for learners with reading difficulties. Egyptian Informatics Journal, 14(3), 195–204. doi:10.1016/j.eij.2013.09.001

Nandhini, K., & Balasundaram, S. R. (2014). Extracting easy to understand summary using differential evolution algorithm. Swarm and Evolutionary Computation, 1–9. doi:10.1016/j.swevo.2013.12.004

Porselvi, A., & Gunasundari, S. (2013). Survey on web page visual summarization. International Journal of Emerging Technology and Advanced Engineering, 3(1), 26–32.

Rafi, M., & Shaikh, M. S. (2010). An improved semantic similarity measure for document clustering based on topic maps. Computer Science Department Karachi Pakistan.

Rajman, M. (1998). Text mining – knowledge extraction from unstructured textual data. In In Proceedings of the 6th

Copyright @ 2015 IlmuKomputer.Com

Conference of International Federation of Classification Societies.
Satya, K. P. N. V, & Murthy, J. V. R. (2012). Clustering Based On Cosine Similarity Measure. International Journal of Engineering Science & Advanced Technology, 2(3), 508–512.
Silber, H. G. (2002). an Intermediate Representation for Automatic Text Summarization. Association for Computational Linguistic, 28, 1–11.
Smith, C., Danielsson, H., & Arne, J. (2011). Cohesion in Automatically Created Summaries. Santa Anna IT Research

BIOGRAPHY OF AUTHORS

Rizki Darmawan. Received M.Kom from STMIK ERESHA, Jakarta. He is an IT professional. His current research interests include information retrieval and machine learning.

Romi Satria Wahono. Received B.Eng and M.Eng degrees in Computer Science respectively from Saitama University, Japan, and Ph.D in Software Engineering from Universiti Teknikal Malaysia Melaka. He is a lecturer at the Graduate School of Computer Science, Dian Nuswantoro University, Indonesia. He is also a founder and chief executive officer of Brainmatics, Inc., a software development company in Indonesia. His current research interests include software engineering and machine learning. Professional member of the ACM and IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Penerapan Algoritma Genetika untuk Optimasi Parameter pada Support Vector Machine untuk Meningkatkan Prediksi Pemasaran Langsung

Ispandi
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
ispandi.ipd@bsi.ac.id

Romi Satria Wahono
Faculty of Computer Science, Dian Nuswantoro University
romi@romisatriawahono.net

Abstrak: Pemasaran langsung adalah proses mengidentifikasi potensi pembeli produk tertentu dan mempromosikan produk dengan sesuai. pelaksanaan pemasaran langsung dari waktu ke waktu menghasilkan data dan informasi dalam bentuk laporan yang perlu di analisis oleh manajer dalam rangka mendukung keputusan. Namun itu adalah tugas yang sulit bagi manajer untuk menganalisis data yang kompleks yang luas. Kesulitan ini menyebabkan perkembangan teknik intelijen bisnis, yang bertujuan mengklasifikasi pengetahuan yang berguna untuk mendukung pengambilan keputusan. Algoritma support vector machine mampu mengatasi data set yang berdimensi tinggi, mengatasi masalah klasifikasi dan regresi dengan linier ataupun nonlinier kernel, yang dapat menjadi satu kemampuan algoritma untuk klasifikasi serta regresi, namun support vector machine memiliki masalah dalam pemilihan parameter yang sesuai untuk meningkatkan optimasi. Untuk mengatasi masalah tersebut diperlukan metode algoritma genetika untuk pemilihan parameter yang sesuai pada metode support vector machine. Beberapa eksperimen dilakukan untuk mendapatkan akurasi yang optimal dan metode yang di usulkan adalah penerapan algoritma untuk optimasi pada parameter pada support vector machine. Hasil penelitian menunjukan, eksperimen dengan menggunakan metode support vector machine dan algoritma genetika yang digunakan untuk melakukan optimasi parameter C, γ dan ε dengan tiga jenis kernel. Kernel pertama tipe kernel dot dengan akurasi sebesar 85,59%, AUC sebesar 0,911 yang kedua tipe kernel radial dengan akurasi sebesar 98.89%, AUC sebesar 0,981 dan yang ketiga dengan tipe kernel Polynomial dengan akurasi sebesar 98.67% dan AUC sebesar 0.938. Hasil eksperimen tersebut menunjukan pengujian data set bank menggunakan penerapan algoritma genetika pada support vector machine menunjukan hasil yang lebih akurat dari penelitian sebelumnya untuk prediksi pemasaran langsung.

Kata Kunci: Optimasi Parameter, Pemasaran Langsung, *Support Vector Machine*, Algoritma Genetika.

1 PENDAHULUAN

Pemasaran langsung adalah proses mengidentifikasi potensi pembeli produk tertentu dan mempromosikan produk dengan sesuai (Wang, 2013). Pelaksanaan pemasaran langsung dari waktu ke waktu menghasilkan data dan informasi dalam bentuk laporan yang perlu di analisis oleh manajer dalam rangka mendukung keputusan. Namun, itu adalah tugas yang sulit bagi manusia untuk menganalisis data yang kompleks yang luas (Turban, 2012). Kesulitan ini menyebabkan perkembangan teknik intelijen bisnis, yang bertujuan mengekstraksi pengetahuan yang berguna untuk mendukung pengambilan keputusan.

Pemasaran langsung adalah sistem pemasaran interaktif yang menggunakan berbagai saluran untuk menargetkan pelanggan potensial (Talla, Leus, & Spieksma, 2011). Pemasaran langsung lainnya juga menggunakan e-mail pemasaran, telemarketing, broadcast fax, dan kupon. Dalam pemasaran langsung, menggunakan catatan respon pelanggan yang tersedia saat ini, dapat memperkirakan jumlah tanggapan atau tingkat respons secara keseluruhan, dan penggunaan informasi tersebut dalam membuat keputusan manajerial.

Salah satu cara yang efektif untuk menganalisa laporan dari kampanye sebelumnya dan serupa dalam mencari tren dan pola adalah melalui intelejen bisnis dan teknik data mining, untuk membangun model dan kemudian ekstrak pengetahuan (Witten, 2011). Intelejen bisnis adalah sebuah konsep luas yang mencangkup data mining yang terdiri dalam ekstraksi pengetahuan dari data mentah.

Selain itu, marketing memiliki sedikit pengetahuan tentang data mining, maka ada kebutuhan untuk mengembangkan suatu kerangka kerja yang disederhanakan untuk membantu marketing dalam memanfaatkan metode data mining untuk pemasaran langsung. Beberapa studi yang di lakukan untuk memprediksi pemasaran dengan menggunakan metode komputasi antara lain: support vector machine (SVM) (Moro & Laureano, 2012) dan multi layer perceptron (MLP) (Elsalamony & Elsayad, 2013).

Multi layer perceptron (MLP) diketahui berfungsi untuk memprediksi dan mengklasifikasi sebuah masalah yang rumit, yang memungkinkan pengakuan terhadap data yang besar (Elsalamony & Elsayad, 2013). Tetapi kinerja MLP tergantung pada parameter, bobot dan fungsi pengalihan, banyak variable dan *overfitting* (Kahrizi & Hashemi, 2014).

Suppor vector machine (SVM) bekerja lebih baik daripada MLP, dengan standar pembelajaran backpropagation, kinerja SVM lebih unggul, hal ini disebabkan karena kemampuan generalisasi support vector machine berdimensi tinggi dalam ruang highdimensional (Martinez, Sanchez, & Velez, 2010). Support vector machine digunakan untuk klasifikasi pola, pemetaan dalam ruang input dengan nonlinear merubah ke ruang berdimensi tinggi, di mana masalah linear klasifikasi menjadi optimal (Ren, 2012).

SVM dapat mengatasi masalah klasifikasi dan regresi dengan linier ataupun nonlinier kernel yang dapat menjadi satu kemampuan algoritma pembelajaran untuk klasifikasi (Kara,

Copyright @ 2015 IlmuKomputer.Com

Acar, & Kaan, 2011). Selain memiliki banyak kemampuan yang telah disebutkan diatas, metode SVM juga memiliki kelemahan pada sulitnya pemilihan parameter SVM yang optimal, keakuratan klasifikasi atau regresi ditentukan oleh sekelompok parameter yang sesuai (Xiang, 2013).

Kinerja SVM sangat tergantung pada pilihan yang memadai dari nilai-nilai parameter, termasuk misalnya, kernel dan parameter regularisasi. Pemilihan parameter SVM umumnya sebagai masalah optimasi di mana teknik pencarian digunakan untuk menemukan konfigurasi parameter yang memaksimalkan kinerja SVM (Rossi & Soares, 2012).

Ada banyak teknik optimasi yang telah digunakan untuk mengoptimasi parameter pada machine learning, seperti algoritma genetika (GA) (Ilhan & Tezel, 2013) dan particle swarm optimization (PSO) (Khoshahval, Minuchehr, & Zolfaghari, 2011).

Karena konsep sederhana, implementasi mudah, dan konvergensi cepat, particle swarm optimization (PSO) dapat diterapkan untuk berbagai aplikasi di berbagai bidang untuk memecahkan masalah optimasi (Liu, Tian, Chen, & Li, 2013). PSO, sebagai alat optimasi, yang dapat membantu menentukan parameter optimum. Tetapi PSO memiliki ketergantungan yang sensitif pada parameter yang digunakan (Yusup, Zain, Zaiton, & Hashim, 2012).

GA adalah menemukan popularitas sebagai alat desain karena fleksibilitas, intuitif dan kemampuan untuk memecahkan sangat non-linear, optimasi mixed integer masalah (Khoshahval et al., 2011). Algoritma Genetika digunakan untuk mengoptimasi parameter yang optimal dengan ruang lingkup yang besar, dengan pemilihan parameter yang tepat algoritma genetika akan lebih optimal (Wang et al., 2013). Algoritma genetika memiiki kelemahan yaitu pemilihan parameter yang salah dapat mengurangi akurasi yang dihasilkan.

Metode algoritma genetika digunakan untuk mengoptimalkan parameter dan untuk lebih menemukan bagian parameter yang dioptimalkan (Wang et al., 2014). Namun semua komponen algoritma genetika bersifat random atau acak menghasilkan solusi yang dihasilkan berbeda-beda. Algoritma genetika di terapkan pada optimasi parameter untuk support vector machine, sehingga hasil yang diperoleh adalah pemilihan optimasi parameter yang sesuai (Zhao, Fu, Ji, Tang, & Zhou, 2011). Permasalahan yang sering dihadapi oleh algoritma Genetika adalah memperoleh solusi optimal setelah serangkaian melakukan perulangan, tetapi kejadian ini dapat dihindari dengan memilih nilai-nilai parameter yang tepat.

Dari uraian diatas, pada penelitian ini algoritma genetika akan di terapkan untuk optimasi parameter pada support vector machine.

2 PENELITIAN TERKAIT

Data set bank dalam jumlah besar yang dihasilkan setiap hari di banyak lembaga. Data set bank dapat digunakan untuk membangun dan memelihara hubungan langsung dengan pelanggan untuk menargetkan mereka secara individu untuk penawaran tertentu. Selain itu, marketing memiliki sedikit pengetahuan tentang data mining, maka ada kebutuhan untuk mengembangkan suatu kerangka kerja yang disederhanakan untuk membantu marketing dalam memanfaatkan metode data mining untuk pemasaran langsung.

Penelitian yang dilakukan (Elsalamony & Elsayad, 2013) Dalam penelitian ini menggunakan data set bank Suppor vector machine (SVM) bekerja lebih baik daripada Multi Layer Perceptron MLP, dengan standar pembelajaran backpropagation, kinerja SVM lebih unggul, hal ini disebabkan karena kemampuan generalisasi support vector machine berdimensi tinggi dalam ruang highdimensional.

Metode SVM juga memiliki kelemahan pada sulitnya pemilihan parameter SVM yang optimal, keakuratan klasifikasi atau regresi ditentukan oleh sekelompok parameter yang sesuai (Moro & Laureano, 2012). Pemilihan parameter SVM umumnya sebagai masalah optimasi di mana teknik pencarian digunakan untuk menemukan konfigurasi parameter yang memaksimalkan kinerja SVM.

Karena konsep sederhana, implementasi mudah, dan konvergensi cepat, particle swarm optimization (PSO) dapat diterapkan untuk berbagai aplikasi di berbagai bidang untuk memecahkan masalah optimasi (Vieira & Mendonc, 2013). PSO sebagai alat optimasi yang dapat membantu menentukan parameter optimum, tetapi PSO memiliki ketergantungan yang sensitif pada parameter yang digunakan.

Algoritma genetika digunakan untuk mengoptimasi parameter yang optimal dengan ruang lingkup yang besar, dengan pemilihan parameter yang tepat algoritma genetika akan lebih optimal (Wang et al., 2013). Algoritma genetika memiliki kelemahan yaitu pemilihan parameter yang salah dapat mengurangi akurasi yang dihasilkan. Permasalahan yang sering dihadapi oleh algoritma genetika adalah memperoleh solusi optimal setelah serangkaian melakukan perulangan, tetapi kejadian ini dapat dihindari dengan memilih nilai-nilai parameter yang tepat.

Dari permasalahan pada penelitian-penelitian di atas disimpulkan bahwa untuk mengolah data set bank adalah sebuah masalah yang rumit karena merupakan data yang kompleks. berdasarkan analisa bahwa metode gabungan dua metode atau lebih (e*nsemble*) menunjukan hasil yang lebih baik dibanding metose individual. SVM yang mampu mengatasi masalah klasifikasi dan regresi dengan linear maupun nonlinear kernel yang dapat menjadi satu kemampuan algoritma, di mana masalah linear klasifikasi menjadi optimal, sedangkan algoritma genetika digunakan untuk mengoptimasi parameter yang optimal dengan ruang lingkup yang besar, dengan pemilihan parameter yang tepat algoritma genetika akan lebih optimal. Oleh karena itu pada penelitian ini diusulkan mengunakan metode e*nsemble* dengan menggunkan algoritma genetika untuk mengoptimasi parameter SVM dengan kombinasi kernel yang berbeda.

3 METODE YANG DIUSULKAN

3.1 Algoritma Genetika

Algoritma genetika adalah suatu teknik optimasi yang didasarkan pada prinsip genetika dan seleksi alam. algoritma genetika merupakan metode pencarian yang disesuaikan dengan prosess generika dari orginisme biologi yang berdasar pada teori evolusi Charles Darwin (Shukla, Tiwari, & Kala, 2010). Algoritma genetika terinspirasi dari mekanisme seleksi alam, dimana individu yang lebih kuat kemungkinan akan menjadi pemenang dalam lingkungan yang kompetitif dan solusi yang optimal dapat diperoleh dan diwakilkan oleh pemenang akhir dari permainan genetika (Haupt & Haupt, 2004). Pada algoritma genetika tersedia solusi yang diterapkan pada sebuah populasi individu yang masing-masing mewakili

Copyright @ 2015 IlmuKomputer.Com

solusi yang mungkin. Setiap solusi yang mungkin disebut dengan kromosom.

Algoritma genetika menggunakan analogi secara langsung dari kebiasaan yang alami yaitu seleksi alam. Algoritma ini berkerja dengan sebuah populasi yang terdiri dari individu-individu, yang masing-masing individu merepresentasikan sebuah solusi yang mungkin bagi persoalan yang ada. Dalam kaitan ini, individu dilambangkan dengan dengan sebuah nilai fitness yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada.

Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi. Untuk menggunakan algoritma genetika, solusi permasalahan direpresentasikan sebagai khromosom (Weise, 2009). Tiga aspek yang penting untuk penggunaan algoritma genetik:

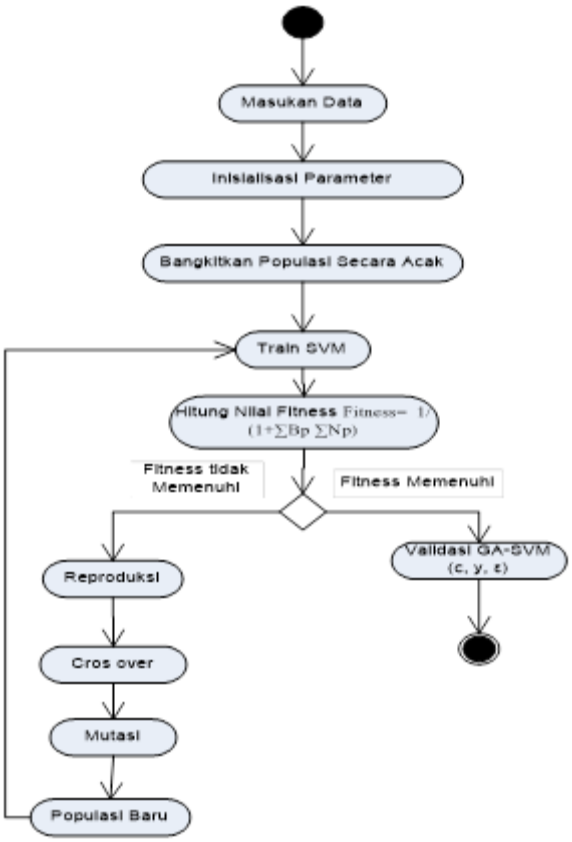
1. Defenisi fungsi fitness
2. Defenisi dan implementasi representasi genetik
3. Defenisi dan implementasi operasi genetik

3.2 Support Vector Machine

Support vector macine adalah salah satu metode klasifikasi dua kelas dan teori ini di dasarkan pada gagasan resiko struktural, support vector machine menggunakan fungsi kernel untuk memetakan data input ke ruang berdimensi tinggi dan menemukan hyper plane optimal untuk memisahkan data dua kelas (Aydin, Karakose, & Akin, 2011). Untuk mendapatkan optimal hyper plane yang memisahkan dua kelas berbeda pada ruang vector. Optimal hyper plane adalah jarak terjauh dari hyper plane kedua kelas tersebut. Pada permasalahan pemisahan secara linear, optimal hyper plane dapat memisahkan dua kelas yang berbeda dengan baik dan vector-vector yang terdekat dengan optimal hyper plane disebut support vector (Wu et al., 2007).

3.3 Support Vector Machine dan Algoritma Genetika

Gambar 1 menggambarkan metode algoritma yang diusulkan dalam penelitian ini. Pada pengolahan data awal, inisialisasi parameter kernel C, y dan ε, kemudiam bangkitkan populasi dari kromosom dibangkitkan secara acak. Ukuran populasi di set ke 5, selanjutnya training SVM, kemudian evaluasi fitnes. Pada tahap ini fitness dari setiap kromosom dievaluasi, setelah itu cek nilai fitness, jika kondisi terpenuhi berhenti, selain itu lakukan reproduksi. Pada tahap ini populasi baru dibuat dengan perulangan mengikuti langkah-langkah sebelum populasi baru selesai, kemudian lakukan crossover. Dengan probabilitas crossover, crossover dari induk dibuat untuk membentuk offspring's (anak). Pada cross over, kromosom dipasangkan secara random, kemudian lakukan mutasi. Setelah operasi crossover berhasil, string sebagai subyekan untuk operasi mutasi, hal ini untuk mencegah runtuhnya seluruh solusi dari populasi menjadi local optimum dari penyelesaian masalah. Variabel dalam string yang akan bermutasi dipilih secara acak, kemudian Populasi baru terbentuk, ulangi langkah train SVM.



Gambar 1. Penggabungan Support Vector Machine dan Algoritma Genetika

4 HASIL DAN PEMBAHASAN

Penelitian yang dilakukan menggunakan komputer dengan spesifikasi CPU Intel Core i3 1.6GHz, RAM 2GB, dan sistem operasi Microsoft Windows 7 Professional 64-bit. Aplikasi yang digunakan adalah RapidMiner 5.2. Penelitian ini menggunakan dataset Bank. Dataset ini didapat dari UCI Machine Learning Repository.

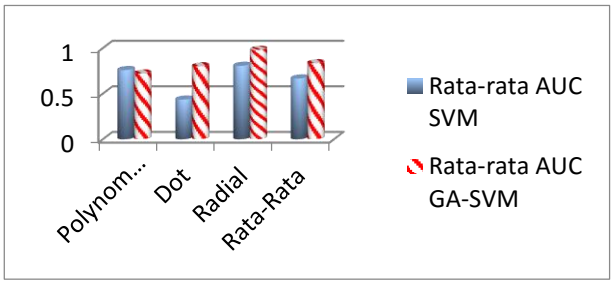
Setelah eksperimen yang dilakukan dengan penerapan model GA-SVM dilakukan uji beda dengan membandingkan akurasi prediksi pemasaran langsung pada data testing support vector machine sebelum dan sesudah dioptimasi dengan algoritma genetika berdasarkan nilai akurasi. Pengujian penerapan algoritma genetika yang digunakan untuk melakukan optimasi parameter C, γ dan ε pada metode support vector machine dalam prediksi pemasaran langsung yang dilakukan dengan menggunakan tiga tipe kernel yaitu kernel dot, radial, dan polynomial.

Berdasarkan hasil eksperimen dan analisis data dalam penelitian ini, maka dapat diperoleh perbedaan rata-rata nilai AUC pada pengujian model SVM sebelum dan sesudah dilakukan optimasi parameter SVM dengan menggunakan algoritma genetika data yang digunakan pada pemasaran langsung. Perbandingan yang dihasilkan dapat memberikan informasi/gambaran tentang perbandingan rata-rata tingkat akurasi pada penerpan model tersebut. Tingkat perbandingan rata-rata nilai AUC dapat dilihat pada Tabel 2 dan Gambar 2.

Copyright @ 2015 IlmuKomputer.Com

Tabel 1. Perbandingan Rata-rata Nilai AUC Pada Tipe Kernel Dot, Polinomial dan Radial.

Tipe Kernel	Rata-rata AUC	
	SVM	GA-SVM
Polynomial	0,7595	0,7217
Dot	0,4354	0,801
Radial	0,806	0,981
Rata-Rata	0,666966667	0,834566667



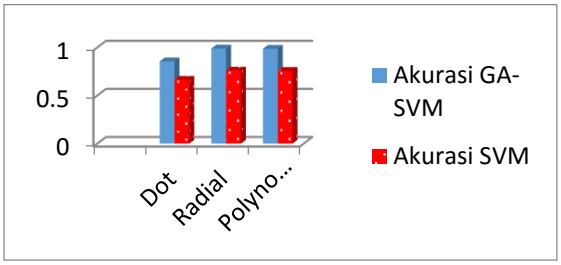
Gambar 2. Grafik Perbandingan Rata-rata Nilai AUC SVM dengan GA-SVM

Berdasarkan Tabel 1, menunjukkan bahwa pengujian penggunaan model GA-SVM pada tipe kernel dot, radial dan polinomial untuk melakukan prediksi pemasaran langsung memiliki rata-rata AUC lebih tinggi jika dibandingkan dengan model SVM, ini menunjukan bahwa pada penggunaan tipe kernel polynomial, dot dan radial dapat meningkatkan peningkatan nilai akurasi prediksi pemasaran langsung pada model SVM setelah dilakukan optmimasi parameter dengan algoritma genetika.

Hasil eksperimen dan analisis data dalam penelitian ini, dapat diperoleh perbandingan nilai akurasi tertinggi pada pengujian model SVM sebelum dan sesudah dilakukan optimasi parameter dengan algoritma genetika pada data testing yang diterapkan pada prediksi pemasaran langsung. Perbandingan ini dapat memberikan gambaran tentang tingkat akurasi terbaik pada penerapan model tersebut. Perbandingan terbaik pada model tersebut dapat dilihat pada Tabel 3 dan Gambar 3.

Tabel 2. Perbandingan Akurasi Terbaik antara GA-SVM Dengan SVM

Tipe Kernel	GA-SVM				SVM			
	Parameter			Akurasi	Parameter			Akurasi
	C	γ	ε		C	γ	ε	
Dot	61.628	0.1893	0.5542	85.59%	863.347	0.1754	0.5579	66,49
Radial	92.073	0.03036	0.54485	98.89%	92.073	0.03036	0.54485	76.05
Polynomial	31.746	0.9568	0.72688	98.67%	31.746	0.9568	0.72688	75.61



Gambar 3. Grafik Perbandingan Terbaik Prediksi Pemasaran Langsung dengan Metode GA-SVM dan SVM

Mengacu Tabel 2, menunjukkan bahwa penerapan model SVM sebelum dilakukan optmiasi parameter nilai akurasi tertinggi terjadi pada tipe kernel Dot dengan nilai C = 863.347, γ = 0.1754 dan ε = 0.5579 dengan nilai akurasi sebesar 66.49%, nilai akurasi tertinggi pada tipe kernel radial terjadi pada nilai C = 92.073, γ = 0.03036 dan ε = 0.54485 dengan nilai akurasi sebesar 76.05% dan nilai akurasi tertinggi pada tipe kernel Polynomial dengan c = 31.746, γ = 0.9568 dan ε = 0.72688 dengan nilai akurasi sebesar 75,61%. Sedangkan penerapan SVM setelah dilakukan optimasi parameter dengan menggunakan GA nilai akurasi tertinggi pada tipe kernel dot terjadi pada nilai c = 61.628, γ = 0.1893, dan ε = 0.5542 dengan tingkat akurasi sebesar 85.59%, untuk nilai akurasi tertinggi pada tipe kernel radial terjadi pada nilai c = 92.073 γ = 03036 dan ε = 0.54485 dengan nilai akurasi sebesar 98,89% dan untuk nilai akurasi tertinggi pada tipe kernel polynomial terjadi pada nilai C = 31.746 γ = 0.09568 dan ε = 0.72688 dengan nilai akurasi sebesar 98.67%.

Hasil menunjukan support vector machine diperoleh pada parameter c = 31.746 gamma = 0.9568 dan epsilon = 0.1887 dengan nilai akurasi sebesar 57.43% dan nilai AUC sebesar 0.7268. Sedangkan pada metode support vector machine yang dipadu dengan algoritma genetika hasil terbaika diperoleh pada parameter c = 31.746 gamma = 0.9568 dan epsilon = 0.18 dengan tingkat akurasi 98,67% dan nilai AUC sebesar 0,938.

5 KESIMPULAN

Pengujian penerapan algoritma genetika yang digunakan untuk melakukan optimasi parameter C, γ dan ε pada metode support vector machine dalam prediksi pemasaran langsung yang dilakukan dengan menggunakan tiga tipe kernel yaitu kernel dot, radial, dan polynomial.

Berdasarkan hasil uji beda menunjukkan bahwa ada perbedaan yang signifikan pada pada nilai rata-rata AUC hasil eksperimen SVM sebelum dan sesudah dilakukan optimasi parameter C, γ dan ε dengan algoritma genetika. Sehingga dapat disimpulkan bahwa penerapan model SVM yang dioptimasi parameter C, γ dan ε dengan algoritma genetika meningkatkan akurasi dalam prediksi pemasaran langsung. Dengan nilai masing-masing parameter c = 92.073 γ = 03036 dan ε = 0.54485 dengan nilai akurasi sebesar 98,89%.

Berdasarkan hasil pengujian yang dilakukan untuk memecahkan masalah prediksi pemasaran langsung, dapat disimpulkan bahwa eksperimen dengan tingkat akurasi tertinggi pada metode SVM dengan tipe kernel radial sebelum dilakukan optimasi parameter dengan nilai akurasi sebesar 76.05%. Berikutnya dilakukan penerapan algoritma genetika untuk optimasi parameter c, γ dan ε dengan tipe kernel radial dengan nilai akurasi sebesar 98.89%.

Copyright @ 2015 IlmuKomputer.Com

REFERENSI

Aydin, I., Karakose, M., & Akin, E. (2011). A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Applied Soft Computing*, *11*, 120–129.

Elsalamony, H. A., & Elsayad, A. M. (2013). Bank Direct Marketing Based on Neural Network and C5 . 0 Models. *International Journal of Engineering and Advanced Technology (IJEAT)*, *2*(6).

Frias-Martinez, E., Sanchez, A., & Velez, J. (2010). Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition. *Engineering Applications of Artificial Intelligence*, *19*(6), 693–704.

Ilhan, I., & Tezel, G. (2013). A genetic algorithm-support vector machine method with parameter optimization for selecting the tag SNPs. *Journal of Biomedical Informatics*, *46*(2), 328–40.

Kahrizi, A., & Hashemi, H. (2014). Neuron curve as a tool for performance evaluation of MLP and RBF architecture in first break picking of seismic data. *Journal of Applied Geophysics*, *108*, 159–166.

Kara, Y., Acar, M., & Kaan, Ö. (2011). Expert Systems with Applications Predicting direction of stock price index movement using artificial neural networks and support vector machines : The sample of the Istanbul Stock Exchange. *Expert Systems With Applications*, *38*(5), 5311–5319.

Khoshahval, F., Minuchehr, H., & Zolfaghari, a. (2011). Performance evaluation of PSO and GA in PWR core loading pattern optimization. *Nuclear Engineering and Design*, *241*(3), 799–808.

Liu, H., Tian, H., Chen, C., & Li, Y. (2013). Electrical Power and Energy Systems An experimental investigation of two Wavelet-MLP hybrid frameworks for wind speed prediction using GA and PSO optimization, *52*, 161–173.

Moro, S., & Laureano, R. M. S. (2012). Using Data Mining for Bank Direct Marketing: An application of the CRISP-DM methodology. *European Simulation and Modelling Conference*, (Figure 1), 117–121.

Ren, J. (2012). ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Knowledge-Based Systems*, *26*, 144–153.

Rossi, L. D., & Soares, C. (2012). Neurocomputing Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, *75*, 3–13.

Shi, Y., Tian, Y., & Kou, G. (2011). *Optimization Based Data Mining Theory and Applications*. (xx, Ed.). Springer London.

Shukla, A., Tiwari, R., & Kala, R. (2010). *Real Life Application of Soft Computing*. CRC Press.

Talla, F., Leus, R. &, & Spieksma, F. C. R. (2011). Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms. *European Journal of Operational Research*, *210*(3), 670–683.

Turban, E. (2012). *Information Technology for Management*. (B. L. Golub, Ed.) (8th ed.). United States of America: John Wiley & Sons, Inc.

Vieira, S. M., & Mendonc, L. F. (2013). Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients, *13*, 3494–3504.

Wang, J. (2013). Data Mining Framework for Direct Marketing : A Case Study of Bank Marketing. *International Journal of Computer Science and Issues*, *10*(2), 198–203.

Wang, Y., Chen, X., Jiang, W., Li, L., Li, W., Yang, L., … Li, X. (2013). Predicting human microRNA precursors based on an optimized feature subset generated by GA-SVM. *Genomics*, *98*(2), 73–8.

Wang, Y., Li, Y., Wang, Q., Lv, Y., Wang, S., Chen, X., … Li, X. (2014). Computational identification of human long intergenic non-coding RNAs using a GA-SVM algorithm. *Gene*, *533*(1), 94–9.

Witten, I. H. (2011). *Data Mining Practical Machine Learning Tools and Techniques* (3rd ed.). USA: Elsevier.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., … Steinberg, D. (2007). *Top 10 algorithms in data mining*. *Knowledge and Information Systems* (Vol. 14, pp. 1–37).

Xiang, C. (2013). A Chaotic Time Series Forecasting Model Based on Parameters Simultaneous Optimization Algorithm. *Journal of Information and Computational Science*, *10*(15), 4917–4930.

Yusup, N., Zain, A. M., Zaiton, S., & Hashim, M. (2012). Procedia Engineering Overview of PSO for Optimizing Process Parameters of Machining.

Zhao, M., Fu, C., Ji, L., Tang, K., & Zhou, M. (2011). Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, *38*(5), 5197–5204.

BIOGRAFI PENULIS

Ispandi. Memperoleh gelar M.Kom dari Sekolah Tinggi Manajemen Ilmu Komputer Nusa Mandiri, Jakarta. Staf pengajar di salah satu Perguruan Tinggi Swasta. Minat penelitian saat ini pada bidang data mining..

Romi Satria Wahono. Memperoleh Gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiai ilmiah ACM, PMI dan IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Integrasi Metode Information Gain Untuk Seleksi Fitur dan Adaboost Untuk Mengurangi Bias Pada Analisis Sentimen Review Restoran Menggunakan Algoritma Naïve Bayes

Lila Dini Utami
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
lila.ldu@gmail.com

Romi Satria Wahono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
romi@brainmatics.com

Abstrak: Internet merupakan bagian penting dari kehidupan sehari-hari. Saat ini, tidak hanya dari anggota keluarga dan teman-teman, tetapi juga dari orang asing yang berlokasi diseluruh dunia yang mungkin telah mengunjungi restoran tertentu. Konsumen dapat memberikan pendapat mereka yang sudah tersedia secara online. Ulasan yang terlalu banyak akan memakan banyak waktu dan pada akhirnya akan menjadi bias. Klasifikasi sentimen bertujuan untuk mengatasi masalah ini dengan cara mengklasifikasikan ulasan pengguna ke pendapat positif atau negatif. Algoritma Naïve Bayes (NB) adalah tekhnik *machine learning* yang populer untuk klasifikasi teks, karena sangat sederhana, efisien dan memiliki performa yang baik pada banyak domain. Namun, naïve bayes memiliki kekurangan yaitu sangat sensitif pada fitur yang terlalu banyak, sehingga membuat akurasi menjadi rendah. Oleh karena itu, dalam penelitian ini menggunakan Information Gain (IG) untuk seleksi fitur dan metode adaboost untuk mengurangi bias agar dapat meningkatkan akurasi algoritma naïve bayes. Penelitian ini menghasilkan klasifikasi teks dalam bentuk positif dan negatif dari *review* restoran. Pengukuran naïve baye berdasarkan akurasi sebelum dan sesudah penambahan metode seleksi fitur. Validasi dilakukan dengan menggunakan 10 *fold cross validation*. Sedangkan pengukuran akurasi diukur dengan *confusion matrix* dan kurva ROC. Hasil penelitian menunjukkan peningkatan akurasi naïve bayes dari 73.00% jadi 81.50% dan nilai AUC dari 0.500 jadi 0.887. Sehingga dapat disimpulkan bahwa integrasi metode information gain dan adaboost pada analisis sentimen *review* restoran ini mampu meningkatkan akurasi algoritma naïve bayes.

Kata Kunci: analis sentimen, *review*restoran, klasifikasi teks, adaboost, information gain, naïve bayes.

1 PENDAHULUAN

Pertumbuhan jaringan sosial yang ada saat ini, membuat konsumen menggunakan konten dalam media untuk membuat keputusan yang lebih baik. Lebih banyak konsumen yang melihat pendapat dari konsumen lain sebelum memilih sebuah restoran. Di sisi lain, untuk restoran, sejumlah besar informasi publik yang tersedia bisa dijadikan sebagai bahan intropeksi untuk menjadikan restoran yang lebih baik (Reyes & Rosso, 2012). Beberapa konsumen menuangkan opini atau pengalaman mereka melalui media sosial seperti Facebook, Twitter, atau situs media yang lainnya. *Review* restoran yang dibuat secara online adalah saluran yang menghubungkan pengunjung satu dengan pengunjung lainnya. Hal ini merupakan layanan penyaringan yang dirancang untuk membantu konsumen. Hasil pencarian biasanya disajikan sebagai daftar restoran yang cocok, ditampilkan dengan singkat melalui sebuah gambar yang disertakan nama restoran, alamat serta *review* keseluruhan makanan dan layanan dan sebuah *hyperlink* ke halaman web yang berdedikasi restoran (Zhang, Ye, Law, & Li, 2010). Jika membaca *review* tersebut secara keseluruhan bisa memakan waktu dan sebaliknya jika hanya sedikit *review* yang dibaca, evaluasi akan bias. Klasifikasi sentimen bertujuan untuk mengatasi masalah ini dengan secara otomatis mengelompokkan *review* pengguna menjadi opini positif atau negatif.

Ada beberapa penelitian yang sudah dilakukan dalam hal pengklasifikasian sentimen terhadap *review* yang tersedia, diantaranya adalah penelitian oleh Kang, Yoo & Han, yang menggunakan algoritma naïve bayes dan mengkombinasikan kata sifat dengan N-grams (Kang, Yoo, & Han, 2012b). Lalu ada pula penelitian dari Zhang, Ye, Zhang & Li, dimana pengklasifikasian sentimen pada *review* restoran di internet yang ditulis dalam bahasa Canton menggunakan algoritma klasifikasi naïve bayes dan Support Vector Machine (SVM) (Zhang, Ye, Zhang, & Li, 2011). Sedangkan penelitian oleh Yulan He, menggunakan information gain dan naïve bayes untuk mempelajari ulasan pelanggan tentang sebuah film dan produk lainnya (He & Zhou, 2011).

Naïve bayes banyak digunakan untuk klasifikasi teks dalam *machine learning* yang didasarkan pada fitur probabilitas (Zhang & Gao, 2011). Naïve bayes sangat sederhana dan efisien. Sebagai teknologi *preprocessing* yang penting dalam klasiseleksi fitur dapat meningkatkan skalabilitas, efisiensi dan akurasi dari klasifikasi teks. Secara umum, metode seleksi fitur yang baik harus mempertimbangkan domain dan algoritma karakteristik. Sebagai *classifier*, naïve bayes sangat sederhana dan efisien serta sangat sensitif terhadap seleksi fitur (Chen, Huang, Tian, & Qu, 2009). Klasifikasi positif yang muncul 10% lebih tinggi dari akurasi klasifikasi negatif dan tampak beberapa kasus seperti star atau bintang dengan *review* yang tidak cocok. Algoritma naïve bayes diusulkan dan diukur melalui eksperimen komparatif dengan Unigrams dan Bigrams sebagai fiturnya. Dalam hal ini, naïve bayes membuktikan tingkat akurasi yang bagus saat klasifikasi dianggap seimbang (Kang, Yoo, & Han, 2012a). Akan tetapi, akurasi menjadi tidak akurat saat menghadapi sentimen klasifikasi yang kompleks.

Karena ketersediaan teks dalam bentuk digital menjamur dan meningkatnya kebutuhan untuk mengakses dengan cara yang fleksibel, klasifikasi teks menjadi tugas dasar dan penting. Meskipun sederhana, algoritma naïve bayes merupakan algoritma populer untuk klasifikasi teks (Ye, Zhang, & Law, 2009). Akan tetapi, masalah utama untuk

Copyright @ 2015 IlmuKomputer.Com

klasifikasi teks adalah dimensi tinggi dari ruang fitur. Hal ini sangat sering karena domain teks memiliki beberapa puluhan ribu fitur. Kebanyakan dari fitur ini tidak relevan dan bermanfaat bagi klasifikasi teks. Bahkan beberapa fitur mungkin mengurangi akurasi klasifikasi. Selain itu, sejumlah besar fitur dapat memperlambat proses klasifikasi (Chen et al., 2009).

Tingkatan lain yang umumnya ditemukan dalam pendekatan klasifikasi sentimen adalah seleksi fitur. Seleksi fitur bisa membuat pengklasifikasi baik lebih efisien dan efektif dengan mengurangi jumlah data yang dianalisa, maupun mengidentifikasi fitur yang sesuai untuk dipertimbangkan dalam proses pembelajaran (Moraes, Valiati, & Neto, 2013). Menurut John, Kohavi, dan Pfleger dalam Chen, ada dua jenis utama metode seleksi fitur dalam *machine learning*: wrapper dan filter. Wrapper menggunakan akurasi klasifikasi dari beberapa algoritma sebagai fungsi evaluasinya (Chen et al., 2009). Metode filter terdiri dari *document frequency*, *mutual information*, information gain, dan *chi-square*. Information gain sering lebih unggul dibandingkan yang lain. Information gain mengukur berapa banyak informasi kehadiran dan ketidakhadiran dari suatu kata yang berperan untuk membuat keputusan klasifikasi yang benar dalam *class* apapun. Informastion gain adalah salah satu pendekatan filter yang sukses dalam pengklasifikasian teks (Uysal & Gunal, 2012).

Sementara itu, menurut Hu (Hu & Hu, 2005), adaboost adalah algoritma yang ide dasarnya adalah untuk memilih dan menggabungkan sekelompok pengklasifikasi lemah untuk membentuk klasifikasi yang kuat. Adaboost adalah algoritma yang iteratif menghasilkan pengklasifikasi dan kemudian menggabungkan mereka untuk membangun klasifikasi utama (Kim, Hahn, & Zhang, 2000). Algoritma adaboost iteratif bekerja pada klasifikasi naïve bayes dengan bobot normal dan mengklasifikasikan masukan yang diberikan ke dalam kelas yang berbeda dengan beberapa atribut (Korada, Kumar, & Deekshitulu, 2012). Adaboost dirancang khusus untuk klasifikasi. Adaboost adalah algoritma pembelajaran yang dapat digunakan untuk meningkatkan akurasi untuk setiap pembelajaran algoritma yang lemah. Algoritma adaboost digunakan untuk meningkatkan akurasi lemah klasifikasi naïve bayes.

Pada penelitian ini menggunakan algoritma naïve bayes disertai information gain sebagai metode seleksi fitur dan metode adaboost sebagai teknik untuk memperbaiki tingkat klasifikasi yang diterapkan untuk mengklasifikasikan teks pada komentar dari *review* suatu restoran untuk meningkatkan akurasi analisa sentimen.

2 PENELITIAN TERKAIT

Ada beberapa penelitian yang menggunakan algoritma naïve bayes sebagai pengklasifikasi, metode adaboost, atau information gain sebagai seleksi fitur dalam klasifikasi teks analisa sentiment pada *review*, diantaranya:Penelitian yang dilakukan oleh Zhang, Ye, Zhang, dan Li mengenai analisa sentimen pada *review* restoran yang ditulis dalam bahasa Canton (Zhang et al., 2011b). Ulasan diambil dari situs www.openrice.com yang terdiri dari 1500 *review* positif dan 1500 *review* negatif. Dua penutur asli dilatih untuk ulasan ini dan didapatkanlah *review* yang sesuai dan digunakan untuk proses klasifikasi terdiri dari 900 *review* positif dan 900 *review* negatif. Sebagai langkah awal, peneliti melakukan seleksi fitur dengan cara mensubtitusi kalimat yang memiliki makna yang sama. Setelah subtitusi selesai, peneliti mengkombinasikan kata sifat dengan n-grams untuk melihat sentimen dalam teks. Algoritma *featureselection* yang digunakan adalah information gain. *Classifier* yang digunakan adalah support vector machine dan naïve bayes.

Sementara itu, penelitian yang dilakukan oleh Kang, Yoo, dan Han mengenai analisa sentimen pada *review* restoran. Sekitar 70.000 dokumen dikumpulkan dari pencarian situs restoran (Kang et al., 2012). Sisi positif dan negatif dikumpulkan dan diklasifikasikan sebelum sentimen analisis dilakukan. Ulasan yang terpilih adalah 5700 *review* positif dan 5700 *review* negatif. Untuk *textprocessing*, peneliti menggunakan *tokenization* dan melakukan pemilihan *review* yang mencakup kata-kata sentimen terkait dengan *review* restoran menggunakan n-grams. *Classifier* yang digunakan adalah support vector machine dan naïve bayes.

Dan penelitian yang dilakukan oleh He dan Zhou mengenai analisa sentimen pada *review* film, buku, DVD, dan barang elektronik (He & Zhou, 2011). Untuk *review* film diambil dari website IMDB dan *review* buku, DVD, dan barang elektronik diperoleh dari www.amazon.com sebanyak 100 *review* positif dan 1000 *review* negatif. Ulasan berisi peringkat terstruktur (bintang) dan teks. Untuk *textprocessing*, peneliti menggunakan *tokenization* dan melakukan pemilihan *review* yang mencakup kata-kata sentimen terkait dengan review menggunakan pengklasifikasi Lexicon Labeling, Heuristic Labeling, Self-labeled instance, Self-learned Features, dan Oracle Labeling. *Classifier* yang digunakan adalah naïve bayes dan support vector machine.

3 METODE YANG DIUSULKAN

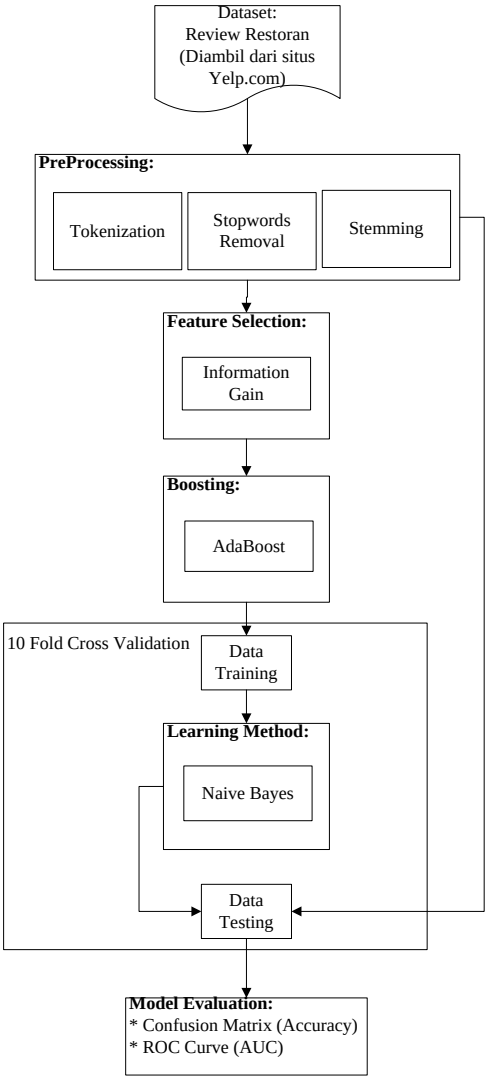
Penelitian ini menggunakan data review restoran yang berada di New York, yang diambil dari situs http://www.yelp.com/nyc. *Review* restoran yang digunakan hanya 200 *review* restoran yang terdiri dari 100 *review* positif dan 100 *review* negatif. Data tersebut masih berupa sekumpulan teks yang terpisah dalam bentuk dokumen. Data *review* positif disatukan dalam satu folder dan diberi nama positif, sedangkan data *review* negatif disatukan dalam satu folder dan diberi nama negatif.

Pre processing yang dilakukan, diantaranya adalah:

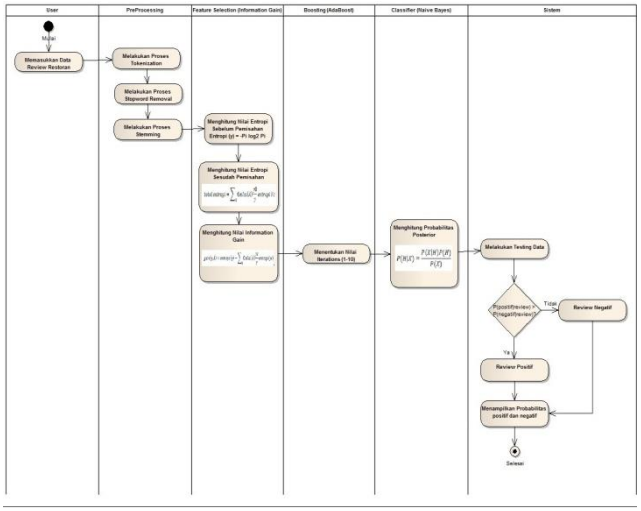
a. *Tokenization*
   Dalam proses tokenization ini, semua kata yang ada di dalam tiap dokumen dikumpulkan dan dihilangkan tanda bacanya, serta dihilangkan jika terdapat simbol atau apapun yang bukan huruf
b. *StopwordsRemoval*
   Dalam proses ini, kata-kata yang tidak relevan akan dihapus, seperti kata the, of, for, with yang merupakan kata-kata yang tidak mempunyai makna tersendiri jika dipisahkan dengan kata yang lain dan tidak terkait dengan dengan kata sifat yang berhubungan dengan sentimen.
c. *Stemming*
   Dalam proses ini kata-kata akan dikelompokkan ke dalam beberapa kelompok yang memiliki kata dasaryang sama, seperti *drug*, *drugged*, dan *drugs* di mana kata dasar dari semuanya adalah kata *drug*.

Feature selection yang peneliti usulkan adalah metode dengan jenis filter, yakni information gain dan metode *boosting* yaitu adaboost, yang digunakan secara integrasi agar akurasi algoritma naïve bayes dapat meningkat. Penelitian ini nantinya menghasilkan akurasidan nilai AUC. Lihat Gambar 1 untuk model yang diusulkan secara detail dan ringkas, sementara itu Gambar 2 adalah model yang diusulkan berbentuk *activity diagram*.

Copyright @ 2015 IlmuKomputer.Com



Gambar 1. Model yang Diusulkan



Gambar 2.*Activity Diagram* Model yang Diusulkan

4 HASIL PENELITIAN

Proses eksperimen ini menggunakan aplikasi *RapidMiner* 5.2. Untuk pengujian model dilakukan menggunakan dataset *review* restoran. Spesifikasi komputer yang digunakan untuk eksperimen ini dapat dilihat pada Tabel 1.

Tabel 1 Spesifikasi Komputer yang Digunakan

Processor	*Intel(R) Celeron(R) CPU 874 @1.10GHz*
Memori	4.00 GB
Harddisk	320 GB
Sistem Operasi	*Microsoft Windows 7*
Aplikasi	*RapidMiner 5.2*

Proses klasifikasi di sini adalah untuk menentukan sebuah kalimat sebagai anggota *class* positif atau *class* negatif berdasarkan nilai perhitungan probabilitas dari rumus bayes yang lebih besar. Jika hasil probabilitas kalimat tersebut untuk *class* positif lebih besar dari pada *class* negatif, maka kalimat tersebut termasuk ke dalam *class* positif. Jika probabilitas untuk *class* positif lebih kecil dari pada *class* negatif, maka kalimat tersebut termasuk ke dalam *class* negatif. Penulis hanya menampilkan 10 dokumen dari keseluruhan 200 data training dan 4 kata yang berhubungan dengan sentimen dan yang paling sering muncul, yaitu *bad*, *good*, *delicious* dan *disappoint*. *Bad*, muncul sebanyak 21 kali yaitu dalam *review* positif sebanyak 4 kali dan *review* negatif sebanyak 17 kali. *Good*, muncul sebanyak 91 kali yaitu dalam *review* positif sebanyak 50 kali dan *review* negatif sebanyak 41 kali. *Delicious*, muncul sebanyak 25 kali yaitu dalam *review* positif sebanyak 22 kali dan *review* negatif sebanyak 3 kali. *Disappoint*, muncul sebanyak 33 kali yaitu dalam *review* positif sebanyak 3 kali dan *review* negatif sebanyak 30 kali. Kehadiran kata di dalam suatu dokumen akan diwakili oleh angka 1 dan angka 0 jika kata tersebut tidak muncul di dalam dokumen.

Tabel 2 Hasil Klasifikasi Teks

Dokumen Ke-	***Bad***	***Delicious***	***Good***	***Dissapoint***	***Class***
1	0	2	3	0	Positif
2	0	1	1	0	Positif
3	0	2	1	0	Positif
101	1	0	3	2	Negatif
102	2	1	1	1	Negatif
103	1	0	4	0	Negatif

Probabilitas bayes yang dijabarkan adalah probabilitas untuk dokumen ke 103.

4. Hitung probabilitas bersyarat (*likelihood*) dokumen ke 103 pada *class* positif dan negatif.
   Untuk *class* positif:
   P(103|positif)=P(bad=1|positif)xP(delicious=0|positif)xP(good=4|positif) x P(dissapoint=0|positif)

$$P(103|positif) = \frac{0}{6} x \frac{5}{6} x \frac{5}{6} x \frac{0}{6}$$

=0 x 0,833 x 0,833 x 0
=0

P(103|negatif)=P(bad=1|negatif)xP(delicious=0|negatif)xP(good=4|negatif) x P(dissapoint=0|negatif)

$$P(103|negatif) = \frac{4}{5} x \frac{1}{5} x \frac{8}{5} x \frac{3}{5}$$

=0,8 x 0,2 x 1,6 x 0,6
=0,1536

Copyright @ 2015 IlmuKomputer.Com

5. Probabilitas prior dari *class* positif dan negatif dihitung dengan proporsi dokumen pada tiap *class*:

$$\text{P(positif)} = \frac{3}{6} = 0{,}5$$
$$\text{P(negatif)} = \frac{2}{6} = 0{,}333$$

6. Hitung probabilitas posterior dengan memasukkan rumus Bayes dan menghilangkan penyebut P(103):

$$\text{P(positif|103)} = \frac{(0)(0{,}5)}{P(103)} = 0$$
$$\text{P(negatif|103)} = \frac{(0{,}1536)(0{,}333)}{P(103)} = 0{,}0511488$$

Berdasarkan probabilitas diatas, maka dapat disimpulkan bahwa dokumen ke 103 termasuk dalam *class* negatif, karena P(positif|103) lebih kecil dari pada P(negatif|103).

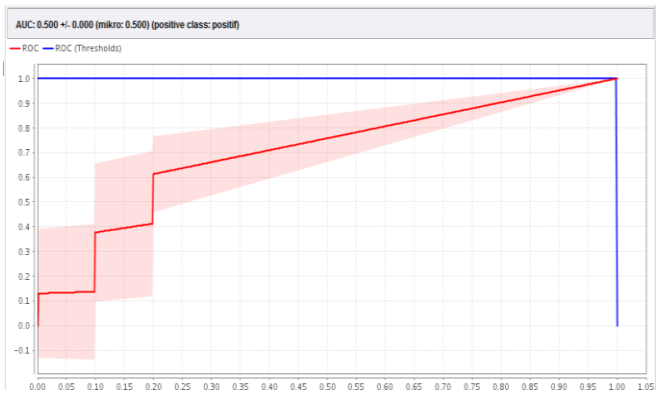
Dari sebanyak 200 data *review* restoran yaitu 100 *review* positif dan 100 *review* negatif, sebanyak 89 data diprediksi sesuai yaitu negatif, dan sebanyak 11 data diprediksi negatif tetapi ternyata positif, 57 data diprediksi sesuai yaitu positif dan 43 data diprediksi positif tetapi ternyata negatif. Hasil yang diperoleh dengan menggunakan algoritma NB adalah nilai *accuracy* = 73.00% seperti pada tabel 3 dan AUC = 0.500, seperti pada Gambar 3.

Tabel 3*ConfusionMatrix* Algoritma NB

***Accuracy*: 73.00% +/- 9.34% (mikro: 73.00%)**			
	truenegative	***truepositive***	***classprecision***
Prediction negative	**89**	**43**	**67.42%**
Prediction positive	**11**	**57**	**83.82%**
classrecall	**89.00%**	**57.00%**	

Nilai accuracy dari *confusionmatrix* tersebut adalah sebagai berikut:

$$accuracy = \frac{(TN + TP)}{(TN + FN + TP + FP)}$$
$$accuracy = \frac{(89 + 57)}{(89 + 11 + 57 + 43)}$$
$$accuracy = \frac{146}{200} = 0.73 = 73.00\%$$



Gambar 3 Grafik Area *UnderCurve* (AUC) Algoritma Naïve Bayes

Penelitian ini menggunakan metode filter yaitu information gain, dimana data yang diolah diberikan bobot dari information gain untuk meningkatkan akurasi algoritmanaïve bayes. Penelitian ini menggunakan operator *selectbyweight* dengan memilih parameter *weightrelation*=top k, dan k=10. Dimana nanti akan dihasilkan 10 atribut teratas. 10 atribut yang terpilih akan ditampilkan bobotnya masing-masing, untuk lebih jelasnya dapat dilihat pada Tabel 4

Tabel 4. Sepuluh Fitur Teratas dan Bobotnya

Atribut	**Bobot**
overpr	0.575
want	0.576
review	0.593
favorit	0.708
amaz	0.713
delici	0.713
good	0.767
definit	0.911
disappoint	1

Bobot diatas adalah bobot yang sudah di-*generate* oleh operator *selectbyweight*. Karena hasilnya masih ada angka 0, maka atribut yang ditampilkan bobotnya dari masing-masing dokumen hanya yang mempunyai bobot 1. Diantara 10 atribut diatas, hanya kata dissapoint yang memiliki bobot 1. Tabel 5 menunjukkan atribut tersebut didalam dokumen dalam bentuk *vector*.

Tabel 5 Atribut Dalam Bentuk *Vector*

No	**Dokumen Ke-**	***Dissapoint***	***Class***
1	12	2	Negatif
2	17	2	Negatif
3	28	2	Negatif
4	96	2	Negatif
5	23	1	Negatif
6	64	1	Positif
7	76	1	Positif
8	149	1	Positif
9	64	1	Positif
10	76	1	Positif

1. Cari nilai entropi sebelum pemisahan:
   y berisi 200 data dengan 100 keputusan positif dan 100 keputusan negatif.
   $Entropy(y) = -Pilog2Pi$
   $Entropy(y) =$ entropi[100,100]
   $$= -\frac{100}{200}\log2\left(\frac{100}{200}\right) - \frac{100}{200}\log2\left(\frac{100}{200}\right) = 1$$

2. Cari nilai entropi setelah pemisahan:
   Untuk atribut *dissapoint*,
   Nilai (positif)=[0,1]
   y=[100,100]
   y0=[97,70]
   y1=[3,30]

   a. *dissapoint* = 0

$$\text{entropy}[97,70] = -\frac{97}{167}\log2\left(\frac{97}{167}\right) - \frac{70}{157}\log2\left(\frac{70}{167}\right) = 0{,}29032$$

b. *dissapoint* = 1

$$\text{entropy}[3,30] = -\frac{3}{33}\log2\left(\frac{3}{33}\right) - \frac{30}{33}\log2\left(\frac{30}{33}\right) = 0{,}1323$$

3. Cari nilai information gain

$$\text{gain}(y, A) = entropi(y) \sum_{0} \in nilai(A)\frac{yc}{y} entropi(yc)$$

$$= \text{entropi}(y) - \frac{167}{200} entropi(y0) - \frac{3}{200} entropi(y1)$$

$$= 1 - \left(\frac{167}{200}\right) 0{,}29032 - \frac{33}{200} 0{,}1323 = 0{,}73335$$

Pengukuran dengan *confusion matrix* di sini akan menampilkan perbandingan dari hasil akurasi model naïve bayes sebelum ditambahkan seleksi fiturinformation gain dan metode adaboost yang bisa dilihat pada Tabel 6 dan setelah ditambahkan seleksi fiturinformation gain dan metode adaboost yang bisa dilihat pada Tabel 7.

Tabel 6 *ConfusionMatrix* Algoritma NaïveBayes Sebelum Penambahan Seleksi Fitur Information Gain dan Metode Adaboost

Accuracy: 70.00% +/- 8.66% (mikro: 70.00%)			
	true negative	***true positif***	***class precision***
Prediction negative	**89**	**49**	**64.49%**
Prediction positive	**11**	**51**	**82.26%**
class recall	**89.00%**	**51.00%**	

Tabel 7 *ConfusionMatrix* Algoritma NaïveBayes Sesudah Penambahan Seleksi Fitur Information Gain dan Metode Adaboost

***Accuracy*:99.50%**			
	true negative	***true positif***	***class precision***
Prediction negative	99	0	100%
Prediction positive	1	100	99.01%
class recall	99.00%	100.00%	

$$akurasi = \frac{100 + 99}{100 + 0 + 1 + 99} = \frac{199}{200} = 0.995 = 99.50\%$$

Hasil pengujian *confusion matrix* di atas diketahui bahwa menggunakan algoritma naïve bayes mempunyai akurasi hanya 70.00% sedangkan algoritma naïve bayes dengan seleksi fiturinformation gain dan metode adaboost memiliki tingkat akurasi yang lebih tinggi yaitu 99.50%. Akurasi naik 29.50% dari yang sebelumnya.

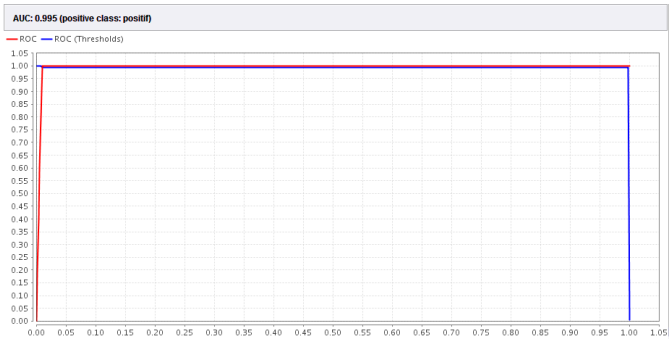
Grafik ROC akan membentuk garis dimana garis tersebut menunjukkan hasil prediksidari model klasifikasi yang digunakan. Apabila garis tersebut berada di atas diagonal grafik maka hasil klasifikasi bernilai baik (*good classification*), sedangkan garis yang berada di bawah diagonal grafik menghasilkan nilai klasifikasi yang buruk (*poor classification*). Garis yang menempel pada sumbu Y menunjukkan grafik tersebut menunjukkan klasifikasi yang baik (Gorunescu, 2011).

Dari grafik ROC didapatkan pula nilai AUC (Area Under the ROC Curve) untuk menganalisa hasil prediksiklasifikasi. Penentuan hasil prediksi klasifikasi dilihat dari batasan nilai AUC sebagai berikut (Gorunescu, 2011):

1. Nilai AUC 0.90-1.00 = *excellent classification*
2. Nilai AUC 0.80-0.90 = *good classification*
3. Nilai AUC 0.70-0.80 = *fair classification*
4. Nilai AUC 0.60-0.70 = *poor classification*
5. Nilai AUC 0.50-0.60 = *failure*

Berikut adalah tampilan kurva ROC yang akan dihitung nilai AUC-nya. Gambar 4 adalah kurva ROC untuk model naïve bayes sebelum menggunakan metode adaboost dan seleksi fitur IG dan gambar 5 adalah kurva ROC untuk model naïve bayes setelah menggunakan metode adaboost dan seleksi fitur information gain.

Gambar 4 Kurva AUC untuk Algoritma NaïveBayes Sebelum Menggunakan Seleksi Fitur Information Gain dan Metode Adaboost



Gambar 5 Kurva AUC Untuk Algoritma NaïveBayes Sesudah Menggunakan Seleksi Fitur Information Gain dan Metode Adaboost

Grafik diatas yang menunjukkan bahwa algoritma naïve bayes hanya memiliki nilai AUC 0.500 yang artinya *failure* (gagal) dibandingkan dengan algoritma naïve bayes yang menggunakan seleksi fitur information gain dan metode adaboost yang memiliki nilai AUC 0.995 yang artinya *excellent classification*.

Pada pengujian ini, menggunakan klasifikasi naïve bayes, algortima *feature selection* information gain dan teknik boosting yaitu metode adaboost. Pengujian dilakukan uji coba dengan melakukan optimalisasi perulangan (*iterations*). Tabel 8 adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai *accuracy* dan AUC.

Copyright @ 2015 IlmuKomputer.Com

Tabel 8 Pengujian Indikator

Iterations	*Accuracy*	AUC
1	80,50%	0,805
2	80,50%	0,850
3	80,50%	0,870
4	80,50%	0,878
5	80,50%	0,882
6	81,50%	0,887
7	80,50%	0,890
8	80,50%	0,890
9	80,50%	0,890
10	80,50%	0,890

Dari semua *iterations* yang di uji, *accuracy* tertinggi adalah pada saat *iterations* = 6, yaitu nilai *accuracy* = 81,50% dan nilai AUC = 0.887.

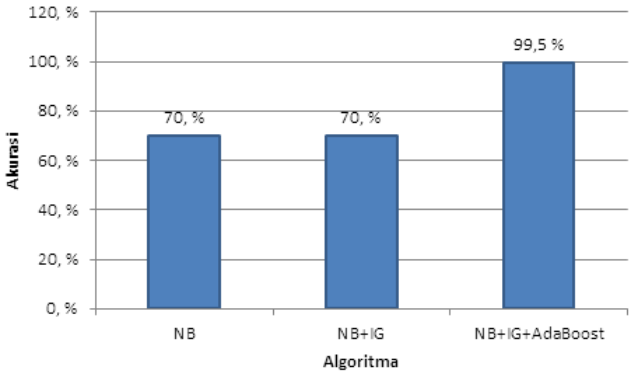
Dengan memiliki model klasifikasi teks pada *review*, pembaca dapat dengan mudah mengidentifikasi mana *review* yang positif maupun yang negatif. Dari data *review* yang sudah ada, dipisahkan menjadi kata-kata, lalu diberikan bobot pada masing-masing kata tersebut. Dapat dilihat kata mana saja yang berhubungan dengan sentimen yang sering muncul dan mempunyai bobot paling tinggi. Dengan demikian dapat diketahui *review* tersebut positif atau negatif.

Dalam penelitian ini, menunjukkan seberapa baik model yang terbentuk. Tanpa menggunakan metode seleksi fitur, algoritma naïve bayes sendiri sudah menghasilkan akurasi sebesar 70.00% dan nilai AUC 0.500. Akurasi tersebut masih kurang akurat, sehingga perlu ditingkatkan lagi menggunakan seleksi fitur yaitu information gain dan teknik *boosting* yaitu metode adaboost. Setelah menggunakan metode adaboost dan information gain, akurasi algoritma naïve bayes meningkat menjadi 99.50% dan nilai AUC 0.995. seperti yang bisa dilihat pada Tabel 9.

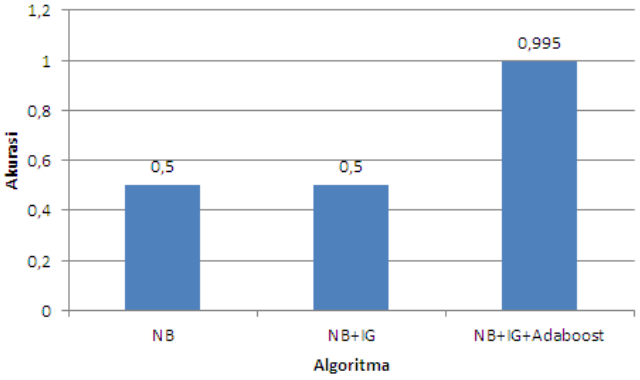
Tabel 9 Perbandingan Model Algoritma Naïve Bayes Sebelum dan Sesudah Menggunakan Seleksi Fitur Information Gain dan Metode Adaboost

	Algoritma Naive Bayes	**Algoritma Naive Bayes Information Gain + AdaBoost**
Sukses prediksi positif	51	100
Sukses prediksi negatif	89	99
Akurasi model	70.00%	99.50%
AUC	0.500	0.995

Berdasarkan hasil evaluasi di atas diketahui bahwa algoritma naïve bayes yang menggunakan seleksi fitur information gain dan metode *boosting* adaboost, mampu meningkatkan tingkat akurasi *review* restoran. Gambar 6 memperlihatkan tingkat akurasi yang meningkat dalam bentuk sebuah grafik. Sedangkan Gambar 7 memperlihatkan nilai AUC.



Gambar 6 Grafik Akurasi Algoritma NaïveBayes Sebelum dan Sesudah Menggunakan Seleksi Fitur Information Gain dan Metode Adaboost



Gambar 7 Grafik Nilai AUC Algoritma NaïveBayes Sebelum dan Sesudah Menggunakan Seleksi Fitur Information Gain dan Metode Adaboost

Penelitian mengenai ulasan *review* terkadang kedapatan perbedaan antara ulasan yang dibuat konsumen secara online dengan editor ulasan demi menarik konsumen untuk lama restoran tersebut (Zhang et al., 2010). Dalam beberapa kasus, ditemukan sentimen campuran, di mana sentimen positif memiliki ungkapan yang sama dengan sentimen negatif. Kasus seperti itu, aspek polaritas domain tidak bisa hanya dianggap sebatas positif atau negatif. Salah satu cara alternatif adalah untuk menetapkan nilai disetiap kemungkinan aspek polaritas yang sama yang diungkapkan dalam review tersebut. (Zhu, Wang, Zhu, Tsou, & Ma, 2011). Penelitian ini menunjukkan bahwa naïve bayes memerlukan sejumlah dukungan untuk meningkatkan akurasi tingkat klasifikasi. Tinggi atau tidaknya tingkat akurasi tergantung oleh jumlah fitur yang ada (Zhang et al., 2011b). Menurut hasil, kinerja naïve bayes meningkat dengan bantuan algortima adaboost dan menghasilkan hasil yang akurat dengan mengurangi kesalahan misklasifikasi dengan meningkatkan iterations (Korada, Kumar, & Deekshitulu, 2012). Metode information gain menunjukkan hasil yang memuaskan dalam filtering sebuah istilah (Moraes, Valiati, & Neto, 2013).

Dari pengolahan data yang sudah dilakukan dengan metode *boosting* yaitu adaboost dan seleksi fitur yaitu information gain, terbukti dapat meningkatkan akurasi algoritma naïve bayes. Data *review* restoran dapat diklasifikasi dengan baik ke dalam bentuk positif dan negatif.

5 KESIMPULAN

Naïve bayes merupakan salah satu pengklasifikasi yang mengklasifikasikan suatu teks, salah satu contoh yakni review restoran. Naïve bayes sangat sederhana dan efisien,

Copyright @ 2015 IlmuKomputer.Com

juga sangat populer digunakan untuk klasifikasi teks dan memiliki performa yang baik pada banyak domain.

Pengolahan data yang dilakukan ada 3 tahap, yakni naïve bayes, naïve bayes dan information gain, dan naïve bayes, information gain, dan adaboost. Dan ternyata, jika hanya naïve bayes saja yang digunakan, akurasi hanya mencapai 70% dan AUC=0,500. Sama halnya jika naïve bayes disertai dengan information gain, akurasi yang dicapaipun hanya 70% dan AUC=0,500, itu membuktikan bahwa information gain tidak mempengaruhi akurasi terhadap naïve bayes. Akan tetapi, jika naïve bayes dan information gain disertai pula dengan adaboost, akurasi meningkat 29,5% menjadi 99,5% dan AUC=0,995.

REFERENCES

Ali, W., Shamsuddin, S. M., & Ismail, A. S. (2012). Intelligent Naïve Bayes-based approaches for Web proxy caching. *Knowledge-Based Systems*, *31*, 162–175.

Bauer, E. (1999). An Empirical Comparison of Voting Classification Algorithms : Bagging , Boosting , and Variants. *Machine Learning Research*, *139*, 105–139.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, *36*, 5432–5435.

Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques*. Berlin.

He, Y., & Zhou, D. (2011). Self-training from labeled features for sentiment analysis. *Information Processing & Management*, *47*, 606–616.

Huang, J., Rogers, S., & Joo, E. (2013). Improving Restaurants. *Information System*, 1–5.

Kang, H., Yoo, S. J., & Han, D. (2012a). Expert Systems with Applications Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems With Applications*, *39*(5), 6000–6010.

Kang, H., Yoo, S. J., & Han, D. (2012b). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, *39*, 6000–6010.

Korada, N. K., Kumar, N. S. P., & Deekshitulu, Y. V. N. H. (2012). Implementation of NBian Classifier and Ada-Boost Algorithm Using Maize Expert System. *International Journal of Information Sciences and Techniques*, *2*, 63–75.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Toronto: Morgan and Claypool

Muthia, D. A. (2013). Analisis Sentimen Pada Review Buku Menggunakan Algoritma. *Sistem InformasiSistem Informasi*, 1–9.

Reyes, A., & Rosso, P. (2012). Making objective decisions from subjective data : Detecting irony in customer reviews. *Decision Support Systems*, *53*, 754–760.

Sharma, A., & Dey, S. (2012). A Comparative Study of Feature Selection and Machine Learing Techniques for Sentiment Analysis. *Information Search and Retrieval*, 1–7

Wang, R. (2012). Adaboost for Feature Selection, Classification and Its Relation with SVM, A Review. *Physics Procedia*, *25*, 800–807.

Wayan, N. (2013). Naïve Bayes Classifier Dan Support Vector Machines Untuk Sentiment Analysis. *Sistem Informasi*, 2–4.

Wu, X. (2009). The Top Ten Algorithms in Data Mining. Boca Raton: Taylor and Francis

Zhang, & Gao, F. (2011). An Improvement to NB for Text Classification. *Procedia Engineering*, *15*, 2160–2164.

Zhang, Ye, Q., Zhang, Z., & Li, Y. (2011). Sentiment classification of Internet restaurant reviews written in Cantonese. *Expert Systems with Applications*, *38*, 7674–7682.

Zhang, Z., Ye, Q., Law, R., & Li, Y. (2010). The impact of e-word-of-mouth on the online popularity of restaurants: A comparison of consumer reviews and editor reviews. *International Journal of Hospitality Management*, *29*, 694–700.

BIOGRAFI PENULIS

Lila Dini Utami. Lahir pada tanggal 28 Juni 1988 di Jakarta. Memperoleh gelar Sarjana Komputer (S.Kom) dari STMIK Nusa Mandiri Jakarta (Jurusan Sistem Informasi) pada tahun 2011. Serta memperoleh gelar M.Kom dari Pascasarjana STMIK Nusa Mandiri pada tahun 2014 (Jurusan Ilmu Komputer).

Romi Satria Wahono. Memperoleh Gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang sofwater engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dia Nuswantoro. Merupakan pendiri dan CEO PT. Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Integrasi Discrete Wavelet Transform dan Singular Value Decomposition pada *Watermark*ing Citra untuk Perlindungan Hak Cipta

Jaya Chandra
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
Email: jaya_1184@yahoo.com

Romi Satria Wahono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Email: romi@brainmatics.com

Abstrak: Tren masalah *watermarking* pada sekarang ini adalah bagaimana mengoptimalkan *trade-off* antara *imperceptibility* (visibilitas) citra ter-*watermark* terhadap pengaruh distorsi dan *robustness* terhadap penyisipan *watermark*. Masalah menggunakan kekuatan penyisipan berdasarkan *Single Scaling Factor (SSF*) atau *Multiple Scaling Factor (MSF)* juga ditemukan. Penelitian ini mengusulkan metode penyisipan *watermark* untuk perlindungan hak cipta pada citra dan algoritma ekstraksi citra ter-*watermark* yang dioptimalkan dengan penggabungan *Discrete Wavelet Transform* (*DWT)* dan *Singular Value Decomposition (SVD)*. Nilai-nilai *singular* dari *LL3* koefisien *sub-band* dari citra *host* dimodifikasi menggunakan nilai tunggal citra *watermark* biner menggunakan *MSFs*. Kontribusi utama dari skema yang diusulkan adalah aplikasi *DWT-SVD* untuk mengidentifikasi beberapa faktor skala yang optimal. Hasil penelitian menunjukkan bahwa skema yang diusulkan menghasilkan nilai *Peak Signal to Noise Ratio (PSNR)* yang tinggi, yang menunjukkan bahwa kualitas visual gambar yang baik pada masalah citra *watermarking* telah mengoptimalkan *trade-off*. *Trade-off* antara *imperceptibility* (visibilitas) citra ter-*watermark* terhadap pengaruh distorsi dan *robustness* citra ter-*watermark* terhadap operasi pengolahan citra. Nilai *PSNR* yang didapat pada citra yang diujikan: *baboon*=53,184; *boat*=53,328; *cameraman*=53,700; *lena*=53,668; *man*=53,328; dan *pepper* sebesar 52,662. Delapan perlakuan khusus pada hasil citra ter-*watermark* diujikan dan diekstraksi kembali yaitu *JPEG 5%, Noise 5%, Gaussian filter 3x3, Sharpening, Histogram Equalization, Scaling 512-256, Gray Quantitation 1bit*, dan *Cropping 1/8*. Hasil dari perlakuan khusus kemudian diukur nilai *Normalized Cross-Correlation (NC)* yang menghasilkan rata-rata semua citra diperoleh sebesar 0,999 dari satu. Hasil penelitian dari metode yang diusulkan lebih unggul nilai PSNR dan NC dari penelitian sebelumnya. Jadi dapat disimpulkan bahwa penerapan dengan metode *DWT-SVD* ini mampu menghasilkan citra yang *robust* namun memiliki tingkat *imperceptibilit*y yang cukup tinggi.

Keywords: *Image Watermarking, Discrete Wavelet Transform, Singular Value Decomposition, Normalized Cross Correlation, Robustness*.

1 PENDAHULUAN

Dengan meningkatnya pertumbuhan penggunaan internet, citra atau gambar digital dapat menyebar ke seluruh dunia hanya dengan satu klik pada tombol mouse. Hal ini menyebabkan kerentanan citra digital (Qiao & Nahrstedt, 1998) dan menghasilkan pertanyaan logis pada hak ciptanya (Mohammad, Alhaj, & Shaltaf, 2008), lalu ada berbagai teknik keamanan informasi yang dapat menangani masalah hak cipta.

Watermarking citra digital adalah proses penyisipan informasi digital atau penanaman kode tertentu seperti gambar logo atau gambar rahasia (Run, Horng, Lai, Kao, & Chen, 2012) ke dalam citra sedemikian rupa sehingga citra yang ter-*watermark* atau citra yang sudah disisipkan suatu kode tertentu (Mohammad et al., 2008), dapat dideteksi atau diekstrak serta tanpa menurunkan kualitas citra tersebut.

Dalam *image watermarking*, terdapat dua kriteria utama yang wajib dipenuhi. Ini adalah (1) *imperceptibility* citra yang tertanam *watermark* dan (2) *robustness* atau ketahanan skema penyisipan *watermark*. Berdasarkan kriteria tersebut, teknik *watermarking* dapat secara luas diklasifikasikan menjadi tiga kelompok: *robust* (kuat), *fragile* (rapuh) dan *semi-fragile* (semi-rapuh) (Cox, Kilian, Leighton, & Shamoon, 1997).

Pada penelitian sebelumnya (J. C. Liu & Chen, 2001; Nikolaidis & Nikolaidis, 1998) *fragile watermarking* dilakukan pada citra domain spasial, *watermark* secara langsung dimasukkan ke dalam permukaan citra dengan mengubah nilai piksel. Hal ini menyebabkan penerapan yang mudah dan biaya operasi yang rendah, akan tetapi umumnya tidak kuat dalam menghadapi serangan atau modifikasi yang sah. Sebaliknya metode domain frekuensi mengubah gambar kedalam domain frekuensi dan kemudian memodifikasi koefisien frekuensi untuk menanamkan *watermark* sehingga kuat terhadap serangan.

Ada banyak tehnik dalam mengubah domain *watermarking* seperti: *Discrete Cosine Transform (DCT)* (Briassouli & Strintzis, 2004; Hernández, Amado, & Pérez-González, 2000; Patra, Phua, & Bornand, 2010), *Singular Value Decomposition (SVD)* (Ali, Ahn, & Pant, 2014; Aslantas, 2009; Chang, Tsai, & Lin, 2005; Dogan, Tuncer, Avci, & Gulten, 2011; Jia, 2014; Lai, 2011b; Run et al., 2012) dan *Discrete Wavelete Transform (DWT)* (Ali & Ahn, 2014; Lai & Tsai, 2010; Olkkonen, 2011; Van Fleet, 2011; M.-S. Wang & Chen, 2009; Xianghong, Lu, Lianjie, & Yamei, 2004)

Analisis frekuensi domain digunakan untuk menentukan lokasi yang mungkin untuk penyisipan koefisien *watermark* dan untuk memastikan ketahanan atau *robustness* yang lebih kuat dalam algoritma penyisipan. Mata manusia lebih sensitif terhadap frekuensi rendah dan menengah pada pita koefisien, oleh karena itu, teknik transform domain bekerja dengan baik jika *watermark* tertanam dalam koefisien frekuensi rendah dari citra (Cox et al., 1997; Nikolaidis & Nikolaidis, 1998). Selain itu, pada penelitian sebelumnya diantara metode transformasi domain yang ada (Ali & Ahn, 2014; Lai & Tsai, 2010; M.-S.

Copyright @ 2015 IlmuKomputer.Com

Wang & Chen, 2009), DWT lebih baik dalam mencapai *robust watermarking* dan *imperceptibility* yang mengarah pada kualitas hasil citra yang baik.

Selama beberapa tahun terakhir, *SVD* digunakan sebagai metode baru untuk *watermarking* (Ali et al., 2014; Jia, 2014; Lai, 2011b; Run et al., 2012), membawa cara pandang yang baru dari suatu citra dan informasi struktural yang sangat penting untuk prediksi kualitas citra. Modifikasi dalam vektor tunggal berhubungan dengan nilai tunggal, dimana secara dasar merupakan perwakilan dalam pencahayaan citra tersebut.

Algoritma evolusioner seperti *Particle Swarm Optimization (PSO)* (Fındık, Babaoğlu, & Ülker, 2010; Run et al., 2012; Y.-R. Wang, Lin, & Yang, 2011), *Genetic Algorithm (GA)* (Kumsawat, Attakitmongcol, & Srikaew, 2005; Shieh, Huang, Wang, & Pan, 2004), *Bacterial foraging* (Huang, Chen, & Abraham, 2010) telah banyak digunakan untuk *watermarking* citra. Kebanyakan teknik evolusi yang ada digunakan untuk mengidentifikasi koefisien citra dalam mengubah domain untuk menanamkan *watermark* (Huang et al., 2010; Shieh et al., 2004; Y.-R. Wang et al., 2011).

Seperti disebutkan di atas, masalah menemukan nilai optimal *Multiple Scaling Factors (MSFs)* dapat diselesaikan dengan menggabungkan teknik evolusi dengan teknik transfomasi (Ishtiaq, Sikandar, Jaffar, & Khan, 2010; Lai, 2011a; Loukhaoukha, 2011) telah menggunakan algoritma *tiny genetic* (*Tiny-GA*) dengan *SVD* untuk menemukan nilai *MSFs*.

Dalam penelitian ini kami mengusulkan metode *Discrete Wavelet Transform* yang dipadu dengan *Singular Value Decomposition* untuk melakukan optimalisasi nilai *MSFs* dari *Discrete Wavelet Transform* dengan *Singular Value Decomposition* berkaitan dengan rentannya citra ter-*watermark* terhadap distorsi sehingga menyulitkan proses ekstraksi citra dan pada akhirnya dapat mengurangi kualitas *watermark* citra hasil dari ekstraksi tersebut.

Paper ini disusun sebagai berikut: pada bagian 2 paper-paper terkait dijelaskan. Pada bagian 3, metode yang diusulkan disajikan. Hasil percobaan perbandingan antara metode yang diusulkan dengan metode lainnya disajikan pada bagian 4. Akhirnya, kesimpulan dari penelitian kami disajikan pada bagian terakhir.

2 PENELITIAN TERKAIT

SVD digunakan untuk *watermarking* (F. L. F. Liu & Liu, 2008). Dalam algoritma ini, mereka menghitung nilai-nilai singular dari citra *host* dan kemudian memodifikasinya dengan menambahkan *watermark*. Mereka juga menerapkan transformasi *SVD* pada matriks yang dihasilkan untuk menemukan nilai-nilai tunggal yang dimodifikasi. Nilai-nilai singular digabungkan dengan *watermark* untuk mendapatkan gambar *watermark*. Sedangkan untuk ekstraksi *watermark*, digunakan proses terbalik (*reversible*). *Watermarking* berbasis *SVD* telah diusulkan oleh berbagai peneliti (Ali et al., 2014; Aslantas, 2009; F. L. F. Liu & Liu, 2008; Mohammad et al., 2008; Patra et al., 2010) menggunakan nilai konstan skala faktor tunggal *(SSF)*.

Penelitian konvensional pada *watermark*ing citra terbatas pada penggunaan formulasi matematika standar seperti: *DCT, DWT, SVD*, dan varian hibrid lainnya seperti: *DCT-DWT, DCT-SVD*, dan *DWT-SVD*. *Watermark* disisipkan ke dalam citra *host* dengan menggunakan persamaan matematika yang secara tradisional memakai kekuatan penyisipan berdasarkan *single scaling value*. Kekuatan penyisipan atau faktor skala adalah jumlah modifikasi yang disebabkan oleh *watermark* di media aslinya. Dalam *watermarking* citra digital, umumnya satu atau nilai konstan faktor skala digunakan untuk menanamkan *watermark* dalam seluruh *host* citra.

PSO diterapkan (Ishtiaq et al., 2010) untuk menemukan *MSFs* dalam domain *DCT*. Mereka menggunakan *PSNR* sebagai fungsi tujuan untuk mengevaluasi setiap partikel. Kelemahan utama dari algoritma ini adalah bahwa ia hanya berfokus pada kualitas visual gambar *watermark* tanpa memperhitungkan faktor *robustness* (ketahanan) *Multi Objective Ant Colony Optimization (MOACO)* pada domain *LWT-SVD* (Loukhaoukha, 2013) untuk menemukan nilai-nilai *MSF*s. Fungsi tujuan mereka adalah formulasi *exponential weighted* sebagai berikut:

linier dapat ditulis:

$$F_{obj}(x) = \sum_{i=1}^{T+2}(e^{p.w} - 1)e^{p(F(X)-F0)} \quad (1)$$

dimana p, w dan F_0 adalah konstanta positif, F (X) adalah vektor nilai-nilai obyektif dan T adalah jumlah yang dipilih operasi pengolahan citra. *MSFs* (Ishtiaq et al., 2010; Loukhaoukha, 2013) skema *watermarking* berbasis *MOACO* melebihi skema *SSF watermark*ing yang berbeda dalam hal

Penelitian yang dilakukan oleh Xianghong et al (2004) menggunakan algoritma penyisipan *watermark* berdasarkan karakteristik *DWT* dan *VT*, dan juga menggunakan *properti Human Visual System (HVS)*. Xianghong bereksperimen mengunakan enam citra abu-abu (256 x 256) yaitu: *Baboon, Boat, Lena, Cameraman, Man, Peppers* dan 1 citra *binary* (32 x 32) sebagai citra yang disisipkan.

Sedangkan Loukhaoukha (2011) menggunakan algoritma penyisipan *watermark* berdasarkan *Liftlet Wavelet Transform (LWT)* dan dengan *Singular Value Decomposition (SVD)* digunakan untuk mencari *Single Scaling Factor (SSF*) dari biner *watermark* yang ditanam dalam dalam *sub-band* untuk mencapai ketahanan (*robustness*) yang optimal tanpa kehilangan transparansi *watermark* (*imperceptibility*).

Hasil percobaan Loukhaoukha menunjukkan bahwa untuk mencapai tingkat *robustness* yang tertinggi tanpa mengurangi *imperceptibility* diperlukan *Multi Scaling Factor (MSF)*. *LWT+SVD* dengan *MSF* pada skema *watermark*ing melebihi *SSF* dalam hal *Imperceptibility* dan *Robustness*.

Ishtiaq et al (2010) menggunakan algoritma penyisipan *watermark* berdasarkan karakteristik *Discrete Cosine Transform (DCT)* dan *Particle Swarm Optimization* (*PSO). PSO* digunakan untuk menentukan skala faktor *watermark* yang optimal, dalam *PSO* setiap partisi mewakili satu solusi lengkap, dalam hal ini partisi yang dimaksud adalah koefisien dari tiap-tiap *watermark* yang terpilih.

Melihat dari hasil penelitian (Ishtiaq et al., 2010), disimpulkan bahwa hasil percobaan Ishtiaq menunjukkan model penelitian dengan *Particle Swarm Optimization* digunakan untuk mengoptimalkan kekuatan *watermark* dalam domain *DCT*. Metode yang diusulkan menunjukkan hasil yang lebih baik terhadap serangan yang berbeda, seperti pemberian *noise, low-pass filter, high-pass filter, filter median* dan *cropping*.

Berdasarkan uraian diatas terdapat perbedaan metode *watermarking* yang digunakan, namun data citra yang digunakan adalah sama yaitu data standar enam citra yaitu: *'Baboon', 'Boat', 'Lena', 'Cameraman', 'Man', 'Peppers'*, dan 1 citra sebagai logo atau *watermark* yang akan digunakan dengan ukuran 256x256 pixel. Masalah penelitian yang dihadapi adalah sulitnya menentukan nilai parameter *MSF* yang optimal agar citra ter-*watermark* dapat mempunyai ketahanan *(robustness*) terhadap serangan dan sekaligus dapat

Copyright @ 2015 IlmuKomputer.Com

sama atau mirip dengan aslinya atau biasa disebut memiliki *imperceptibility* yang tinggi.

Sebagai alat evaluasi yang digunakan pada beberapa penelitian diatas adalah menggunakan *Peak Signal to Noise Ratio (PSNR)* dan *Normalized Cross-Correlation (NC)*.

Peak Signal to Noise Ratio adalah pengukur yang banyak digunakan untuk mengukur tingkat kemiripan antara citra asli dengan citra hasil konstruksi (Cheddad, Condell, Curran, & Mc Kevitt, 2010). *PSNR* digunakan untuk mengukur kualitas gambar (M.-S. Wang & Chen, 2009). Persamaan *PSNR* dinyatakan dalam satuan dB:

$$PSNR = 10log_{10}\left(\frac{l^2_{max}}{MSE}\right) \quad (2)$$

Dimana $l^2_{\max}$ adalah nilai pixel maksimum yang mungkin dari image *l*, dan *MSE* adalah *Mean Square Error* yang didefinisikan sebagai:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(S_{xy} - l_{xy})^2 \quad (3)$$

Dimana x dan y adalah koordinat citra, M dan N adalah dimensi dari citra, S_{xy} adalah *watermark*, dan l_{xy} adalah citra *host*.

PSNR sering dinyatakan pada skala logaritmik dalam desibel (dB). Jika nilai *PSNR* jatuh dibawah 30dB menunjukkan kualitas citra yang cukup rendah (Cheddad et al., 2010), jika diatas atau sama dengan 40db maka menunjukkan kualitas citra yang tinggi.

NC (*Normalized Cross-Correlation*) merupakan salah satu alat ukur yang digunakan untuk menguji ketahanan (*robustness)* pada suatu citra. Citra ter-*watermark* diuji dengan sebelumnya melakukan beberapa serangan *malicious* dan *non malicious* untuk proteksi citra digital (Qiao & Nahrstedt, 1998; Run et al., 2012; Tan & Liu, 2008; M.-S. Wang & Chen, 2007). Berikut adalah persamaan matematikanya:

$$NC(W, W') = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[W(i,j).W'(i,j)]}{\sum_{i=1}^{m}\sum_{j=1}^{n}(W(i,j))^2} \quad (4)$$

Dimana *W* adalah *watermark* asli, dan W^1 adalah *watermark* hasil ekstraksi citra ter-*watermark*.

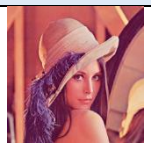
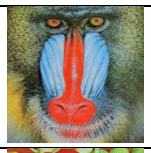
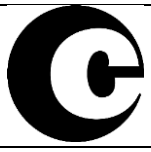
Pada penelitian ini kami akan menerapkan metode *Discrete Wavelet Transform* yang dipadu dengan *Singular Value Decomposition* untuk melakukan optimalisasi nilai *MSFs* dari *Discrete Wavelet Transform* dengan *Singular Value Decomposition* berkaitan dengan rentannya citra ter-*watermark* terhadap distorsi sehingga menyulitkan proses ekstraksi citra dan pada akhirnya dapat mengurangi kualitas *watermark* citra hasil dari ekstraksi tersebut. Kemudian untuk membuktikan kehandalan metode yang telah diusulkan, dengan metode-metode *watermarking* lainnya untuk perlindungan hak cipta citra. Penelitian ini membandingkan hasil dengan penelitian terkait (Loukhaoukha, 2011) dan (Ishtiaq et al., 2010) pengukurannya dalam *Normalized Cross Correlation (NC)* dan *Peak Signal Noise Ratio (PSNR)*. Berdasarkan percobaan, dapat disimpulkan *SVD* telah terbukti menjadi metode yang sukses dan *sufficient* untuk *penyisipan* dan ekstraksi *watermark* pada citra digital dalam mencapai citra ter-*watermark* yang robust dan mempunyai tingkat *imperceptibility* yang tinggi.

3 METODE YANG DIUSULKAN

Pada penelitian ini, data yang digunakan adalah *dataset* citra standar, citra yang digunakan adalah Standar enam Citra: *Lena*, *Baboon*, *Cameraman*, *Peppers*, *Boat*, dan *Man*. Data citra yang digunakan dalam penelitian ini dapat diperoleh melalui situs: *http://www.imageprocessingplace.com/downloads_V3/root_downloads/image_databases/standard_test_images.zip*.

Seperti pada Tabel 1 data citra yang berupa gambar ini memiliki ekstensi *.TIFF.yang dapat dilihat pada Tabel 1.

Tabel 1. *Dataset* Standar Enam Citra

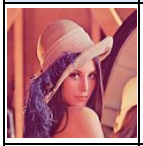
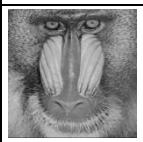
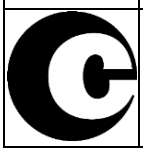
Data citra	Ekstension	Format	Ukuran
	*.TIFF	RGB	512 x 512 pixel
	*.TIFF	RGB	512 x 512 pixel
	*.TIFF	RGB	512 x 512 pixel
	*.TIFF	Grey-scale	512 x 512 pixel
	*.TIFF	Grey-scale	512 x 512 pixel
	*.TIFF	Grey-scale	512 x 512 pixel
	*.PNG	Grey-scale	721 x 721 pixel

Pada Tabel 1 terlihat enam Data Citra Standar dengan satu citra logo yang akan dijadikan *watermark*. Ekstension Citra Standar awal adalah .TIFF dengan *format Red Grey Black (RGB)* yaitu citra *Lena*, *Baboon*, *Peppers* dan ada beberapa dengan format *Grey Scale* yaitu citra *Cameraman*, *Man*, dan *Boat*, dengan ukuran seluruh citra standar sebesar 512 x 512 *pixel* dan satu logo *watermark* dengan ukuran 721 x 721 pixel.

Pada tahap awal pengolahan data (*preprocessing*), kami melakukan konversi untuk citra yang berbentuk RGB diubah kedalam bentuk *grey-scale* dengan ekstension *.jpg, dengan ukuran 256x256. Logo yang akan disisipkan juga terlebih dahulu konversikan menjadi 256x256, dengan ekstension *.jpg. hasil konversi citra dari format RGB ke dalam *grey-scale*, konversi ukuran 256x256, dan ekstension *.jpg dapat dilihat pada Tabel 2.

Copyright @ 2015 IlmuKomputer.Com

Tabel 2. Data Standar 6 Citra Setelah Dilakukan *Preprocessing*

Input awal	*Output*			
Data citra	Data citra	Ekstensi	Format	Ukuran
		*.jpg	*Grey scale*	256x256 pixel
		*.jpg	*Grey scale*	256x256 pixel
		*.jpg	*Grey scale*	256x256 pixel
		*.jpg	*Grey scale*	256x256 pixel
		*.jpg	*Grey scale*	256x256 pixel
		*.jpg	*Grey scale*	256x256 pixel
		*.jpg	*Grey scale*	256x256 pixel

Selanjutnya kami mengusulkan metode yang disebut DWT+SVD pada fitur penyisipan *watermark* yang mana SVD digunakan untuk mengoptimasi parameter MSF pada DWT untuk mendapatkan hasil citra ter-*watermark* yang *robust* dan *impercept*.

Multi Scaling Factor (MSF) merupakan parameter yang mengontrol *trade-off* antara *imperceptibility* dan *robustness*. Dalam penelitian ini *MSF* ditentukan dengan menggunakan *SVD*. Berikut adalah algoritma dalam penentuan *MSF* pada penyisipan *watermark*:

1. Input *host* dan *watermark*
2. Cari *LL, HL, LH, HH* dengan rumus *DWT2(host)*
3. Cari *w_LL, w_HL, w_LH, w_HH* dengan rumus *DWT2(watermark)*
4. Cari *U,S,V* dengan rumus *SVD(HH)*
5. Cari *Uw,Sw,Vw* dengan rumus *SVD(w_HH)*
6. Cari nilai *msf* dengan rumus *max(sum(Vw*Uw)/(Vw*Uw))/10*.

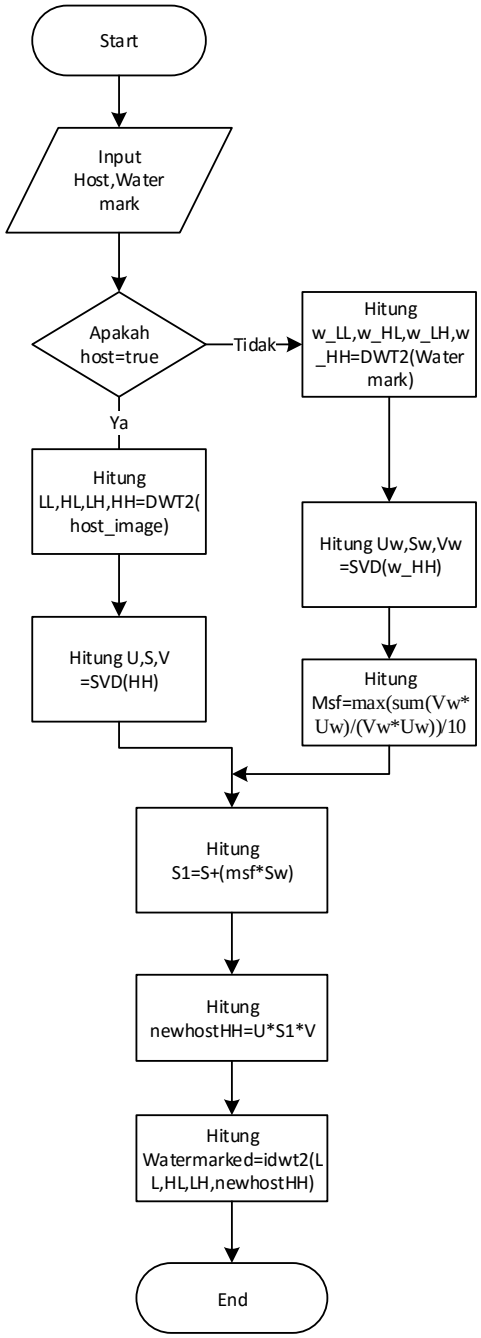
Pada Gambar 1. Terlihat proses penyisipan pada *watermark*ing citra. Tahapan proses penyisipan *watermark* dari metode *DWT-SVD* adalah sebagai berikut:

1. Input *host* dan *watermark*
2. Cari *LL, HL, LH, HH* dengan rumus *DWT2(host)*
3. Cari *w_LL, w_HL, w_LH, w_HH* dengan rumus *DWT2(watermark)*
4. Cari *U,S,V* dengan rumus *SVD(HH)*
5. Cari *Uw,Sw,Vw* dengan rumus *SVD(w_HH)*
6. Cari nilai *msf* dengan rumus *max(sum(Vw*Uw)/(Vw*Uw))/10*
7. Tentukan dan cari koefisien *S1* dengan rumus *S+(msf*Sw)*
8. Cari *newhostHH* dengan rumus *U*S1*V*
9. Kemudian cari citra ter-*watermark* dengan rumus *IDWT2(LL, HL, LH, newhostHH)*

SVD adalah teknik numerik digunakan untuk diagonal matriks dalam analisis numerik dan merupakan sebuah algoritma yang dikembangkan untuk berbagai aplikasi. Dekomposisi nilai tunggal atau *Singular Value Decomposition* merupakan turunan dari teori aljabar linier (Tan & Liu, 2008). *SVD* dapat direpresentasikan secara matematis sebagai berikut:

$$A = U\,S\,V^T \tag{5}$$

Dimana *U* dan *V* adalah matriks ortogonal, dimana kolom *U* adalah vektor tunggal kiri dan kolom *V* adalah vektor tunggal kanan dari matriks persegi *A*.



Gambar 1. Metode Penelitian Penyisipan yang Diusulkan

Proses penyisipan logo *watermark* pada citra *host* dimulai dengan input citra yang akan menjadi inang (*host*), lalu dilanjutkan dengan memasukkan logo *watermark* yang merupakan citra logo yang akan ditanam atau disisipkan pada citra *host*. Kemudian dilakukan proses pemecahan koefisien citra *host* dengan discrete wavelet transform didapat empat sub-band. Setiap tingkat dekomposisi dari *(DWT)* memisahkan gambar menjadi empat *sub-band* yaitu komponen pendekatan resolusi yang lebih rendah *(LL)* dan tiga lainnya sesuai dengan horizontal *(HL)*, vertikal *(LH)* dan diagonal *(HH)* komponen rinci.

Proses selanjutnya pada penyisipan *watermark* adalah pemecahan koefisien citra *watermark* yang akan menjadi logo yang ditanam dalam citra *host* yaitu dengan menerapkan *Discrete Wavelet Transform (DWT)* pada citra logo yang akan menjadi *watermark*. Kemudian dilakukan proses pencarian nilai-nilai *singular* dengan *Singular Value Decomposition* untuk menemukan signifikansinya dalam pengolahan citra sebagai citra digital dapat berupa matriks entri skala negatif atau positif pada citra *host*. Pencarian nilai *U* dan *V* pada citra *host* adalah matriks ortogonal, dimana kolom *U* adalah vektor tunggal kiri citra *host* dan kolom *V* adalah vektor tunggal kanan dari matriks citra *host*. *S* adalah matriks citra *host* diagonal dari *singular value* dalam urutan menurun.

Kemudian dilakukan proses pencarian nilai-nilai *singular* dengan *Singular Value Decomposition* pada citra logo. Penentuan nilai *U*w dan *V*w pada citra logo adalah matriks ortogonal citra logo, dimana kolom *Uw* adalah vektor tunggal kiri citra logo dan kolom *Vw* adalah vektor tunggal kanan dari matriks citra logo. *S*w adalah matriks citra logo diagonal dari *singular value* dalam urutan menurun.

Setelah didapat nilai vektor tunggal kiri citra logo (*Uw*) dan nilai vektor tunggal kanan citra logo (*Vw*). Ditentukan nilai parameter skala faktor yaitu dengan mengalikan nilai vektor tunggal kiri dan kanan, kemudian dijumlahkan dan dicari nilai maksimumnya, selanjutnya dibagi dengan hasil kali nilai vektor tunggal kiri dan kanan citra logo, kemudian dibagi sepuluh. Didapat nilai skala faktor yang akan digunakan selanjutnya pada penentuan koefisien matriks *S1*.

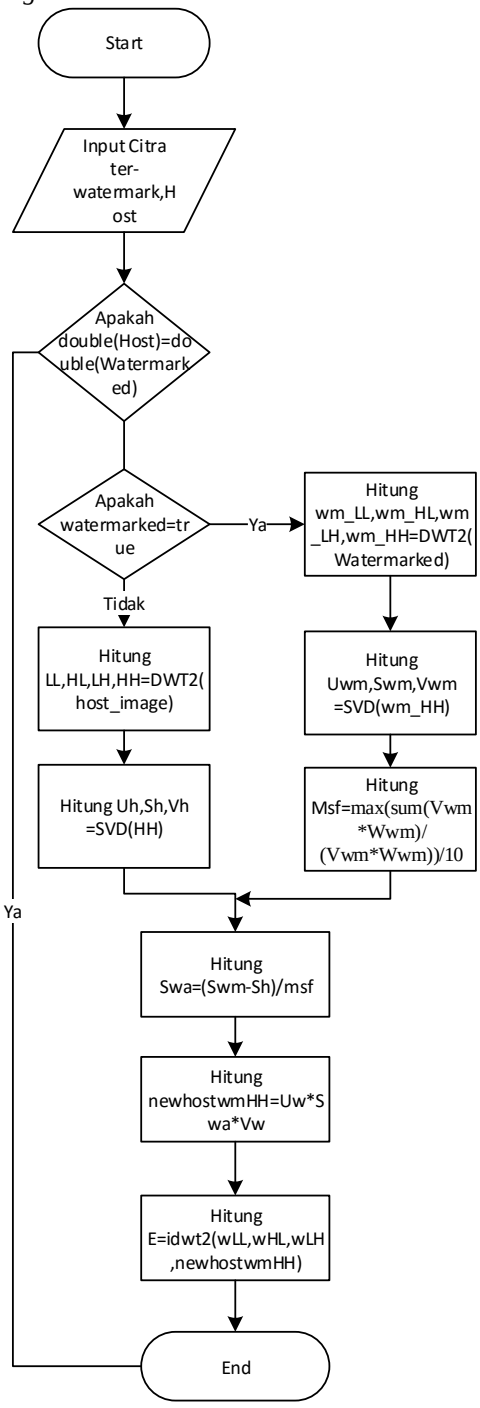
Selanjutnya matriks *S1* digunakan untuk penentuan koefisien baru untuk citra ter-*watermark* (*newhostHH*). Proses terakhir adalah dengan menerapkan *inverse Discrete Wavelet Transform* pada 4 *sub-band LL, HL, LH*, dan *sub-band* baru citra ter-*watermark* (*newhostHH*).

Setelah tahapan penyisipan *watermark* terdapat tahap ekstraksi *watermark*, tahap ini berfungsi untuk mengeluarkan *watermark* dari citra ter-*watermark*. Hasil dari proses ini adalah citra atau logo yang tersimpan atau tersembunyi dalam suatu citra. Metode yang digunakan untuk proses ekstraksi dengan metode *DWT-SVD*.

Terdapat tahapan proses ekstraksi citra ter-*watermark* dengan metode *DWT-SVD* adalah sebagai berikut:

1. *Input host, watermark* dan *watermarked*
2. Tentukan apakah citra ter-*watermark* terdapat *watermark* didalamnya dengan rumus *if double(host)=double(watermarked)* jika Ya langsung *end*, jika Tidak lanjut ke tahap tiga
3. Cari *LL, HL, LH, HH* dengan rumus *DWT2(host)*
4. Cari *wm_LL, wm_HL, wm_LH, wm_HH* dengan rumus *DWT2(watermarked)*
5. Cari *Uh,Sh,Vh* dengan rumus *SVD(HH)*
6. Cari *Uw,Sw,Vw* dengan rumus *SVD(w_HH)*
7. Cari *Uwm,Swm,Vwm* dengan rumus *SVD(wm_HH)*
8. Cari nilai *msf* dengan rumus *max(sum(Vwm*Wwm)/(Vwm*Wwm))/10*
9. Tentukan dan cari nilai koefisien *Swa* dengan rumus *(Swm-Sh/msf)*
10. Cari *newhostwmHH* dengan rumus *Uw*Swa*Vw*
11. Kemudian cari citra *watermark* terekstrak dengan rumus *IDWT2(wLL, wHL, wLH, newhostwmHH)*

Pada Gambar 2. Terlihat proses ekstraksi pada *watermarking* citra



Gambar 2. Metode Penelitian Ekstraksi yang Diusulkan

Selain proses penyisipan dan ekstraksi normal, untuk menguji ketahanan citra ter-*watermark* hasil metode *DWT-SVD*, dilakukan pekerjaan modifikasi beberapa serangan pada citra ter-*watermark* diantaranya: JPEG 5%, *Noise 5%*, *gausian filter*, *sharpening*, *histogram equalization*, *scaling*, dan *gray-scale quantization satu bit.*

Copyright @ 2015 IlmuKomputer.Com

4 HASIL EKSPERIMEN

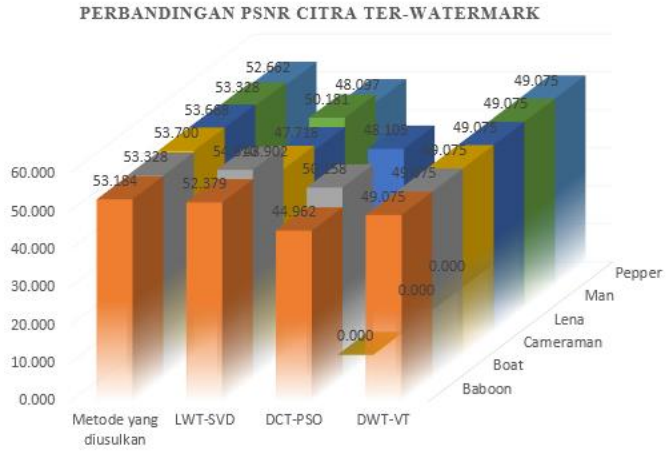
Eksperimen dilakukan menggunakan komputer personal Intel Core i5, 4GB RAM, 320GB HDD, sistem operasi Windows 7, dan Matlab R2014b.

Penelitian ini dilakukan dalam tiga tahapan. Tahap pertama setelah mendapatkan hasil citra ter-*watermark* metode *DWT+SVD*, kami membandingkan hasil eksperimen antara citra asli dengan citra ter-*watermark*. Kemudian kami melakukan pengukuran PSNR untuk menguji kemiripan antara citra asli dengan citra ter-*watermark*.

Tahap kedua, untuk mengidentifikasi bahwa citra ter-*watermark* hasil dari penerapan *watermarking* citra metode *DWT+SVD* dapat diekstrak, kami melakukan ekstraksi citra ter-*watermark*. Kemudian kami mengukurnya, pengukuran dilakukan antara *watermark* asli dengan *watermark* hasil dari ekstraksi dengan *PSNR* untuk menguji kemiripan antara citra logo asli dengan citra logo hasil ekstraksi *watermark*.

Tahap ketiga untuk mengetahui kekuatan (*robust*) model *DWT+SVD* pada *watermarking* citra, kami memberikan beberapa serangan pada citra yang tertanam *watermark*, serangan atau pemberian distorsi berupa: *JPEG 5%*, *Noise 5%*, *Gaussian filter 3x3*, *Sharpening*, *Histogram Equalization*, *Scaling 512-256*, *Gray Quantitation 1bit*, dan *Cropping 1/8*.

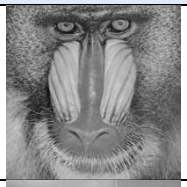
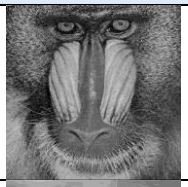
Hasil penerapan metode *DWT+SVD* pada penelitian ini pada citra yang sudah diberi *watermark* (citra ter-*watermark*) dibandingkan dengan metode lainnya. Data citra yang digunakan untuk penelitian dan pengujian adalah sama dengan yang digunakan beberapa peneliti lain dalam bidang yang sama yaitu *watermarking*. Pada penelitian ini membandingkan hasil dari metode yang diajukan *DWT-SVD* dengan metode *LWT-SVD* (Loukhaoukha, 2011), metode *DCT-PSO* (Ishtiaq et al., 2010), dan metode *DWT-VT* (Xianghong et al., 2004).



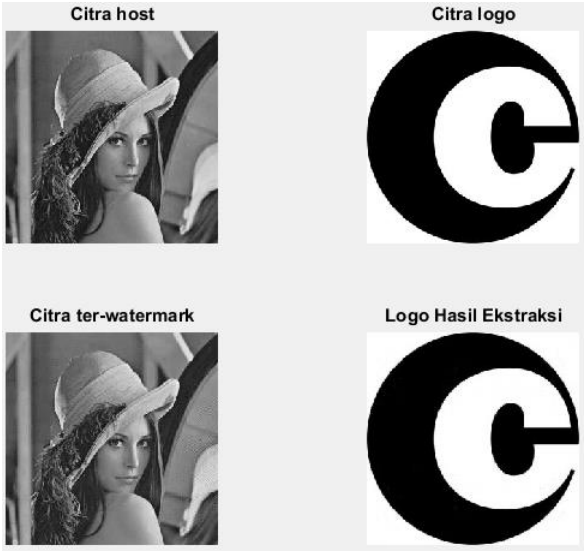
Gambar 3. Grafik Perbandingan PSNR Citra Ter-*watermark* Dengan Metode Lain

Pada tahapan pertama, eksperimen pada masing-masing citra dilakukan dengan penyisipan logo *watermark*. Hasil perhitungan pengujian citra ter-*watermark* dengan *PSNR* dan *NC* pada masing-masing citra terujikan dan dirangkumkan dalam Tabel 3. Pada Tabel 3. nilai *NC* yang diperoleh pada masing-masing citra yang ujikan adalah 1 (satu), sedangkan nilai *PSNR* bervariasi, citra *'Baboon'* sebesar 53,184 dB, *'Boat'* sebesar 53,328 dB, '*Cameraman'* sebesar 53,700 dB, *'Lena'* sebesar 53,668 dB, *'Man'* sebesar 53,328 dB dan *'Pepper'* sebesar 52,662 dB seperti yang ditampilkan pada Tabel 3.

Tabel 3. Hasil Eksperimen Citra Ter-*watermark* Dengan PSNR dan NC

Citra *host*	Citra ter-*watermark*	*PSNR*	*NC*
		53,184	1
		53,328	1
		53,700	1
		53,668	1
		53,328	1
		52,662	1

Pada tahapan kedua kami mengekstrak citra ter-*watermark* yang telah tertanam citra logo pada proses penyisipan sebelumnya. Tampilan citra *host*, citra logo, citra ter-*watermark* dan logo hasil ekstraksi dapat dilihat pada Gambar 4.



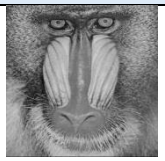
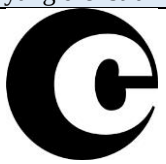
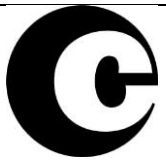
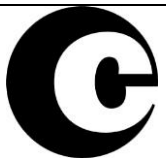
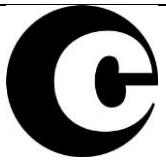
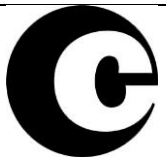
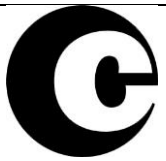
Gambar 4. Gambar Citra *Host*, Citra Logo, Citra Ter-*watermark* dan Logo Hasil Ekstraksi

Copyright @ 2015 IlmuKomputer.Com

Pada gambar 4 terlihat bahwa antara citra ter-*watermark* dan citra *host* nyaris tidak ada perbedaan. Begitu juga pada antara citra logo dan logo hasil ekstraksi sangat mirip sekali. Namun dalam *watermarking* kita dapat mengukur tingkat kemiripan antara citra asli dengan citra turunan dengan *Peak Signal To Noise Ratio (PSNR).*

Hasil perhitungan *MSE, PSNR* dan *NC* pada antara citra logo dengan citra hasil ekstraksi *watermarking* dapat dilihat pada Tabel 4.

Tabel 4. Hasil Eksperimen Ekstraksi Citra Ter-*watermark* DWT+SVD

Citra *host*	Citra Logo yang diekstrak	*MSE*	*PSNR*	*NC*
		2,466	44,2108	1
		5,535	40,699	1
		4,769	41,345	1
		5,732	40,5476	1
		4,900	41,2282	1
		4,623	41,4807	1

Berdasarkan Tabel 4. Logo yang diekstrak hampir mirip dengan aslinya, jika dilihat dengan kasat mata, maka dipastikan tidak ada perbedaannya. Akan tetapi dalam *watermarking* terdapat cara untuk mengukur kemiripan antara citra asli dengan citra turunannya yaitu dengan menentukan nilai *Peak Signal To Noise Ration (PSNR)*. Pada Tabel 4 nilai *PSNR* rata-rata diatas 40. Jika nilai *PSNR* jatuh dibawah 30dB menunjukkan kualitas citra yang cukup rendah (Cheddad et al., 2010), jika diatas atau sama dengan 40db maka menunjukkan kualitas citra yang tinggi.

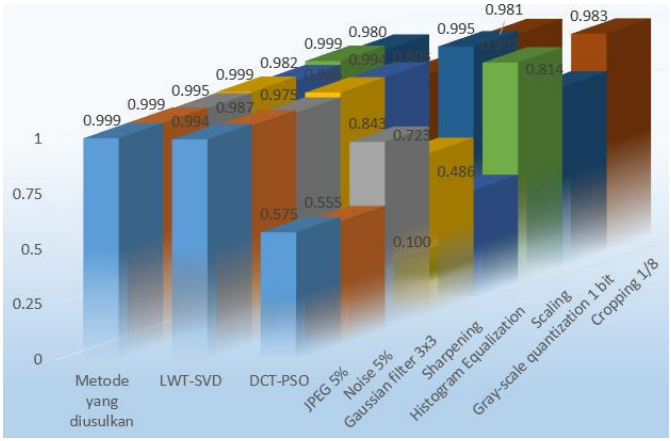
Pada tahapan ketiga, untuk menguji ketahanan citra yang sudah ditanam *watermark*, maka pada penelitian ini peneliti mencoba menerapkan beberapa serangan pada citra yang tertanam *watermark*, serangan atau pemberian distorsi berupa: *JPEG 5%, Noise 5%, Gaussian filter 3x3, Sharpening, Histogram Equalization, Scaling 512-256, Gray Quantitation 1bit,* dan *Cropping 1/8.* Pada Tabel 5. Terlihat hasil pemberian distorsi pada citra ter-*watermark.*

Tabel 5. *Normalized Cross-Correlation* (*NC*) Citra Ter-*watermark* Setelah Diberikan Distorsi

Citra *host*	a	b	c	d	e	f	g	h
	0,999	0,989	0,996	0,999	0,828	0,999	0,934	0,879
	0,999	0,999	0,995	0,999	0,982	0,999	0,980	0,803
	0,999	0,986	0,994	0,999	0,764	0,999	0,990	0,649
	0,999	0,989	0,996	0,999	0,793	0,999	0,910	0,880
	0,999	0,963	0,995	0,999	0,265	0,999	0,836	0,648
	0,999	0,985	0,996	0,999	0,705	0,999	0,935	0,965

Pada Tabel 5, kolom a adalah nilai *NC* setelah pemberian kompresi JPEG 5%, kolom b adalah nilai *NC* setelah pemberian *noise* 5%, kolom c adalah nilai *NC* setelah pemberian *gaussian filter* 3x3, kolom d adalah nilai *NC* setelah pemberian *sharpening*, kolom e adalah nilai NC setelah pemberian histogram, kolom f adalah nilai NC setelah pemberian *scaling*, kolom g adalah nilai NC setelah pemberian *gray scale* quantization 1 bit dan kolom h adalah nilai NC setelah pemberian *cropping* 1/8.

Terlihat pada Tabel 5, bahwa metode yang diusulkan mendapat nilai *NC* tertinggi yaitu sebesar 0,999 pada citra *'Baboon'* yang diujikan dengan beberapa serangan distorsi. Terbukti bahwa metode yang diusulkan kuat terhadap kompresi JPEG, *sharpening* dan *scaling* dengan mendapat nilai *NC* sebesar 0,999. Selanjutnya diikuti oleh distorsi n*oise* 5% dengan nilai *NC* sebesar 0,989.



Gambar 5. Grafik Perbandingan NC Citra Ter-*watermark* Distorsi Dengan Metode Lain

Pada Gambar 5, terlihat nilai *Normalized Cross-Correlation (NC)* dengan menerapkan metode yang diusulkan memperoleh nilai tertinggi pada beberapa perlakuan khusus

Copyright @ 2015 IlmuKomputer.Com

(distorsi). Pada perlakuan kompresi JPEG 5% dari semua citra yang diujikan menghasilkan nilai 0,999, dimana nilai tersebut diatas rata-rata dari hasil metode penelitian lainnya. Begitu juga dengan perlakukan *Noise* 5%, *Gaussian Filter* 3x3, dan *Sharpening* nilai *NC* cukup memuaskan, akan tetapi pada *Histogram filter*, *Grey-Scale Quantization*, dan *Cropping* selisih tipis dengan metode penelitian lainnya.

5 KESIMPULAN

Penerapan integrasi dari algoritma *Discrete Wavelet Transform (DWT)* dan *Singular Value Decomposition (SVD)* pada *watermarking* citra diusulkan untuk penentuan parameter *MSF* pada *watermarking* citra terbukti dapat meningkatkan kemiripan (*imperceptibility)* dan ketahanan (r*obustness)* pada citra ter-*watermark*. Pada metode *SVD* nilai faktor skala dilakukan untuk mengidentifikasi letak posisi koefisien mana yang memiliki nilai posisi yang optimal, jika *SVD* diterapkan untuk mengoptimasi koefisien parameter pada *DWT*. Komparasi nilai *PSNR* dan *NC* dari beberapa metode *watermarking* citra dilakukan untuk membuktikan kehandalan metode yang telah diusulkan. Hasil eksperimen membuktikan bahwa metode yang diusulkan *DWT+SVD* memiliki nilai *PSNR* dan *NC* yang lebih baik dari pada metode *watermarking* citra lainnya.

REFERENSI

Ali, M., & Ahn, C. W. (2014). An optimized *watermark*ing technique based on self-adaptive de in DWT-SVD transform domain. *Signal Processing*, *94*, 545–556.

Ali, M., Ahn, C. W., & Pant, M. (2014). A robust image *watermark*ing technique using SVD and differential evolution in DCT domain. *Optik - International Journal for Light and Electron Optics*, *125*(1), 428–434.

Aslantas, V. (2009). An optimal robust digital image *watermark*ing based on SVD using differential evolution algorithm. *Optics Communications*, *282*(5), 769–777.

Briassouli, A., & Strintzis, M. G. (2004). Locally optimum nonlinearities for DCT *watermark* detection. *IEEE Transactions on Image Processing*, *13*(12), 1604–1617.

Chang, C.-C., Tsai, P., & Lin, C.-C. (2005). SVD-based digital image *watermark*ing scheme. *Pattern Recognition Letters*, *26*, 1577–1586.

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal Processing*, *90*(3), 727–752.

Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. (1997). Secure spread spectrum *watermark*ing for multimedia. *IEEE Transactions on Image Processing*, *6*(12), 1673–1687.

Dogan, S., Tuncer, T., Avci, E., & Gulten, A. (2011). A robust color image *watermark*ing with Singular Value Decomposition method. *Advances in Engineering Software*, *42*(6), 336–346.

Fındık, O., Babaoğlu, İ., & Ülker, E. (2010). A color image *watermark*ing scheme based on hybrid classification method: Particle swarm optimization and k-nearest neighbor algorithm. *Optics Communications*, *283*(24), 4916–4922.

Hernández, J. R., Amado, M., & Pérez-González, F. (2000). DCT-domain *watermark*ing techniques for still images: detector performance analysis and a new structure. *IEEE Transactions on Image Processing*, *9*(1), 55–68.

Huang, H., Chen, Y., & Abraham, A. (2010). Optimized *watermark*ing using swarm-based bacterial foraging. *Journal of Information Hiding and Multimedia Signal Processing*, *1*(1), 51–58.

Ishtiaq, M., Sikandar, B., Jaffar, M. A., & Khan, A. (2010). Adaptive *Watermark* Strength Selection using Particle Swarm Optimization. *ICIC Express Letters*, *4*(5), 1–6.

Jia, S. L. (2014). A novel blind color images *watermark*ing based on SVD. *Optik*, *125*, 2868–2874.

Kumsawat, P., Attakitmongcol, K., & Srikaew, a. (2005). A new approach for optimization in image *watermark*ing by using genetic algorithms. *IEEE Transactions on Signal Processing*, *53*(12), 4707–4719.

Lai, C.-C. (2011a). A digital *watermark*ing scheme based on singular value decomposition and tiny genetic algorithm. *Digital Signal Processing*, *21*(4), 522–527.

Lai, C.-C. (2011b). An improved SVD-based *watermark*ing scheme using human visual characteristics. *Optics Communications*, *284*(4), 938–944.

Lai, C.-C., & Tsai, C.-C. (2010). Digital Image *Watermark*ing Using Discrete Wavelet Transform and Singular Value Decomposition. *IEEE Transactions on Instrumentation and Measurement*, *59*(11), 3060–3063.

Liu, F. L. F., & Liu, Y. L. Y. (2008). A *Watermark*ing Algorithm for Digital Image Based on DCT and SVD. *2008 Congress on Image and Signal Processing*, *1*, 380–383.

Liu, J. C., & Chen, S. Y. (2001). Fast two-layer image *watermark*ing without referring to the original image and *watermark*. *Image and Vision Computing*, *19*, 1083–1097.

Loukhaoukha, K. (2011). Optimal Image *Watermark*ing Algorithm Based on LWT-SVD via Multi-objective Ant Colony Optimization, *2*(4), 303–319.

Loukhaoukha, K. (2013). Image *Watermark*ing Algorithm Based on Multiobjective Ant Colony Optimization and Singular Value Decomposition, *2013*.

Mohammad, A. a., Alhaj, A., & Shaltaf, S. (2008). An improved SVD-based *watermark*ing scheme for protecting rightful ownership. *Signal Processing*, *88*, 2158–2180.

Nikolaidis, N., & Nikolaidis, N. (1998). Robust image *watermark*ing in the spatial domain. *Signal Processing*, *66*, 385–403.

Olkkonen, H. (2011). *Discrete Wavelete Transform :Algorithms And Application*. (H. Olkkonen, Ed.). Croatia: InTech.

Patra, J. C., Phua, J. E., & Bornand, C. (2010). A novel DCT domain CRT-based *watermark*ing scheme for image authentication surviving JPEG compression. *Digital Signal Processing: A Review Journal*, *20*(6), 1597–1611.

Qiao, L., & Nahrstedt, K. (1998). *Watermark*ing Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights. *Journal of Visual Communication and Image Representation*, *9*(3), 194–210.

Run, R. S., Horng, S. J., Lai, J. L., Kao, T. W., & Chen, R. J. (2012). An improved SVD-based *watermark*ing technique for copyright protection. *Expert Systems with Applications*, *39*(1), 673–689.

Shieh, C. S., Huang, H. C., Wang, F. H., & Pan, J. S. (2004). Genetic *watermark*ing based on transform-domain techniques. *Pattern Recognition*, *37*(3), 555–565.

Tan, T., & Liu, R. (2008). An improved SVD-based *watermark*ing scheme for protecting rightful ownership. *Signal Processing*, *88*(9), 2158–2180.

Van Fleet, P. J. (2011). *Discrete Wavelet Transformations: An Elementary Approach with Applications*. *Discrete Wavelet Transformations: An Elementary Approach with Applications*. John Wiley & Sons, Inc.

Wang, M.-S., & Chen, W.-C. (2007). Digital image copyright protection scheme based on visual cryptography and singular value decomposition. *Optical Engineering*, *46*(4), 067006.

Wang, M.-S., & Chen, W.-C. (2009). A hybrid DWT-SVD copyright protection scheme based on k-means clustering and visual cryptography. *Computer Standards & Interfaces*, *31*(4), 757–762.

Wang, Y.-R., Lin, W.-H., & Yang, L. (2011). An intelligent *watermark*ing method based on particle swarm optimization. *Expert Systems with Applications*, *38*(7), 8024–8029.

Xianghong, T. X. T., Lu, L. L. L., Lianjie, Y. L. Y., & Yamei, N. Y. N. (2004). A digital *watermark*ing scheme based on DWT and vector transform. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, 635–638.

Copyright @ 2015 IlmuKomputer.Com

BIOGRAFI PENULIS

Jaya Chandra. Memperoleh gelar M.Kom dari Sekolah Tinggi Manajemen Ilmu Komputer Nusa Mandiri, Jakarta. Staff IT di salah satu Perusahaan IT Swasta, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada saat ini pada bidang image processing, soft computing dan game programming.

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Penerapan *Naive Bayes* untuk Mengurangi Data *Noise* pada Klasifikasi Multi Kelas dengan *Decision Tree*

Al Riza Khadafy
Program Studi Ilmu Komputer, STMIK Nusa Mandiri Jakarta
rizacalm@gmail.com

Romi Satria Wahono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
romi@romisatriawahono.net

Abstrak: Selama beberapa dekade terakhir, cukup banyak algoritma *data mining* yang telah diusulkan oleh peneliti kecerdasan komputasi untuk memecahkan masalah klasifikasi di dunia nyata. Di antara metode-metode *data mining* lainnya, *Decision Tree* (DT) memiliki berbagai keunggulan diantaranya sederhana untuk dipahami, mudah untuk diterapkan, membutuhkan sedikit pengetahuan, mampu menangani data numerik dan kategorikal, tangguh, dan dapat menangani *dataset* yang besar. Banyak *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas yang ada di dunia memiliki *noise* atau mengandung *error*. Algoritma pengklasifikasi DT memiliki keunggulan dalam menyelesaikan masalah klasifikasi, namun data *noise* yang terdapat pada *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas dapat mengurangi akurasi pada klasifikasinya. Masalah data *noise* pada *dataset* tersebut akan diselesaikan dengan menerapkan pengklasifikasi *Naive Bayes* (NB) untuk menemukan *instance* yang mengandung *noise* dan menghapusnya sebelum diproses oleh pengklasifikasi DT. Pengujian metode yang diusulkan dilakukan dengan delapan *dataset* uji dari UCI (*University of California, Irvine*) *machine learning repository* dan dibandingkan dengan algoritma pengklasifikasi DT. Hasil akurasi yang didapat menunjukkan bahwa algoritma yang diusulkan DT+NB lebih unggul dari algoritma DT, dengan nilai akurasi untuk masing-masing *dataset* uji seperti *Breast Cancer* 96.59% (meningkat 21,06%), *Diabetes* 92,32% (meningkat 18,49%), *Glass* 87,50% (meningkat 20,68%), *Iris* 97,22% (meningkat 1,22%), *Soybean* 95,28% (meningkat 3,77%), *Vote* 98,98% (meningkat 2,66%), *Image Segmentation* 99,10% (meningkat 3,36%), dan *Tic-tac-toe* 93,85% (meningkat 9,30%). Dengan demikian dapat disimpulkan bahwa penerapan NB terbukti dapat menangani data *noise* pada *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas sehingga akurasi pada algoritma klasifikasi DT meningkat.

Keywords: data *noise*, pengklasifikasi *Naive Bayes*, pengklasifikasi *Decision Tree*

1 PENDAHULUAN

Selama beberapa dekade terakhir, cukup banyak algoritma *data mining* yang telah diusulkan oleh peneliti kecerdasan komputasi untuk memecahkan masalah klasifikasi di dunia nyata (Farid et al., 2013; Liao, Chu, & Hsiao, 2012; Ngai, Xiu, & Chau, 2009). Secara umum, klasifikasi adalah fungsi *data mining* yang menggambarkan dan membedakan kelas data atau konsep. Tujuan dari klasifikasi adalah untuk secara akurat memprediksi label kelas dari *instance* yang nilai atributnya diketahui, tapi nilai kelasnya tidak diketahui. Beberapa algoritma *data mining* yang sering digunakan untuk klasifikasi diantaranya adalah *Decision Tree* dan *Naive Bayes*.

Decision Tree (DT) atau pohon keputusan adalah algoritma klasifikasi yang banyak digunakan dalam *data mining* seperti ID3 (Quinlan, 1986), ID4 (Utgoff, 1989), ID5 (Utgoff, 1989), C4.5 (Quinlan, 1993), C5.0 (Bujlow, Riaz, & Pedersen, 2012), dan CART (Breiman, Friedman, Olshen, & Stone, 1984). Tujuan dari DT adalah untuk membuat model yang dapat memprediksi nilai dari sebuah kelas target pada *test instance* yang tidak terlihat berdasarkan beberapa fitur masukan (Loh & Shih, 1997; Safavian & Landgrebe, 1991; Turney, 1995). Di antara metode-metode *data mining* lainnya, DT memiliki berbagai keunggulan diantaranya sederhana untuk dipahami, mudah untuk diterapkan, membutuhkan sedikit pengetahuan, mampu menangani data numerik dan kategorikal, tangguh, dan dapat menangani *dataset* yang besar (Han, Kamber, & Pei, 2012).

Berbagai metode terkait algoritma pengklasifikasi DT telah dikembangkan pada beberapa penelitian, diantaranya adalah *Decision Tree Using Fast Splitting Attribute Selection* (DTFS) (Franco-Arcega, Carrasco-Ochoa, Sanchez-Diaz, & Martinez-Trinidad, 2011), *Classification by Clustering* (CbC) (Aviad & Roy, 2011), C4.5 dengan pendekatan *One-Against-All* untuk meningkatkan akurasi klasifikasi pada masalah klasifikasi multi kelas (Polat & Gunes, 2009), penanganan eksepsi pada DT (Balamurugan & Rajaram, 2009), *Associative Classification Tree* (ACT) (Chen & Hung, 2009), *Fuzzy Decision Tree Gini Index Based* (G-FDT) (Chandra & Paul Varghese, 2009), dan *Co-Evolving Decision Tree* (Aitkenhead, 2008).

Performa algoritma *data mining* dalam banyak kasus tergantung pada kualitas *dataset*, karena data *training* berkualitas rendah dapat menyebabkan klasifikasi yang lemah (Han et al., 2012). Dengan demikian, dibutuhkan teknik *data preprocessing* untuk mempersiapkan data yang akan diproses. Hal ini dapat meningkatkan kualitas data, sehingga membantu untuk meningkatkan akurasi dan efisiensi proses *data mining*. Beberapa teknik *data preprocessing* diantaranya adalah *data cleaning*: menghapus data yang mengandung *error*, *data integration*: menggabungkan data dari berbagai sumber, *data transformation*: normalisasi data, dan *data reduction*: mengurangi ukuran data dengan menggabungkan dan menghilangkan fitur yang berlebihan.

Naive Bayes (NB) adalah algoritma klasifikasi probabilitas sederhana yang berdasarkan pada teorema Bayes, asumsi bebas yang kuat (*naive*), dan model fitur independen (Farid, Rahman, & Rahman, 2011; Farid & Rahman, 2010; Lee & Isa, 2010). NB juga merupakan algoritma klasifikasi yang utama pada *data mining* dan banyak diterapkan dalam masalah klasifikasi di dunia nyata karena memiliki performa klasifikasi yang

Copyright @ 2015 IlmuKomputer.Com

tinggi. Mirip dengan DT, algoritma pengklasifikasi NB juga memiliki beberapa keunggulan seperti mudah digunakan, hanya membutuhkan satu kali *scan* data *training*, penanganan nilai atribut yang hilang, dan data kontinu (Han et al., 2012).

Banyak *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas yang ada di dunia memiliki *noise* atau mengandung *error*, hal ini dapat menyebabkan berkurangnya akurasi pada klasifikasi DT (Han et al., 2012; Polat & Gunes, 2009; Quinlan, 1986). *Instance* yang mengandung *error* pada *dataset* menyebabkan salah klasifikasi saat diproses oleh algoritma pengklasifikasi NB. Dengan demikian, algoritma pengklasifikasi NB dapat digunakan untuk menemukan *instance* yang bermasalah pada *dataset*.

Pada penelitian ini algoritma pengklasifikasi NB akan digunakan untuk menemukan *instance* yang bermasalah pada data *training* dan menghapusnya sebelum algoritma DT membuat pohon keputusan agar akurasi klasifikasinya meningkat.

2 PENELITIAN TERKAIT

Polat dan Gunes melakukan penelitian pada tahun 2009, yaitu mereka menggabungkan algoritma pengklasifikasi C4.5 dengan pendekatan *One-Against-All* untuk memecahkan masalah klasifikasi multi kelas. Pada penelitian tersebut digunakan *dataset* dari UCI *machine learning repository*, diantaranya *dataset Dermatology*, *Image Segmentation*, dan *Lymphography*. Pertama algoritma C4.5 dijalankan pada setiap *dataset* menggunakan 10-*fold cross validation* dan mendapatkan hasil akurasi 84,48%, 88,79%, dan 80,11% pada masing-masing *dataset*. Kemudian algoritma usulan dijalankan pada setiap *dataset* menggunakan 10-*fold cross validation* dan mendapatkan hasil akurasi yang lebih tinggi yaitu 96,71%, 95,18%, dan 87,95% pada masing-masing *dataset*.

Penelitian yang dilakukan Aitkenhead pada tahun 2008, yaitu dengan mengembangkan pendekatan evolusioner pada algoritma *Decision Tree* untuk mengatasi masalah data *noise* dan masalah kombinasi data kuantitatif dan kualitatif pada *dataset* yang dapat menyulitkan proses kategorisasi kelas. Pada penelitian tersebut digunakan *dataset Glass Chemistry* dan *Car Costing*. Pengujian dilakukan dengan menjalankan algoritma usulan pada setiap *dataset*, kemudian dibandingkan dengan algoritma C4.5 dan didapatkan hasil akurasi yang lebih tinggi yaitu 0,824 dan 0,892 pada masing-masing *dataset*.

Penelitian yang dilakukan Balamurugan dan Rajaram pada tahun 2009, yaitu mereka melakukan perbaikan pada algoritma *Decision Tree* dengan menambahkan prosedur penghitungan *Maximum Influence Factor* (MIF) untuk mengatasi masalah kegagalan dalam pemilihan atribut yang akan di-*split* yang dapat menyebabkan label kelas dipilih secara acak. *Dataset* yang digunakan dalam penelitian tersebut diantaranya *Blood Transfusion*, *Teaching Assistant Evaluation*, *SPECT Heart*, *Haberman's Survival*, *Contraceptive Method Choice*, *Hayes Roth*, *Concrete*, *Forest-fires*, *Solarflare 1*, dan *Solarflare 2*. Pengujian dilakukan dengan menjalankan algoritma usulan pada setiap *dataset* kemudian dilakukan perbandingan dengan algoritma lain seperti C4.5, NB, K-NN. Pada penelitian tersebut didapatkan nilai akurasi lebih tinggi yaitu 85,16 %, 77,78 %, 71,70 %, 78,79 %, 77,50 %, 76,74 %, 76,74 %, 75,68 %, 77,09% pada masing-masing *dataset*.

Penelitian yang dilakukan Chandra dan Paul Varghese pada tahun 2009, yaitu mereka melakukan perbaikan terhadap algoritma *Decision Tree* untuk mengatasi masalah pemilihan *splitting attribute* yang dapat menyebabkan *misclassification*. Perbaikan yang dilakukan adalah dengan menggunakan teknik *Fuzzy Decision Tree Algorithm Gini Index Based* (G-FDT). *Dataset* yang digunakan dalam penelitian tersebut diantaranya *Haberman*, *Iris*, *Balanced Scale*, *Liver*, *Diabetes*, *Wincosin BC*, *Echocardiogram*, *Wine*, *Ionosphere*, *Glass*, *Vehicle Silhouette*, *Heart Stat Log*, *Smoking*, *Contraceptive Method Choice*. Pengujian dilakukan dengan menjalankan algoritma usulan pada setiap *dataset* kemudian dilakukan perbandingan dengan algoritma *Supervised Learning In Quest* (SLIQ). Pada penelitian tersebut didapatkan hasil akurasi dan kecepatan algoritma yang lebih tinggi.

3 METODE USULAN

Untuk menangani masalah data *noise* pada klasifikasi *Decision Tree* (DT), diusulkan metode dengan pendekatan klasifikasi *Naive Bayes* (NB) untuk menemukan *instance* yang bermasalah atau mengandung *noise* kemudian menghapus *instance* tersebut. *Pseudocode* algoritma usulan ditunjukkan pada Gambar 1.

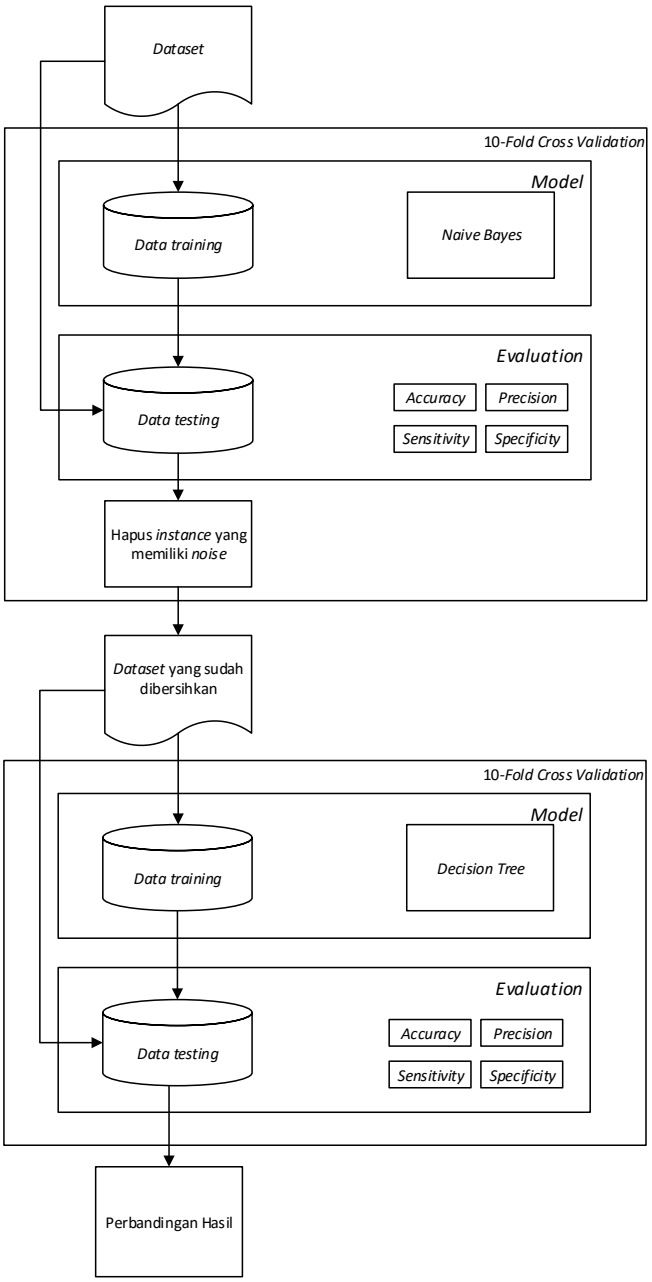
```
    Input: D = {x1, x2, ..., xn} // dataset training
    Output: T, Decision tree. // model decision tree
    Metode:
1   for each class, Ci ∈ D, do
2       Find the prior probabilities, P(Ci).
3   end for
4   for each attribute value, Aij ∈ D, do
5       Find the class conditional probabilities, P(Aij|Ci).
6   end for
7   for each training instance, xi ∈ D, do
8       Find the posterior probabilities, P(Ci|xi)
9       if xi misclassified, do
10          Remove xi from D; // hapus instance yang salah klasifikasi
11      end if
12  end for
13  T = ∅;
14  Determine best splitting attribute;
15  T = Create the root node and label it with the splitting attribute;
16  T = Add arc to the root node for each split predicate and label;
17  for each arc do
18      D = Dataset created by applying splitting predicate to D;
19      if stopping point reached for this path,
20          T' = Create a leaf node and label it with an appropriate class;
21      else
22          T' = DTBuild(D);
23      end if
24      T = Add T' to arc;
25  end for
```

Gambar 1. *Pseudocode* Algoritma DT + NB

Perancangan metode yang diusulkan yaitu dengan menerapkan algoritma pengklasifikasi NB untuk mengurangi *noise* pada klasifikasi multi kelas dengan DT. Dimulai dengan membagi *dataset* menjadi data *training* dan data *testing* dengan menggunakan metode 10-*fold cross validation*, kemudian menerapkan algoritma pengklasifikasi NB untuk menemukan dan kemudian menghapus *instance* yang memiliki *noise*. Kemudian *dataset* yang sudah dibersihkan dari *instance* yang memiliki *noise* tersebut diproses menggunakan algoritma DT untuk menghasilkan pohon keputusan. Selanjutnya hasil evaluasi model diukur nilai *accuracy*, *precision*, *sensitivity*, dan *specificity*. Gambar 2 menampilkan metode yang diusulkan.

Metode yang diusulkan diawali dengan membagi *dataset* menjadi data *training* dan data *testing* dengan menggunakan 10-*fold cross validation*, yaitu dengan membagi data 90% untuk proses *training* dan 10% untuk proses *testing*. Data *training* diproses dengan menggunakan algoritma pengklasifikasi NB untuk menghasilkan model klasifikasi. Kemudian dengan model klasifikasi tersebut dilakukan *testing*. Selanjutnya *instance* yang ditemukan salah klasifikasi atau *misclassified* dihapus dari *dataset*.

Copyright @ 2015 IlmuKomputer.Com

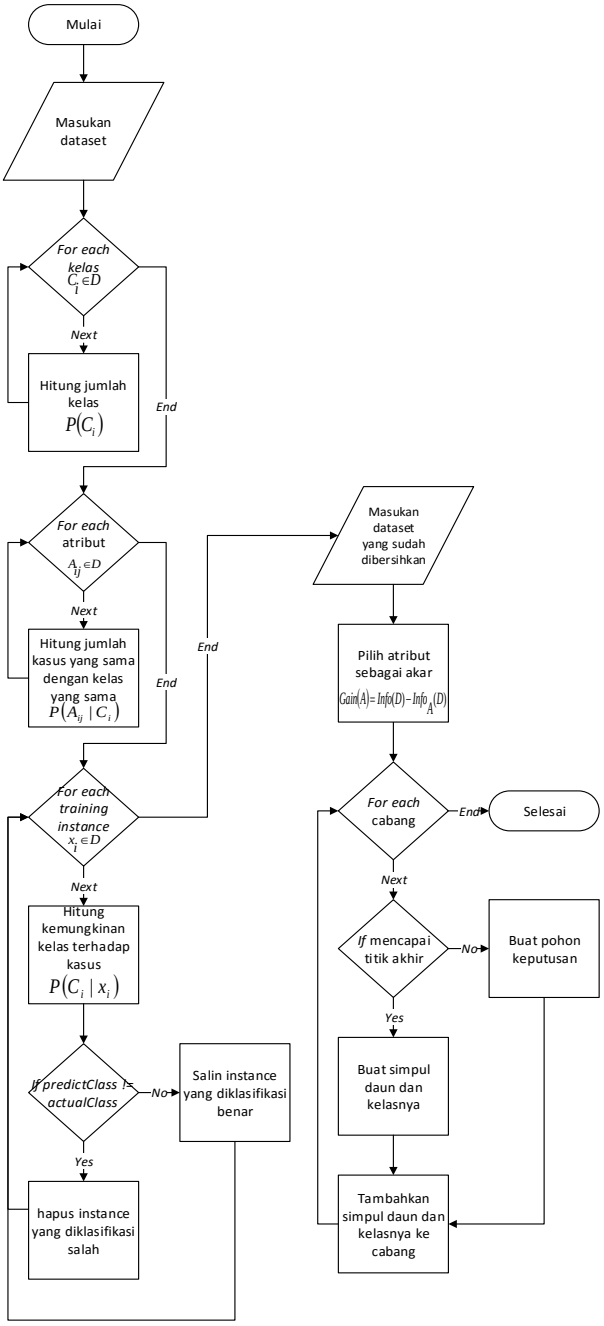


Gambar 2. Metode yang Diusulkan

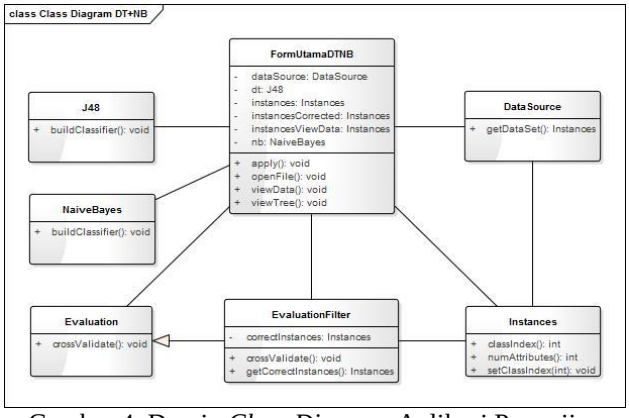
Dataset yang sudah dibersihkan dari *instance* yang salah klasifikasi kemudian dibagi menjadi data *training* dan data *testing* dengan menggunakan 10-*fold cross validation*, selanjutnya data *training* diproses dengan algoritma DT untuk menghasilkan pohon keputusan. Kemudian dengan model pohon keputusan tersebut dilakukan *testing*. Hasil validasi dari proses digunakan untuk mengukur kinerja algoritma dari metode yang diusulkan. Langkah-langkah pada penerapan algoritma pengklasifikasi NB untuk mengurangi *noise* pada klasifikasi multi kelas dengan DT ditunjukkan pada Gambar 3.

Proses eksperimen dan pengujian metode pada penelitian ini menggunakan antarmuka pengguna atau *user interface* (UI) dari aplikasi yang dikembangkan untuk mengukur kinerja metode yang diusulkan.

Aplikasi didesain menggunakan bahasa pemrograman Java dengan menggunakan *library* Weka. Rancangan dalam bentuk *class diagram* ditunjukkan pada Gambar 4 dan rancangan *form* utama UI aplikasi ditunjukkan pada Gambar 5.

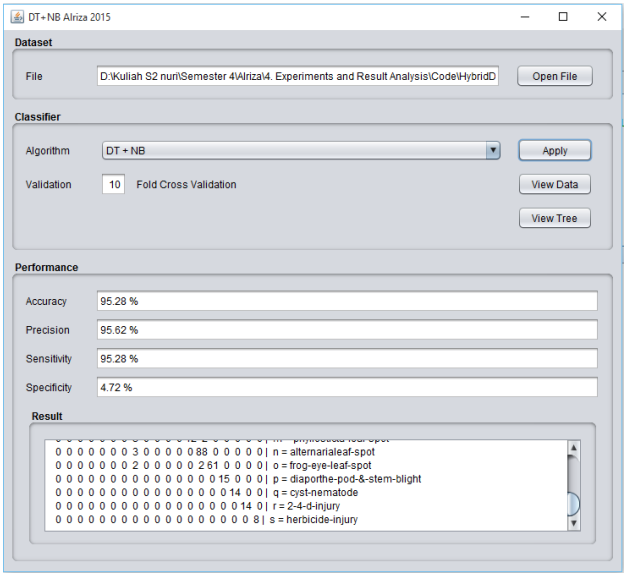


Gambar 3. *Flowchart* Metode yang Diusulkan



Gambar 4. Desain *Class Diagram* Aplikasi Pengujian

UI aplikasi memiliki tiga bagian yaitu *Dataset*, *Classifier*, dan *Performance*. Bagian *Dataset* memiliki tombol *Open File* berfungsi untuk memilih *file dataset*. Bagian *Classifier* memiliki *combo box Algorithm* untuk memilih algoritma yang akan digunakan, tombol *Apply* untuk memproses algoritma yang dipilih, tombol *View Data* untuk melihat *dataset* yang dipilih, tombol *View Tree* untuk melihat model pohon keputusan yang dihasilkan, *text box Validation* untuk menentukan jumlah *k-fold cross validation*. Bagian *Performance* memiliki *text box Accuracy*, *Precision*, *Sensitivity*, *Specificity*, dan *text area Result* yang menampilkan hasil kinerja klasifikasi.



Gambar 5. Aplikasi Pengujian

Dalam penelitian ini digunakan komputer untuk melakukan proses perhitungan terhadap metode yang diusulkan dengan spesifikasi komputer yang ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi Komputer

Processor	Intel Core i5-4210U 1,7 GHz
Memory	6 GB
Harddisk	1 TB
Operating System	Windows 10
Application	Java - Netbeans IDE 8.02

Pengukuran kinerja model menggunakan tabel *confusion matrix*. Pada tabel *confusion matrix* berisi nilai *false positive* (FP), *false negative* (FN), *true positive* (TP), dan *true negative* (TN). Kinerja yang diukur termasuk akurasi secara umum seperti *accuracy*, *precision*, *sensitivity*, dan *specificity*. Validasi yang dilakukan adalah dengan menggunakan 10-*fold cross validation* dimana *dataset* akan dibagi dalam dua segmen, data *training* dan data *testing* menjadi 10 bagian. Kinerja model akan dibandingkan antara algoritma *Decision Tree* (DT) + *Naive Bayes* (NB) dengan DT.

4 HASIL PENELITIAN

Eksperimen dilakukan dengan menggunakan laptop Dell 5000 *series* dengan *processor* Intel Core i5-4210U @ 1,7 GHz 2.40 GHz, *memory* 6 GB, *harddisk* 1 TB, dan menggunakan sistem operasi Windows 10 64-bit. Eksperimen ini juga menggunakan perangkat lunak Weka 3.6 untuk menganalisa penghitungan, dan menggunakan Netbeans IDE 8.02 dengan bahasa pemrograman Java dalam pengembangan aplikasi untuk menguji hasil perhitungan. Metode yang digunakan adalah dengan menerapkan algoritma pengklasifikasi *Naive Bayes* (NB) untuk mengurangi *noise* pada klasifikasi multi kelas dengan *Decision Tree* (DT). Algoritma pengklasifikasi NB digunakan untuk menemukan dan menghilangkan *instance* yang mengandung *noise*, sehingga akurasi pada klasifikasi yang dihasilkan oleh algoritma DT dapat meningkat.

Data yang digunakan pada penelitian ini adalah delapan *dataset* uji dari *University of California Irvine* (UCI) *machine learning repository* yang diperoleh melalui situs http://archive.ics.uci.edu/ml. *Dataset* tersebut digunakan oleh banyak peneliti untuk melakukan pengujian metode yang dibuat. *Dataset* tersebut juga bersifat publik dan dapat digunakan oleh siapa saja. *Dataset* yang digunakan dalam penelitian ini terdiri atas *dataset* yang memiliki dua kelas dan *dataset* yang memiliki lebih dari dua kelas atau multi kelas. Delapan *dataset* yang digunakan adalah sebagai berikut:

1. Data kanker payudara (*Breast Cancer*)
2. Data pasien diabetes (*Diabetes*)
3. Data klasifikasi kaca (*Glass*)
4. Data tanaman iris (*Iris*)
5. Data kacang kedelai (*Soybean*)
6. Data *voting* kongres di Amerika Serikat tahun 1984 (*Vote*)
7. Data segmentasi gambar (*Image Segmentation*)
8. Data permainan tic-tac-toe (*Tic-tac-toe*)

Dataset Breast Cancer adalah kumpulan data terkait klasifikasi penyakit kanker payudara, atribut yang dimiliki bertipe *nominal*, terdiri dari 286 *instances*, 10 atribut, dan 2 kelas.

Dataset Diabetes adalah kumpulan data terkait klasifikasi penyakit diabetes, atribut yang dimiliki bertipe *real*, terdiri dari 768 *instances*, 9 atribut, dan 2 kelas.

Dataset Glass adalah kumpulan data terkait klasifikasi tipe *glass* atau kaca, atribut yang dimiliki bertipe *real*, terdiri dari 214 *instances*, 10 atribut, dan 6 kelas.

Dataset Iris adalah kumpulan data terkait klasifikasi tanaman *iris*, atribut yang dimiliki bertipe *real*, terdiri dari 150 *instances*, 5 atribut, dan 3 kelas.

Dataset Soybean adalah kumpulan data terkait klasifikasi penyakit tanaman kedelai, atribut yang dimiliki bertipe *nominal*, terdiri dari 683 *instances*, 36 atribut, dan 19 kelas.

Dataset Vote adalah kumpulan data terkait klasifikasi pemilih dalam pemungutan suara di Amerika Serikat pada tahun 1984, atribut yang dimiliki bertipe *nominal*, terdiri dari 435 *instances*, 17 atribut, dan 2 kelas.

Dataset Image Segmentation adalah kumpulan data terkait klasifikasi gambar alam terbuka, atribut yang dimiliki bertipe *real*, terdiri dari 1500 *instances*, 20 atribut, dan 7 kelas.

Dataset Tic-tac-toe adalah kumpulan data terkait permainan bulat-silang, atribut yang dimiliki bertipe *nominal*, terdiri dari 958 *instances*, 10 atribut, dan 2 kelas. Tabel 2 menjelaskan spesifikasi dari delapan *dataset* UCI *machine learning repository*.

Tabel 2. Spesifikasi Delapan *Dataset* UCI *Machine Learning Repository*

Dataset	Jumlah atribut	Tipe atribut	Jumlah *instance*	Jumlah kelas
Breast cancer	10	*Nominal*	286	2
Diabetes	9	*Real*	768	2
Glass	10	*Real*	214	6
Iris	5	*Real*	150	3
Soybean	36	*Nominal*	683	19
Vote	17	*Nominal*	435	2
Image Segmentation	20	*Real*	1500	7
Tic-tac-toe	10	*Nominal*	958	2

Copyright @ 2015 IlmuKomputer.Com

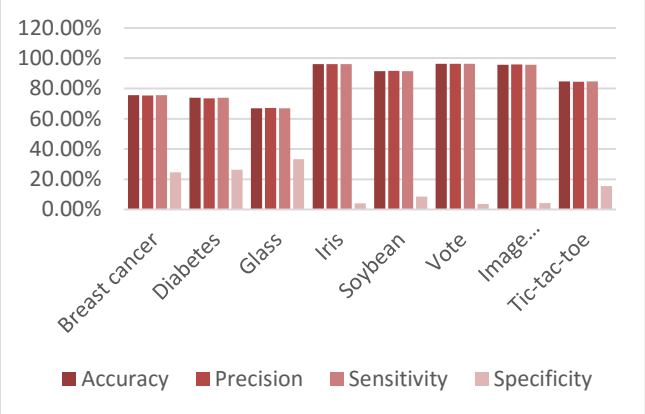
Eksperimen dalam penelitian dilakukan dalam dua metode, yaitu dengan menggunakan metode *Decision Tree* dan metode *Decision Tree* (DT) yang diintegrasikan dengan *Naive Bayes* (NB) atau DT+NB.

Pada eksperimen pertama ini percobaan dilakukan dengan menguji delapan *dataset* UCI *machine learning repository* menggunakan algoritma DT. Teknik validasi yang digunakan adalah 10-*fold cross validation*, dengan membagi *dataset* menjadi 10 bagian. Dari 10 bagian data tersebut, 9 bagian dijadikan data *training*, 1 bagian sisanya dijadikan data *testing*.

Berdasarkan hasil eksperimen, dilakukan perbandingan kinerja *Decision Tree* (DT) dengan *Decision Tree* dan *Naive Bayes* (DT + NB) untuk mengetahui algoritma klasifikasi yang terbaik. Pengukuran dilakukan dengan menguji delapan *dataset* dari UCI *machine learning repository* (*Breast Cancer*, *Diabetes*, *Glass*, *Iris*, *Soybean*, *Vote*, *Image Segmentation*, *Tic-tac-toe*). Hasil pengukuran algoritma klasifikasi dapat dilihat pada Tabel 3 dan grafik perbandingannya pada Gambar 6 untuk semua *dataset* dengan menggunakan algoritma DT, pada Tabel 4 dan Gambar 7 untuk semua *dataset* dengan menggunakan algoritma DT+NB.

Tabel 3. Hasil Pengukuran Algoritma Klasifikasi DT pada Semua Dataset Uji

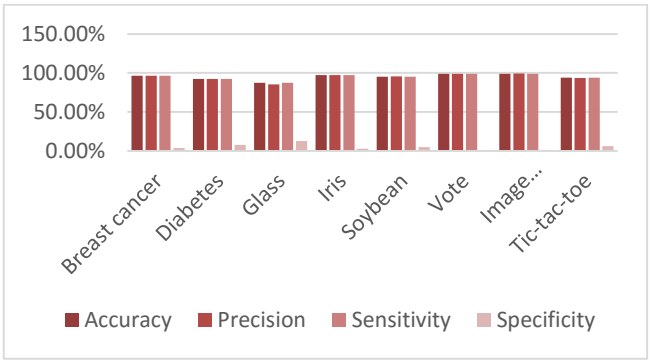
Dataset Training	*Accuracy*	*Precision*	*Sensitivity*	*Specificity*
Breast Cancer	75,52%	75,24%	75,52%	24,48%
Diabetes	73,83%	73,52%	73,83%	26,17%
Glass	66,82%	67,04%	66,82%	33,18%
Iris	96,00%	96,04%	96,00%	4,00%
Soybean	91,51%	91,65%	91,51%	8,49%
Vote	96,32%	96,32%	96,32%	3,68%
Image Segmentation	95,73%	95,78%	95,73%	4,27%
Tic-tac-toe	84,55%	84,49%	84,55%	15,45%



Gambar 6. Grafik Kinerja Algoritma Klasifikasi DT pada Semua Dataset Uji

Tabel 4. Hasil Pengukuran Algoritma Klasifikasi DT + NB pada Semua Dataset Uji

Dataset Training	*Accuracy*	*Precision*	*Sensitivity*	*Specificity*
Breast cancer	96,59%	96,63%	96,59%	3,41%
Diabetes	92,32%	92,34%	92,32%	7,68%
Glass	87,50%	85,46%	87,50%	12,50%
Iris	97,22%	97,25%	97,22%	2,78%
Soybean	95,28%	95,62%	95,28%	4,72%
Vote	98,98%	98,98%	98,98%	1,02%
Image Segmentation	99,10%	99,11%	99,10%	0,90%
Tic-tac-toe	93,85%	93,74%	93,85%	6,15%



Gambar 7. Grafik Kinerja Algoritma Klasifikasi DT + NB pada Semua Dataset Uji

Selanjutnya dilakukan Uji beda dengan metode statistik yang digunakan untuk menguji hipotesis pada algoritma DT dengan algoritma DT + NB.

H_0: Tidak ada perbedaan antara nilai rata-rata *accuracy* DT dengan DT + NB

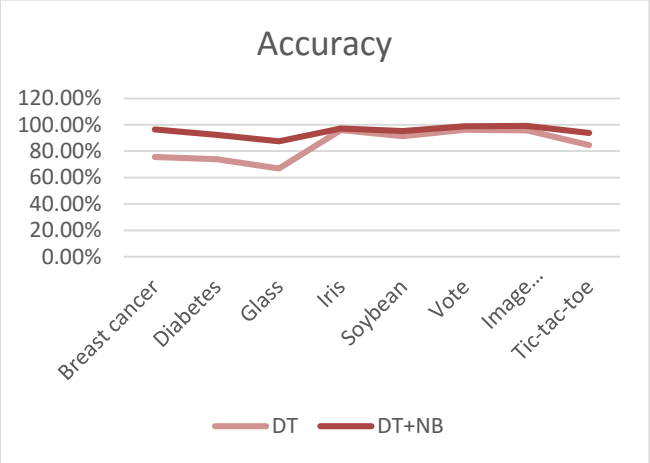
H_1: Ada perbedaan antara nilai rata-rata *accuracy* DT dengan DT + NB

Perbedaan nilai *accuracy* antara DT dengan DT + NB disajikan dalam Tabel 5 dan Gambar 8.

Tabel 5. Perbandingan *Accuracy* DT dengan DT + NB

Dataset Training	DT	DT + NB
Breast Cancer	75,52%	96,59%
Diabetes	73,83%	92,32%
Glass	66,82%	87,50%
Iris	96,00%	97,22%
Soybean	91,51%	95,28%
Vote	96,32%	98,98%
Image Segmentation	95,73%	99,10%
Tic-tac-toe	84,55%	93,85%

Copyright @ 2015 IlmuKomputer.Com



Gambar 8. Grafik Perbandingan *Accuracy* DT dengan DT + NB

Hasil analisis dengan menggunakan uji *t-Test Paired Two Sample for Means* disajikan dalam Tabel 6.

Tabel 6. Hasil Uji Beda Statistik *Accuracy* DT dengan DT + NB

	DT	*DT + NB*
Mean	0,850362	**0,95104**
Variance	0,013599	0,001497
Observations	8	8
Pearson Correlation	0,845275	
Hypothesized Mean Difference	0	
Df	7	
t Stat	-3,29507	
P(T<=t) one-tail	0,006605	
t Critical one-tail	1,894579	
P(T<=t) two-tail	**0,01321**	
t Critical two-tail	2,364624	

Pada Tabel 6 dapat dilihat bahwa nilai rata-rata *accuracy* dari algoritma DT + NB lebih tinggi dibandingkan algoritma DT sebesar 0,95104. Dalam uji beda statistik nilai *alpha* ditentukan sebesar 0,05, jika nilai p lebih kecil dibandingkan *alpha* ($p < 0,05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara algoritma yang dibandingkan, namun bila nilai p lebih besar dibanding *alpha* ($p > 0,05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara algoritma yang dibandingkan. Pada Tabel 4.36 dapat diketahui bahwa nilai P(T<=t) adalah 0,01321, ini menunjukkan bahwa nilai p lebih kecil daripada nilai *alpha* (0,01321 < 0,05) sehingga hipotesis H_0 ditolak dan H_1 diterima. Dengan demikian dapat disimpulkan bahwa ada perbedaan yang signifikan antara algoritma DT dengan DT + NB.

5 KESIMPULAN

Dalam penelitian ini algoritma pengklasifikasi *Naive Bayes* (NB) digunakan untuk menemukan dan menghilangkan *instance* yang mengandung *noise*, sehingga akurasi pada klasifikasi yang dihasilkan oleh algoritma *Decision Tree* (DT) dapat meningkat. Pengujian dilakukan pada delapan *dataset* dari UCI *machine learning repository* dengan menggunakan algoritma yang diusulkan dan algoritma DT. *Dataset* yang digunakan dalam pengujian terdiri atas *dataset* yang memiliki dua kelas dan *dataset* yang memiliki lebih dari dua kelas atau multi kelas.

Berdasarkan hasil eksperimen dan evaluasi pada penelitian ini, secara umum dapat disimpulkan bahwa penerapan algoritma pengklasifikasi NB dapat mengurangi data *noise* pada *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas sehingga akurasi klasifikasi algoritma DT dapat meningkat. Hasil akurasi yang didapat menunjukkan bahwa metode yang diusulkan DT+NB lebih unggul dari metode DT, dengan nilai akurasi untuk masing-masing *dataset* uji seperti *Breast Cancer* 96,59% (meningkat 21,06%), *Diabetes* 92,32% (meningkat 18,49%), *Glass* 87,50% (meningkat 20,68%), *Iris* 97,22% (meningkat 1,22%), *Soybean* 95,28% (meningkat 3,77%), *Vote* 98,98% (meningkat 2,66%), *Image Segmentation* 99,10% (meningkat 3,36%), dan *Tic-tac-toe* 93,85% (meningkat 9,30%). Perbandingan nilai akurasi dilakukan dengan uji t atau *t-Test* antara metode DT dengan metode yang diusulkan DT + NB untuk mendapatkan nilai perbedaan akurasi signifikan antara kedua metode tersebut. Dari hasil perbandingan didapatkan nilai P(T<=t) adalah 0,01321, ini menunjukkan bahwa nilai p lebih kecil daripada nilai *alpha* (0,01321 < 0,05). Dengan demikian dapat disimpulkan bahwa ada perbedaan akurasi yang signifikan antara metode DT dengan DT + NB.

REFERENSI

Aggarwal, C. C. (2015). *Data Mining, The Textbook*. Springer Berlin Heidelberg.

Aitkenhead, M. J. (2008). A co-evolving decision tree classification method. *Expert Systems with Applications*, *34*(1), 18–25.

Aviad, B., & Roy, G. (2011). Classification by clustering decision tree-like classifier based on adjusted clusters. *Expert Systems with Applications*, *38*(7), 8220–8228.

Balamurugan, S. A. A., & Rajaram, R. (2009). Effective solution for unhandled exception in decision tree induction algorithms. *Expert Systems with Applications*, *36*(10), 12113–12119.

Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2008). *Thesis Guide - A Guide for Students in Computer Science and Information Systems (2nd ed)*. Springer-Verlag.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. *Chapman and Hall/CRC* (1st ed., Vol. 19). Chapman and Hall/CRC.

Bujlow, T., Riaz, T., & Pedersen, J. M. (2012). A method for classification of network traffic based on C5.0 machine learning algorithm. *2012 International Conference on Computing, Networking and Communications, ICNC'12*, 237–241.

Chandra, B., & Paul Varghese, P. (2009). Fuzzifying Gini Index based decision trees. *Expert Systems with Applications*, *36*(4), 8549–8559.

Chen, Y. L., & Hung, L. T. H. (2009). Using decision trees to summarize associative classification rules. *Expert Systems with Applications*, *36*, 2338–2351.

Dawson, C. W. (2009). *Projects in Computing and Information Systems A Student's Guide (2nd ed)*. *Great Britain: Pearson Education*.

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Farid, D. M., & Rahman, M. Z. (2010). Anomaly network intrusion detection based on improved self adaptive Bayesian algorithm. *Journal of Computers*, *5*(1), 23–31.

Copyright @ 2015 IlmuKomputer.Com

Farid, D. M., Rahman, M. Z., & Rahman, C. M. (2011). Adaptive Intrusion Detection based on Boosting and Naive Bayesian Classifier. *International Journal of Computer Applications*, *24*(3), 12–19.

Farid, D. M., Zhang, L., Hossain, A., Rahman, C. M., Strachan, R., Sexton, G., & Dahal, K. (2013). An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, *40*(15), 5895–5906.

Franco-Arcega, A., Carrasco-Ochoa, J. a., Sanchez-Diaz, G., & Martinez-Trinidad, J. F. (2011). Decision tree induction using a fast splitting attribute selection for large datasets. *Expert Systems with Applications*, *38*(11), 14290–14300.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*.

Jamain, A., & Hand, D. J. (2008). Mining Supervised Classification Performance Studies: A Meta-Analytic Investigation. *Journal of Classification*, *25*(1), 87–112.

Larose Daniel T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley Interscience.

Lee, L. H., & Isa, D. (2010). Automatically computed document dependent weighting factor facility for Naïve Bayes classification. *Expert Systems with Applications*, *37*(12), 8471–8478.

Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*(12), 11303–11311.

Loh, W.-Y., & Shih, Y.-S. (1997). Split Selection Methods for Classification Trees. *Statistica Sinica*, *7*(4), 815–840.

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*. *Data Mining and Knowledge Discovery Handbook* (2nd ed.). New York: Springer-Verlag.

Ngai, E. W. T., Xiu, L., & Chau, D. C. K. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, *36*, 2592–2602.

Polat, K., & Gunes, S. (2009). A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications*, *36*, 1587–1592.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. California: Morgan Kaufmann.

Safavian, S. R., & Landgrebe, D. (1991). A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3).

Turney, P. (1995). Cost-Sensitive Classification : Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research*, *2*, 369–409.

Utgoff, P. E. (1989). Incremental Induction of Decision Trees. *Machine Learning*, *4*(2), 161–186.

Witten, I. H., Eibe, F., & Hall, M. A. (2011). *Data mining : practical machine learning tools and techniques.—3rd ed*. *Morgan Kaufmann* (3rd ed.). Morgan Kaufmann.

BIOGRAFI PENULIS

Al Riza Khadafy. Memperoleh gelar S.Kom pada jurusan Sistem Informasi dari STMIK Nusa Mandiri, Jakarta dan gelar M.Kom pada jurusan Ilmu Komputer dari Pascasarjana STMIK Nusa Mandiri Jakarta. Bekerja sebagai staff IT di perusahaan swasta di Jakarta. Minat penelitian pada saat ini meliputi bidang *data mining* dan *machine learning*.

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Pendiri dan CEO PT Brainmatics, perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiai ilmiah ACM, PMI dan IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Comparative Analysis of Mamdani, Sugeno And Tsukamoto Method of Fuzzy Inference System for Air Conditioner Energy Saving

Aep Saepullah
Faculty of Enginering, Information Technology Departement, University of Muhammadiyah Tangerang
mister.aep@gmail.com

Romi Satria Wahono
Faculty of Computer Science, Dian Nuswantoro University
romi@romisatriawahono.net

Abstract: Air Conditioner (AC) nowadays is one of the electrical equipment commonly used in human daily life to reduce the heat, especially for communities who live in the hot weather area. But in the other side, air conditioner usage has a shortage such as a huge electrical energy consumption of air conditioning and it reach 90% of the total electrical energy that was needed by a household, and that especially happen when operated at the peak load electricity time or around 17:00 until 22:00, and it will cause a deficit of power supplies for use by other household appliances. In this paper will be conducted analysis and comparison between Mamdani, Sugeno and Tsukamoto method on fuzzy inference systems to find a best method in terms of reduction in electrical energy consumption of air conditioner by using Room Temperature and Humidity as input variables and Compressor speed as output variable. In this research, experiments was performed by using crisp input of room temperature 11^{O}C, 21% humidity, room temperature 14^{O}C, 41% humidity, room temperature 27^{O}C, 44% humidity and room temperature 33^{O}C, 68% humidity. The results of experiments showed that the best method in terms of reduction in electrical energy consumption of air conditioning system is a method of Tsukamoto where the average electrical energy efficiency achieved by 74,2775%.

Keywords: Air Conditioner, Energy Saving, Fuzzy Inference System, Best Method.

1 INTRODUCTION

Air conditioner known as one electrical equipment commonly used in human daily life to reduce heat, especially for communities who live in the hot weather area. Air conditioner use to change the room air temperature to make people feel comfortable because the air conditioner able to change the temperature it self and humidity due to user desire.

The Indonesian electricity energy presently not sufficient to support for all the human being activity, it can proof by commonly rotating blackouts happen in some area in Indonesia, so that realy necessary to save energy and use energy efficiently as possible (Sudirman, 2011). Due to the data who has been release by Indonesia Ministry of Energy and Mineral Resources (PUSDATIN KESDM, 2011) show that the electrical energy consumption of household appliance in 2011 reach 59.309 GWh is equal approximately 41,1% from total of 148.359 GWh, where the consumption of air conditioner it self take 40 % from all electrical energy that supply for household scope.

Reducing the use of electrical energy and providing comfort room (optimal temperature) are two important considerations in the design of air conditioning systems (Nasution, 2008). Proper cooling load claulations (SNI, 2000) will be able to ensure as much as posible the attention of energy saving opportunities at the planning stage.

Air conditioner energy saving research has been made with fuzzy logic Mamdani model implementation and optimized by genetic algorithm achieve energy saving for 31,5% (Parameshwaran, Karunakaran, Iniyan, & Samuel, 2008) , and further research mention that air conditioner using wich controlled by fuzzy logic control Sugeno model and optimized by genetic algorithm achieve energy saving for 23,8% with temperature set in 23^{0}C. (Wang, 2009)

Some other research claim that fuzzy logic control with Sugeno model using on energy saving of air conditioner energy consumption are better than proportional integral derivative (PID) control, by performing measurements over a periode of two hours achieve energy saving for 22,97% with temperature set at 20^{0}C (Nasution, 2011). further some research about comparison between Mamdani model of fuzzy logic and neuro fuzzy for control the air conditioner to obtain energy saving achieve 20% of energy saving by using fuzzy logic and achieve 40% of energy saving by using neuro fuzzy to control the air conditioner (Kaur & Kaur, 2012), in this study also expressed that fuzzy logic can be used for the process of setting up a non linear or hard to do with conventional systems (Kaur & Kaur, 2012). Fuzzy logic also allows the implementation arrangements in accordance with the feeling that possessed by humans (Kaur & Kaur, 2012).

2 FUZZY LOGIC

Fuzzy theory was first introduced by Dr. Lotfi Zadeh in 1965 from the University of California, to develop qualitative concept that has no precise boundaries, for example there is no clear or definite value that represents the boundary between normal and low, normal or high and (Sivanandam, Sumathi, & Deepa, 2007). Fuzzy logic is an appropriate way to map an input space into a space of output, similar to the black box to do something to compute a solution, the value of output (Prats, 2001).

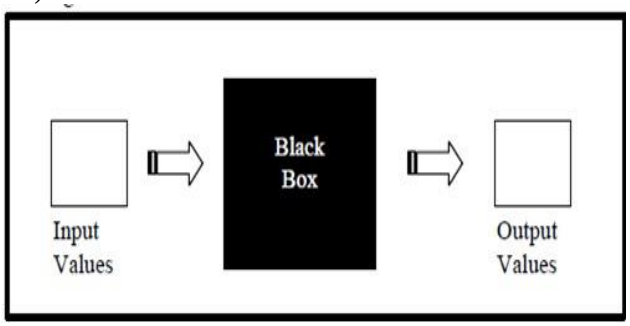


Figure 1 Input – Output System

Copyright @ 2015 IlmuKomputer.Com

Fuzzy set has 2 atribute, wich is:
a. Linguistic.
b. Numeric.

Some few things need to know for understanding fuzzy system is:
a. Fuzzy variable.
b. Fuzzy sets.
c. Universe of discourse.
d. Domain

This study used fuzzy inference system specification as follows:
a. Room Temperature variable divided into four sets, wich is:
Very low = [0 15]
Low = [10 30]
High = [25 35]
Very high = [30 45]

Membership function for room temperature as input variable is on figure 2.

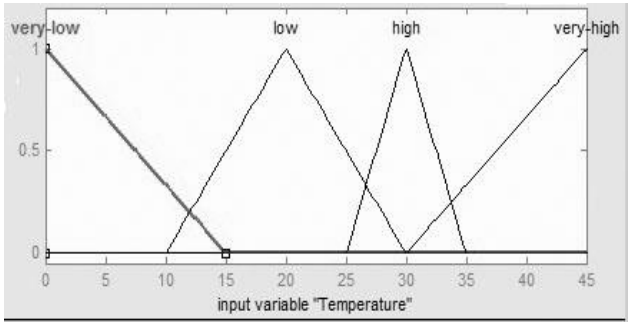


Figure 2. Room Temperature Variable

b. Humidity variable divided into four sets, wich is:
Dry = [0 30]
Comfortable = [20 50]
Humid = [40 70]
Sticky = [60 100]

Membership function for Humidity as input variable is on figure 3.

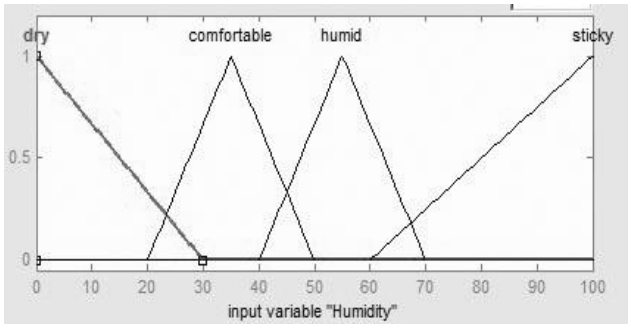


Figure 3. Humidity Variable

The output variable of Mamdani and Tsukamoto model is:
a. Compressor Speed variable wich divided into four sets, namely:
Off = [0]
Low = [30 60]
Medium = [50 80]
Fast = [70 100]

Membership function for Compressor speed Humidity as output variable is shown on figure 4 .

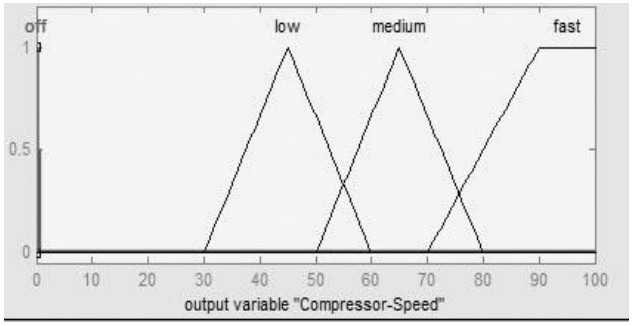


Figure 4. Compressor Speed Variable

The Sugeno model output variable, fuzzy sets and domains is shown in table 1

Table 1 Sugeno Model Output Variable

Compressor Speed	Constant Value
Off	0
Low	0.3333
Medium	0.6667
Fast	1

3 FUZZIFICATION

In this study used 4 times experiments using sampling data as shown in table 2

Table 2 Data Sampling

Room Temperature	Humidity
11^{O}C	21%
14^{O}C	41%
27^{O}C	44%
33^{O}C	68%

Fuzzification for Room Temperature variable with crisp input 11^{O}C shown on figure 5:

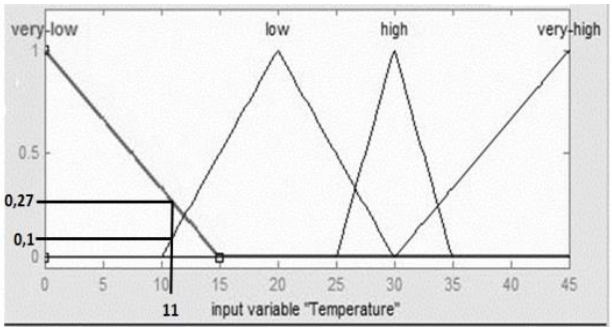


Figure 5. Room Temperature Fuzzification With Crisp Input 11^{O}C

$$\mu[x] = \begin{cases} 0; & x \le a, x \ge c \\ (x-a)/(b-a) & a < x \le b \\ (c-x)/(c-b) & b < x \le c \end{cases}$$

$$\mu\text{Very Low}[x] = \begin{cases} 0; & x \ge 15 \\ (15-11)/(15-0) & 0 < x \le 15 \end{cases}$$

$$\mu\text{Very Low}[x] = 0.27$$

$$\mu\text{Low}[x] = \begin{cases} 0; & x \le 10 \\ (11-10)/(20-10) & 10 < x \le 20 \end{cases}$$

$$\mu\text{Low}[x] = 0.1$$

Copyright @ 2015 IlmuKomputer.Com

Fuzzification for Humidity variable with crisp input 21% shown on figure 6:

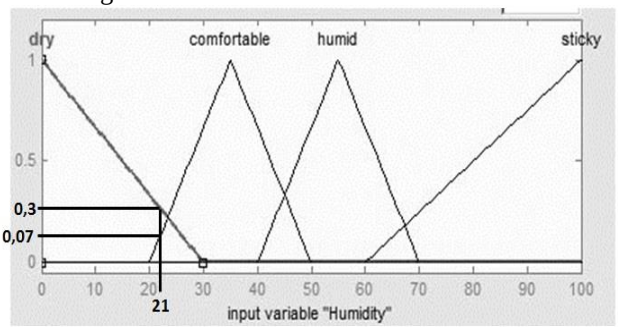


Figure 6. Humidity Fuzzification With Crisp Input 21%

$$\mu[x] = \begin{cases} 0; & x \le a, x \ge c \\ (x-a)/(b-a) & a \le x \le b \\ (c-x)/(c-b) & b \le x \le c \end{cases}$$

$$\mu Dry[x] = \begin{cases} 0; & x \ge 30 \\ (30-21)/(30-0) & 0 \le x \le 30 \end{cases}$$

$$\mu Dry[x] = 0.3$$

$$\mu Comportable[x] = \begin{cases} 0; & x \le 20 \\ (21-20)/(35-20) & 20 \le x \le 35 \end{cases}$$

$$\mu Comportable[x] = 0.07$$

4 INFERENCE

Rule base used in this study shown in table 3

Tabel 3 Rule Base

Humidity / Room Temperature	Dry	Comfortable	Humid	Sticky
Very Low	Off	Off	Off	Low
Low	Off	Off	Low	Medium
High	Low	Medium	Fast	Fast
Very High	Medium	Fast	Fast	Fast

5 DEFUZZIFICATION

A. TSUKAMOTO METHOD

The experiment results of Tsukamoto method using crisp input of room temperature 14^{O}C and 41% of humidity, the motor speed reaches 5,63%, so it can be conclude that the motor speed only take 81,635 Rpm with the energy saving achieves approximately 94,37.

A. Rule 1

If Room Temperature Very Low and Humidity Comfortable then Speed Off

α_1 = μTempVeryLow ∩ μHumComfortable
= min(μTempVeryLow[11], HumComfortable[21])
= min (0,07; 0,6)
= 0,07

If Room Temperature Very Low (0,07) and Humidity Comfortable (0,6) then Speed Off (0,07)

Z_1 = (Z-0) / 0 = 0,07
Z_1 = 0

B. Rule 2

If Room Temperature Very Low and Humidity Humid then Speed Off

α_2 = μTempVery Low ∩ μHumHumid
= min (μTempVery Low[14], μHumHumid[41])
= min (0,07; 0,06)
= 0,06

If Room Temperature Very Low (0,07) and Humidity Humid (0,06) then Speed Off (0,06)

Z_2 = (Z-0) / 0 = 0,06
Z_2 = 0

C. Rule 3

If Room Temperature Low and Humidity Comfortable then Speed Off

α_3 = μTempLow ∩ μHumComfortable
= min (μTempLow[14], μHumComfortable[41])
= min (0,3; 0,6)
= 0,3

If Room Temperature Low (0,3) and Humidity Comfortable (0,6) then Speed Off (0,3)

Z_3 = (Z-0) / 0 = 0,3
Z_3 = 0

D. Rule 4

If Room Temperature Low and Humidity Humid then Speed Low

α_4 = μTempLow ∩ μHumHumid
= min (μTempLow[14], μHumHumid[41])
= min (0,3; 0,06)
= 0,06

If Room Temperature Low (0,3) and Humidity Humid (0,06) then Speed Low (0,06)

Z_4 = (Z-45) / 15 = 0,06
Z_4 = 45,99

$$\frac{\propto 1 * Z1 + \propto 2 * Z2 + \propto 3 * Z3 + \propto 4 * Z4}{\propto 1 + \propto 2 + \propto 3 + \propto 4}$$

$$\frac{0,07 * 0 + 0,06 * 0 + 0,3 * 0 + 0,06 * 45,99}{0,07 + 0,06 + 0,3 + 0,06}$$

$$= \frac{0 + 0 + 0 + 2,7594}{0,49}$$

$$= 5,63$$

B. SUGENO METHOD

The experiment of Sugeno method as shown in figure 7 with crisp input of Room Temperature 14^{O}C and Humidity 41% the compressor speed reaches 3,7 %. It can be concluded that the motor speed take approximately 53,65 Rpm with energy saving achieves approximately 96,3 %, the results of calculations using Sugeno method showed better results than calculations using the Tsukamoto.

Copyright @ 2015 IlmuKomputer.Com

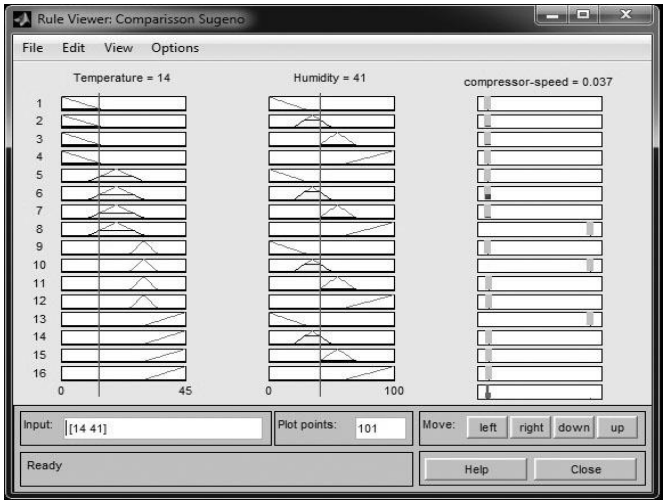


Figure 7. Sugeno Method Defuzzification With Crisp Input Of Room Temperature Is 14ºc And Humidity Is 41%

C. MAMDANI METHOD

The experiment of Mamdani method as shown in figure 8 with crisp input of Room Temperature 14^{O}C and Humidity 41% the compressor speed reaches 37,3 %. It can be concluded that the motor speed take approximately 540,85 Rpm with energy saving achieves approximately 62,7%.

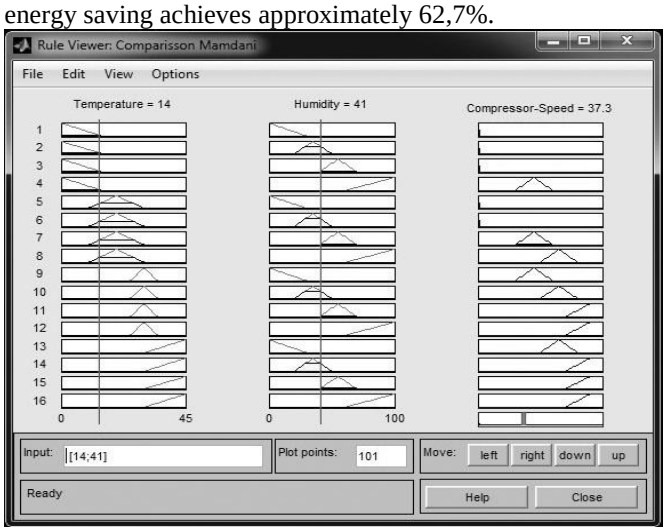


Figure 8. Mamdani Method Defuzzification With Crisp Input Of Room Temperature Is 14^{O}C And Humidity Is 41%

Due to compressor motor that was design for dynamicly, then to obtain a valid result all of the simulation results will be added together to find the average value of the results taken from the output variable to find wich method are better according to the simulations. All of the simulations results are shown in table 4

Table 4 Simulation Results

Method / Crisp Input	Mamdani	Sugeno	Tsukamoto
Room Temperature 11^{O}C Humidity 21%	0 %	0 %	0 %
Room Temperature 14^{O}C Humidity 41%	37,3 %	3,7 %	5,63 %
Room Temperature 27^{O}C Humidity 44%	65,1 %	26,2 %	41,9 %
Room Temperature 33^{O}C Humidity 68%	86,2 %	100 %	55,36 %
Average	47,15 %	32,475 %	25,7225 %
Efficiency	52,85 %	67,525 %	74,2775 %

From the above results can be calculated by the amount of electrical energy reduction calculation as shown in Table 5

Table 5 Calculation Of The Electrical Energy Consumption Of Air Conditoning

Method	Motor rotation averages (Rpm)	Cooling energy consumption averages (Kwh)
Tsukamoto	372,97625 Rpm	29,125 Kwh
Sugeno	470,8875 Rpm	36,771 Kwh
Mamdani	683,675 Rpm	53,387 Kwh

6 CONCLUSION

Based on experiments conducted in this study it can be concluded that the method of Mamdani, Sugeno and Tsukamoto proved to be used in air-conditioning for the reduction of electrical energy consumption, but the results are varied. The results of the first experiment using a room temperature setting 27^{O}C and humidity of 44% with a 34.9% yield, calculated by using Mamdani shows results that are much smaller in terms of energy savings when compared with calculations using the Tsukamoto with a level of energy savings 58.099 % and Sugeno method with the level of energy savings of 73.8%.

The results of the second experiment using the settings 33^{O}C room temperature and humidity of 68% calculated by using Mamdani shows energy savings rate of 13.8%, calculated by using Mamdani shows results that are much smaller in terms of energy savings when compared to calculations using Tsukamoto method with the level of energy savings of 31.176% and Mamdani method is better than the Sugeno method with high energy savings at 0%. From the three methods were compared, the best method in terms of reduction of electrical energy consumption is Tsukamoto method with average savings of 74.2775%.

REFERENCES

Armendariz, J., Ortega-Estrada, C., Mar-Luna, F., & Cesaretti, E. (2013). Dual-Axis Solar Tracking Controller Based on Fuzzy-Rules Emulated Networks and Astronomical Yearbook Records. *Proceedings of the World Congress on Engineering*.

Bendib, T., Djeffal, F., Arar, D., & Meguellati, M. (2013). Fuzzy-Logic-based Approach for Organic Solar Cell Parameters

Copyright @ 2015 IlmuKomputer.Com

Extraction. *Proceedings of the World Congress on Engineering*.

Ibrahim, M., & Ibrahim, H. (2012). Comparison Between, Fuzzy and P&O Control for MPPT for Photovoltaic System Using Boost Converter. *Journal of Energy Technologies and Policy*.

Kaur, A., & Kaur, A. (2012). Comparison Of Fuzzy Logic And Neuro Fuzzy Algorithms For Air Conditioning System. *International Journal of Soft Computing and Engineering (IJSCE)*.

Kaur, A., & Kaur, A. (2012). Comparison of Mamdani Fuzzy model and Neuro fuzzy model for air conditioning systems. *International Journal of Computer Science and Information Technologies (IJCSIT)*.

Kaur, A., & Kaur, A. (2012). Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference System For Air Conditioning System. *International Journal of Soft Computing and Engineering*.

Nasution, H. (2008). Development of a Fuzzy Logic Controller Algorithm for Air-conditioning System. *Telkomnika*.

Nasution, H. (2011). Development of a Fuzzy Logic Controller Algorithm for Air-conditioning System. *Telkomnika*.

Parameshwaran, R., Karunakaran, R., Iniyan, S., & Samuel, A. A. (2008). Optimization of Energy Conservation Potential for VAV Air Conditioning System using Fuzzy based Genetic Algorithm. *International Journal of Engineering and Natural Sciences (IJNES)*.

Prats, P. J. (2001). *Development And Testing Of A Number Of Matlab Based Fuzzy System Applications.* Warwick: School of Engineering, University of Warwick.

PUSDATIN KESDM. (2011). *Statistik Listrik.* Jakarta: Kementerian Energi Dan Sumber Daya Mineral.

Sheraz, M., & Abido, M. (2013). An Efficient Fuzzy Logic Based Maximum Power point Tracking Controller for Photovoltaic Systems. *International Conference on Renewable Energies and Power Quality*. ICREPQ.

Sivanandam, S. N., Sumathi, S., & Deepa, S. N. (2007). *Introduction to Fuzzy Logic Using MATLAB.* Verlag Berlin Heidelberg: Springer.

SNI. (2000). *Nomor 03-6390-2000.* Badan SNI.

Sudirman. (2011). Pengaruh Fuzzy Logic Control Dibandingkan Dengan Kontrol Konvensional Terhadap Konsumsi Energi Listrik Pada Air Conditioning. *International Journal of Engineering and Natural Sciences*, 171-176.

Usta Ö , M., Akyazi, & Altas , İ. (2011). Design and Performance of Solar Tracking System with Fuzzy Logic Controller. *International Advanced Technologies Symposium*.

Wang, F. e. (2009). Evaluation And Optimization Of Air Conditioner Energy Saving Control Considering Indoor Thermal Comfort. *Eleventh International IBPSA Conference.* Glasgow, Scotland: IBPSA.

BIOGRAPHY OF AUTHORS

Aep Saepullah, Received Bachelor of Computer Science degrees in Software Engineering from Raharja School of Computer Science, and Master of Computer Science degrees in Software Engineering respectively from Eresha School of Computer Science. He is curently a lecturer at the Faculty of Engineering, majoring in Information Technology Departement at University of Muhammadiyah Tangerang. His research interest is in Inteligent System, Computer Security and Data Mining.

Romi Satria Wahono. Received B.Eng and M.Eng degrees in Software Engineering respectively from Saitama University, Japan, and Ph.D in Software Engineering and Machine Learning from Universiti Teknikal Malaysia Melaka. He is a lecturer at the Faculty of Computer Science, Dian Nuswantoro University, Indonesia. He is also a founder and CEO of Brainmatics, Inc., a software development company in Indonesia. His current research interests include software engineering and machine learning. Professional member of the ACM, PMI and IEEE Computer Society.

Copyright @ 2015 IlmuKomputer.Com

Penanganan Fitur Kontinyu dengan *Feature Discretization* Berbasis *Expectation Maximization Clustering* untuk Klasifikasi *Spam Email* Menggunakan Algoritma ID3

Safuan, Romi Satria Wahono, dan Catur Supriyanto
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
soft_sfn@yahoo.com, romi@romisatriawahono.net,caturs@research.dinus.ac.id

Abstrak: Pemanfaatan jaringan internet saat ini berkembang begitu pesatnya, salah satunya adalah pengiriman surat elektronik atau *email.* Akhir-akhir ini ramai diperbincangkan adanya *spam email*. *Spam email* adalah email yang tidak diminta dan tidak diinginkan dari orang asing yang dikirim dalam jumlah besar ke *mailing list*, biasanya beberapa dengan sifat komersial. Adanya *spam* ini mengurangi produktivitas karyawan karena harus meluangkan waktu untuk menghapus pesan *spam*. Untuk mengatasi permasalahan tersebut dibutuhkan sebuah filter email yang akan mendeteksi keberadaan spam sehingga tidak dimunculkan pada *inbox mail.* Banyak peneliti yang mencoba untuk membuat filter email dengan berbagai macam metode, tetapi belum ada yang menghasilkan akurasi maksimal. Pada penelitian ini akan dilakukan klasifikasi dengan menggunakan algoritma *Decision Tree Iterative Dicotomizer 3* (ID3) karena ID3 merupakan algoritma yang paling banyak digunakan di pohon keputusan, terkenal dengan kecepatan tinggi dalam klasifikasi, kemampuan belajar yang kuat dan konstruksi mudah. Tetapi ID3 tidak dapat menangani fitur kontinyu sehingga proses klasifikasi tidak bisa dilakukan. Pada penelitian ini, *feature discretization* berbasis *Expectation Maximization* (EM) *Clustering* digunakan untuk merubah fitur kontinyu menjadi fitur diskrit, sehingga proses klasifikasi *spam email* bisa dilakukan. Hasil eksperimen menunjukkan ID3 dapat melakukan klasifikasi *spam email* dengan akurasi 91,96% jika menggunakan data training 90%. Terjadi peningkatan sebesar 28,05% dibandingkan dengan klasifikasi ID3 menggunakan *binning*.

Kata kunci: Klasifikasi, *Spam email*, ID3, *Feature Discretization, Expectation Maximization Clustering*

1 PENDAHULUAN

Pemanfaatan jaringan internet saat ini berkembang begitu pesatnya, salah satunya adalah pengiriman surat atau pesan. Jalur internet sudah menggantikan pengiriman surat konvensional menjadi surat elektronik atau *email.* Dengan menggunakan *email*, pengiriman pesan dapat dilakukan dengan cepat antar negara di seluruh dunia. Akhir-akhir ini ramai diperbincangkan adanya *spam email*. *Spam* adalah email yang tidak diminta dan tidak diinginkan dari orang asing yang dikirim dalam jumlah besar ke *mailing list*, biasanya dengan beberapa sifat komersial dan dikirim dalam jumlah besar (Saad, Darwish, & Faraj, 2012). Beberapa berpendapat bahwa definisi ini harus dibatasi untuk situasi di mana penerima memilih untuk menerima email ini misalnya mencari pekerjaan atau mahasiswa penelitian yang sedang melakukan penelitian.

Menurut sebuah laporan yang diterbitkan oleh McAfee Maret 2009 (Allias, Megat, Noor, & Ismail, 2014), biaya kehilangan produktivitas per hari untuk pengguna kira-kira sama dengan $0,50. Hitungan ini berdasarkan dari aktivitas pengguna yang menghabiskan 30 detik untuk menangani dua pesan *spam* setiap hari. Oleh karena itu, produktivitas per karyawan yang hilang per tahun karena *spam* kira-kira sama dengan $182,50.

Dengan adanya masalah *spam* tersebut, ada banyak literatur yang diusulkan untuk menyaring *spam email*. Saadat Nazirova *et al.* (Nazirova, 2011) membagi filter *spam* berdasarkan teknik filternya menjadi 2 kategori yaitu metode untuk mencegah penyebaran spam dan metode untuk mencegah penerimaan spam. Metode pencegahan penyebaran spam diantaranya adalah peraturan pemerintah yang membatasi distribusi spam, pengembangan protokol email menggunakan otentikasi pengirim dan pemblokiran server email yang mendistribusikan spam. Metode pencegahan penerimaan spam dibagi menjadi 2 kategori yaitu penyaringan dengan menggunakan pendekatan teori dan berdasarkan area filtrasi (dari sisi server dan pengguna).

Salah satu cara penyaringan *spam* adalah menggunakan pendekatan teori berbasis pembelajaran. Beberapa penelitian yang pernah dilakukan dengan menggunakan algoritma klasifikasi berbasis pembelajaran adalah *Naïve Bayes* (NB) (Çiltik & Güngör, 2008) (Marsono, El-Kharashi, & Gebali, 2008), *Support Vector Machine* (SVM) (Sculley & Wachman, 2007), *Artifial Neural Networking* (ANN) (Wu, 2009), *Logistic Regression* (LR) (Jorgensen, Zhou, & Inge, 2008), *K-Nearest Neigbor* (KNN) (Méndez, Glez-Peña, Fdez-Riverola, Díaz, & Corchado, 2009) dan *Decission Tree* (Sheu, 2009).

Klasifikasi adalah proses menemukan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep, dengan tujuan menggunakan model tersebut supaya mampu memprediksi kelas objek dimana label kelasnya tidak diketahui (Han & Kamber, 2006). Di bidang klasifikasi, ada banyak cabang yang berkembang yaitu pohon keputusan, klasifikasi Bayesian , jaringan saraf dan algoritma genetika (Tsai, Lee, & Yang, 2008). Di antara cabang tersebut, pohon keputusan telah menjadi alat yang popular untuk beberapa alasan: (a) dibandingkan dengan jaringan saraf atau pendekatan berbasis bayesian, pohon keputusan lebih mudah diinterpretasikan oleh manusia; (b) lebih efisien untuk data pelatihan yang besar dibanding dari jaringan saraf yang akan memerlukan banyak waktu pada ribuan iterasi; (c) algoritma pohon keputusan tidak memerlukan pengetahuan domain atau pengetahuan sebelumnya; dan, (d) akan menampilkan akurasi klasifikasi lebih baik dibandingkan dengan teknik lain.

Aman Kumar Sharma *et al.* (Sharma, 2011) membandingkan akurasi empat algoritma *Decision Tree* yaitu *Iterative Dichotomiser 3* (ID3), J48, *Simple Classification And Regression Tree* (CART) and *Alternating Decision Tree* (ADTree). CART menunjukkan hasil yang hampir sama dengan J48. ADTree dan ID3 menunjukkan akurasi kecil dibandingkan dengan CART dan J48. Hal ini menunjukkan bahwa algoritma J48 lebih disukai dibanding CART, ADTree dan ID3 dalam klasifikasi email spam yang mana ketepatan klasifikasi menjadi sangat penting.

Chakraborty *et al.*(Chakraborty & Mondal, 2012) melakukan penelitian dengan membandingkan dan

Copyright @ 2015 IlmuKomputer.Com

menganalisa tiga jenis teknikklasifikasi pohon keputusan yaitu *Naïve Bayes Tree* (NBT), C 4.5 (atau J48) dan *Logistik Model Tree* (LMT) untuk filtrasi spam. Hasil eksperimen menunjukkan bahwa LMT mempunyai akurasi sekitar 86% dan tingkat *false positif* jauh lebih rendah dari NBT dan J48. NBT membutuhkan waktu pelatihan tertinggi di antara semua klasifikasi pohon keputusan yang diteliti tapi memiliki *false positive rate* diantara J48 dan LMT. J48 memerlukan waktu pelatihan dan jumlah waktu berjalan paling sedikit di antara NBT dan LMT pada dataset yang sama.

Pada penelitian ini akan digunakan algoritma *decision tree* ID3 karena algoritma ini lebih baik dibanding algoritma *Decision Tree* yang lain seperti C4.5, CHAID dan CART (Sheu, 2009). Algoritma ID3 paling banyak digunakan di pohon keputusan (Jin & De-lin, 2009), terkenal dengan kecepatan tinggi dalam klasifikasi, kemampuan belajar yang kuat dan konstruksi mudah (Liu Yuxun & Xie Niuniu, 2010). Tetapi ID3 mempunyai kelemahan yaitu tidak dapat mengklasifikasikan fitur kontinyu dalam dataset (Jearanaitanakij, 2005) (Al-Ibrahim, 2011) sehingga proses klasifikasi tidak dapat dilakukan.

ID3 dirancang untuk menangani data pelatihan dengan nilai atribut diskrit dan simbolik (Al-Ibrahim, 2011). Untuk mengatasi masalah fitur kontinyu, *feature discretization* (FD) telah diusulkan dengan tujuan memperoleh representasi dari dataset yang lebih memadai untuk pembelajaran. Penggunaan teknik ini telah terbukti baik untuk meningkatkan akurasi klasifikasi dan menurunkan penggunaan memori (Ferreira & Figueiredo, 2012). FD berfungsi untuk merubah fitur kontinyu (real) menjadi fitur diskret (Dash, Paramguru, & Dash, 2011) (Madhu, Rajinikanth, & Govardhan, 2014), membagi nilai menjadi interval yang lebih kecil (Senthilkumar, Karthikeyan, Manjula, & Krishnamoorthy, 2012) (Al-Ibrahim, 2011) dan meningkatkan performa algoritma (Wijaya, 2008) sehingga lebih cocok digunakan untuk menangani masalah atribut / fitur kontinyu pada ID3.

Ferreira *et al.* (Ferreira & Figueiredo, 2014) (Ferreira & Figueiredo, 2012) membagi FD menjadi 2 kategori yaitu *supervised* dan *unsupervised*. Kategori *supervised* terdiri dari beberapa teknik misalnya *information entropy minimization* (IEM), ChiSquare, *bayesian belief networks* (BBN) dan *class-attribute interdependence maximization* (CAIM). Sedangkan kategori *unsupervised* terdiri dari *equal-interval binning* (EIB), *equal-frequency binning* (EFB) dan *proportional k-interval discretization* (PkID).

Gennady Agre *et al.* (Agre & Peev, 2002) dalam penelitiannya membandingkan 2 metode diskritisasi yaitu *entropy based discretization* Fayyad dan Irani *(supervised)* dengan *equal width binning* dan *equal frequency binning (unsupervised)* menggunakan 2 algoritma mesin pembelajaran *Simple Bayesian Classifier* (SBC) dan *Symbolic Nearest Mean Classifier* (SNMC). Hasil eksperimen menunjukkan bahwa dua metode diskritisasi *unsupervised* mempunyai performa lebih baik (terutama *equal frequency binning*) daripada metode *supervised entropy based discretization* yang diusulkan oleh Fayyad dan Irani.

Ankit Gupta *et al.*(Gupta, Mehrotra, & Mohan, 2010) membandingkan 2 metode diskritisasi berbasis *clustering* yaitu *Shared Nearest Neighbor* (SNN) dan K-Means yang digabung dengan *minimum entropy–maximum description length* (ME–MDL) menggunakan 3 algoritma klasifikasi *supervised* yaitu NB, SVM dan *Maximum Entropy*. Berdasarkan percobaan pada 11 dataset yang diamati, jika jumlah cluster yang diinginkan adalah sama dengan jumlah kelas atau jumlah kelas + 1, maka kinerja klasifikasi lebih baik dengan menggunakan ME-MDL. K-Means memberikan kinerja yang lebih baik dari SNN.

Penelitian yang dilakukan oleh Yong Gyu Junga et al (Jung, Kang, & Heo, 2014) membandingkan kinerja dari algorithma K-means and *Expectation Maximization* (EM). Dari percobaan yang telah dilakukan menunjukkan bahwa kecepatan pemrosesan K-means lebih lambat dibanding EM, tapi akurasi klasifikasi data adalah 94,7467% yang merupakan 7,3171% lebih baik dari yang didapat oleh EM. Tentu, ketidaktelitian dari K-means lebih rendah dibandingkan dengan yang ada pada algoritma EM. Secara keseluruhan, optimasi lebih lanjut harus diperkenalkan untuk mengurangi waktu.

Pada penelitian ini akan dilakukan proses klasifikasi *spam email* menggunakan algoritma ID3 dengan metode *feature discretization* berbasis EM *clustering*, karena EM dapat melakukan pengelompokan pada data yang mempunyai banyak rentang nilai yang berbeda secara signifikan (Ladysz, 2004) dan secara umum dapat diterapkan untuk fitur kontinyu dan kategori (I. Witten, 2011).

2 PENELITIAN TERKAIT

Salah satu masalah pada klasifikasi spam yaitu banyaknya atribut yang dihasilkan dari kata yang ada pada email. Banyak metode yang diusulkan untuk mengatasi masalah klasifikasi spam tersebut. Seperti penelitian yang dilakukan (Kumar, Poonkuzhali, & Sudhakar, 2012) yaitu dengan melakukan perbandingan pada beberapa algoritma data mining untuk klasifikasi spam. Algoritma klasifikasi yang dibandingkan adalah C4.5, C-PLS, C-RT, CS-CRT, CS-MC4, CS-SVC,ID3, K-NN, LDA, Log Reg TRILLS, Multi Layer Perceptron, Multilogical Logistic Regression, Naïve Bayes Continuous, PLS-DA, PLS-LDA, Random Tree dan SVM.Eksperimen dengan menggunakan fitur seleksi *fisher filtering*, *Relief filtering*, *Runs filtering* dan *Stepwise discriminant analysis.* Klasifikasi *Random Tree* dianggap sebagai pengklasifikasi terbaik, karena menghasilkan akurasi 99% melalui seleksi fitur *fisher filtering*.

Selain itu, penelitian (Chakraborty & Mondal, 2012) dilakukan dengan membandingkan dan menganalisa tiga jenis teknik klasifikasi pohon keputusan yang pada dasarnya pengklasifikasi data mining yaitu *Naïve Bayes Tree* (NBT), C 4.5 (atau J48) dan *Logistik Model Tree* (LMT) untuk filtrasi spam. Sebelum dataset diterapkan pada algoritma yang diuji, dilakukan *preprocessing* (pemrosesan awal) dengan menggunakan seleksi fitur. Hasil eksperimen menunjukkan bahwa LMT mempunyai akurasi sekitar 86% dan tingkat *false positif* jauh lebih rendah dari NBT dan J48.

Analisis komparatif dilakukan pada penelitian (Hamsapriya, T., 2012) dengan beberapa algoritma klasifikasi yaitu *Multilayer Perceptron* (MLP), J48 dan *Naïve Bayes* (NB). Hasil penelitian menunjukkan bahwa algoritma klasifikasi yang sama menghasilkan performa yang berbeda ketika dijalankan pada dataset yang sama tetapi menggunakan perangkat lunak yang berbeda. Selanjutnya teramati bahwa pada dataset ini untuk MLP menghasilkan tingkat kesalahan yang sangat baik dibandingkan dengan algoritma lain.

Penelitian yang dilakukan oleh (Gupta et al., 2010) dilakukan untuk mengatasi masalah diskritisasi dari variabel kontinyu untuk algoritma klasifikasi mesin pembelajaran. Teknik yang digunakan yaitu K-means *clustering* and *shared nearest neighbor* (SNN) *clustering* digabung dengan *minimum entropy–maximum description length* (ME–MDL) menggunakan 3 algoritma klasifikasi *supervised* yaitu NB, SVM dan *Maximum Entropy*. Hasil penelitian menunjukkan,

Copyright @ 2015 IlmuKomputer.Com

jika SVM digabungkan dengan SNN atau K-means pada jumlah cluster sama yang sama dengan jumlah kelas atau jumlah kelas + 1, hasilnya tidak lebih baik dari ME-MDL. Sulit untuk menilai algoritma *clustering* yang lebih baik karena di 7 dari 11 kasus K-means lebih baik daripada SNN.

Penelitian yang dilakukan oleh Chharia *et.al* untuk klasifikasi *spam email* dengan mengkombinasikan beberapa algoritma klasifikasi, menggunakan diversifikasi dengan mengatur fitur dan pengklasifikasi berbeda yaitu *Multinomial Naïve Bayes with Modified Absolute Discount Smoothing Method* (MNB-MAD), *Naïve Bayes* (NB), *Bagging, CART, C4.5, ADTree, Random Forest* (Rnd), *Functional Trees* (FT) *dan SimpleLogistis* (SL). Data yang digunakan adalah *SpamAssassin corpus* dan *Enron corpus*. Hasil penelitian menunjukkan, akurasi yang dicapai oleh metode ensamble yang diusulkan pada *Spamassasin corpus* sebesar 96,4% sedangkan pada pada *Enron corpus* sebesar 98,6%.

Penelitian yang dilakukan oleh Ali Al Ibrahim *et.al* dilakukan untuk mengatasi masalah fitur kontinyu untuk algoritma ID3. Teknik yang digunakan adalah *Continuous Inductive Learning Algorithm* (CILA), yaitu sebuah algoritma baru yang mengadopsi algoritma ID3 untuk mendiskrit fitur kontinyu yang dibuat oleh Ali Al Ibrahim. Hasil penelitian menunjukkan, CILA dapat secara otomatis memilih interval angka yang berbeda dengan teknik diskritisasi yang lain. Waktu yang dibutuhkan juga lebih singkat dibanding dengan metode diskrit *unsupervised* yang lain.

3 METODE YANG DIUSULKAN

Data yang digunakan pada penelitian ini bersumber pada database spam email yang bersumber dari UCI *repository of machine learning database. Spambase* terdiri dari terdiri dari total 4601 *e-mail*, dimana 1813 (39.4%) adalah *spam* dan 2788 (60.6%) adalah *non-spam*. Koleksi *spam email* berasal dari HP *email* dan *spam email* individu. Koleksi *non-spam email* berasal dari *email* kantor dan *email* perseorangan. Setiap *email* telah dianalisa dan terdapat 58 atribut (57 atribut input dan 1 atribut target atau kelas) yang menjelaskan tentang *spam email.* Rincian dari atribut tersebut adalah:

1. 48 atribut bertipe *continuous* dengan *range* 0-100 yang beranggotakan kata. Kata yang dimaksud antara lain:

Make	*address*	*all*	*3d*	*Our*	*Over*
Remove	*Internet*	*Order*	*mail*	*Receive*	*Will*
People	*Report*	*Addresses*	*Free*	*Business*	*Email*
You	*Credit*	*Your*	*Font*	*000*	*Money*
Hp	*Hpl*	*George*	*650*	*Lab*	*Labs*
telnet	*857*	*Data*	*415*	*85*	*Technology*
1999	*Parts*	*Pm*	*Direct*	*Cs*	*Meeting*
Original	*Project*	*Re*	*Edu*	*Table*	*Conference*

   Dengan prosentase:

$$\frac{\text{Jumlah kata yang muncul pada email}}{\text{Total kesuluruhan kata pada email}} \text{ x } 100\ \% \qquad (1)$$

2. 6 atribut bertipe *continuous* dengan *range* 0-100 yang beranggotakan karakter:
   ";" "(" "["
   "!" "$" "#"

   Dengan prosentase seperti pada persamaan (1).

3. 1 atribut *bertipe continous real* dengan nilai minimal 1, yang berisi rata-rata deret huruf kapital yang tidak bisa dipecahkan.
4. 1 atribut bertipe *continous real* dengan nilai minimal 1, yang berisi nilai terpanjang deret huruf kapital yang tidak bisa dipecahkan.
5. 1 atribut bertipe *continous real* dengan nilai minimal 1, yang berisi nilai jumlah deret huruf kapital yang tidak bisa dipecahkan.
6. 1 atribut bertipe *nomimal* dengan nilai 0 atau 1, yang berisi data target / kelas.

Metode klasifikasi yang diusulkan adalah menggunakan algoritma ID3 dengan diskrit fitur berbasis EM *clustering* untuk menangani fitur kontinyu pada dataset spam email. Evaluasi dilakukan dengan mengukur tingkat akurasi dan efisiensi.

Algoritma ID3 berusaha membangun *decision tree* (pohon keputusan) secara *top-down* (dari atas ke bawah) dengan mengevaluasi semua atribut yang ada menggunakan suatu ukuran statistik (yang banyak digunakan adalah *information gain*) untuk mengukur efektifitas suatu atribut dalam mengklasifikasikan kumpulan sampel data.

Untuk menghitung *information gain*, terlebih dahulu harus memahami suatu aturan lain yang disebut *entropy*. Di dalam bidang *Information Theory*, kita sering menggunakan *entropy* sebagai suatu parameter untuk mengukur heterogenitas (keberagaman) dari suatu kumpulan sampel data. Jika kumpulan sampel data semakin heterogen, maka nilai *entropy*-nya semakin besar. Secara matematis, *entropy* dirumuskan sebagai berikut:

$$Entropy = \sum_{i}^{c} -p_i \log_2 p_i \qquad (2)$$

dimana c adalah jumlah nilai yang ada pada atribut target (jumlah kelas klasifikasi). Sedangkan p_i menyatakan jumlah sampel untuk kelas *i*.

Setelah mendapatkan nilai entropy untuk suatu kumpulan sampel data, maka kita dapat mengukur efektifitas suatu atribut dalam mengklasifikasikan data. Ukuran efektifitas ini disebut sebagai *information gain*. Secara matematis, information gain dari suatu atribut A, dituliskan sebagai berikut:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (3)$$

dimana:

A	: atribut
V	: menyatakan suatu nilai yang mungkin untuk atribut *A*
Values(A)	: himpunan nilai-nilai yang mungkin untuk atribut *A*
|Sv|	: jumlah sampel untuk nilai *v*
|S|	: jumlah seluruh data
Entropy (Sv)	: entropy untuk sampel-sampel yang memiliki nilai *v*.

Secara ringkas, langkah kerja Algoritma ID3 dapat digambarkan sebagai berikut:

1. Penghitungan IG dari setiap atribut
2. Pemilihan atribut yang memiliki nilai IG terbesar
3. Pembentukan simpul yang berisi atribut tersebut
4. Ulangi proses perhitungan *information gain* akan terus dilaksanakan sampai semua data telah termasuk dalam kelas yang sama. Atribut yang telah dipilih tidak diikutkan lagi dalam perhitungan nilai IG.

Copyright @ 2015 IlmuKomputer.Com

Expectation Maximization (EM) adalah algoritma perbaikan iteratif populer yang dapat digunakan untuk menemukan perkiraan parameter (Han & Kamber, 2006). EM merupakan salah satu metode untuk menemukan estimasi *maximum likelihood* dari sebuah dataset dengan distribusi tertentu. EM termasuk algoritma *partitional* yang berbasiskan model yang menggunakan perhitungan probabilitas, bukan jarak seperti umumnya algoritma *clustering* yang lainnya (Chharia & Gupta, 2013). Jika pada algoritma K-means, parameter utamanya adalah *centroid*, maka untuk EM parameter utamanya adalah q_{mk} dan α_k untuk mendapatkan nilai r_{nk} yaitu probabilitas dokumen n masuk ke klaster k atau probabilitas klaster k beranggotakan dokumen n.
Langkah-langkah algoritma EM adalah sebagai berikut:

1. *Guess Model Parameter*
   Proses ini adalah melakukan penebakan nilai probabilitas data terhadap sebuah klaster. Langkah *guess* pertama adalah *guess probability* data klaster sebagai *model parame*ter. Inisialisasi nilai probabilitas pada data kata dilakukan secara random/ acak. Untuk probabilitas klaster, totalnya harus selalu bernilai 1.

Tabel 1*Guess Model Parameter*

Y (klaster)	X1	X2	X3	X4	P(Y)
0	0,1	0,3	0,8	0,8	0,7
1	0,2	0,3	0,1	0,1	0,2
2	0,7	0,4	0,1	0,1	0,1

Dimana pada tahap ini akan ditebak nilai parameter q_{mk} dan α_k.

2. *Expectation Step*

$$r_{nk} = \frac{\alpha_k(\prod_{t_m \in d_n} q_{mk})(\prod_{t_m \ni d_n}(1-q_{mk}) \ )}{\sum_{k=1}^{K} \alpha_k(\prod_{t_m \in d_n} q_{mk})(\prod_{t_m \ni d_n}(1-q_{mk}) \ )} \quad (4)$$

dimana:
- r_{nk} adalah nilai probabilitas setiap dokumen n terhadap masing-masing *cluster* atau nilai probabilitas *cluster* k terhadap sebuah dokumen
- $\alpha_k(\prod_{t_m \in d_n} q_{mk})(\prod_{t_m \ni d_n}(1-q_{mk}) \ )$adalah probabilitas total term terhadap sebuah klaster
- $\sum_{k=1}^{K} \alpha_k(\prod_{t_m \in d_n} q_{mk})(\prod_{t_m \ni d_n}(1-q_{mk}) \ )$adalah nilai total probabilitas semua term terhadap semua klaster.

Setelah r_{nk}didapat, maka akan dihitung *Frequency Counts*

$$\sum_{n=1}^{N} r_{nk} \, \mathrm{I}(t_m \in d_n) \quad (5)$$

3. *Maximization Step*

$$q_{mk} = \frac{\sum_{n=1}^{N} r_{nk} \mathrm{I}(t_m \in d_n)}{\sum_{n=1}^{N} r_{nk}} \quad (6)$$

dimana:
- q_{mk} adalah nilai probabilitas term *m* terhadap sebuah klaster dimana term *m* tersebut merupakan anggota dari suatu dokumen *n*.
- $\sum_{n=1}^{N} r_{nk} \, \mathrm{I}(t_m \in d_n)$ adalah *frequency Counts*, probabilitas klaster k terhadap semua dokumen yang mempunyai term m sebagai anggotanya (nilai *term* m = 1).
- $\sum_{n=1}^{N} r_{nk}$ adalah probabilitas sebuah *cluster* k terhadap semua dokumen.

Kemudian dihitung probabilitas sebuah klaster k:

$$\alpha_k = \frac{\sum_{n=1}^{N} r_{nk}}{N} \quad (7)$$

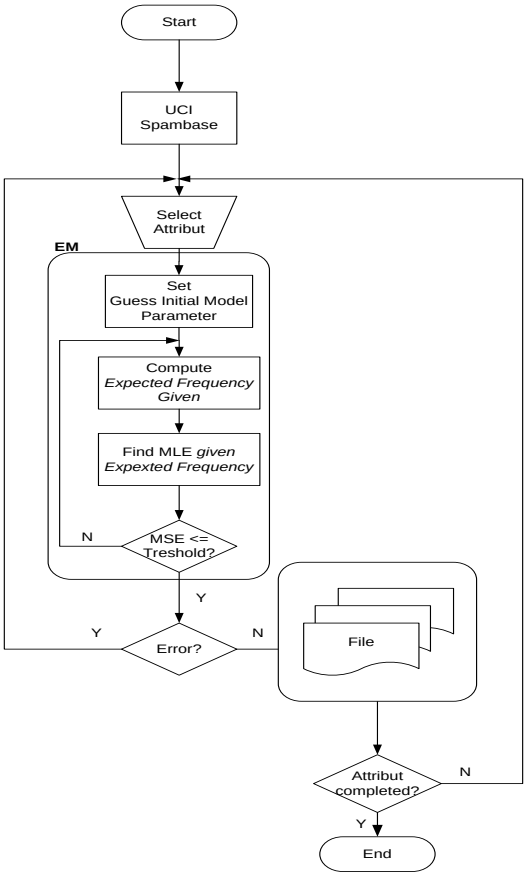
dimana *N* adalah probabilitas total klaster

4. Ulangi langkah 2 dan 3 sampai *Convergence*.
   Nilai probabilitas klaster data bersifat *Convergence* jika *update* probabilitas data terhadap klaster data tidak berubah-ubah lagi. Dengan kata lain nilai probabilitas dokumen terhadap sebuah klaster sudah bernilai 1.
   Langkah 1: Tentukan nilai *threshold*. Semakin kecil nilai *threshold* maka semakin dekat dengan *convergence*. Dalam hal ini nilai *threshold* nya adalah nol.
   Langkah 2: Hitung nilai *Means Square Error* dengan menggunakan rumus:

$$MSE(\vartheta) = E[(\vartheta - \theta)^2] \quad (8)$$

   Langkah 3: Bandingkan Nilai MSE dengan *threshold*
   Jika $MSE \leq threshold$ maka *convergence* dan iterasi berhenti.

Tahap *feature discretization* adalah memproses tiap fitur pada spambase dengan algoritma EM yang menghasilkan *output* sebuah file. Jumlah file yang terbentuk sesuai dengan jumlah fitur yang berhasil diproses oleh EM. Hasil proses ini digabung menjadi satu untuk diproses pada tahap klasifikasi. Proses *feature discretization* dapat dilihat pada Gambar 1.



Gambar 1 Proses *Feature Discretization*

Akurasi yang dihasilkan dihitung menggunakan *confusion matrix*. Perhitungan pada *confusion matrix* dihitung berdasarkan prediksi *positif* yang benar (*True Positif*), prediksi *positif* yang salah (*False Positif*), prediksi negatif yang benar (*True Negatif*) dan prediksi *negatif* yang salah (*False Negatif*).

$$Akurasi = \frac{TP + TN}{TP+FP+TN+FN} \quad (9)$$

Semakin tinggi nilai akurasinya, semakin baik pula metode yang dihasilkan

4 HASIL EKSPERIMEN

Eksperimen dilakukan dengan menggunakan komputer dengan spesifikasi processor Intel Celeron M560 2.13 GHz CPU, 2 GB RAM, dan sistem operasi Microsoft Windows 7 *Professional* 32-bit. *Software* yang digunakan adalah bahasa pemrograman PHP dan RapidMiner 5.2.

Tabel 2 Data UCI *Spambase*

word_freq_make	word_freq_address	word_freq_all	word_freq_3d	word_freq_our	word_freq_over	Class
0	0,64	0,64	0	0,32	0	1
0,21	0,28	0,5	0	0,14	0,28	1
0,06	0	0,71	0	1,23	0,19	1
0	0	0	0	0,63	0	1
0	0	0	0	0,63	0	1
0,3	0	0,3	0	0	0	0
0,96	0	0	0	0,32	0	0
0	0	0,65	0	0	0	0
0,31	0	0,62	0	0	0,31	0

Pada eksperimen ini, data yang digunakan adalah 4601 data email spambase dari UCI Machine *Learning repository* yang terdiri dari 57 atribut kontinyu dan 1 atribut nominal berisi data target/kelas. Nilai yang terdapat pada masing-masing fitur adalah prosentase munculnya kata dibandingkan dengan total keseluruhan data pada email. Sedangkan pada fitur kelas hanya ada 2 nilai yaitu 0 dan 1, 0 menunjukkan label *ham* (bukan *spam*) sedangkan label 1 menunjukkan label *spam* seperti terlihat pada Gambar1.

Pada tahap FDdata spambase diproses menggunakan algoritma EM *clustering* pada setiap fitur selain fitur kelas dan membentuk sebuah file. File yang dihasilkan kemudian digabungkan sehingga terbentuk sebuah file. Untuk lebih jelasnya dapt dilihat pada Tabel 3 dan Tabel 4.

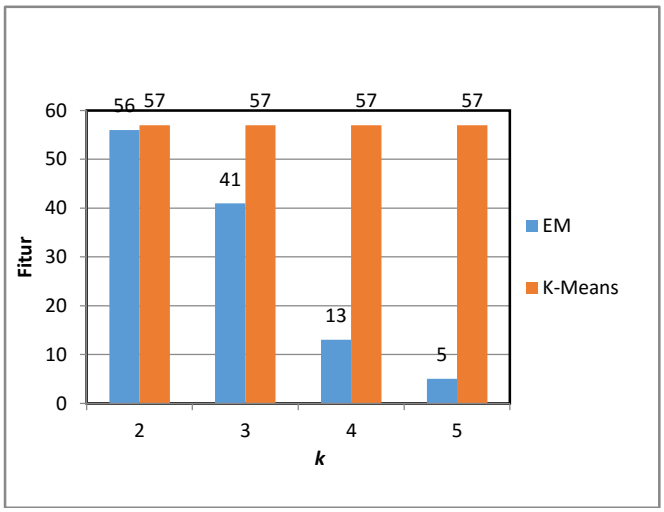
Tabel 3 Data Hasil FD

capital_run_length_average	cluster_0_probability	cluster_1_probability	cluster_2_probability	cluster_3_probability	cluster
3,8	,0	1,0	,0	,0	cluster_1
5,1	,0	,9	,0	,1	cluster_1
9,8	,0	,0	,0	1,0	cluster_3
3,5	,0	1,0	,0	,0	cluster_1
2,5	,7	,3	,0	,0	cluster_0
9,7	,0	,0	,0	1,0	cluster_3
1,7	,9	,1	,0	,0	cluster_0
4,7	,0	1,0	,0	,0	cluster_1

Tabel 4 Data Hasil Penggabungan

capital_ run_ length_ average	capital_ run_ length_ longest	capital_ run_ length_ total	Char _freq_ !	Class
cluster_1	cluster_1	cluster_2	cluster_2	Spam
cluster_1	cluster_1	cluster_2	cluster_0	Spam
cluster_3	cluster_2	cluster_3	cluster_0	Spam
cluster_1	cluster_1	cluster_1	cluster_0	Spam
cluster_1	cluster_1	cluster_1	cluster_0	Spam
cluster_1	cluster_0	cluster_0	cluster_1	Spam
cluster_0	cluster_0	cluster_1	cluster_0	Spam
cluster_0	cluster_0	cluster_0	cluster_1	Spam
cluster_0	cluster_0	cluster_0	cluster_0	Spam
cluster_0	cluster_1	cluster_1	cluster_0	Spam
cluster_0	cluster_0	cluster_0	cluster_1	Ham
cluster_0	cluster_0	cluster_0	cluster_2	Ham
cluster_0	cluster_0	cluster_0	cluster_1	Ham
cluster_0	cluster_0	cluster_0	cluster_1	Ham
cluster_0	cluster_0	cluster_1	cluster_1	Ham
cluster_0	cluster_0	cluster_0	cluster_0	Ham
cluster_0	cluster_0	cluster_1	cluster_1	Ham
cluster_0	cluster_0	cluster_1	cluster_1	Ham
cluster_0	cluster_0	cluster_1	cluster_0	Spam
cluster_0	cluster_0	cluster_0	cluster_1	Spam
cluster_0	cluster_0	cluster_0	cluster_0	Spam
cluster_0	cluster_1	cluster_1	cluster_0	Spam
cluster_0	cluster_0	cluster_1	cluster_1	Ham

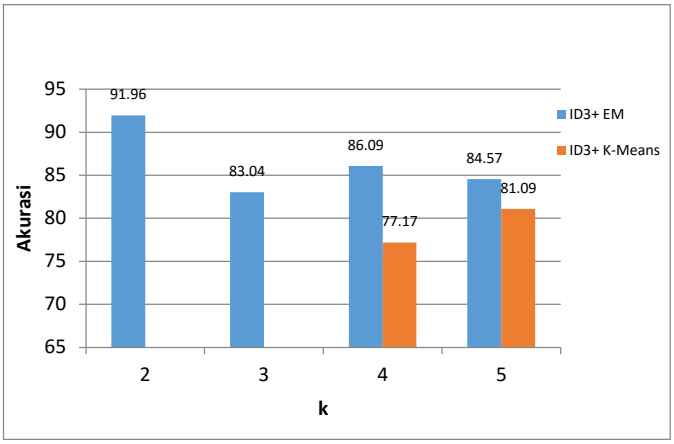
Eksperimen pertama dilakukan untuk mengetahui seberapa besar pengaruh proses FD menggunakan EM terhadap klasifikasi *spam email* dibandingkan dengan algoritma K-Means yang merupakan metode *clustering* yang sering digunakan. Pada eksperimen dengan EM, jika dilakukan perubahan pada nilai *k* akan terjadi penurunan jumlah fitur. Semakin besar nilai *k* akan semakin kecil jumlah fitur diskrit yang dihasilkan. Hasil ekperimen dapat dilihat pada Gambar 2.



Gambar 2 Grafik Hubungan antara Nilai *k* dan Jumlah Fitur Diskrit yang Dihasilkan

Eksperimen selanjutnya dilakukan proses klasifikasi dengan algoritma ID3 menggunakan data training sebesar 70%. Dari Gambar 3 dapat diketahui bahwa akurasi klasifikasi ID3 menggunakan FD dan EM mampu mengungguli K-Means pada klasifikasi *spam email.*

Copyright @ 2015 IlmuKomputer.Com



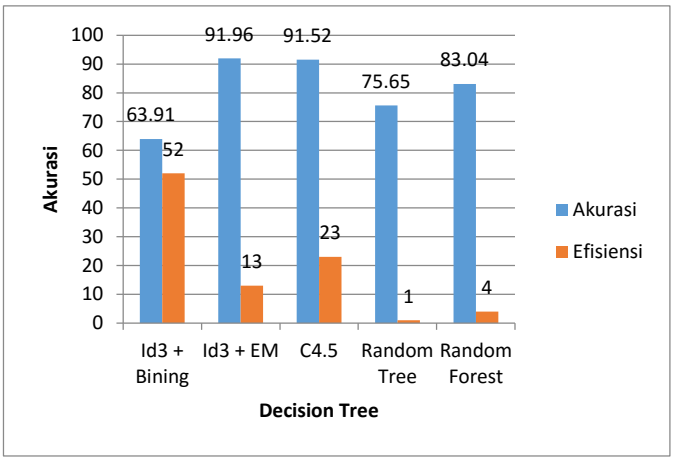
Gambar 3 Grafik Habungan antara Nilai Klaster *k* dengan Akurasi

Eksperimen tahap berikutnya yaitu dengan merubah rasio data yang digunakan pada proses training dengan menggunakan jumlah fitur sebanyak 56, dikarenakan pada eksperimen *feature discretization* akurasi tertinggi ID3+EM dicapai pada jumlah fitur tersebut. Rasio dirubah mulai dari 50%, 60%, 70%, 80% dan 90%. Hasil eksperimen menunjukkan ada kenaikan akurasi sesuai dengan peningkatan rasio data, seperti terlihat pada Tabel 5.

Tabel 5 Hasil Ekperimen ID3-EM dengan Merubah Rasio Data Training

Data training(%)	50	60	70	80	90
Efisiensi (sec)	16	12	12	13	13
Akurasi (%)	89,78	91,52	90,29	91,25	91,96

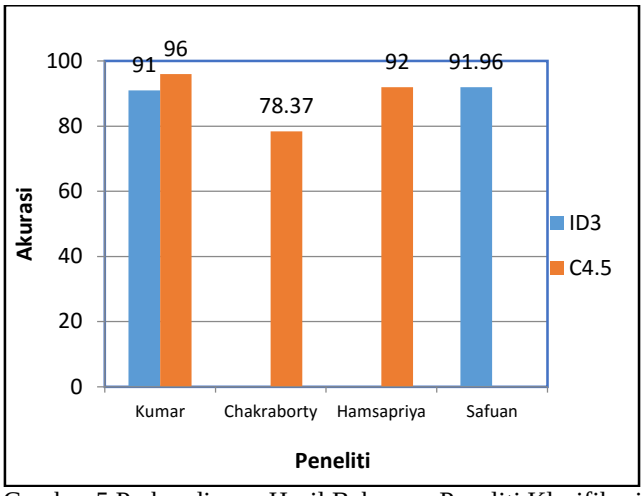
Eksperimen dilanjutkan dengan klasifikasi menggunakan algoritma *decision tree*(DT) yang lain yaitu algoritma C4.5, *Random Forest* dan *Random Tree*. Parameter yang digunakan adalah data training 90% karena pada ekperimen ini didapatkan akurasi ID3+EM yang maksimal. Hasil eksperimen dapat dilihat pada Tabel 4.15.



Gambar 4 Grafik Perbandingan ID3-EM dengan DT

Hasil eksperimen menunjukkan bahwa akurasi klasifikasi menggunakan algoritma ID3 dengan proses FD menggunakan EM memiliki akurasi yang lebih tinggi dan waktu proses yang lebih kecil dibandingkan dengan algoritma C4.5. Tetapi *random forest* dan *random tree* mempunyai waktu proses yang lebih cepat dibanding ID3.

Untuk membuktikan bahwa penelitian yang dilakukan mempunyai kontribuasi terhadap penelitian, maka dilakukan perbandingan dengan peneliti terdahulu yang sudah melakukan penelitian pada klasifikasi spam email, yaitu Kumar *et.al*, Chakraborty *et. al* dan Hamsapriya *et. al.* Perbandingan hasil penelitian dapat dilihat pada data grafik pada Gambar 5.



Gambar 5 Perbandingan Hasil Beberapa Peneliti Klasifikasi *Spam Email*

Dari grafik pada Gambar 5 dapat dilihat bahwa ID3 dari penelitian yang telah dilakukan memiliki akurasi yang lebih tinggi dibanding dengan ID3 pada penelitian Kumar *et.al*. ID3 juga mengungguli akurasi algoritma C4.5 pada penelitian yang dilakukan oleh Chakraborty *et.al*, hampir seimbang (selisih 0,04) dengan hasil penelitian Hamsapriya *et.al* tetapi masih lebih rendah dibanding akurasi C4.5 pada penelitian yang dilakukan oleh Kumar *et.al.* Hal ini dimungkinkan karena penggunaan fitur seleksi dan fitur reduksi pada penelitian Kumar *et.al* dapat menemukan fitur yang benar-benar berpengaruh pada klasifikasi. Sedangkan *feature discretization* pada penelitian ini hanya merubah fitur kontinyu menjadi diskrit saja tanpa memilih fitur yang berpengaruh pada klasifikasi. Ini diperlihatkan dengan berkurangnya akurasi walaupun fiturnya lebih sedikit seperti terlihat pada Gambar 3. Jadi pada penelitian ini jumlah fitur yang kecil tidak menambah akurasi klasifikasi seperti yang dihasilkan oleh fitur seleksi pada penelitian Kumar *et.al* sehingga perlu dilakukan proses fitur seleksi dulu sebelum proses *feature discretization* dilakukan.

5 KESIMPULAN

Dari hasil pengujian diatas, dapat disimpulkan bahwa penerapan *feature discretization* berbasis EM *clustering* dapat mengubah fitur kontinyu menjadi diskrit sehingga klasifikasi *spam email* dengan algoritma ID3 dapat dilakukan dan akurasinya meningkat dibanding penggunaan FD selain EM.

Hasil percobaan menunjukkan bahwa dalam klasifikasi *spam email*, ID3 dapat menghasilkan akurasi 91,96% dengan menggunakan jumlah data training 90% dan jumlah fitur sebanyak 56 yang dihasilkan dari nilai *k* = 2 pada EM. Hasil eksperimen juga menunjukkan bahwa akurasi ID3+EM meningkat sebesar 28,05% dibandingkan dengan ID3+ *binning.* Metode *binning* adalah sebuah metode diskritisasi yang umum digunakan dengan memeriksa "nilai tetangga", yaitu dengan mengurutkan dari yang terkecil sampai dengan yang terbesar kemudian dipartisi ke dalam beberapa *bin.*

FD dengan EM *clustering* terbukti dapat meningkatkan akurasi pada algoritma ID3. Namun ada beberapa faktor yang dapat dicoba untuk penelitian selanjutnya, agar dapat menghasilkan metode yang lebih baik lagi, yaitu:

1. Pada penelitian selanjutnya mungkin bisa menggunakan dataset email selain *spambase* UCI seperti *SpamAssassin*

Copyright @ 2015 IlmuKomputer.Com

corpus dan *Enron corpus* untuk mencoba performa metode FD dengan EM ini.

2. Penerapan *pruning* pada ID3 untuk meningkatkan akurasi klasifikasi. *Pruning* adalah proses yang dilakukan untuk memotong atau menghilangkan beberapa cabang (*branches*) yang tidak diperlukan. Cabang atau node yang tidak diperlukan dapat menyebabkan ukuran *tree* menjadi sangat besar yang disebut *over-fitting*. *Over-fitting* akan menyebabkan terjadinya misklasifikasi, sehingga tingkat akurasi klasifikasi menjadi rendah

REFERENSI

Agre, G., & Peev, S. (2002). On Supervised and Unsupervised Discretization. *Methods*, *2*(2).

Al-Ibrahim, A. (2011). Discretization of Continuous Attributes in Supervised Learning algorithms. *The Research Bulletin of Jordan ACM - ISWSA*, *7952*(Iv).

Allias, N., Megat, M. N., Noor, M., & Ismail, M. N. (2014). A hybrid Gini PSO-SVM feature selection based on Taguchi method. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication - ICUIMC '14* (pp. 1–5). New York, New York, USA: ACM Press. http://doi.org/10.1145/2557977.2557999

Chakraborty, S., & Mondal, B. (2012). Spam Mail Filtering Technique using Different Decision Tree Classifiers through Data Mining Approach - A Comparative Performance Analysis. *International Journal of Computer Applications*, *47*(16), 26–31.

Chharia, A., & Gupta, R. K. (2013). Email classifier: An ensemble using probability and rules. In *2013 Sixth International Conference on Contemporary Computing (IC3)* (pp. 130–136). IEEE. http://doi.org/10.1109/IC3.2013.6612176

Çiltik, A., & Güngör, T. (2008). Time-efficient spam e-mail filtering using n-gram models. *Pattern Recognition Letters*, *29*(1), 19–33. http://doi.org/10.1016/j.patrec.2007.07.018

Dash, R., Paramguru, R. L., & Dash, R. (2011). Comparative Analysis of Supervised and Unsupervised Discretization Techniques. *International Journal of Advances in Science and Technology*, 29–37.

Ferreira, A. J., & Figueiredo, M. a T. (2012). An unsupervised approach to feature discretization and selection. *Pattern Recognition*, *45*(9), 3048–3060. http://doi.org/10.1016/j.patcog.2011.12.008

Ferreira, A. J., & Figueiredo, M. a T. (2014). Incremental filter and wrapper approaches for feature discretization. *Neurocomputing*, *123*, 60–74. http://doi.org/10.1016/j.neucom.2012.10.036

Gupta, A., Mehrotra, K. G., & Mohan, C. (2010). A clustering-based discretization for supervised learning. *Statistics & Probability Letters*, *80*(9-10), 816–824. http://doi.org/10.1016/j.spl.2010.01.015

Hamsapriya, T., D. K. R. and M. R. C. (2012). A Comparative Study of Supervised Machine Learning Techniques for Spam E-mail Filtering. In *2012 Fourth International Conference on Computational Intelligence and Communication Networks* (Vol. 6948, pp. 506–512). IEEE. http://doi.org/10.1109/CICN.2012.14

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. *Morgan Kaufmann Publishers is an imprint of Elsevier* (Vol. 54). http://doi.org/10.1007/978-3-642-19721-5

I. Witten, E. F. (2011). Data Mining : Practical Machine Learning Tools and Techniques. *Morgan Kaufmann Publishers Is an Imprint of Elsevier*.

Jearanaitanakij, K. (2005). Classifying Continuous Data Set by ID3 Algorithm. In *2005 5th International Conference on Information Communications & Signal Processing* (pp. 1048–1051). IEEE. http://doi.org/10.1109/ICICS.2005.1689212

Jin, C., & De-lin, L. (2009). An Improved ID3 Decision Tree Algorithm. *Proceedings of 2009 4th International Conference on Computer Science & Education*, 127–130.

Jorgensen, Z., Zhou, Y., & Inge, M. (2008). A Multiple Instance Learning Strategy for Combating Good Word Attacks on Spam Filters. *Journal of Machine Learning Research*, *8*, 1115–1146. Retrieved from http://jmlr.csail.mit.edu/papers/volume9/jorgensen08a/jorgensen08a.pdf

Jung, Y. G., Kang, M. S., & Heo, J. (2014). Clustering performance comparison using *K* -means and expectation maximization algorithms. *Biotechnology & Biotechnological Equipment*, *28*(sup1), S44–S48. http://doi.org/10.1080/13102818.2014.949045

Kumar, R. K., Poonkuzhali, G., & Sudhakar, P. (2012). Comparative Study on Email Spam Classifier using Data Mining Techniques. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, *I*.

Ladysz, R. (2004). Clustering of Envolving Time Series Data.

Liu Yuxun, & Xie Niuniu. (2010). Improved ID3 algorithm. In *2010 3rd International Conference on Computer Science and Information Technology*. http://doi.org/10.1109/ICCSIT.2010.5564765

Madhu, G., Rajinikanth, T. V, & Govardhan, A. (2014). Feature Selection Algorithm with Discretization and PSO Search Methods for Continuous Attributes. *International Journal of Computer Science and Information Technologies*, *5*(2), 1398–1402.

Marsono, M. N., El-Kharashi, M. W., & Gebali, F. (2008). Binary LNS-based naïve Bayes inference engine for spam control: noise analysis and FPGA implementation. *IET Computers & Digital Techniques*, *2*(1), 56. http://doi.org/10.1049/iet-cdt:20050180

Méndez, J. R., Glez-Peña, D., Fdez-Riverola, F., Díaz, F., & Corchado, J. M. (2009). Managing irrelevant knowledge in CBR models for unsolicited e-mail classification. *Expert Systems with Applications*, *36*(2), 1601–1614. http://doi.org/10.1016/j.eswa.2007.11.037

Nazirova, S. (2011). Survey on Spam Filtering Techniques. *Communications and Network*, *03*(03), 153–160. http://doi.org/10.4236/cn.2011.33019

Saad, O., Darwish, A., & Faraj, R. (2012). A survey of machine learning techniques for Spam filtering. *Journal of Computer Science*, *12*(2), 66–73.

Sculley, D., & Wachman, G. M. (2007). Relaxed online SVMs for spam filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07* (Vol. 36, p. 415). New York, New York, USA: ACM Press. http://doi.org/10.1145/1277741.1277813

Senthilkumar, J., Karthikeyan, S., Manjula, D., & Krishnamoorthy, R. (2012). Web Service Based Feature Selection and Discretization with Efficiency. *2012 IEEE Sixth International Conference on Semantic Computing*, 269–276. http://doi.org/10.1109/ICSC.2012.51

Copyright @ 2015 IlmuKomputer.Com

Sharma, A. K., Sahni, S. (2011). A Comparative Study of Classification Algorithms for Spam Email Data Analysis. *International Journal on Computer Science and Engineering (IJCSE)*, (May), 1890–1895.

Sheu, J. J. (2009). An efficient two-phase spam filtering method based on e-mails categorization. *International Journal of Network Security*, *9*(1), 34–43.

Tsai, C.-J., Lee, C.-I., & Yang, W.-P. (2008). A discretization algorithm based on Class-Attribute Contingency Coefficient. *Information Sciences*, *178*(3), 714–731. http://doi.org/10.1016/j.ins.2007.09.004

Wijaya, A., Wahono, R.S. (2008). Two-Step Cluster based Feature Discretization of Naïve Bayes for Outlier Detection in Intrinsic Plagiarism Detection. *Journal of Intelligent Systems*, (February 2015), 2–9.

Wu, C.-H. (2009). Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. *Expert Systems with Applications*, *36*(3), 4321–4330. http://doi.org/10.1016/j.eswa.2008.03.002

BIOGRAFI PENULIS

Safuan. Lahir pada tanggal 28 Februari 1972 di Kota Semarang, Jawa Tengah. Memperoleh gelar Sarjana Komputer (S.Kom) dari Jurusan Sistem Komputer, STEKOM, Semarang pada tahun 2007. Serta memperoleh gelar M.Kom dari Fakultas Ilmu Komputer, Universitas Dian Nuswantoro pada tahun 2015.

Romi Satria Wahono. Memperoleh Gelar B.Eng dan M.Eng pada fakultas Computer Science, Saitama University, Japan, dan Ph.D pada fakultas Software Engineering, Universiti Teknikal Malaysia Melaka. Mengajar di fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Indonesia. Merupakan pendiri dan CEO Brainmatics, sebuah perusahaan yang bergerak di bidang *software development*, Indonesia. Bidang minat penelitian adalah *Software Engineering* dan *Machine Learning*. Profesional member dari ACM dan asosiasi ilmiah IEEE.

Catur Supriyanto. Dosen di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia. Menerima gelar master dari Universiti Teknikal Malaysia Melaka (UTEM), Malaysia. Minat penelitiannya adalah *information retrieval, machine learning, soft computing* dan *intelligent system.*

Copyright @ 2015 IlmuKomputer.Com