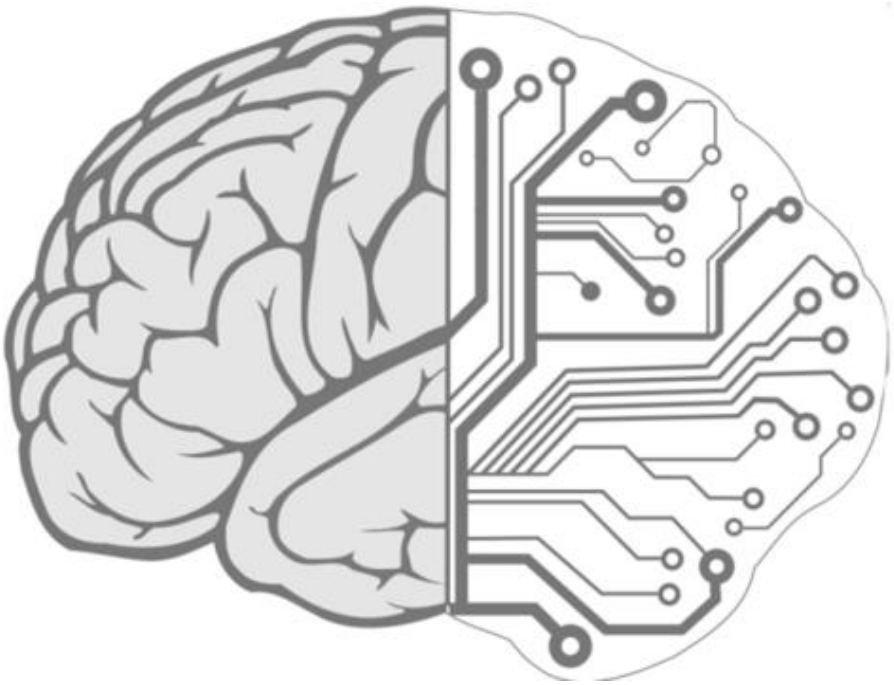
Journal of Intelligent Systems



IlmuKomputer.Com
ikatlah ilmu dengan menuliskannya

Editorial Board

Contents

REGULAR PAPERS

Two-Step Cluster based Feature Discretization of Naïve Bayes for Outlier Detection in Intrinsic Plagiarism Detection

Adi Wijaya
Graduate School of Informatics Engineering, STMIK Eresha
Email:adiwjj@gmail.com

Romi Satria Wahono
Faculty of Computer Science, Dian Nuswantoro University
Email: romi@brainmatics.com

Abstract: Intrinsic plagiarism detection is the task of analyzing a document with respect to undeclared changes in writing style which treated as outliers. Naive Bayes is often used to outlier detection. However, Naive Bayes has assumption that the values of continuous feature are normally distributed where this condition is strongly violated that caused low classification performance. Discretization of continuous feature can improve the performance of Naïve Bayes. In this study, feature discretization based on Two-Step Cluster for Naïve Bayes has been proposed. The proposed method using tf-idf and query language model as feature creator and False Positive/False Negative (FP/FN) threshold which aims to improve the accuracy and evaluated using PAN PC 2009 dataset. The result indicated that the proposed method with discrete feature outperform the result from continuous feature for all evaluation, such as recall, precision, f-measure and accuracy. The using of FP/FN threshold affects the result as well since it can decrease FP and FN; thus, increase all evaluation.

Keywords: intrinsic plagiarism detection, naïve bayes, feature discretization, two-step cluster

1 INTRODUCTION

The problem of plagiarism has recently increased because of the digital era of resources available on the web (Alzahrani, Salim, & Abraham, 2012). As a result, automated plagiarism analysis and detection receives increasing attention especially in academia (Maurer & Kappe, 2006). Intrinsic plagiarism detection (IPD), introduced by Meyer zu Eissen and Stein (2006), more ambitious since no reference corpus is given (Meyer zu Eissen, Stein, & Kulig, 2007) (Tschuggnall & Specht, 2012). IPD is a method for discovering plagiarism by analyzing a document with respect to undeclared changes in writing style (Stein, Lipka, & Prettenhofer, 2011). Since significant deviations in writing style are treated as outliers (Oberreuter & Velásquez, 2013); so, in IPD, outlier detection is important step.

Many studies have been published related to quantify writing style then detect its deviation writing style, such as using character n-gram profile as stylometric feature (Stamatatos, 2009b), word n-gram and word frequency (Oberreuter & Velásquez, 2013) and grammar tree as syntactical feature (Tschuggnall & Specht, 2012). Their approach still not produces excellent result due to unable to detect writing style change as outlier because writing style with small change (Stamatatos, 2009b), writing style change in short text (Oberreuter & Velásquez, 2013) and sentences with few words do not have a significant grammar tree and are therefore not detected (Tschuggnall & Specht, 2012).

This lack of outlier detection need to be solved and machine learning approach can be used for outlier detection (Chandola, Banerjee, & Kumar, 2009). One of algorithm to tackle this problem is Naïve Bayes (NB) since NB is often used to outlier, anomaly or novelty detection (Alan & Catal, 2011; Bahrepour, Zhang, Meratnia, & Havinga, 2009; Kamra, Terzi, & Bertino, 2007; Lepora et al., 2010). NB is fast, easy to implement with the simple structure, effective (Taheri & Mammadov, 2013). NB classifier continues to be a popular learning algorithm for data mining applications due to its simplicity and linear run-time (Hall, 2007).

However, NB has assumption that the values of continuous attributes are normally distributed within each class (Baron, 2014; Jamain & Hand, 2005; Soria, Garibaldi, Ambrogi, Biganzoli, & Ellis, 2011; Wong, 2012) where in many real-world data sets, this condition is strongly violated (Soria et al., 2011) and caused low classification performance (Yang & Webb, 2008).

With empirical evidence, discretization of continuous attributes can simplify data and improve the efficiency of inductive learning algorithms (Li, Deng, Feng, & Fan, 2011). Many discretization methods have been proposed to improve the performance of NB classifiers in terms of both time and accuracy (Ferreira & Figueiredo, 2012; Tsai, Lee, & Yang, 2008; Wong, 2012).

In this research, we propose the combination of Two-Step Cluster (TSC) and NB for improving the accuracy of outlier detection in IPD. TSC is applied to deal with the feature discretization of NB. TSC is chosen due to the ability to handle both continuous and categorical variables (Michailidou, Maheras, Arseni-Papadimititriou, Kolyva-Machera, & Anagnostopoulou, 2008; Satish & Bharadhwaj, 2010a), and in a single run, this procedure helps to identify the variables that significantly differentiate the segments from one another (Satish & Bharadhwaj, 2010a; Wu et al., 2006). TSC promises to solve at least some of these problems (e.g., the ability to deal with mixed-type variables and large data sets, automatic determination of the optimum number of clusters, and variables which may not be normally distributed) (Michailidou et al., 2008).

This paper is organized as follows. In section 2, the related works are explained. In section 3, the proposed method is presented. The experimental results of comparing the proposed method with others are presented in section 4. Finally, our work of this paper is summarized in the last section.

2 RELATED WORKS

Many studies have been published in which the IPD problem is further investigated both statistical based (Oberreuter & Velásquez, 2013; Stamatatos, 2009b; Tschuggnall & Specht, 2012) or machine learning based (Seaward & Matwin, 2009; Curran, 2010; Stein et al., 2011). To date, statistical based is dominated the research in IPD, but recently, machine learning based is trending since the result is promising.

Stamatatos (2009b) using character n-gram profile as stylometric feature that effective for quantifying writing style (Kanaris & Stamatatos, 2007; Koppel, Schler, & Argamon, 2009), robust to noisy text (Kanaris & Stamatatos, 2007) and language independent (Stamatatos, 2009a). This approach attempts to quantify the style variation within a document using character n-gram profiles and a style-change function based on an appropriate dissimilarity measure originally proposed for author identification. In the 1st International Competition on Plagiarism Detection 2009, this method was the first winner with precision, recall and overall are 0.2321, 0.4607 and 0.2462 respectively (Oberreuter & Velásquez, 2013).

Oberreuter & Velásquez (2013) showing that the usage of words can be analyzed and utilized to detect variations in style with great accuracy at the cost of detecting fewer cases (Oberreuter & Velásquez, 2013). They use word n-gram and word frequency as writing style quantification. In The 3rd International Competition on Plagiarism Detection 2011, they was the first winner with precision is 0.34, recall is 0.31 and overall is 0.33 (Oberreuter & Velásquez, 2013).

Tschuggnall & Specht (2012) using syntactical feature, namely the grammar used by an author, able to identify passages that might have been plagiarized due to the assumption is that the way of constructing sentences is significantly different for individual authors (Tschuggnall & Specht, 2012). They use grammar base as syntactical feature for writing style quantification and pq-gram-distance to identify the distance between two grammar trees. By comparing grammar trees and applying statistics the algorithm searches for significant different sentences and marks them as suspicious. The result is precision and recall value of about 32%.

Seaward & Matwin (2009) using three main components of its learning scheme, i.e. data pre-processors, learning algorithm and feature selector. In data pre-processor, the model use Kolmogorov Complexity (KC) measure as style feature. While in learning algorithm, it use Support Vector Machine (SVM) and NN and chi-square feature evaluator as feature selector. KC is used to describe the complexity or degree of randomness of a binary string and can be computed using any lossless compression algorithm. Their model use run-length encoding and Lempel-Ziv compression to create 10 complexity features, i.e. Adjective complexity, adjective count, global topic word complexity, verb word complexity, passive word complexity, active word complexity, preposition count, stop word count, average word length per sentence and local topic word complexity. Performance of the model shows that NN still better than SVM. NN outperform in precision (0.548) and f-measure (0.603) but lower recall (0.671) compared with SVM (recall=0.671, precision=0.521 and f-measure=0.587).

Curran (2010) using 3 components of its learning scheme. Data pre-processor is used in order to create an appropriate data feature that will feed to the model. Its main classifier is Neural Network (NN) and using Neuro-evolution of Augmenting Topologies (NEAT) as parameter optimizer of NN. The NEAT system evolves both NN structures and weights by incrementally increasing the complexity of NN. The model creates ten data features from their data pre-processor. A number of stylometric features were chosen for their study, such as: number of punctuation marks, sentence length (number of characters), sentence word frequency class, number of prepositions, number of syllables per word, average word length, number of stop-words, Gunning Fog index, Flesch index and Kincaid index. The model result 60% of accuracy for the plagiarized class, meaning that 60% of plagiarized sentences are recognized as being plagiarized.

Stein et al (2011) propose meta learning method as outlier post-processing and analyze the degradation in the quality of the model fitting process form its classifier, SVM. They use 3 kinds of stylometric features, such as lexical feature (character based), lexical feature (word based) and syntactic feature. In Meta learning method, they use 2 approaches, heuristic voting and unmasking. Heuristic voting is the estimation and use of acceptance and rejection thresholds based on the number of classified outlier sections and Unmasking measures the increase of a sequence of reconstruction errors, starting with a good reconstruction which then is successively impaired. The use of unmasking is considering a style outlier analysis as a heuristic to compile a potentially plagiarized and sufficiently large auxiliary document. The best result is using unmasking method as Meta learning. The result shows that collection of short plagiarized and light impurity has lowest precision (0.66), moderate recall (0.572) and moderate f-measure (0.67) among other collection while the best result is collection of long document and strong impurity with precision, recall and f-measure are 0.98, 0.60 and 0.74 respectively.

3 PROPOSED METHOD

We propose a method called TSC-FD+NB, which is short for Two-Step Cluster based feature discretization for Naïve Bayes (NB) to achieve better detection performance of outlier detection in intrinsic plagiarism detection. The proposed method evaluated using dataset PAN PC 2009 and using term weighting (tf-idf) and 3 functions in query language model (QLM) based on Ponte and Croft (1998) model, i.e.: mean term frequency of term in documents where it is occurs, risk for term in document and probability of producing the query for given document as document quantification. Figure 1 shows block diagram of the proposed method.

The aim of discretization by using Two-Step Cluster (TSC) is to improve NB classification performance since its tends to be better when continuous features are discretized (Dougherty, 1995). Discretization is a popular approach to handling continuous feature in machine learning (Yang & Webb, 2002); discretization of continuous features can simplify data, more compact and improve the efficiency of inductive learning algorithms. TSC is one of clustering algorithm that developed firstly by Chiu et al. (2001) and designed to handle very large data sets, is provided by the statistical package SPSS. Two-Step cluster able to handle both continuous and categorical variables (Michailidou et al., 2008; Satish & Bharadhwaj, 2010a).

As shown in Figure 1, after processed dataset which is dataset have been quantified is feeding, TSC is used for create class label from processed data. This class label is consisting of two labels, 1 and 2 which mean 1 for plagiarized term and 2 for plagiarism-free term; since there is no information about class label for each term. After class label is defined, prior probability of each class label for NB calculation is conducted

which is needed in NB calculation. The next step is feature discretization where each features discrete with 4 classes. The idea is the same with create 4-binning or 4 quartile in equal width or equal frequency interval binning as unsupervised discretization method. After that, NB calculation for discrete feature is conducted and resulting status of each term with four statuses may be resulted, i.e.: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). If the status is FP or FN, so the next activity is status adjustment following condition as shown in Table 1. If status is not FP or not FN, final status is the same with previous status.

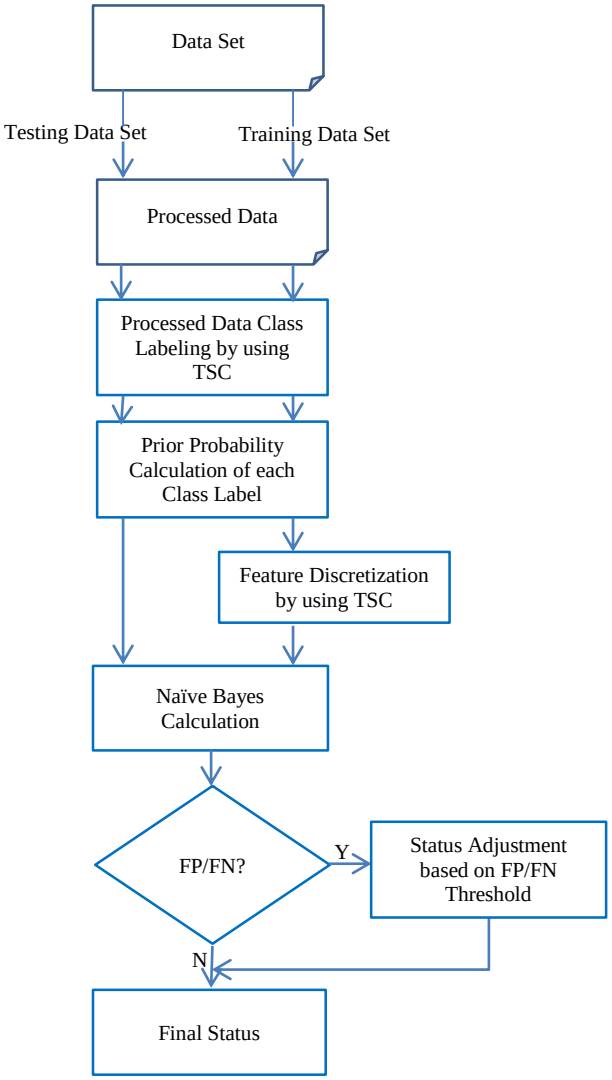


Figure 1. Block Diagram of TSC-FD+NB Method

Table 1. Status Adjustment Condition

First Status	Threshold Comparison	Adjusted Guess	Adjusted Status
FP	$[P(C_2\|x) > P(C_1\|x)]$ <= Threshold	1	TN
	$[P(C_2\|x) > P(C_1\|x)]$ > Threshold	2	FP
FN	$[P(C_1\|x) > P(C_2\|x)]$ <= Threshold	2	TP
	$[P(C_1\|x) > P(C_2\|x)]$ > Threshold	1	FN

In this study, the evaluation method using the classifier's effectiveness (Oberreuter & Velásquez, 2013). The results of this process will produce a confusion matrix that contains the value true positive (TP), true negative (TN), false positive (FP) and false negative (FN) as shown in Table 2. The main result is model accuracy and common information retrieval measurement, such as: recall, precision and f-measure. It calculated based on confusion matrix that produces from the model. Based on confusion matrix, the measurement calculation are as follows:

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - Measure = \frac{2\ x\ Precision\ x\ Recall}{Precison + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Table 2. Confusion Matrix

	Actual	
Guess	Plagiarized	Plagiarism-free
Plagiarized	TP	FP
Plagiarism-free	FN	TN

The main result which is accuracy of all the models will be compared with statistical test both parametric and non-parametric test. T-test as one of parametric test is used to compare between two models if their sample distribution is normal or Wilcoxon Signed Ranks test, one of non-parametric test, if their sample distribution is not normal. While comparison between multi models using Friedman test, one of non-parametric test, to verify whether there is a significant difference between the proposed methods as Demsar (2006) suggested Friedman test for multi classifier or model comparisons. After that, post-hoc test conducted using Nemenyi Test to detect which models significantly has different result since Friedman Test only show whether there is different or no. Therefore, parametric test and non-parametric test are used in this study.

4 EXPERIMENTAL RESULTS

The experiments are conducted using a computing platform based on Intel Core 2 Duo 2.2 GHz CPU, 2 GB RAM, and Microsoft Windows XP SP2 32-bit operating system. The development environment is MS Visual Basic 6, PHP and MySql as database server.

First of all, we conducted experiments on PAN PC 2009 with continuous feature. The experimental results are reported in Table 3. NB with continuous feature perform not so good since has low in all result. FN/FN threshold doesn't change the recall since the difference between posterior still higher than FP/FN thresholds.

Table 3. Results on NB with Continuous Feature

FP/FN Threshold	R	P	F	Accuracy
0	0.167	0.061	0.090	0.788
0.001	0.167	0.062	0.090	0.790
0.005	0.167	0.063	0.091	0.793
0.01	0.167	0.064	0.092	0.794
0.05	**0.167**	**0.071**	**0.099**	**0.811**

In the next experiment, we implemented NB with discrete feature on PAN PC 2009 dataset. The experimental result is shown in Table 4. The improved model is highlighted with boldfaced print. NB with discrete feature model perform excellent rather than continuous feature. FP/FN threshold also increase both recall and precision; thus, F-measure and accuracy increase as well.

The result of comparison of recall, precision, f-measure and accuracy can be described in Figure 2, Figure 3, Figure 4 and Figure 5 respectively. As shown in Figure 2, recall of NB with discrete feature overcome recall in continuous feature. Recall in NB with discrete feature linearly increasing when FP/FN threshold increase. Although NB with discrete feature recall in FP/FN Threshold = 0 is lower than continuous feature, as the increase in the FP/FN threshold, NB with discrete feature recall is increase and overcome the continuous feature with the best recall is 0.247.

Table 4. Results on NB with Discrete Feature

FP/FN Threshold	R	P	F	Accuracy
0	0.156	0.057	0.083	0.786
0.001	0.172	0.073	0.102	0.812
0.005	0.182	0.119	0.144	0.865
0.01	0.186	0.187	0.186	0.899
0.05	**0.247**	**1.000**	**0.397**	**0.953**

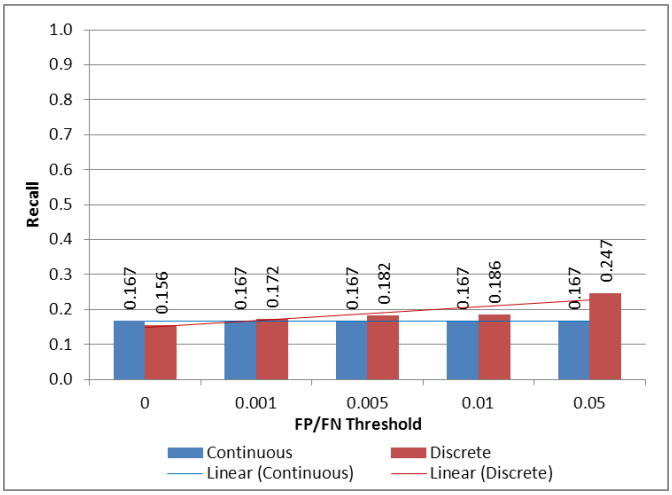


Figure 2. NB with Continuous vs. Discrete Feature's Recall

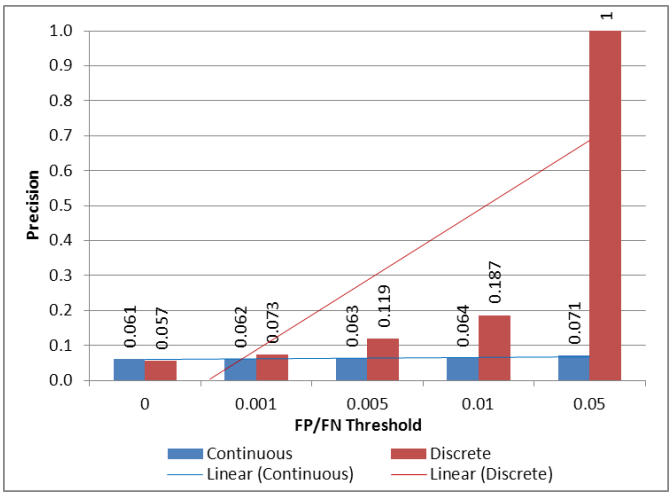


Figure 3. NB with Continuous vs. Discrete Feature's Precision

As shown in Figure 3, precision of NB with discrete feature is overcome continuous feature in almost models except model with FP/FN Threshold = 0. Precision in NB with discrete feature linearly increase when FP/FN threshold increase with the best precision is 1 because in this model value of FP=0. As shown in Figure 3, precision of NB with discrete feature increased sharply compared to continuous feature.

F-measure is harmonic mean between recall and precision. As shown in Figure 4, f-measure of NB with continuous feature tends to constant while f-measure of NB with discrete feature is increases as FP/FN threshold increase. This is because recall and precision of NB with discrete feature increase sharply compared to continuous feature. The best result is model with FP/FN Threshold = 0.05 both NB with continuous feature and NB with discrete feature. The rising of F-measure of NB with continuous feature about 0.109 times from the lowest result model with FP/FN threshold while in NB with discrete feature, it rises 3.754 times from the lowest result. This because precision of NB with discrete feature increase sharply compared to continuous feature.

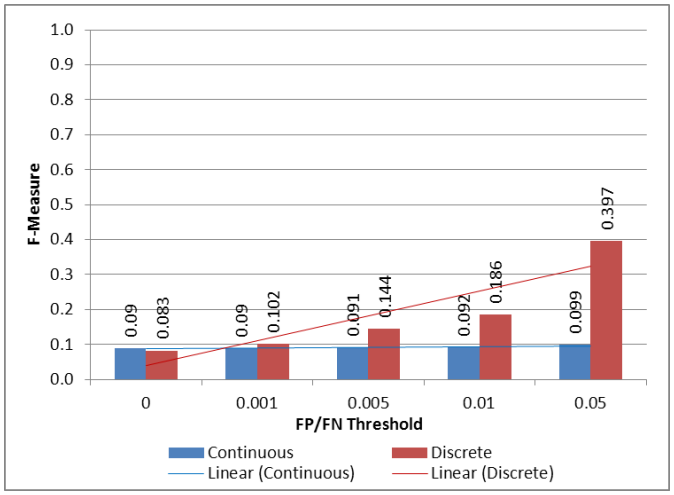


Figure 4. NB with Continuous vs. Discrete Feature's F-measure

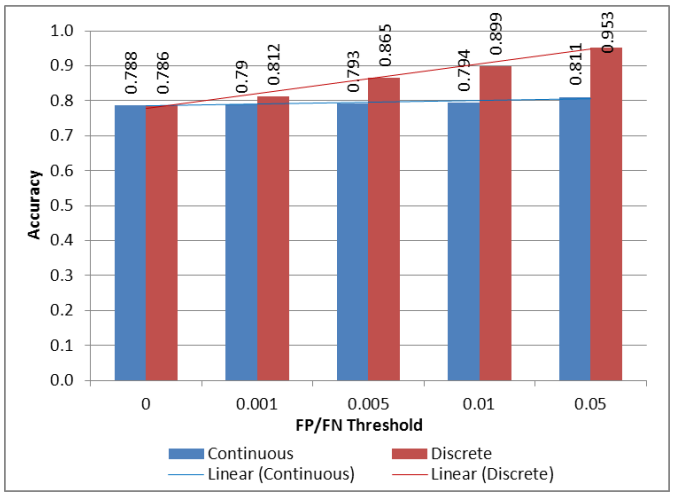


Figure 5. NB with Continuous vs. Discrete Feature's Accuracy

As shown in Figure 5, accuracy of NB with discrete feature is overcome NB with continuous feature with linear increment while accuracy of NB with continuous feature tent to constant. The best accuracy is 0.953 reached by model of NB with discrete feature with FP/FN threshold 0.05. Accuracy of NB with continuous feature increases only 0.029 times from the lowest result, model with FP/FN Threshold = 0 while in NB with discrete feature, it increases about 0.212 times.

This result indicates that FP/FP threshold is effective enough to increase all measurement in NB with discrete feature while not effective in NB with continuous feature. Threshold is one of scheme that many researchers used to improve plagiarism detection such as in case the similarity score, if above a threshold, the detected plagiarism case is considered true and otherwise (Stamatatos, 2011b); lowering FP value so less time is required to filter out FPs by the evaluator (Chen et al., 2010) and for the last purpose is obtaining a reasonable trade-off between precision and recall (Oberreuter & Velásquez, 2013). In this study, the result also confirm some studies that recall is tend to lower for short document since false negatives are relatively short documents (Stamatatos, 2009b) and also recall still low for document with little portion of plagiarizes passage (Oberreuter & Velásquez, 2013).

The results of both methods are compared in order to verify whether a significant difference between NB with continuous feature and the proposed TSC-FD+NB method. We compare between methods with the same FP/FN thresholds. Since there is model with no normal distribution of data, so comparison is used Wilcoxon signed ranks test. Table 5 shows the result.

Table 5. Wilcoxon Signed Rank Test for Continuous Feature's Accuracy vs. Discrete Feature's Accuracy

	Negative Ranks	Positive Ranks	Z	Asymp. Sig. (2-tailed)
c_th_0 vs d_th_0	13.53 (20/30)	18.28 (9/30)	-1.147	**0.252**
c_th_0.001 vs. d_th_0.001	15.66 (25/30)	10.88 (4/30)	-3.763	**0**
c_th_0.005 vs. d_th_0.005	15 (29/30)	0 (0/30)	-4.703	**0**
c_th_0.01 vs. d_th_0.01	15.00 (29/30)	0 (0/30)	-4.703	**0**
c_th_0.05 vs. d_th_0.05	15 (29/30)	0 (0/30)	-4.703	**0**

As shown in Table 5, there is 5 pair of comparisons. In Pair 1, NB with continuous feature with FP/FN Threshold = 0 (c_th_0) is compare to NB with discrete feature with FN/FP Threshold = 0.001 (d_th_0). P-value = 0.252 (> 0.005), so H_0 is failed to be rejected. This indicates that there is no difference accuracy between c_th_0 and d_th_0. Another consideration is negative ranks and positive ranks, which indicates that d_th_0 accuracy is higher than c_th_0 accuracy with 20 cases out of 30 cases (negative ranks) while positive ranks is 9 cases of 30 cases, which means there is 9 cases in c_th_0 that its accuracy is higher than d_th_0.

In Pair 2, NB with continuous feature with FP/FN Threshold = 0.001 (c_th_0.001) is compare to NB with discrete feature with FP/FN Threshold = 0.001 (d_th_0.001). P-value = 0.000 (<0.005), so H_0 is rejected. This indicates that there is difference accuracy between c_th_0.001 and d_th_0.001. Another consideration is negative ranks and positive ranks, which indicates that d_th_0.001 accuracy is higher than c_th_0.001 accuracy with 25 cases out of 30 cases (negative ranks) while positive ranks is 4 cases of 30 cases, which means there is only 4 cases in c_th_0.001 that its accuracy is higher than d_th_0.001. Another pairs are having the same result. All of them indicate that there is difference accuracy where p-value = 0.000 and negative ranks is 29 cases out of 30 cases, which means accuracy of 29 cases in d_th_0.005, d_th_0.01 and d_th_0.05 are higher than c_th_0.005, c_th_0.01 and c_th_0.05 respectively.

Finally, Demsar recommends the Friedman test for classifier comparisons, which relies on less restrictive assumptions (Demsar, 2006). Based on this recommendation, the Friedman test is employed in this study to compare the accuracy of the different models. At first comparison, NB with various FP/FN thresholds are tested with Friedman test both continuous feature and discrete feature. Table 6 shows Friedman test result.

As shown in Table 6, p-value < 0.0001 which is smaller than significance level (α = 0.05). The null hypothesis, H_0 is that the samples come from the same population. Since p-value < 0.05, so H_0 is rejected. This means that sample comes from different population. It indicates that there is difference between models, but Friedman test doesn't provide which

model is different. To answer that question, post-hoc test is used, which is in this case is Nemenyi test. According to the Nemenyi test the performance of two models is significantly different if the corresponding mean ranks differ by at least the critical difference (Li et al., 2011). Nemenyi test is similar to the Tukey test for ANOVA and is used when all classifiers are compared to each other (Demsar, 2006).

Table 6. Friedman Test of All Models Accuracy

Q (Observed value)	234.3890
Q (Critical value)	16.9190
DF	9
p-value (Two-tailed)	< 0.0001
alpha	0.05

Table 7. Multiple pairwise comparisons using Nemenyi's procedure of All Models Accuraccy

Sample*)	Frequency	Sum of ranks	Mean of ranks
(1) c_0	30	43.5000	1.4500
(2) c_0.001	30	81.5000	2.7167
(3) d_0	30	106.5000	3.5500
(4) c_0.005	30	111.5000	3.7167
(5) c_0.01	30	134.0000	4.4667
(6) d_0.001	30	178.0000	5.9333
(7) c_0.05	30	194.0000	6.4667
(8) d_0.005	30	237.0000	7.9000
(9) d_0.01	30	267.5000	8.9167
(10) d_0.05	30	296.5000	9.8833

Table 7 described the mean rank based procedures Nemenyi. The mean of ranks is obtained from the comparison between the models, the higher the rank, the higher the point, and then divided by the number of data samples. The Nemenyi test calculates all pairwise comparisons between different models and checks which models' performance differences exceed the critical difference (CD), which are 2.4732.

The significant difference table of the Nemenyi test is shown in Table 8. Model of NB with discrete feature with FP/FN threshold is the most different with other models about 7 differences. In continuous feature, model of NB with FP/FN threshold 0.005 and 0.01 have fewest difference with other models with 3 differences. In discrete feature, model of NB with FP/FN threshold 0 and 0.001 have fewest difference with other models with 3 differences as well.

Table 8. Significant Differences of All Models Accuracy

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1)	N	N	N	**Y**	**Y**	N	**Y**	**Y**	**Y**	**Y**
(2)	N	N	N	N	**Y**	N	**Y**	**Y**	**Y**	**Y**
(3)	N	N	N	N	**Y**	N	N	**Y**	**Y**	**Y**
(4)	**Y**	N	N	N	N	N	N	**Y**	**Y**	**Y**
(5)	**Y**	**Y**	**Y**	N	N	**Y**	N	N	N	**Y**
(6)	N	N	N	N	**Y**	N	N	**Y**	**Y**	**Y**
(7)	**Y**	**Y**	N	N	N	N	N	N	**Y**	**Y**
(8)	**Y**	**Y**	**Y**	**Y**	N	**Y**	N	N	N	N
(9)	**Y**	**Y**	**Y**	**Y**	N	**Y**	**Y**	N	N	N
(10)	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	N	N	N

As shown in Table 7 and 8, model of NB with discrete feature with FP/FN Threshold = 0.05 outperform other models followed by model of NB with discrete feature with FP/FN Threshold = 0.01, model of NB with discrete feature with FP/FN Threshold = 0.005, model of NB with continuous feature with FP/FN Threshold = 0.05 and model of NB with discrete feature with FP/FN Threshold = 0.001 in the second, third, fourth and fifth rank respectively. Overall model of NB with discrete feature outperform model of NB with continuous feature. This result confirmed Webb (2001), that explain why discretization can be effective for NB classifier.

These results prove that NB with discrete feature is better than NB with continuous feature. The discretization, referring to Yang and Webb (2003) can be effective for NB classifier. TSC as the method of discretization is promising method because support both continuous and categorical variables, and in a single run; this procedure helps to identify the variables that significantly differentiate the segments from one another (Satish & Bharadhwaj, 2010b), automatic determination of the optimum number of clusters, and variables which may not be normally distributed (Michailidou et al., 2008). From this study, TSC can be alternative as discretization based on the clustering analysis, such as k-means discretization (Dash, Paramguru, & Dash, 2011; Richhariya & Sharma, 2014) and shared nearest neighbor clustering algorithm (Gupta, Mehrotra, & Mohan, 2010). Thus, TSC based discretization of NB can be used to improve the performance of intrinsic plagiarism detection by detect outlier from short plagiarized passage in a document.

5 CONCLUSION

The experimental result shows that the result models of NB with discrete feature outperform the result from NB with continuous feature for all measurement, such as recall, precision, f-measure and accuracy with significant difference. The using of FP/FN threshold affect the result as well with FP/FN threshold = 0.05 is the best since it can decrease false positive (FP) and false negative (FN) better than other value in all models especially model with discrete feature where the decrement of FP and FN is highest; thus, increase all measurement especially precision and accuracy. Therefore, it can be concluded that feature discretization based on Two-Step Cluster can improve the accuracy of NB for outlier detection in intrinsic plagiarism detection compared to NB with continuous feature.

REFERENCES

Alan, O., & Catal, C. (2011). Thresholds based outlier detection approach for mining class outliers: An empirical case study on software measurement datasets. *Expert Systems with Applications*, *38*(4), 3440–3445.

Alzahrani, S. M., Salim, N., & Abraham, A. (2012). Understanding plagiarism linguistic patterns, textual features, and detection methods. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *42*(2), 133–149.

Bahrepour, M., Zhang, Y., Meratnia, N., & Havinga, P. J. M. (2009). Use of event detection approaches for outlier detection in wireless sensor networks. *2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 439–444.

Baron, G. (2014). Influence of Data Discretization on Efficiency of Bayesian Classifier for Authorship Attribution. *Procedia Computer Science*, *35*, 1112–1121.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–72.

Chen, C., Yeh, J., & Ke, H. (2010). Plagiarism Detection using ROUGE and WordNet, *2*(3), 34–44.

Chiu, T., Fang, D., Chen, J., Wang, Y., & Jeris, C. (2001). A Robust and Scalable Clustering Algorithm for Mixed Type Attributes in Large Database Environment. In *Proceedings of the 7th ACM SIGKDD Internation- al Conference on Knowledge Discovery and Data Mining* (pp. 263–268).

Curran, D. (2010). An evolutionary neural network approach to intrinsic plagiarism detection. In *AICS 2009, LNAI 6206* (pp. 33–40). Springer-Verlag Berlin Heidelberg.

Dash, R., Paramguru, R. L., & Dash, R. (2011). Comparative Analysis of Supervised and Unsupervised Discretization Techniques, *2*(3), 29–37.

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dougherty, J. (1995). Supervised and Unsupervised Discretization of Continuous Features.

Ferreira, A. J., & Figueiredo, M. a. T. (2012). An unsupervised approach to feature discretization and selection. *Pattern Recognition*, *45*(9), 3048–3060.

Gupta, A., Mehrotra, K. G., & Mohan, C. (2010). A clustering-based discretization for supervised learning. *Statistics & Probability Letters*, *80*(9-10), 816–824. doi:10.1016/j.spl.2010.01.015

Hall, M. (2007). A decision tree-based attribute weighting filter for naive Bayes. *Knowledge-Based Systems*, *20*(2), 120–126.

Jamain, A., & Hand, D. J. (2005). The Naive Bayes Mystery: A classification detective story. *Pattern Recognition Letters*, *26*(11), 1752–1760.

Kamra, A., Terzi, E., & Bertino, E. (2007). Information Assurance and Security Detecting anomalous access patterns in relational databases.

Kanaris, I., & Stamatatos, E. (2007). Webpage Genre Identification Using Variable-Length Character n-Grams. In *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)* (pp. 3–10). IEEE.

Koppel, M., Schler, J., & Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, *60*(1), 9–26.

Lepora, N. F., Pearson, M. J., Mitchinson, B., Evans, M., Fox, C., Pipe, A., Prescott, T. J. (2010). Naive Bayes novelty detection for a moving robot with whiskers. *2010 IEEE International Conference on Robotics and Biomimetics*, 131–136.

Li, M., Deng, S., Feng, S., & Fan, J. (2011). An effective discretization based on Class-Attribute Coherence Maximization. *Pattern Recognition Letters*, *32*(15), 1962–1973.

Maurer, H., & Kappe, F. (2006). Plagiarism - A Survey, *12*(8), 1050–1084.

Meyer zu Eissen, S., Stein, B., & Kulig, M. (2007). Plagiarism Detection Without Reference Collections. In *Studies in Classification, Data Analysis, and Knowledge Organization, Advances in Data Analysis* (pp. 359–366). Berlin: Springer.

Michailidou, C., Maheras, P., Arseni-Papadimititriou, a., Kolyva-Machera, F., & Anagnostopoulou, C. (2008). A study of weather types at Athens and Thessaloniki and their relationship to circulation types for the cold-wet period, part I: two-step cluster analysis. *Theoretical and Applied Climatology*, *97*(1-2), 163–177.

Oberreuter, G., & Velásquez, J. D. (2013). Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Systems with Applications*, *40*(9), 3756–3763.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275–281).

Richhariya, V., & Sharma, N. (2014). Optimized Intrusion Detection by CACC Discretization Via Naïve Bayes and K-Means Clustering, *14*(1), 54–58.

Satish, S. M., & Bharadhwaj, S. (2010a). Information Search Behaviour among New Car Buyers: A Two-Step Cluster Analysis. *IIMB Management Review*, *22*(1-2), 2.

Satish, S. M., & Bharadhwaj, S. (2010b). Information search behaviour among new car buyers: A two-step cluster analysis. *IIMB Management Review*, *22*(1-2), 5–15.

Seaward, L., & Matwin, S. (2009). Intrinsic Plagiarism Detection using Complexity Analysis. In *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009)* (pp. 56–61).

Soria, D., Garibaldi, J. M., Ambrogi, F., Biganzoli, E. M., & Ellis, I. O. (2011). A "non-parametric" version of the naive Bayes classifier. *Knowledge-Based Systems*, *24*(6), 775–784.

Stamatatos, E. (2009a). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, *60*(3), 538–556.

Stamatatos, E. (2009b). Intrinsic Plagiarism Detection Using Character n -gram Profiles. In *Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009)* (pp. 38–46).

Stamatatos, E. (2011). Plagiarism Detection Using Stopword n -grams, *62*(12), 2512–2527.

Stein, B., Lipka, N., & Prettenhofer, P. (2011). Intrinsic plagiarism analysis. *Language Resources and Evaluation*, *45*(1), 63–82.

Taheri, S., & Mammadov, M. (2013). Learning the naive Bayes classifier with optimization models. *International Journal of Applied Mathematics and Computer Science*, *23*(4), 787–795.

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.

Tschuggnall, M., & Specht, G¨. (2012). Plag-Inn: Intrinsic Plagiarism Detection Using Grammar Trees. In G. Bouma, A. Ittoo, E. Métais, & H. Wortmann (Eds.), *LNCS-Natural Language Processing and Information Systems* (Vol. 7337, pp. 284–289). Berlin, Heidelberg: Springer Berlin Heidelberg.

Webb, G. I. (2001). On Why Discretization Works for Naive-Bayes Classifiers.

Wong, T.-T. (2012). A hybrid discretization method for naïve Bayesian classifiers. *Pattern Recognition*, *45*(6), 2321–2325.

Wu, H., Zhang, X., Li, X., Liao, P., Li, W., Li, Z., … Pei, F. (2006). Studies on Acute Toxicity of Model Toxins by Proton Magnetic Resonance Spectroscopy of Urine Combined with Two-step Cluster Analysis. *Chinese Journal of Analytical Chemistry*, *34*(1), 21–25.

Yang, Y., & Webb, G. I. (2002). A Comparative Study of Discretization Methods for Naive-Bayes Classifiers. In *Proceedings of PKAW 2002, The 2002 Pacific Rim Knowledge Acquisition Work- shop* (pp. 159–173). Tokyo, Japan.

Yang, Y., & Webb, G. I. (2008). Discretization for naive-Bayes learning: managing discretization bias and variance. *Machine Learning*, *74*(1), 39–74.

BIOGRAPHY OF AUTHORS

Adi Wijaya. Received MKom from STMIK Eresha, Jakarta. He is an IT professional and part-time lecturer at STIKIM, Jakarta. 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, PMI and IEEE Computer Society.

Penerapan Reduksi Region Palsu Berbasis *Mathematical Morphology* pada Algoritma *Adaboost* Untuk Deteksi Plat Nomor Kendaraan Indonesia

Muhammad Faisal Amin
Software Development Department, CV Adcoms Anugrah
Email: faisal.indonesia@gmail.com

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

Abstract: Tahap deteksi plat nomor merupakan langkah yang paling penting dan sulit dalam sistem identifikasi plat nomor. Kondisi plat nomor yang memiliki warna *background* yang mirip dengan warna mobil, dan memiliki variasi yang besar dalam bentuk dan ukuran, menyebabkan deteksi plat nomor menjadi rendah. Kondisi tersebut terjadi pada plat nomor kendaraan pribadi Indonesia. Agar deteksi plat nomor Indonesia menjadi akurat, maka diusulkan untuk menerapkan algoritma adaboost. Dibandingkan dengan metode lain, algoritma adaboost adalah metode terbaik untuk mengatasi masalah yang terjadi pada plat nomor Indonesia. Algoritma adaboost akurat dalam mendeteksi plat nomor tanpa terikat oleh warna, bentuk, dan ukuran. Akan tetapi, akurasi dari algoritma ini rendah ketika terdapat banyak region palsu pada gambar. Oleh karena itu, diusulkan untuk menambahkan proses reduksi region palsu berupa operasi *mathematical morphology* di bagian *online recognizing* algoritma *adaboost*. Hasil eksperimen menunjukkan bahwa integrasi algoritma *adaboost* dan *mathematical morphology* lebih akurat dalam mendeteksi plat nomor Indonesia. Nilai *precision rate* dan *recall rate* masing-masing dari algoritma *adaboost* standard adalah 84,44% dan 84,62%. Setelah algoritma *adaboost* dan *mathematical morphology* diintegrasikan, nilai *precision rate* dan *recall rate* masing-masing naik menjadi 94,47% dan 92,31%.

Keywords: sistem deteksi plat nomor, algoritma adaboost, mathematical morphology

1 PENDAHULUAN

Sistem identifikasi plat nomor kendaraan menjadi trend dalam penelitian sistem transportasi cerdas. Sistem identifikasi plat nomor kendaraan terdiri dari tiga tahap, yaitu deteksi plat nomor, segmentasi karakter, dan pengenalan karakter. Diantara ketiga tahap tersebut, tahap deteksi plat nomor merupakan tahap yang paling penting dan paling sulit (Anishiya & Joans, 2011) (Sun, Li, Xu, & Wang, 2009) (Chang, Chen, Chung, & Chen, 2004). Hal ini disebabkan karena tahap ini dapat mempengaruhi keakuratan sistem secara keseluruhan. Jika tahap ini gagal, maka dipastikan tahap berikutnya pasti juga akan mengalami kegagalan.

Kondisi plat nomor yang yang mempunyai warna background yang mirip dengan warna body mobil (Cui & Xie, 2009), dan variasi yang besar dalam bentuk dan ukuran (Liu, Cui, Shu, & Xin, 2011) menyebabkan deteksi terhadap plat nomor menjadi lebih sulit. Kondisi tersebut terjadi pada plat nomor pribadi kendaraan Indonesia. *Background* plat nomor pribadi kendaraan Indonesia berwarna hitam. Padahal warna hitam adalah salah satu warna mobil paling populer di dunia (Dupont, 2010). Ini artinya, kebanyakan warna *background* plat nomor di Indonesia mirip dengan warna mobil. Masalah lain yang terjadi pada plat nomor pribadi Indonesia, yaitu variasi yang besar baik dalam bentuk dan ukuran. Hal ini terjadi karena terjadi perubahan desain plat nomor kendaraan sejak bulan April 2011. Selain itu, para pemakai kendaraan banyak yang menggunakan plat nomor tidak standard. Gambar 1 menunjukkan contoh variasi dari plat nomor Indonesia.

Metode saat ini yang saat digunakan untuk deteksi plat nomor, yaitu metode berbasis *image processing* dan metode berbasis *machine learning*. Contoh metode deteksi plat nomor berbasis *image processing*, antara lain *color image processing* (Cui & Xie, 2009), deteksi tepi (Suri, Walia, & Verma, 2010), *mathematical morphology* (Anishiya & Joans, 2011), dan sebagainya. Metode berbasis *image processing* cenderung lebih mudah diimplementasikan. Akan tetapi, metode ini tidak kuat terhadap perubahan lingkungan (Zhao, et al., 2010). Berbeda dengan metode berbasis *machine learning* yang lebih kuat terhadap perubahan lingkungan (Zhao, et al., 2010). Contoh metode deteksi plat nomor berbasis *machine learning*, antara lain *neural network* (Sirithinaphong & Chamnongthai, 1998), algoritma *adaboost* (Cui, et al., 2009), dan sebagainya. Dalam mendeteksi objek, algoritma *adaboost* mempunyai keunggulan baik di akurasi dan kecepatan (Viola & Jones, 2004). Algoritma ini memiliki akurasi yang tinggi, seperti neural network (NN) tapi lebih cepat dari NN (Viola & Jones, 2004). Algoritma *adaboost* akurat dalam mendeteksi plat nomor dan tidak terikat dengan ukuran, warna, dan posisi plat nomor (Zhang, Shen, Xiao, & Li, 2010). Algoritma ini sesuai untuk plat nomor Indonesia yang memiliki warna mirip dengan warna mobil, dan memiliki variasi yang besar dalam bentuk dan ukuran. Akan tetapi, akurasi deteksi metode ini rendah ketika terdapat banyak region palsu pada gambar input (Wu & Ai, 2008).

Gambar 1. Plat Nomor Kendaraan Indonesia

Untuk mengatasi kekurangan pada algoritma *adaboost* tersebut, diusulkan proses reduksi region palsu pada bagian *online recognizing* algoritma *adaboost*. Proses reduksi region palsu ini sebenarnya diadopsi dari tahap *rough detection* metode deteksi plat nomor berbasis *image processing*. Metode

yang akan digunakan untuk reduksi region palsu adalah operasi *mathematical morphology*. Alasan memilih operasi *mathematical morphology* karena metode ini mampu melakukan analisis pada gambar yang sensitif terhadap bentuk tertentu (Abolghasemi & Ahmadyfard, 2007) dan kontras kecerahan (Sulehria, Zhang, & Irfan, 2007). Sedangkan varian algoritma *adaboost* yang akan digunakan adalah *gentle adaboost*. Alasan memilih algoritma *gentle adaboost* adalah karena algoritma ini memiliki kinerja terbaik dalam deteksi plat nomor dibandingkan varian algoritma *adaboost* yang lain (Cui, et al., 2009). Diharapkan integrasi algoritma *adaboost* dan *mathematical morphology* dapat lebih akurat dalam mendeteksi plat nomor kendaraan Indonesia.

Paper ini disusun sebagai berikut. Pada bagian 2, penelitian terkait dijelaskan. Pada bagian 3 model yang diusulkan dijelaskan. Hasil eksperimen dijelaskan pada bagian 4. Ringkasan pekerjaan pada paper dijelaskan pada bagian terakhir.

2 PENELITIAN TERKAIT

Penelitian yang dilakukan oleh Dlagnekov (Dlagnekov, 2004) menggunakan ekstraksi fitur berbasis algoritma adaboost untuk mendeteksi plat nomor. Penelitian ini merupakan penelitian pertama yang mengusulkan algoritma *adaboost* untuk deteksi plat nomor. Fitur yang mereka gunakan adalah varian lain dari *haar feature*, yaitu *x-derivative*, *y-derivative*, *variance*, dan *x-derivative*. Metode yang mereka usulkan memperoleh nilai *recall rate* sebesar 95,5% dan *false positive rate* sebesar 5,7%.

Penelitian yang dilakukan oleh Cui, et al (Cui, et al., 2009) melakukan komparasi terhadap tiga algoritma *adaboost*, yaitu *dicrete adaboost*, *real adaboost*, dan *gentle adaboost* untuk deteksi plat nomor. Mereka mengomparasi tingkat deteksi dan tingkat *false positive* ketiga algoritma *adaboost* tersebut dengan beberapa setting pengaturan yang berbeda, seperti subwindow size dan jumlah layer *cascade*. Hasil penelitian mereka menyebutkan bahwa *gentle adaboost* memiliki kinerja yang paling baik berdasarkan *ROC curve*.

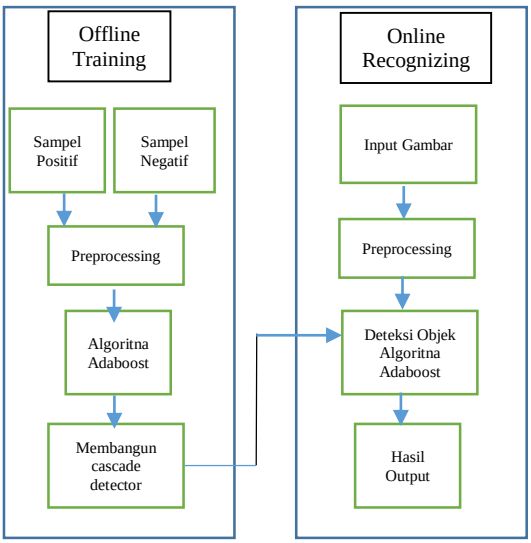
Penelitian yang dilakukan oleh Zhang, Shen, Xiao, dan Li (Zhang, Shen, Xiao, & Li, 2010) menggunakan *global feature* dan *local feature* berbasis *adaboost* untuk mendeteksi plat nomor. Kedua *feature* ini dikombinasikan dalam *cascade detector*. Metode yang mereka usulkan memperoleh nilai *recall rate* sebesar 93,5%.

Algoritma *adaboost* akurat dalam mendeteksi plat nomor dan tidak terikat dengan ukuran, warna, dan posisi plat nomor (Zhang, Shen, Xiao, & Li, 2010). Namun, akurasi algoritma ini ini rendah ketika terdapat banyak region palsu pada gambar input (Wu & Ai, 2008). Oleh karena itu, pada penelitian diusulkan proses reduksi region palsu pada bagian *online recognizing* algoritma *adaboost* menggunakan operasi *mathematical morphology*.

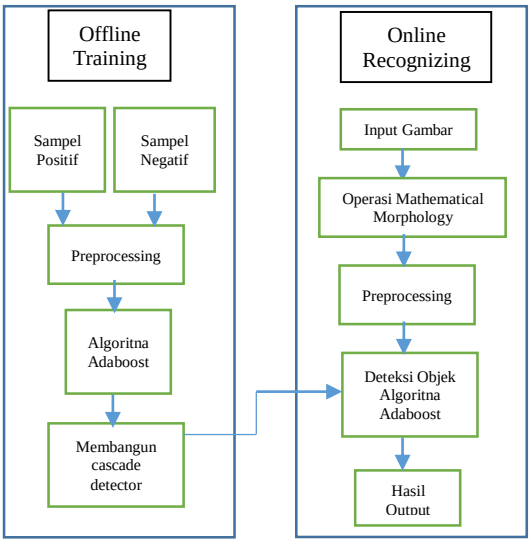
3 MODEL YANG DIUSULKAN

Kami mengusulkan mengintegrasikan operasi mathematical morphology pada bagian *online recognizing* algoritma adaboost. Operasi *mathematical morhology* yang digunakan adalah *top-hat transfrom* dan *dilation*. Operasi *mathematical morphology* ini akan mereduksi region palsu pada gambar input. Setelah itu, *Cascade detector* tinggal memindai kandidat region plat nomor tersebut untuk menentukan region kandidat plat nomor yang valid. Pada Gambar 2 dan Gambar 3 ditampilkan masing-masing skema algoritma *adaboost* standard dan model yang diusulkan.

Model standard dari algoritma *adaboost* terdiri dari dua bagian, yaitu bagian offline training dan bagian *online recognizing*. Bagian offline training adalah bagian proses pelatihan data yang tidak bekerja secara *realtime*. Bagian ini meliputi penginputan sampel gambar positif dan sampel gambar negatif, preprocessing, pelatihan data oleh algoritma *adaboost* sampai membangun detektor. Setelah detektor terbentuk kita bisa melakukan pendeteksian secara *realtime*/*online recognizing* terhadap data pengujian. Sebelum melakukan pendeteksian dengan algoritma *adaboost*, terlebih dahulu data pengujian sudah harus mengalami preprocessing. Kemudian algoritma *adaboost* hasil pelatihan data berupa detektor akan melakukan pendeteksian objek dan menampilkan hasilnya.



Gambar 2. Skema Algoritma Adaboost Standard



Gambar 3. Skema Model yang Diusulkan

Pada penelitian ini diusulkan untuk melakukan reduksi region palsu dengan operasi-operasi *mathematical morphology* pada bagian *online recognizing*. Hasil output dari operasi *mathematical morphology* adalah region kandidat-kandidat plat nomor. Hal ini menjadikan kerja algoritma *adaboost* menjadi lebih ringan karena tidak perlu memindai gambar secara keseluruhan. Diharapkan dengan reduksi region palsu dengan operasi *mathematical morphology* ini akurasi deteksi

algoritma *adaboost* meningkat dan menurunkan tingkat kesalahan deteksi.

Sebelum proses ekstraksi *feature* terlebih dahulu dilakukan *preprocessing*. Hal ini dilakukan untuk mengurangi kompleksitas proses selanjutnya. Terlebih dahulu gambar RGB dikonversi ke *grayscale* dengan persamaan:

$$\boldsymbol{Gray = 0.0229R + 0.587G + 0.114B} \quad (1)$$

Setelah itu semua gambar untuk pelatihan data mengalami proses normalisasi variance. Ingat bahwa:

$$\boldsymbol{\sigma^2 = m^2 - \frac{1}{N}\sum x^2} \quad (2)$$

Di mana σ adalah standard deviasi, m adalah mean , dan x adalah nilai pixel pada gambar.

Feature yang diekstrak oleh algoritma *adaboost* adalah *haar feature*. Nilai *haar feature* adalah perbedaan antara jumlah piksel dalam daerah hitam dan daerah putih sehingga dapat mencerminkan perubahan skala pada gambar grayscale. *Haar feature* diekstrak menggunakan *integral image*. Kemudian algoritma *adaboost* akan melakukan seleksi terhadap *feature* dan melakukan pembobotan untuk membentuk *classifier* lemah. Gabungan dari *classifier* lemah membentuk *classifier* kuat. Pada Gambar 4 ditunjukkan skema detail algoritma adaboost.

Haar feature dihitung melalui *integral image*. Nilai pada *integral image* pada setiap titik (x,y) dapat dinyatakan sebagai jumlah titik piksel atas kiri saat ini dan dihitung melalui persamaan berikut:

$$\boldsymbol{ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y')} \quad (3)$$

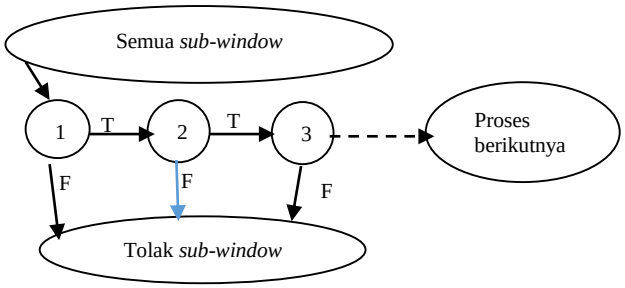
Dimana ii(x,y) adalah integral image yang telah dihitung dan i(x',y') adalah gambar grayscale original.

Varian Algoritma *adaboost* yang akan digunakan adalah *gentle adaboost*. Adapun langkah demi langkah algoritma *gentle adaboost* adalah sebagai berikut:

1. Dimulai dengan inisialisasi bobot Wi = 1/N,I = 1,2,…,N, F(x) = 0
2. Ulangi untuk m = 1,2,…,M:
   A. Ulangi untuk m = 1,2,…,M:Neg = menset nilai sample negatif
   B. Sesuaikan dengan fungsi regresi $f_m(x)$ dengan pembobotan least-squares dari y_i ke x_i dengan bobot w_i
   C. Update $F(x) \leftarrow F(x) + f_m(x)$
   D. Update $w_i(x) \leftarrow w, \exp(- y_i f_m(x_i))$ dan normalisasi ulang
3. Output classifier
   $sign[F(x)] = sign\left[\sum_{m=1}^{M} f_m\right]$

Proses berikutnya adalah membentuk *cascade detector*. Tahapan dalam *cascade* ini dibangun oleh pelatihan *classifier* menggunakan *adaboost* dan kemudian menyesuaikan nilai ambang batas untuk meminimalkan negatif palsu. Pada Gambar 5 ditunjukkan *cacade detector* . Algoritma pelatihan untuk membangun *cascade detector* adalah sebagai berikut:

1. Pengguna memilih nilai untuk f, yaitu tingkat maksimum *false positive* dapat diterima per tahap dan d, yaitu tingkat deteksi minimum yang dapat diterima per tahap.
2. Pengguna memilih target keseluruhan dari tingkat *false positive* F_{target}
3. Pos = menset nilai sample positif
4. Neg = menset nilai sample negatif
5. $F_0 = 1.0$, $D_0 = 1.0$, i = 0
6. i = 0
7. While $F_i > F_{target}$
   A. $i \leftarrow i + 1$
   B. $n_i = 0$; $F_i = F_i\text{-}1$
   C. While $F_i > f \times F_i\text{-}1$
      a. $n_i \leftarrow n_i+1$
      b. Gunakan Pos dan Neg untuk melatih *classifier* dengan fitur n_i menggunakan *adaboost*
      c. Lakukan evaluasi terhadap *classifier cascade* saat ini untuk menentukan F_i, D_i
      d. Menurunkan *threshold* untuk *classifier* ke i sampai *classifier* saat ini memiliki tingkat deteksi setidaknya d X D_{i-1}
   D. $N \leftarrow \boldsymbol{\varphi}$
   E. If $F_{i+1} > F$ target maka lakukan evaluasi pada *cascade detector* saat ini menjadi bukan plat nomor dan menempatkan setiap pendeteksian palsu ke dalam Neg



Gambar 4. Skema Algoritma Adaboost

Operasi *mathematical morphology* bekerja pada gambar biner dan dapat diperluas ke gambar *grayscale*. Terlebih dahulu gambar RGB dikonversi ke *grayscale*. Setelah gambar dikonversi menjadi gambar *grayscale*, kontras dari gambar *foreground* akan ditingkatkan. *Top-hat transform* merupakan hasil subtraksi gambar input dengan gambar yang telah mengalami operasi *opening*. Operasi ini menekan *background* gelap dan menyoroti *foreground* sehingga kontras gambar *foreground* meningkat. *Top-hat transform* dapat dituliskan dengan persamaan berikut ini:

$$\boldsymbol{TopHat(src) = src - open(src)} \quad (4)$$

Gambar *grayscale* yang kontrasnya sudah ditingkatkan akan diubah ke gambar biner. Gambar biner tersebut perlu ditebalkan. Hal ini dilakukan untuk mengantisipasi hasil biner yang kurang baik dan menggabungkan region berdekatan pada gambar. Di sini operasi dilation akan diterapkan dan hasil outputnya adalah berupa kandidat plat nomor. *Dilation* dapat dituliskan dengan persamaan berikut ini:

$$\boldsymbol{D(A,B) = A \oplus B} \quad (5)$$

Dimana A adalah gambar input dan B adalah *structuring element*.

4 HASIL EKSPERIMEN

13 layer *cascade detector* akan dilatih dengan algoritma *adaboost* berdasarkan gambar sampel positif dan gambar sampel negatif. Jumlah sampel gambar positif adalah 2720 gambar dan jumlah sampel gambar negatif adalah 2864.. Adapun setting parameter lain yang dilakukan pada algoritma *adaboost*, yaitu *boosting* = *gentle adaboost*, *min hit rate* = 0.995000, *max false alarn* = 0.500000, *mode* = *ALL*, *subwindow size* = 40x15, 43x16, 47x15, dan 50x19. Pada eksperimen ini akan dicoba empat nilai *subwindow size* yang berbeda untuk menemukan akurasi deteksi yang terbaik. Nilai subwindow size yang digunakan, yaitu 40x15, 43x16, 47x15, dan 50x19.

Setelah tahap training selesai, maka akan terbentuk *classifier* berupa 4 *cascade detector* dengan *subwindow size* 40x15, 43x16, 47x15, dan 50x19. Masing-masing *cascade detector* itu akan diuji untuk mendeteksi plat nomor pada data pengujian dengan tiga nilai *scale factor* berbeda. Adapun nilai *scale factor* yang digunakan, yaitu 1,1, 1,2, dan 1,3. Dengan demikian, sembilan model *cascade detector* akan diuji untuk mencari akurasi deteksi yang terbaik. Metode pengujian yang akan digunakan, yaitu *precision rate* dan *recall rate*. Berikut ini persamaan untuk menghitung *precision rate* dan *recall rate*.

$$\textbf{Precision rate} = \frac{TP}{TP + FP} \quad (7)$$

$$\textbf{Recall rate} = \frac{TP}{TP + FN} \quad (8)$$

Di mana TP adalah *true positive*, FP adalah *false positive*, dan FN adalah *false negative*.

Setelah ditemukan model pengaturan *subwindow size* dan *scale factor* terbaik, maka model tersebut akan diimplemantasikan dan akan ditambahkan operasi *mathematical morphology* pada bagian *online recognizing*. Operasi *mathematical morphology* akan mereduksi region palsu dan menghasilkan kandidat region plat nomor. Operasi *mathematical morphology* yang digunakan adalah *top-hat transform* dan *dilation*. Adapun untuk nilai *structuring element*(*SE*) dan *anchor point* untuk operasi *top-hat transform* adalah masing-masing 13,3 dan 5,2. Sedanglan nilai *SE* dan *anchor point* untuk operasi dilation adalah masing-masing 10,3 dan 5,2. *Cascade detector* tinggal memindai kandidat region plat nomor tersebut untuk menentukan region kandidat plat nomor yang valid. Pengujian berikutnya adalah mengomparasi classifier hasil training algoritma *adaboost* standard dengan clasifier hasil training dari integrasi algoritma *adaboost* dan *mathematical morphology*. Metode pengujian yang digunakan juga sama, yaitu precision rate dan recall rate. Pada Gambar 5 ditunjukan hasil deteksi dari algoritma *adaboost* terhadap plat nomor Indonesia dengan berbagai variasi plat nomor. Pada Gambar 6 ditunjukkan langkah-langkah hasil deteksi model yang diusulkan, yaitu integrasi algoritma *adaboost* dan *mathematical morphplogy*.

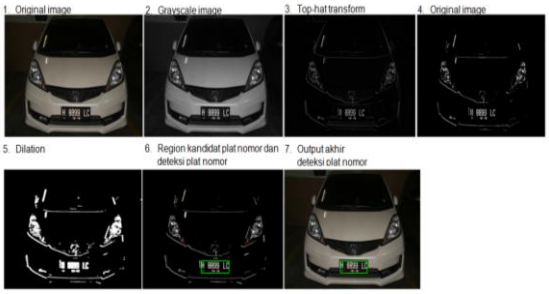
Beberapa hasil deteksi yang kurang akurat dari algoritma *adaboost* tetapi akurat dengan model yang diusulkan ditujukkan pada Gambar 7 dan Gambar 8. Gambar 7 di sebelah kiri menunjukkan deteksi yang kurang akurat dari algoritma *adaboost* (*false positive*) dan gambar sebelah kanan menunjukkan hasil deteksi yang akurat dari model yang diusulkan. Gambar 8 di sebelah kiri menunjukkan deteksi yang kurang akurat dari algoritma *adaboost* (*false negative*) dan gambar sebelah kanan menunjukkan hasil deteksi yang akurat dari model yang diusulkan.

Pada Tabel 1, Tabel 2, Tabel 3, dan Tabel 4 ditunjukkan nilai *precision rate* dan *recall rate* masing-masing *subwindow* berukuran 40x15, 43x16, 47x15, dan 50x19 dengan *tiga scale factor* berbeda.

Beberapa model algoritma *adaboost* dari eksperimen yang telah dilakukan kebanyakan menghasilkan nilai *precision rate* yang rendah, bahkan ada yang nilainya di bawah 20%. Hal ini berarti bahwa terlalu banyak region bukan plat nomor / *false positive* yang terdeteksi sebagai plat nomor oleh *classifier*. Hanya model *subwindow* berukuran 43x16 yang memiliki nilai *precision rate* yang tinggi. Nilai *precision rate* yang diperoleh 84,44%, 89,97%, dan 94,19% dengan masing-masing *scale factor* 1,1, 1,2, dan 1,3.

Untuk nilai *recall rate* secara umum semua model telah mencapai nilai di atas 60%. Empat model yang memperoleh nilai *recall rate* di atas 80%, yaitu *subwindow* berukuran 40x15 *scale factor* 1,1, *subwindow* berukuran 43x16 *scale factor* 1,1, *subwindow* berukuran 47x15 *scale factor* 1,1, dan *subwindow* berukuran 50x19 *scale factor* 1,1 dengan masing-masing nilai 82,74%, 84,62%, 83,99%, dan 89,18%. Hal menarik yang ditemukan, yaitu semua *subwindow* dengan *scale factor* 1,1 memiliki nilai *recall rate* tertinggi dibandingkan dengan *scale factor* 1,2 dan 1,3.

Gambar 5. Hasil Deteksi Algoritma Adaboost



Gambar 6. Langkah-Langkah Deteksi Model yang Diusulkan

Gambar 7. False Positive Tidak Terjadi

Gambar 8. False Negative tidak Terjadi

Tiga model algoritma *adaboost* yang hanya memperoleh nilai yang tinggi pada salah satu parameter, memperoleh nilai *precision rate* yang tinggi tapi rendah pada *recall rate*, dan sebaliknya. Contohnya nilai *precision rate* pada model *subwindow* berukuran 43x16 *scale factor* 1,3 yang memperoleh nilai *precision rate* tertinggi, yaitu 94,19%, tapi memperoleh nilai *recall rate* yang rendah, yaitu 64,03%. Begitu juga dengan model subwindow berukuran 43x16 *scale factor* 1,2 memperoleh nilai *precision rate* yang tinggi, yaitu 89,97%, tapi memperoleh nilai yang rendah pada *recall rate*, yaitu 63,41%. Hal ini berarti bahwa terjadi kesalahan yang tinggi berupa banyaknya region yang kenyataannya plat nomor tidak terdeteksi sebagai plat nomor / *false negative* oleh *classifier*. Begitu juga dengan model *subwindow* berukuran 50x19 memperoleh nilai *recall rate* tertinggi, yaitu 89,18% tapi *drop* pada nilai *precision rate* yang hanya bernilai 19,35%. Hal ini berarti bahwa terjadi kesalahan yang sangat tinggi berupa sangat banyak region bukan plat nomor yang terdeteksi sebagai plat nomor. Terdapat satu model yang memperoleh nilai *precision rate* dan *recall rate* yang sama-sama tinggi, yaitu *subwindow* berukuran 43x16 *scale factor* 1,1. Nilai *precision rate* model ini sebesar 90,48% dan nilai *recall rate* sebesar 88,67%.

Tabel 1. Precision dan Recall Subwindow 40x15

Scale factor	*Hits (TP)*	*Missed (FN)*	*False (FP)*	*Precision*	*Recall*
1.1	398	83	3115	11,20%	82,74%
1.2	329	152	857	27,74%	68,40%
1.3	308	173	1802	15,59%	64,03%

Tabel 2. Precision dan Recall Subwindow 43x16

Scale factor	*Hits (TP)*	*Missed (FN)*	*False (FP)*	*Precision*	*Recall*
1.1	407	74	75	84,44%	84,62%
1.2	305	176	34	89,97%	63,41%
1.3	308	173	19	94,19%	64,03%

Tabel 3. Precicion dan Recall Subwindow 47x15

Scale factor	*Hits (TP)*	*Missed (FN)*	*False (FP)*	*Precision*	*Recall*
1.1	404	77	1461	21,66&	83,99%
1.2	340	141	859	28,36%	70,69%
1.3	298	183	657	31,20%	61,95%

Tabel 4. Precision dan Recall Subwindow 50x19

Scale factor	*Hits (TP)*	*Missed (FN)*	*False (FP)*	*Precision*	*Recall*
1.1	429	12	1788	19,35%	89,18%
1.2	380	101	1013	27,28%	79%
1.3	345	136	879	28,19%	71,73%

Pada penelitian ingin dikembangkan *classifier* yang bernilai tinggi berdasarkan dua parameter pengukuran, yaitu *precision rate* dan *recall rate*. Jika kedua parameter ini nilainya sama-sama tinggi berarti *classifier* yang dibangun memiliki tingkat akurasi deteksi yang tinggi dan memiliki tingkat kesalahan deteksi yang rendah. Dengan alasan itulah model *subwindow* berukuran 43x16 *scale factor* 1,1 ditetapkan padaalgoritma *adaboost*. Setelah itu algoritma *adaboost* dengan model pengaturan terbaik tadi akan ditambahkan operasi *mathematical morphology* di bagian *online recognizing*. Pengujian berikutnya adalah mengomparasi akurasi dari algoritma *adaboost* standard dengan metode integrasi alogoritma *adaboost* dan *mathematical morphology*. Parameter pengukuran yang digunakan juga sama. Pada Tabel 5 ditunjukkan komparasi antara algoritma *adaboost* dengan model yang diusulkan berdasarkan *precision rate* dan *recall rate*.

Dari hasil komparasi antara algoritma *adaboost* standard dan model yang diusulkan berdasarkan dari parameter *precision rate* dan *recall rate* menunjukkan bahwa model yang diusulkan, yaitu integrasi algoritma *adaboost* dan *mathematical morphology* memiliki kinerja deteksi yang lebih akurat. Nilai *precision rate* dan *recall rate* dari metode yang diusulkan masing-masing adalah 94,47% dan 92,31%. Sedangkan algoritma *adaboost* standard memiliki nilai *precision rate* dan *recall rate* yang lebih rendah, yaitu masing-masing 84,44% dan 84,62%. Nilai *precision rate* dan *recall rate* dari integrasi algoritma *adaboost* dan *mathematical morphology* masing-masing lebih tinggi 10,03% dan 7,69% dari nilai *precision rate* dan *recall rate* dari algoritma *adaboost* standard.

Tabel 5. Komparasi Algoritma Adaboost dengan Model yang Diusulkan

Metode	*Hits (TP)*	*Missed (FN)*	*False (FP)*	*Precision*	*Recall*
Algoritma *adaboost*	407	74	75	84,44%	84,62%
Model yang diusulkan	444	37	26	94,47%	92,31%

5 KESIMPULAN

Dari hasil eksperimen yang telah dilakukan pada dplat nomor Indonesia, dapat ditarik kesimpulan bahwa penerapan algoritma *adaboost* dan *mathematical morphology* lebih akurat dalam mendeteksi plat nomor Indonesia yang kebanyakan memiliki warna *background* yang mirip dengan warna mobil serta memiliki variasi yang besar dalam bentuk dan ukuran. Selain itu, dengan mengintegrasikan algoritma *adaboost* dengan operasi *mathematical morphology* pada bagian *online recognizing*, ternyata dapat mengatasi kekurangan dari algoritma *adaboost* yang akurasinya rendah ketika mendeteksi objek pada gambar input yang banyak memiliki region palsu. Nilai *precision rate* dan *recall rate* dari integrasi algoritma *adaboost* dan *mathematical morphology* masing-masing 94,47% dan 92,31%.. Kedua nilai ini lebih tinggi 10,03% dan 7,69% dari nilai *precision rate* dan *recall rate* algoritma *adaboost* standard.

Metode integrasi algoritma *adaboost* dan *mathematical morphology* memang terbukti akurat untuk deteksi plat nomor yang memiliki warna *background* yang mirip dengan warna mobil, memiliki variasi yang besar dalam bentuk dan ukuran, dan terdapat banyak region palsu pada gambar input. Namun, di penelitian yang akan datang, ada beberapa pekerjaan yang perlu dilakukan. Beberapa gambar input yang region plat nomornya memilki cahaya yang sangat terang, tidak terdeteksi sebagai plat nomor oleh *classifier*. Perlu sekali mencari metode yang tepat untuk memecahkan permasalahan tersebut.

REFERENSI

Abolghasemi, V., & Ahmadyfard, A. (2007). Local Enhancement of Car Image for License Plate Detection. *15th European Signal Processing Conference*, (pp. 2179-2183). Poznan.

Anishiya, P., & Joans, S. M. (2011). Number Plate Recognition for Indian Cars Using Morphological Dilation and Erosion with the Aid Of Ocrs. *International Conference on Information and Network Technology*, (pp. 115-119). Singapore.

Chang, S.-L., Chen, L.-S., Chung, Y.-C., & Chen, S.-W. (2004). Automatic license plate recognition . *IEEE Transactions on Intelligent Transportation System, 5*(1), 42-53.

Cui, D., Gu, D., Member, IEEE, Cai, H., & Sun, J. (2009). License Plate Detection Algorithm Based on Gentle AdaBoost Algorithm with a Cascade Structure. *International Conference on Robotics and Biomimetics*, (pp. 1962-1966). Guilin.

Cui, Z., & Xie, M. (2009). A Method for Blue Background White Characters Car License Plate Location. *Computer Science and Information Technology*, (pp. 393-395).

Dlagnekov, L. (2004). *License Plate Detection Using Adaboost.* San Diego.

Dupont. (2010, December). Retrieved from Dupont Web site: http://www2.dupont.com/Media_Center/en_US/daily_news/december/article20101208.html

Liu, Y., Cui, L., Shu, J., & Xin, G. (2011). License Plate Location Method Based on Binary Image Jump and Mathematical Morphology. *International Journal of Digital Content Technology and its Applications*, 259-265.

Sirithinaphong, T., & Chamnongthai, K. (1998). Extraction of Car License Plate Using Motor Vehicle Regulation and Character Pattern Recognition. *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, (pp. 559-562). Chiangmai.

Sulehria, H. K., Zhang, Y., & Irfan, D. (2007). Mathematical Morphology Methodology for Extraction of Vehicle Number Plates. *InternationalL Journal of Computers, 1*(3), 69-73.

Sun, G., Li, G., Xu, L., & Wang, J. (2009, December). The Location and Recognition of Chinese Vehicle License Plates under Complex Backgrounds. *Journal of Multimedia, 4*, 442-449.

Suri, P., Walia, E., & Verma, E. A. (2010, Dec). Vehicle Number Plate Detection using Sobel Edge Detection Technique. *International Journal of Computer Science and Technology, 1*(2), 179-182.

Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, 137-154.

Wu, Y.-W., & Ai, X.-Y. (2008). An Improvement of Face Detection Using AdaBoost with Color Information. *ISECS International Colloquium on Computing, Communication, Control, and Management*, (pp. 317-321).

Zhang, X., Shen, P., Xiao, Y., & Li, B. (2010). License Plate-Location using Adaboost Algorithm. *Information and Automation*, (pp. 2456-2461). Harbin.

Zhao, Y., Gu, J., Liu, C., Han, S., Gao, Y., & Hu, Q. (2010). License Plate Location Based on Haar-like Cascade Classifiers and Edges. *Second WRI Global Congress on Intelligent Systems*, (pp. 102-105).

BIOGRAPHY OF AUTHORS

Muhammad Faisal Amin. Received S.Kom degrees in information system from STMIK Banjarabaru, Indonesia and M.Kom degrees in informatic engineering from Dian Nuswantoro University, Indonesia. He is a lecturer at the Under Graduate School of information technology, Lambung Mangkurat University, Indonesia. He is also a founder and chief executive officer of Adcoms, Inc., a software development company in Indonesia. His current research interests include computer vision 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, PMI and IEEE Computer Society.

Color and Texture Feature Extraction Using Gabor Filter - Local Binary Patterns for Image Segmentation with Fuzzy C-Means

Yanuar Wicaksono, Romi Satria Wahono, and Vincent Suhartono
Faculty of Computer Science, Dian Nuswantoro University
yanuar.wic@gmail.com, romi@brainmatics.com, vincent.suhartono@dsn.dinus.ac.id

Abstract: Image segmentation to be basic for image analysis and recognition process. Segmentation divides the image into several regions based on the unique homogeneous image pixel. Image segmentation classify homogeneous pixels basedon several features such as color, texture and others. Color contains a lot of information and human vision can see thousands of color combinations and intensity compared with grayscale or with black and white (binary). The method is easy to implement to segementation is clustering method such as the Fuzzy C-Means (FCM) algorithm. Features to beextracted image is color and texture, to use the color vector L* a* b* color space and to texture using Gabor filters. However, Gabor filters have poor performance when the image is segmented many micro texture, thus affecting the accuracy of image segmentation. As support in improving the accuracy of the extracted micro texture used method of Local Binary Patterns (LBP). Experimental use of color features compared with grayscales increased 16.54% accuracy rate for texture Gabor filters and 14.57% for filter LBP. While the LBP texture features can help improve the accuracy of image segmentation, although small at 2% on a grayscales and 0.05% on the color space L* a* b*.

Keywords: Texture and Color, Image Segmentation, Local Binary Pattern, Gabor Filter, Fuzzy c-Means

1 INTRODUCTION

Interpret an image to obtain a description of the image through several processes including preprocessing, image segmentation, image analysis, and image interpretation (Perner, 2006). Image segmentation becomes a foundation for process analysis and image recognition. Segmentation divides the image into several regions based on the unique homogeneous image pixel (Gonzalez & Richard, 2002) (Raut, Raghuwanshi, Dharaskar, & Raut, 2009) and became a topic that is still widely studied (Cremers, Rousson, & Deriche, 2006). The purpose of segmentation is to separate the image into several regions so interpreted simply be something that is meaningful and easy to analyze.

Image segmentation classify homogeneous pixels based on several features such as color, texture and others. Color contains a lot of information and human vision can see thousands of color combinations and intensity compared with grayscale or with black and white (binary) (Cheng, Jiang, Sun, & Wang, 2001) (Khan, Adhami, & Bhuiyan, 2009). More complete information of color image and image segmentation is better compared with the scale of gray. The human visual system is not only able to distinguish objects based on color, but texture also has an important role.The texture of the image can be defined as a function of local spatial variation in pixel intensity and orientation on grayscale (Tuceryan & Jain, 1999). The main characteristics of the texture is a repeat pattern of spatial pixels in the image (Abbadeni, Zhou, & Wang, 2000) (Manjunath, Ohm, Vasudevan, & Yamada, 2001) which can be repeated exactly, or as a set of small variations, possibly as a function of position. By combining color and texture features of the image would be helpful in distinguish regions have the same color but different textures, or otherwise.

Approaches based on similarities and differences by Zaher Al Aghbari and Ruba Al-Haj (Aghbari & Al-Haj, 2006) image segmentation methods can be classified into five: threshold, boundary-based, region-based, clustering and combined or mixed methods.Of some of these methods apply the clustering method because it is easy to be applied and produce satisfactory segmentation results. The number of features characters from each image pixel as a vector space enter analyzed the clustering method so that it takes a preprocessing step to extracting features for each pixel. Clustering method that has been widely used, there are two main types of hard clustering and fuzzy clustering. Hard clustering (k-means) method which is simple and easy to use. However, in general a lot of issues such as limited spatial resolution, the minimum contrast, overlapping intensities, noise and intensity inhomogeneities reduce the effectiveness of hard clustering method (Krinidis & Chatzis, 2010). Algorithm fuzzy c-means (FCM) clustering method fuzzy one which gives the value of membership in each group for each pixel.

Feature extraction procedure resulted in a description of an object in terms of measurable parameters that represent relevant properties of the object, and can be used for classification by setting the object to the class (Vandenbroucke, Macaire, & Postaire, 2003). Image features are used for segmentation of color and texture features, although the nature of a separate feature which texture image uses level grayscale while color extracting all the information on the color space.Color is a feature in a three-dimensional color space (3D) RGB, which related to the frequency of the red, green and blue from the spectrum of light.In research Imtnan-Ul-Haque Qazi et al (Qazi, Alata, Burie, Moussa, & Fernandez-Maloigne, 2011) showed that the L* a* b* color space indicates the minimum correlation between luminance and chrominance information as well as the lesser of the color space used for color texture classification based on spatial structure information.

Then for feature extraction method on texture analysis in the past few years a lot of research analysis tektsur introduces Gabor filter method (Clausi & Jernigan, 2000) (Idrissa & Acheroy, 2002) (Zhang, Tan, & Ma, 2002) (Khan, Adhami, & Bhuiyan, 2009) because the cells in the visual cortex simple mammalian brain can be modeled by Gabor functions, so that image analysis by Gabor filter is similar to the human visual perception system.

Research that has combined a color and texture features is done by Khan (Khan, Adhami, & Bhuiyan, 2009) by applying a L* a* b* color space and Gabor filters. Gabor filter calculates pixel values to implement the filter banks on small neighborhoods that Gabor filters work well on macro texture

and micro texture ignored. At the micro-textured images, Gabor filter performance in image segmentation becomes worse (Li & Staunton, 2008). To detect micro texture using Local Binary Patterns (LBP) (Mäenpää & Pietikäinen, 2006) which is based on research conducted by Ma Li and RC Staunton (Li & Staunton, 2008) and Lotfi Tlig et al (Tlig, Sayadi, & Fnaiech, 2012) in the color space grayscale.

This paper is organized as follows. In section 2, the related works are explained. In section 3, the proposed method is presented. The experimental results of comparing the proposed method with others are presented in section 4. Finally, ourwork of this paper is summarized in the last section.

2 RELATED WORKS

Research conducted by Ma Li and R.C. Staunton (Li & Staunton, 2008) a new approach to multi-texture image segmentation based on the formation of an effective texture feature vector with a color spacegrayscale. Vector texture features obtained from the integration between the single Gabor filter with local binary pattern (LBP). The method they propose to obtain a single Gabor filter becomes efficient for a small number of texture classes, but for three or more, can not distinguish the difference in texture. When integrated with LBP image segmentation for the better.

The research conducted by Jesmin F. Khan et al (Khan, Adhami, & Bhuiyan, 2009) an approach to new method of selection scale based on local image properties related to changes in brightness, color, texture and position are taken for each pixel at the selected size with L * a * b *color space. Feature image is measured using a Gabor filter in accordance with the selected adaptive orientation size frequency, and phase for each pixel. To cluster pixels into different regions, the joint distribution of pixel features is modeled by a mixture of Gaussians utilizing three variants of the expectation maximization algorithm (EM). Three different versions of EM used in research with clustering are: (1) Penalized EM, (2) Penalized stochastic EM, and (3) the inverse Penalized EM. Researchers determine the value of the number of models that best suits the natural number of clusters in the image based on the Schwarz criterion, which maximizes the posterior probability of the number of groups given the samples of observation.

Research conducted by Lotfi Tlig et al (Tlig, Sayadi, & Fnaiech, 2012) for texture analysis using Gabor filter which has been widely applied, but Gabor filter has a strong dependence on a number of parameters that affect the performance of texture characterization. Furthermore, Gabor filters can not extract texture features micro also has a negative effect on the clustering process. The approach taken in this study combines the outputs of grating cell operator (GCO) is derived from Gabor filters with local binary pattern (LBP).

In this research, we applying the feature extraction using Gabor filter-LBP as research conducted by Ma Li and RC Staunton (Li & Staunton, 2008) and and Lotfi Tlig et al (Tlig, Sayadi, & Fnaiech, 2012) for image segmentation based on color and texture features that have been proposed by Jesmin F. Khan (Khan, Adhami, & Bhuiyan, 2009).

3 PROPOSED METHOD

We propose a method for extracting color and texture features using Gabor filters -LBP are applied to the data for image segmentation research that secondary data drawn from the Berkeley Segmentation Dataset (BSDS).

The BSDS is a RGB image that will converted to L* a* b* color space, so it has three values L, a and b. From these values represent the values of L grayscale color space to extracting texture feature while the values of a and b represent the color feature.

For macro texture will be extracted using Gabor filters with a formula such as:

$$G_{\lambda\theta\varphi\sigma\gamma}(x,y) = e^{-\frac{\acute{x}^2+\gamma^2\acute{y}^2}{2\sigma^2}} \cos\left(2\pi\frac{\acute{x}}{\lambda}+\varphi\right)$$
$$\acute{x} = x\cos\theta + y\sin\theta$$
$$\acute{y} = -x\sin\theta + y\cos\theta$$

where Lambda (λ) is the wavelength parameters of the sinusoidal factor. λ is the inverse of the frequency of the wave in the Gabor function with a value of f = 1 / λ. Tetha (θ) is the normal orientation of the parallel lines of Gabor function, its value is determined in degrees between 0 and 360. Phi (φ) is the phase offset as a factor in the cosine Gabor function, its value in degrees between -180 and 180 Sigma (σ) standard deviation of the Gaussian factor determines the size of the (linear) support of the gabor function. σvalue can not be determined directly but can be changed only through the value of the bandwidth (b). Gamma (γ) is the spatial aspect ratio that determines the shape of the ellipse of the Gabor function. To γ = 1, the shape is a circle. To γ<1 elongated shape seen in the orientation function.

For micro texture will be extracted using Local Binary Patterns (LBP)witha formula such as:

$$LBP_{P,R}(x,y) = \sum_{P=0}^{P-1} s(g_P - g_C) * 2^P$$
$$s(x)\begin{cases}1\,;x \geq 0\\0\,;x\ <0\end{cases}$$

where gp the neighboring pixels are evenly distributed sample points on the circle a number of P pixels with radius R centered at gc the center pixel.

Integrating the extraction of Gabor filters - LBP and color component a* and b* to be the image features which used as image attributes in clustering using fuzzy c-means (FCM).

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} U_{ij}^{w} \left\|x_i - v_j\right\|^2$$

As shown in Figure 1, the final image segmentation result is obtained from the measures proposed model.

Evaluation of the proposed algorithm with quantitative performance, where segmentation accuracy is estimated using hit rate (Khan, Adhami, & Bhuiyan, 2009). Hit rate is percentage number of pixels classified correctly compared to ground truth data. Ground truth data is label-segmentation of the human hand, which divides the image into some number of segments, where the segments representing parts of the image.

$$hit\ rate = \frac{number\ of\ pixels\ classified\ correctly}{number\ of\ pixels\ ground\ truth}\ x\ 100\%$$

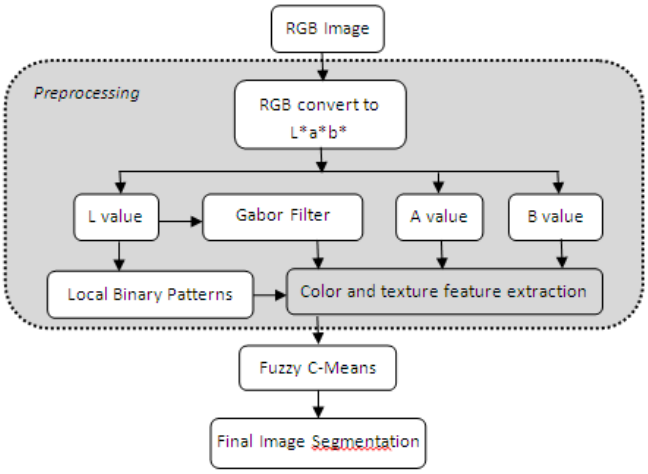


Figure 1.Proposed Method

4 EXPERIMENTAL RESULTS

Experiments and testing the proposed methods using Matlab R2012a. The data has been prepared and separated between training and testing data on preprocessing.

In the first experiment will be the process of transformation to sRGB color models from color space RGB, sRGB to CIE XYZ, CIE XYZ to CIE Lab as a color image extraction process. In this study, using the image processing toolbox of matlab for RBG color model transformation to the L * a * b * with the following command:

```
I_RGB = imread('317080.jpg');
cform = makecform('srgb2lab');
I_lab = applycform(I_RGB,cform);
```

Figure 2. Command Transformation CIE L*a*b* in Matlab

In the Figure 3 can be seen image after transformated from RGB color space to the color space L* a* b*. Taken value L* of the color space L* a* b* as representing the value luminace grayscale image is used for texture feature extraction.

Figure 3. BSDS image 317080 in RGB color space (left), transformation CIE L*a*b* (center) and value L* of CIE L*a*b*

In the process of texture feature extraction using Gabor filters, to build the filter bank must determine the parameters of Gabor functions, including:

a. Frequency (f)

Table 1. Frequency (f), lambda(λ) and sigma (σ)

F	λ	σ
0.18396	5.43590	3.05591
0.21751	4.59759	2.58464
0.23428	4.26846	2.39961
0.24266	4.12095	2.31668
0.24686	4.05096	2.27733
0.25314	3.95031	2.22075
0.25734	3.88595	2.18457
0.26572	3.76331	2.11563
0.28249	3.53989	1.99003
0.31604	3.16418	1.77881

b. Tetha (θ)
In this study using angle as suggested in (Clausi & Jernigan, 2000) are: (θ) = 0º, 30º, 60º, 90º, 120º, 150º

c. Phi (φ)=[0,90]

d. bandwidth (b)=1

e. Gamma (γ)=0.5

with the above parameters will result in the value of Gabor function (gb) which will be convoluted the original image (img) into a Gabor filter.

```
filgb = conv2(img, gb, 'same');
```

Figure 4. Command Convolution in Matlab

For the process of texture feature extraction using Gabor filter delivering sixty texture features. Texture features extraction with Gabor filter the value f = 0.18396, λ = 5.43590, σ = 3.05591 and θ = 0º after getting the value function and convolution with BSDS image as shown in Figure 5.

Figure 5. One of the Gabor filter texture feature extraction

On texture feature extraction using the Local Binary Pattern (LBP) using the L * value as the feature extraction with Gabor filters. L * value of 3 x 3 at BSDS image f (237,157) with the central pixel f (238,158) calculated the value of LBP 198.

Piksel 3 x 3		
196	200	201
202	200	197
201	199	199

Threshold		
0	1	1
1		0
1	0	0

Biner		
1	2	4
128		8
64	32	16

pattern = 11000110
LBP = 2+4+64+128 = 198

Figure 6. Value LBP on BSDS image f (238,158)

To get the value of LBP texture features taken from the histogram by dividing the image blocks with size 3x3.

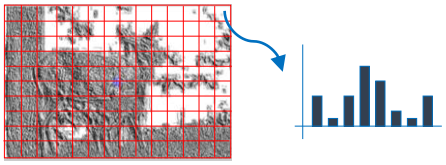


Figure 7. LBP texture feature extraction

Of the features that have been extracted, all combined to form a data vector with attributes, color features a* b*, texture feature using Gabor filters and texture feature using the LBP with records of 151686 pixels segmentation performed by Fuzzy C-Means (FCM).

In BSDS image is a natural image contains many colors and textures so many areas that can be formed in the image segmentation so that the number of clusters is determined two, three and four to produce a segmentation according to the ground truth object. Image segmentation with FCM using fuzzy toolbox in matlab with the command "fcm (data, number_of_clusters)".

Table 2. Data on color and texture features of the image BSDS

i	1	2	3	...	151684	151685	151686
a*	128	128	128	...	138	139	139
b*	128	128	128	...	159	159	158
Gb 1	0.60	0.60	0.61	...	0.47	0.48	0.49
Gb 2	0.43	0.43	0.44	...	0.27	0.24	0.24
...	...	...	...	...	...	...	...
Gb 59	0.51	0.50	0.50	...	0.44	0.59	0.59
Gb 60	0.41	0.41	0.40	...	0.57	0.59	0.52
LBP bn1	0	0	0	...	4	4	4
LBP bn2	3	3	3	...	0	0	0
...	...	...	...	...	...	...	...
LBP bn7	0	0	0	...	0	0	0
LBP bn8	6	6	6	...	1	1	1

Testing the model on the image BSDS by using the ground truth were correctly counted the number of pixels compared to the total number of pixels where the pixels were correctly marked with white color or value 1. From the calculations hit rate in the first experiment in Figure 8 the level of accuracy of 0.7968 (79.68%).

In subsequent experiments carried out in the same way that the segmentation is determined using FCM to cluster two, three and four using color feature grayscale and L* a* b* and texture features Gabor filter and LBP.

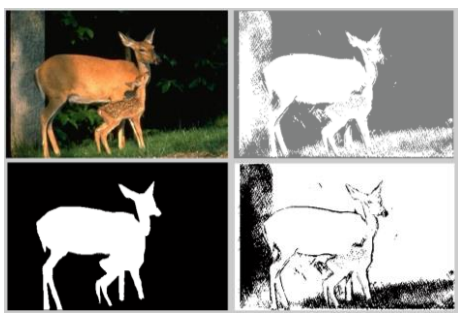


Figure 8. BSDS Image (top left), image segmentation (top right), ground truth (bottom left) and hit rate (bottom right)

Table 3. Experiments and testing models BSDS image

BSDS image	Cluster	Grayscale + Gabor	Grayscale + Gabor + LBP	L*a*b* + Gabor	L*a*b* + Gabor + LBP
317080	2	0.7968	0.7131	0.9582	0.9586
	3	0.5826	0.7490	0.8281	0.8235
	4	0.4908	0.6862	0.5372	0.5524
113016	2	0.5294	0.5208	0.9579	0.9522
	3	0.6484	0.6504	0.9781	0.9784
	4	0.6188	0.7029	0.9755	0.9753
12003	2	0.6855	0.7837	0.7820	0.7764
	3	0.8051	0.8144	0.8176	0.8187
	4	0.5283	0.3763	0.7936	0.7909
296059	2	0.8488	0.7137	0.7445	0.7433
	3	0.6808	0.8875	0.9808	0.9811
	4	0.5064	0.4820	0.6987	0.6007

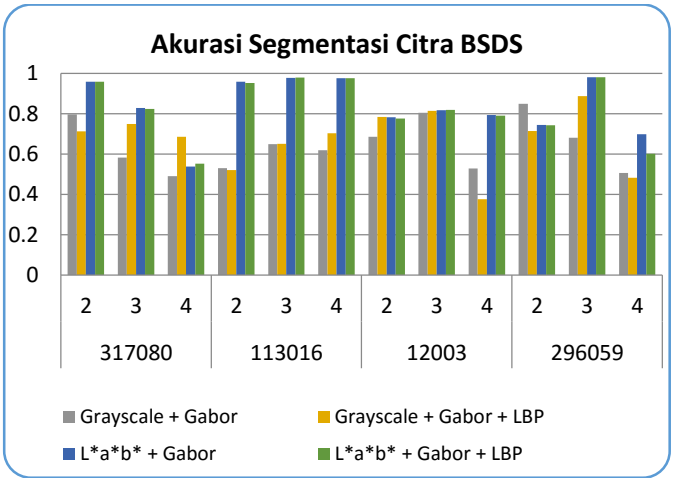


Figure 9. Graphical of accuracy BSDS image segmentation for each cluster

In testing the BSDS image is a natural image with a color features grayscale and the color space L* a* b* combined with texture feature Gabor filters and LBP, was tested on each cluster. Seen from Table 3 a comparison between the features and the cluster is shown in graphical form in Figure 9. Most of the number cluster is value four so levels accuracy to be decreases because the object was split to form a separate region with a central point adjacent to the pixels other than the object. To determine the improvement in the accuracy of each feature is used the highest accuracy value of each image for each feature because its object approaching the ground truth. Then averaged the accuracy of each feature of the image BSDS to each other than the accuracy of the segmentation of the features used in segmentation.

Table 4. The level of accuracy of BSDS image segmentation

Citra BSDS	Grayscale + Gabor	Grayscale + Gabor + LBP	L*a*b* + Gabor	L*a*b* + Gabor + LBP
317080	0.7708	0.7490	0.9584	0.9586
113016	0.6484	0.7029	0.9781	0.9784
12003	0.8051	0.8144	0.8176	0.8187
296059	0.8488	0.8875	0.9808	0.9811
Rata-rata	0.7683	0.7885	0.9337	0.9342

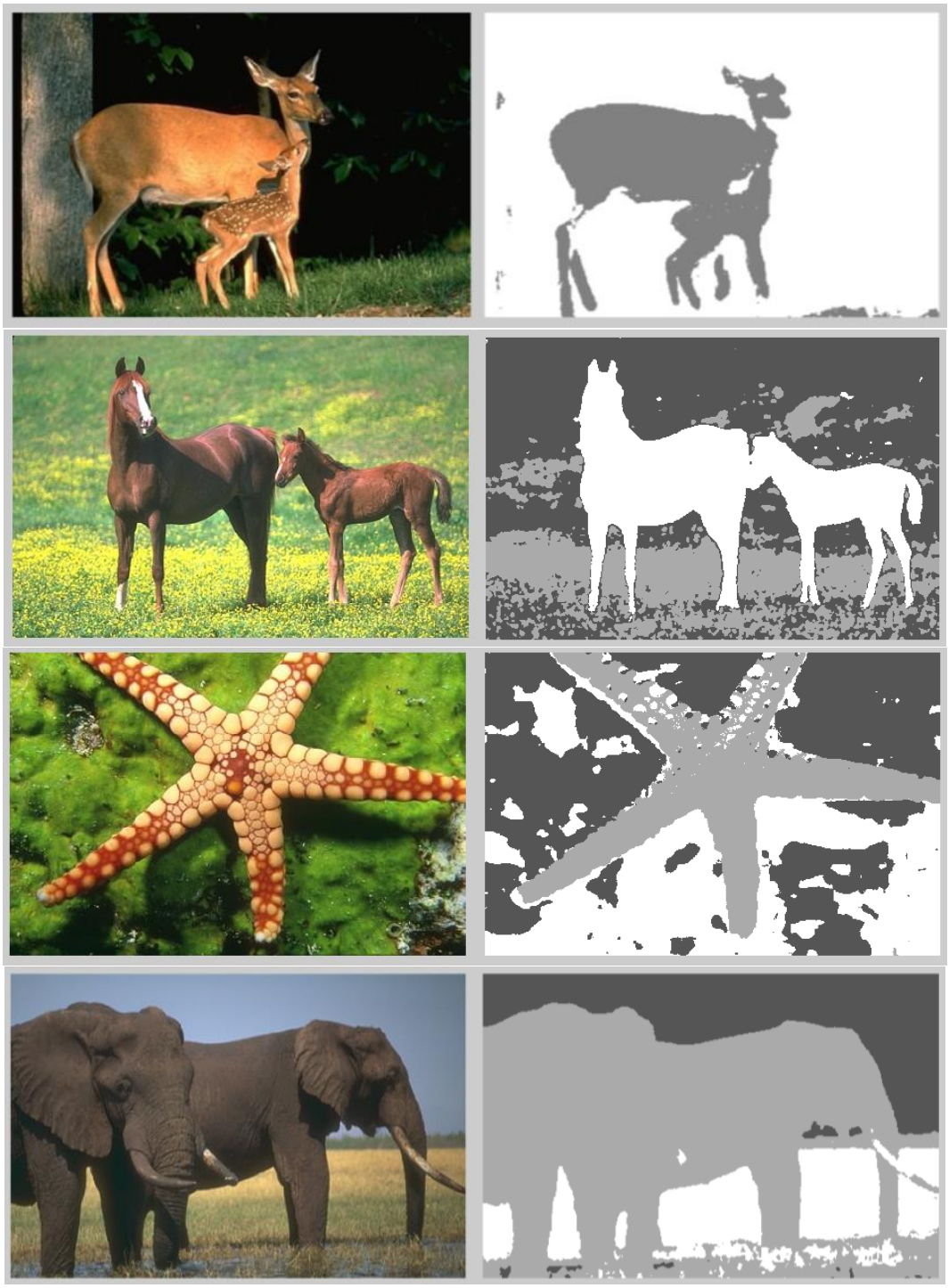


Figure 10. BSDS image (left) and segmentation has a high accuracy rate (right)

5 CONCLUSION

After conducting experiments and testing of BSDS image segmentation using FCM with applying color and texture feature extraction using Gabor filter-LBP then to the use of color features in comparison with the grayscale levels increased 16.54% accuracy for Gabor filter texture of 76.83% to 93.37 % and increased to 14.57% from 78.85% LBP filter becomes 93.42%. While the LBP texture features can help improve the accuracy of image segmentation by 2.02%, although small on color space grayscale from 76.83% to 78.85% and by 0.05% in the color space L* a* b* from 93.37% to 93.42%.

REFERENCES

Abbadeni, N., Zhou, D., & Wang, S. (2000). Computational measures corresponding to perceptual textural features. *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, 897–900.

Aghbari, Z. A., & Al-Haj, R. (2006). Hill-manipulation: An effective algorithm for color image segmentation. *Image and Vision Computing*, *vol. 24*, *no. 8*, 894–903.

Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recognition*, *vol. 34*, *no. 12*, 2259–2281.

Clausi, D. A., & Jernigan, M. E. (2000). Designing Gabor filters for optimal texture separability. *Pattern Recognition*, *vol. 33*, *no. 11*, 1835–1849.

Cremers, D., Rousson, M., & Deriche, R. (2006). A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. *International Journal of Computer Vision*, *vol.72 no.2*, 195–215.

Gonzalez, R. C., & Richard, E. (2002). *Digital Image Processing*, *2nd ed. Upper Saddle River*. New Jersey 07458: Prentice-Hall.

Idrissa, M., & Acheroy, M. (2002). Texture classification using Gabor filters. *Pattern Recognition Letters*, *vol. 23*, *no. 9*, 1095–1102.

Khan, J. F., Adhami, R. R., & Bhuiyan, S. M. (2009). A customized Gabor filter for unsupervised color image segmentation. *Image and Vision Computing*, *vol. 27*, *no. 4*, 489–501.

Krinidis, S., & Chatzis, V. (2010). A robust fuzzy local information C-Means clustering algorithm. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, *vol. 19*, *no. 5*, 1328–37.

Li, M., & Staunton, R. C. (2008). Optimum Gabor filter design and local binary patterns for texture segmentation. *Pattern Recognition Letters*, *vol. 29*, *no. 5*, 664–672.

Mäenpää, T., & Pietikäinen, M. (2006). Texture Analysis With Local Binary Patterns. In C. H. Chen, & P. S. Wang, *Handbook Of Pattern Recognition And Computer Vision*, *3rd ed.* (pp. 197–216). Singapore: World Scientific Publishing Co. Pte. Ltd.

Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., & Yamada, A. (2001). Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, *vol. 11*, *no. 6*, 703–715.

Perner, P. (2006). Case Based Reasoning For Image Analysis And Interpretation. In C. H. Wang, *Handbook Of Pattern Recognition And Computer Vision*, *3rd ed.* (pp. 95–114). Singapore: World Scientific Publishing Co. Pte. Ltd.

Qazi, I.-U.-H., Alata, O., Burie, J.-C., Moussa, A., & Fernandez-Maloigne, C. (2011). Choice of a pertinent color space for color texture characterization using parametric spectral analysis. *Pattern Recognition*, *vol. 44*, *no. 1*, 16–31.

Raut, S. A., Raghuwanshi, M., Dharaskar, R. .., & Raut, A. (2009). Image Segmentation – A State-Of-Art Survey for Prediction. *2009 International Conference on Advanced Computer Control*, 420–424.

Tlig, L., Sayadi, M., & Fnaiech, F. (2012). A new fuzzy segmentation approach based on S-FCM type 2 using LBP-GCO features. *Signal Processing: Image Communication*, *vol. 27*, *no. 6*, 694–708.

Tuceryan, M., & Jain, A. (1999). Texture analysis. In C. H. Chen, L. F. Pau, & P. S. Wang, *Handbook Of Pattern Recognition And Computer Vision*, *2nd ed.* (pp. 207–248). Singapore: World Scientific Publishing Co. Pte. Ltd.

Vandenbroucke, N., Macaire, L., & Postaire, J.-G. (2003). Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis. *Computer Vision and Image Understanding*, *vol. 90*, *no. 2*, 190–216.

Zhang, J., Tan, T., & Ma, L. (2002). Invariant texture segmentation via circular Gabor filters. *in Object recognition supported by user interaction for service robots*, *vol. 2* , 901–904.

BIOGRAPHY OF AUTHORS

Yanuar Wicaksono. Received S.Kom in Informatics Enggenering from AKI University, Semarang-Indonesia, and M.Kom in Informatics Enggenering from Dian Nuswantoro University, Semarang-Indonesia. His current research interests include image processing, and machine learning.

RomiSatriaWahono. Received B.Eng and M.Eng degrees in Computer Science respectively from Saitama University, Japan, and Ph.D in Software Engineering from UniversitiTeknikal Malaysia Melaka. He is a lecturer at the Graduate School of Computer Science, Dian Nuswantoro University, Semarang, 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, PMI and IEEE Computer Society.

Vincent Suhartono. He has received Ing degree on Information Technology and Broadcasting Technology from Fachhochschule Bielefeld, Germany in 1979 and Dipl.-Ing degree on Electronics Technology from Universitaet Bremen, Germany in 1986. And Dr.-Ing from the faculty of Electrical Engineering and Intelligence Control from Universitaet Bremen Germany in 1999. He is a lecturer in the faculty of Computer Science, Dian Nuswantoro University, Semarang. His current research interests include Artificial Intelligence and Robotics.

Pemilihan Parameter Smoothing pada Probabilistic Neural Network dengan Menggunakan Particle Swarm Optimization untuk Pendeteksian Teks Pada Citra

Endah Ekasanti Saputri
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Email: endah.ekasanti@gmail.com

Romi Satria Wahono dan Vincent Suhartono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Email: romi@brainmatics.com, vincent.suhartono@dsn.dinus.ac.id

Abstract: Teks sering dijumpai di berbagai tempat seperti nama jalan, nama toko, spanduk, penunjuk jalan, peringatan, dan lain sebagainya. Deteksi teks terbagi menjadi tiga pendekatan yaitu pendekatan tekstur, pendekatan edge, dan pendekatan Connected Component. Pendekatan tekstur dapat mendeteksi teks dengan baik, namun membutuhkan data training yang banyak. Probabilistic Neural Netwok (PNN) dapat mengatasi permasalahan tersebut. Namun PNN memiliki permasalahan dalam menentukan nilai parameter smoothing yang biasanya dilakukan secara trial and error. Particle Swarm Optimization (PSO) merupakan algoritma optimasi yang dapat menangani permasalahan pada PNN. Pada penelitian ini, PNN digunakan pada pendekatan tekstur guna menangani permasalahan pada pendekatan tekstur, yaitu banyaknya data training yang dibutuhkan. Selain itu, digunakan PSO untuk menentukan parameter smoothing pada PNN agar akurasi yang dihasilkan PNN-PSO lebih baik dari PNN tradisional. Hasil eksperimen menunjukkan PNN dapat mendeteksi teks dengan akurasi 75,42% hanya dengan mengunakan 300 data training, dan menghasilkan 77,75% dengan menggunakan 1500 data training. Sedangkan PNN-PSO dapat menghasilkan akurasi 76,91% dengan menggunakan 300 data training dan 77,89% dengan menggunakan 1500 data training. Maka dapat disimpulkan bahwa PNN dapat mendeteksi teks dengan baik walaupun data training yang digunakan sedikit dan dapat mengatasi permasalahan pada pendekatan tekstur. Sedangkan, PSO dapat menentukan nilai parameter smoothing pada PNN dan menghasilkan akurasi yang lebih baik dari PNN tradisional, yaitu dengan peningkatan akurasi sekitar 0,1% hingga 1,5%. Selain itu, penggunaan PSO pada PNN dapat digunakan dalam menentukan nilai parameter smoothing secara otomatis pada dataset yang berbeda.

Keywords: deteksi teks, pendekatan tekstur, probabilistic neural network, particle swarm optimization, parameter smoothing

1 PENDAHULUAN

Teks atau tulisan sering kita jumpai di berbagai tempat seperti nama jalan, nama toko, spanduk, penunjuk jalan, peringatan, dan lain sebagainya. Teks memiliki peranan penting dalam kehidupan sehari-hari karena teks kaya akan informasi (Karaoglu, Fernando, & Tremeau, 2010) (Epshtein, Ofek, & wexler, 2010) (Meng & Song, 2012) (Angadi & Kodabagi, 2010). Hal ini dapat menjadi masalah pada seorang turis yang tidak mengerti bahasa sekitar, serta seorang yang memiliki masalah penglihatan. Untuk itu, saat ini banyak dibangun suatu perangkat seperti tourist assistance, atau sebuah sistem yang dapat membantu seorang dengan masalah penglihatan untuk dapat mengelilingi kota dan melakukan aktifitas sehari-hari. Situasi perkotaan dapat dianalisa secara simultan dan ditambah dengan algoritma text-to-speech, membuat perangkat tersebut dapat membaca tanda-tanda jalan, label pada pusat perbelanjaan, dan sebagainya (Karaoglu, Fernando, & Tremeau, 2010). Sehingga perangkat tersebut diharapkan dapat membantu seorang yang memiliki masalah penglihatan.

Ketersediaan perangkat yang murah seperti handphone, menjadikan deteksi teks pada citra menjadi luas. Karena pengambilan citra dengan kamera memiliki banyak masalah seperti orientasi, pencahayaan, low resolution, blur, menjadikan deteksi teks menjadi penelitian yang menarik. Dan mendorong para peneliti untuk mencoba mencari solusi untuk menemukan desain sistem sesederhana mungkin yang dapat menangani permasalahan yang ada (Karaoglu, Fernando, & Tremeau, 2010). Selain itu, menurut penelitian (Karaoglu, Fernando, & Tremeau, 2010) (Meng & Song, 2012) (Shivakumara, Phan, & Tan, 2011), deteksi teks pada citra juga memiliki banyak permasalahan seperti ukuran tulisan, jenis tulisan dan warna tulisan yang berbeda, serta background yang kompleks.

Berdasarkan penelitian (Shivakumara, Phan, & Tan, 2011) (Shi, Xiao, Wang, & Zhang, 2012) (Angadi & Kodabagi, 2010), metode deteksi teks pada citra alam, secara umum dapat dikategorikan menjadi tiga, yaitu pendekatan Connected Component (CC), batas tepi (edge based approach), dan tekstur. (Karaoglu, Fernando, & Tremeau, 2010) melakukan penelitian dengan pendekatan CC, namun CC tidak dapat bekerja dengan baik pada garis teks yang memiliki kontras yang rendah (garis teks yang terputus-putus) serta bermasalah pada complex background. Permasalahan kontras yang rendah pada CC ini, dapat ditangani oleh pendekatan batas tepi. (Bai, Yin, & Liu, 2012) melakukan penelitian dengan menggunakan *edge strength* dan *variance of orientation* namun pendekatan ini juga memiliki permasalahan pada tingginya nilai komputasi pada dataset yang besar. Dilain pihak, (Huang, 2012) melakukan penelitian dengan menggunakan pendekatan tekstur yang dapat menangani permasalahan pada pendekatan batas tepi dan *complex background* pada CC dengan cara mempertimbangkan bahwa teks memiliki tekstur yang berbeda. Namun pendekatan tekstur membutuhkan jumlah sample yang besar untuk data training teks dan non-teks.

Pada penelitian ini, akan digunakan pendekatan tekstur dalam mendeteksi teks pada citra alam. Secara umum, ketiga pendekatan tersebut memiliki beberapa tahapan, yaitu

candidate text region extraction, text line localization, feature extraction, training, verifikasi (Karaoglu, Fernando, & Tremeau, 2010). *Feature extraction* yang biasa digunakan oleh para peneliti khususnya pada pendekatan tekstur, adalah Gabor Wavelet. Seperti penelitian yang dilakukan oleh (Shivakumara & Tan, 2010) (Wang & Wang, 2010) (Angadi & Kodabagi, 2010) (Le, Dinh, Kim, & Lee, 2010) (Huang, 2012) (Lee, Lee, Lee, Yuille, & Koch, 2011). Sedangkan pada tahap verifikasi, dibutuhkan suatu algoritma / metode *learning* untuk membedakan citra teks dan non-teks dari citra yang dihasilkan pada tahap sebelumnya.

Algoritma yang biasanya digunakan pada pendekatan texture dalam mendeteksi teks adalah Support Vector Machine(SVM), Random Forest, K-Means, dan ANN. SVM dapat menghasilkan akurasi yang tinggi, dan dapat menangani kasus dengan dimensi besar, namun membutuhkan data training yang besar. Sedangkan Random Forest, memiliki kelebihan pada proses *learning* yang cepat, dapat menangani input variabel yang besar, dan dapat menghasilkan akurasi tinggi. Namun Random Forest ini bermasalah pada waktu komputasi yang lama. Selain itu, K-Means memiliki kelebihan dalam menangani background yang kompleks. Namun bermasalah pada kontras yang rendah dan waktu komputasi yang tinggi.

Probabilistic Neural Network (PNN) merupakan algoritma klasifikasi dan merupakan suatu algoritma ANN yang menggunakan fungsi *probabilistic*, tidak membutuhkan dataset yang besar dalam tahap pembelajarannya, serta memiliki kelebihan yaitu dapat mengatasi permasalahan yang ada pada Back-Propagation(BP) yaitu dapat mengatasi waktu pelatihan yang lama, terjebak pada global minimum, dan sulitnya perancangan arsitektur jaringan (Spech, 1990). Berdasarkan paper yang ada, PNN dapat digunakan untuk mengklasifikasikan secara akurat pada beberapa penelitian serta memiliki beberapa kelebihan dibandingkan dengan BP. Sehingga, algoritma ini diharapkan dapat digunakan untuk memperbaiki kekurangan yang ada pada pendeketan tekstur, yaitu dengan menggunakan PNN pada tahap verifikasi citra. Namun, PNN memiliki masalah pada penentuan parameter smoothing yang biasanya ditentukan dengan cara *trial and error* atau *user_defined* (Yang & Yang, 2012). Untuk itu, diperlukan suatu metode optimasi yang dapat menentukan parameter *smoothing* yang paling optimal pada PNN.

Pada beberapa penelitian, penentuan parameter *smoothing* pada PNN dioptimasi dengan menggunakan Genetic Algorithm(GA) (Yang & Yang, 2012), berdasarkan Centre Neighbor (Liu, Wang, & Cheng, 2011) menggunakan PSO dalam menentukan parameter smoothing pada PNN dan hasil eksperimen menunjukkan tingkat akurasi yang dihasilkan oleh PSO-PNN lebih tinggi dari tradisional PNN, yaitu 63,31% untuk tradisional PNN dan 93,95% untuk PSO-PNN. Sehingga PSO diharapkan dapat mengatasi masalah yang ada pada PNN, yaitu dalam menentukan nilai *smoothing* yang sesuai, sehingga dapat menghasilkan klasifikasi teks yang lebih akurat.

Pada penelitian ini, akan menggunakan pendekatan tekstur, yaitu dengan menerapkan Probabilistic Neural Network(PNN) dan menerapkan Particle Swarm Optimization(PSO) untuk menentukan parameter *smoothing* yang ada pada PNN. Sehingga diharapkan dapat menghasilkan akurasi yang lebih akurat.

2 PENELITIAN TERKAIT

Permasalahan yang ada pada deteksi teks, yaitu banyaknya variasi warna, orientasi, pencahayaan, *low resolution*, *blur*, *noise*. Selain itu, banyak metode yang diusulkan yang belum dapat menangani permasalahan tersebut dengan baik. Seperti penelitian yang dilakukan (Angadi & Kodabagi, 2010). Latar belakang dari penelitian yang dilakukan Angadi adalah kurang efisiennya pendekatan Connected Component (CC) dan pendekatan edge dalam mengatasi gambar yang memiliki *noise*, *low resolution image* dan *complex background*. Penelitian yang dilakukan Angadi, menggunakan Discrete Cosine Transform (DCT) pada tahap *preprocessing*, Homogeneity Function pada tahap *Feature Extraction*, dan pada tahap *classification* menggunakan Discriminant Function, dan menghasilkan akurasi sebesar 96,6%.

Selain itu, (Ji, Xu, Yao, Zhang, Sun, & Liu, 2008) melakukan penelitian berdasarkan permasalahan pada pencahayaan dan kontras yang tidak konsisten pada gambar yang akan dideteksi. Penelitian ini menggunakan Pyramid Haar pada tahap *preprocessing*, Haar Wavelet pada tahap *Feature Extraction*, dan selanjutnya menggunakan Directional Correlation Analysis (DCA) pada tahap *classification*. Penelitian ini menghasilkan precision (p) 0.50, recall (r) 0.79, dan system performance (f) 0.68.

Sedangkan penelitian yang dilakukan oleh (Pan, Liu, & Hou, 2010) dilandasi oleh hasil kandidat teks pada pendekatan CC yang masih terlalu kasar dan dapat mempersulit proses *verification*. Penelitian yang dilakukan Pan, menggunakan boosted classifier dan polynomial classifier pada tahap *preprocessing*, Gabor Wavelet pada tahap Feature Extraction, dan Histogram of Oriented Gradients, Local Binary Pattern, Discrete Cosine Transform pada tahap *classification*.

Pada penelitian ini menggunakan pendekatan texture yang dapat menangani permasalahan pada deteksi teks yaitu pada *low resolution image*, blur, pencahayaan, kontras, variasi warna, ukuran dan jenis tulisan dengan menggunakan gabungan Probabilistic Neural Network (PNN) dan particle Swarm Optimization (PSO) pada tahap classification.

3 METODE YANG DIUSULKAN

Pada penelitian ini, data yang digunakan adalah dataset ICDAR Competition 2003 dan ICDAR Competition 2011. Sedangkan, metode yang diusulkan pada penelitian ini adalah menggunakan pendekatan tekstur. Gambar alam yang akan dideteksi, dipecah atau dipotong-potong menjadi blok-blok kecil dengan ukuran 70x70 pixel dan 100x100 pixel. Kemudian, blok tersebut diubah kedalam *grayscale image* yang selanjutnya diambil cirinya dengan metode *feature extraction* Gabor Wavelet. Setelah itu, ciri dari blok-blok tersebut dikenali sebagai teks atau non teks dengan metode *classification* Probabilistic Neural Network (PNN) yang telah di-*improved* dengan suatu metode optimasi Particle Swarm Optimization (PSO). Masalah pada penelitian ini adalah sulitnya menentukan nilai parameter *smoothing* pada PNN. Sedangkan nilai parameter *smoothing* pada PNN ini sangat menentukan tingginya tingkat akurasi yang dihasilkan oleh PNN dalam mendeteksi teks.

Tahapan pada proses *training* PNN adalah:

1. Menghitung Total Minimum Distance (TMD) pada masing-masing kelas dengan menggunakan Persamaan:

$$\mathrm{TMD} = \sum_{i=1}^{n} |x_1(i) - x_2(i)|$$

2. Kemudian menghitung Parameter Smoothing pada setiap kelas dengan rentang nilai g sesuai inputan *user* menggunakan Persamaan:

$$S = \frac{g * TMD_0}{\text{jumlah pola latih kelas} - n}$$

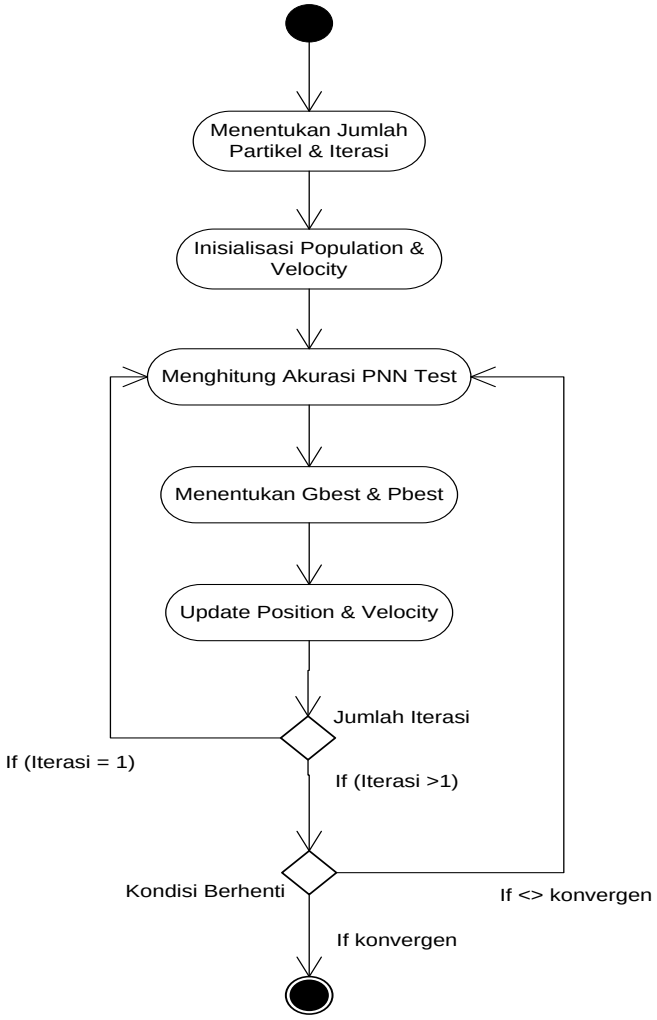
3. Setelah itu, menghitung probabilitas setiap pola terhadap kedua kelas dengan Persamaan :

$$p(x|C_k) = \frac{1}{(2\pi)^{m/2}\sigma_k^m|C_k|} \sum_{\rho_i \in C_k} \exp[-\|x - w_i\|^2/(2\sigma_k^2)]$$

4. Dan yang terakhir, mencari nilai maksimum dari probabilitas kedua kelas pada suatu pola.

Sedangkan pada proses *testing*, PNN menghitung Total Minimum Distance (TMD) antara pola *testing* dengan pola *training* sesuai dengan kelasnya. Dengan menggunakan parameter *smoothing* yang sudah dihitung pada tahap *training*, kemudian PNN menghitung probabilitas pola terhadap kedua. Berdasarkan probabilitas tersebut, dicari nilai maksimum dari probabilitas kedua kelas.

Pemilihan parameter *smoothing* pada Probabilistic Neural Network (PNN) biasanya dilakukan secara *trial and error*. Seperti pada Gambar 1, penelitian ini menggunakan Particle Swarm Optimization (PSO) dalam menentukan nilai dari parameter *smoothing*. Nilai parameter smoothing yang digunakan diperoleh dari perhitungan PSO terhadap akurasi pada PNN Test. Nilai parameter smoothing yang menghasilkan akurasi PNN Test terbaiklah yang digunakan pada proses selanjutnya.



Gambar 1. Diagram Aktifitas PNN-PSO

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*).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

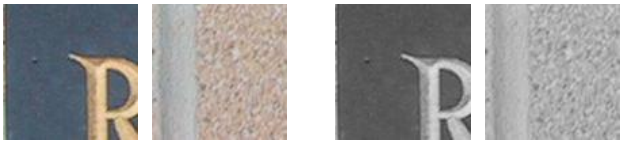
Semakin tinggi nilai akurasinya, semakin baik pula metode yang dihasilkan.

4 HASIL EKSPERIMEN

Eksperimen dilakukan dengan menggunakan Matlab R2009b pada Intel Core 2 Duo, 1GB RAM, 80GB HD, dan system operasi Windows XP SP1.

Pada penelitian ini, gambar yang menjadi inputan awal adalah gambar alam yang memiliki teks atau tulisan di dalamnya, seperti Gambar 2. Kemudian gambar tersebut masuk pada tahap *preprocessing* yaitu pemotongan gambar menjadi 100x100 pixel (Gambar 3a) dan mengubah gambar yang sudah dipotong tersebut menjadi *grayscale image* (Gambar 3b) untuk memperkecil dimensi warna yang terkandung di dalam gambar tersebut.

Gambar 2. Gambar yang mengandung teks



a) b)

Gambar 3. Blok gambar 100x100 pixel

Setelah tahap *preprocessing*, gambar tersebut diambil cirinya dengan metode ekstraksi ciri Gabor Wavelet, yaitu dengan membangun matrix konvolusi, yaitu *real* dan *imaginer*, dan dioperasikan dengan matrix asli. Sehingga menghasilkan matrix ciri yang digunakan pada tahap selanjutnya, yaitu *classification*.

Proses *training* pada PNN menghasilkan nilai *Total Minimum Distance*(TMD) yang digunakan pada proses selanjutnya, yaitu tahap validasi dan *testing*. Seperti pada Tabel 2 berikut, dengan mengubah-ubah jumlah data training yang digunakan, akan berbeda pula nilai TMD yang dihasilkan.

Tabel 2. Total Minimum Distance Hasil Proses Training

Jumlah Data Training	300 (Positif = 100; Negatif = 200)	600 (Positif = 200; Negatif = 400)	900 (Positif = 300; Negatif = 600)	1200 (Positif = 400; Negatif = 800)	1500 (Positif = 500; Negatif = 1000)
TMD kelas 1 (teks)	23.50	40.31	59.25	77.58	97.48
TMD kelas 2 (non-teks)	13.93	23.61	34.36	41.91	51.49

Eksperimen dilakukan dengan melakukan percobaan pada jumlah data yang digunakan pada tahap *training* PNN. Percobaan ini dilakukan untuk mengetahui seberapa berpengaruh jumlah data yang digunakan pada tahap training dan kombinasinya terhadap parameter *smoothing* dengan akurasi yang dihasilkan system.

Setelah dilakukan eksperimen terhadap PNN, nilai parameter *smoothing* yang sesuai pada studikasus dan data *training* yang digunakan adalah antara 0,8 dan 1. Sesuai penelitian yang dilakukan (Liu, Wang, & Cheng, 2011) mengatakan bahwa nilai parameter *smoothing* dapat berubah ubah sesuai dengan data *training* yang digunakan.

Selanjutnya, kita bandingkan akurasi yang dihasilkan. Grafik pada Gambar 7 menunjukkan akurasi yang dihasilkan PNN-PSO lebih baik dari PNN tradisional. Jika dihitung secara rata-rata, akurasi yang dihasilkan PNN adalah 74,7%, sedangkan PNN-PSO menghasilkan akurasi 77,12%.

Tabel 4. Hasil Komparasi PNN dan PNN-PSO

Algoritma	Akurasi
PNN	74,7%
PNN-PSO	77,12%

Gambar 7. Grafik Komparasi PNN dan PNN-PSO

Pemilihan jumlah iterasi dan populasi pada PSO juga mempengaruhi akurasi dan waktu yang dibutuhkan dalam mendeteksi teks. Seperti yang diungkapkan (Jorhedi & Jasni, 2013) (Kennedy & Eberhart, 1995) dalam penelitiannya, bahwa jumlah partikel yang terlalu banyak akan menyebabkan algoritma terjebak pada local optima dan dapat memperberat proses komputasi algoritma yang mengakibatkan lamanya waktu komputasi. Jumlah partikel yang menghasilkan akurasi tertinggi dihasilkan dengan jumlah partikel 100.

Sedangkan, *stopping criteria* yang digunakan pada penelitian ini adalah jumlah iterasi. Jumlah iterasi ini disesuaikan dengan kasus yang ditangani pada suatu penelitian. akurasi tertinggi lebih banyak dihasilkan pada jumlah iterasi sama dengan 1000. Seperti pada penelitian (Jorhedi & Jasni, 2013) (Kennedy & Eberhart, 1995)mengatakan bahwa PSO akan membutuhkan iterasi yang lebih banyak untuk mendapatkan global optimum dan menghindari kegagalan dalam menemukan global optimum. Sebab, semakin banyak iterasi yang digunakan pada suatu kasus, akan membuat pergerakan partikel lebih tersebar keseluruh daerah pencarian.

5 KESIMPULAN

Pada penelitian ini dilakukan pengujian model dengan menggunakan Probabilistic Neural Network (PNN) dan Probabilistic Neural Network & Particle Swarm Optimization (PNN-PSO). Beberapa percobaan dilakukan dengan mengkombinasikan beberapa parameter pada PNN dan PSO untuk mendapatkan akurasi terbaik dalam mendeteksi teks pada gambar alam. Parameter PNN yang diujicoba adalah jumlah data *training*, sedangkan parameter PSO yang diujicoba adalah jumlah partikel pada suatu populasi, dan jumlah maksimum iterasi. Hasil percobaan menunjukkan: 1) dalam mendeteksi teks, PNN dapat menghasilkan akurasi 77,75% dengan menggunakan nilai parameter smoothing 0,8 dan jumlah data training 1500. 2) Sedangkan PNN-PSO dapat menghasilkan akurasi 77,89% pada jumlah data training 1500, jumlah populasi 50, dan maksimum iterasi 1000. Dari hasil pengujian diatas, dapat disimpulkan bahwa: 1) penggunaan PNN pada pendekatan tekstur dalam mendeteksi teks pada gambar alam tidak harus menggunakan data training yang banyak. 2) Penggunaan PSO dalam menentukan parameter smoothing pada PNN dapat meningkatkan akurasi, dan dapat digunakan dalam menentukan parameter *smoothing* secara otomatis pada dataset yang berbeda.

REFERENSI

Angadi, S. A. (2010). Text Region Extraction from Low Resolution Natural Scene Images using Texture Features. *International Advance Computing Conference*, 121-128.

Bai, B., Yin, F., & Liu, C. (2012). A Fast Stroke-Based Method for Text Detection in Video. *IAPR International Workshop on Document Analysis Systems*, 69-73.

Donald, F. S. (1990). Probabilistic Neural Networks. *Neural Network I Pergamon Press pie Original Contribution*, 3.

Epshtein, B., Ofek, E., & Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. *Computer Vision and Pattern Recognition (CVPR)*, 10, 2963-2970.

Huang, X. (2012). Automatic Video Text Detection and Localization Based on Coarseness Texture. *International Conference on Intelligent Computation Technology and Automation*, 1(2).

Jamil, A., Siddiqi, I., Arif, F., & Raza, A. (2011). Edge-based Features for Localization of Artificial Urdu Text in Video Images. *International Conference on Document Analysis and Recognition*, 1120-1124.

Ji, R., Xu, P., Yao, H., Zhang, Z., Sun, X., & Liu, T. (2008). Directional correlation analysis of local Haar binary pattern for text detection. *IEEE International Conference on Multimedia and Expo*, 885-888.

Jorhedi, R, A., & Jasni, J. (2013). Parameter Selection in Particle Swarm Optimization: a survey. *Journal of Experimental & Theoretical Artificial Intelligence*, 25, 527–542.

Karaoglu, S., Fernando, B., Trémeau, A., & Etienne, S. (2010). A Novel Algorithm for Text Detection and Localization in Natural Scene Images. *IEEE Digital Image Computing : Techniques and Applications*, 641-648.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. 1942-1948.

Le, H. P., Dinh, T. N., Kim, S. H., & Lee, G. S. (2010). Text Detection in Binarized Text Images of Korean Signboard by Stroke Width Feature. *IEEE International Conference on Computer and Information Technology (CIT 2010)*, 1588-1592.

Lee, J., Lee, P., Lee, S., Yuille, A., & Koch, C. (2011). AdaBoost for Text Detection in Natural Scene. *International Conference on Document Analysis and Recognition*.

Liu, B., Wang, H., & Cheng, X. (2011). Exchange Rate Forecasting Method Based on Particle Swarm Optimization and Probabilistic Neural Network Model. *International Conference on Network Computing and Information Security*, 288–292.

Meng, Q., & Song, Y. (2012). Text Detection in Natural Scenes with Salient Region. *IAPR International Workshop on Document Analysis System*, 384-388.

Pan, Y,F., Liu, C., & Hou, X. (2010). Fast scene text localization by learning-based filtering and verification. *IEEE International Conference on Image Processing*, 2269-2272.

Shi, C., Xiao, B., Wang, C., & Zhang, Y. (2012). Graph-based Background Suppression For Scene Text Detection. *IAPR International Workshop on Document Analysis System*, 210-214.

Shivakumara, P., & Tan, C. L. (2010). New Wavelet and Color Features for Text Detection in Video. *International Conference on Pattern Recognition.*

Shivakumara, P., Phan, T. Q., & Tan, C. L. (2011). A Laplacian Approach to Multi-Oriented Text Detection in Video. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(2), 412-419.

Wang, C., & Wang, H. (2010). Utilization of Temporal Continuity in Video Text Detection. *International Conference on MultiMedia and Information Technology*, 343-346.

Wen-xu, W. (2011). Fault diagnosis of transformer based on probabilistic neural network. *International Conference on Intelligent Computation Technology and Automation*.

Yang, H. (2012). An Improved Probabilistic Neural Network with GA Optimization. *International Conference on Intelligent Computation Technology and Automation*, 76-79.

Zhang, H., Zhao, K., Song, Z,Y., & Guo, J. (2013). Text extraction from natural scene image: A survey. *Neurocomputing*, 122, 310-323.

Zhao, H., Liu, C., Wang, H., & Li, C. (2010). Classifying ECoG Signals Using Probabilistic Neural Network. *WASE International Conference on Information Engineering.*

BIOGRAFI PENULIS

Endah Ekasanti Saputri. Lahir pada tanggal 27 Oktober 1988 di Kab.Sintang, Kalimantan Barat. Memperoleh gelar Sarjana Teknik(S.T) dari fakultas Teknik Informatika, Institut Teknologi Telkom, Bandung (sekarang: Universitas Telkom) pada tahun 2010. Serta memperoleh gelar M.Kom dari Fakultas Ilmu Komputer, Universitas Dian Nuswantoro pada tahun 2014.

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.

Vincent Suhartono. Memperoleh gelar Ing pada Information Technology and Broadcasting Technology dari Fachhochschule Bielefeld, Germany pada tahun 1979. Dan memperoleh gelar Dipl.-Ing pada Electronics Technology, Universitaet Bremen, Germany pada tahun 1986. Selain itu memperoleh Dr.-Ing dari fakultas Electrical Engineering and Intelligence Control, Bremen Germany pada tahun 1999. Merupakan pengajar di fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang.

Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree

Achmad Bisri
Fakultas Teknik, Universitas Pamulang
Email: achmadbizri@gmail.com

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

Abstract: Universitas Pamulang salah satu perguruan tinggi yang memiliki jumlah mahasiswa yang besar, namun dalam data histori terdapat masalah dengan jumlah kelulusan yang tepat waktu dan terlambat (tidak tepat waktu) yang tidak seimbang. Metode decision tree memiliki kinerja yang baik dalam menangani klasifikasi tepat waktu atau terlambat tetapi decision tree memiliki kelemahan dalam derajat yang tinggi dari ketidakseimbangan kelas (*class imbalance*). Untuk mengatasi masalah tersebut dapat dilakukan dengan sebuah metode yang dapat menyeimbangkan kelas dan meningkatkan akurasi. Adaboost salah satu metode *boosting* yang mampu menyeimbangkan kelas dengan memberikan bobot pada tingkat *error* klasifikasi yang dapat merubah distribusi data. Eksperimen dilakukan dengan menerapkan metode adaboost pada decision tree (DT) untuk mendapatkan hasil yang optimal dan tingkat akurasi yang baik. Hasil ekperimen yang diperoleh dari metode decision tree untuk akurasi sebesar 87,18%, AUC sebesar 0,864, dan RMSE sebesar 0,320, sedangkan hasil dari decision tree dengan adaboost (DTBoost) untuk akurasi sebesar 90,45%, AUC sebesar 0,951, dan RMSE sebesar 0,273, maka dapat disimpulkan dalam penentuan kelulusan mahasiswa dengan metode decision tree dan adaboost terbukti mampu menyelesaikan masalah ketidakseimbangan kelas dan meningkatkan akurasi yang tinggi dan dapat menurunkan tingkat *error* klasifikasi.

Keywords: kelulusan, ketidakseimbangan kelas, decision tree, adaboost

1. PENDAHULUAN

Angka partisipasi mahasiswa di setiap tahun akademik terjadi peningkatan, maka daya tampung dan tingkat kelulusan pun perlu diperhatikan dan menjadi bagian terpenting untuk evaluasi penentuan kelulusan dan dapat dijadikan sebagai bahan pendukung dalam pengambilan keputusan.

Metode klasifikasi banyak digunakan oleh para peneliti seperti Decision Tree (DT) untuk prediksi kelulusan (Undavia, Dolia, & Shah, 2013), Artificial Neural Networks (ANNs) untuk prediksi hasil kelulusan (Karamouzis & Vrettos, 2008), model klasifikasi dengan pemberian bobot menggunakan Algoritma Genetika (GA) (Minaei-Bidgoli, Kashy, Kortemeyer, & Punch, 2013).

Neural network (NN) memiliki kelebihan pada prediksi non-linier, kuat pada *parallel processing* dan kemampuan untuk mentoleransi kesalahan, tetapi memiliki kelemahan pada perlunya data training yang besar, *over-fitting*, rendahnya konvergensi, dan sifatnya yang *local optimum* (Capparuccia, Leone, & Marchitto, 2007). Decision tree (DT) dapat memecahkan masalah neural network yaitu menangani *over-fitting*, menangani atribut yang kontinu, memilih yang tepat untuk *attribute selection*, menangani *training data* dengan nilai atribut yang hilang, dan meningkat efisiensi komputasi (Quinlan, 1993), pada umumnya tingkat keberhasilan dari *decision tree* difokuskan pada dataset yang relatif seimbang (Cieslak, Hoens, Chawla, & Kegelmeyer, 2012), tetapi decision tree memiliki kelemahan misalnya *entropy* dan *gini* ketika dataset memiliki derajat yang tinggi dari *class imbalance* (Cieslak, Hoens, Chawla, & Kegelmeyer, 2012).

Distribusi *class imbalance* dari sebuah *dataset* yang telah menimbulkan kesulitan yang serius pada sebagian besar algoritma pembelajaran *classifier*, yang mengasumsikan bahwa distribusi yang relatif seimbang (Sun, Kamel, Wong, & Wang, 2007). Distribusi *class imbalance* dapat ditandai sebagai sesuatu yang memiliki lebih banyak kasus dari beberapa *class* yang lain, masalah keseimbangan adalah salah satu dimana satu *class* diwakili oleh sampel yang besar, sedangkan yang lainnya hanya diwakili oleh beberapa sampel (Sun, Kamel, Wong, & Wang, 2007), pada umumnya klasifikasi standar memiliki kinerja yang buruk pada dataset *imbalance* karena mereka dirancang untuk menjeneralisasi dari data *training* dan hasil dari hipotesis yang paling sederhana adalah yang paling sesuai dengan data.

Penanggulangan *class imbalance* dalam pendistribusian secara signifikan dapat dibantu oleh metode sampling (Hulse & Khoshgoftaar, 2009). Salah satu kebutuhan untuk memodifikasi distribusi data, dikondisikan pada fungsi evaluasi. *Re-sampling*, dengan mengembangkan *class* minoritas (positif) atau mengempiskan *class* mayoritas (negatif) dari sebuah dataset yang diberikan, telah menjadi standar *de-facto* untuk mengatasi *class imbalance* dalam berbagai domain (Chawla, Cieslak, Hall, & Joshi, 2008). Beberapa teknik untuk mengatasi *class imbalance* seperti *over-sampling* cenderung mengurangi jumlah pemangkasan yang terjadi, sedangkan *under-sampling* sering membuat pemangkasan yang tidak perlu (Drummond & Holte, 2003), Drummond dan Holte dengan menggunakan metode C4.5 menemukan, bahwa *under-sampling* mayoritas adalah lebih efektif dalam menangani masalah *class imbalance* dan *over-sampling* minoritas sering menghasilkan sedikit atau tidak ada perubahan dalam kinerja. Selain itu juga mereka mencari bahwa *over-sampling* dapat dibuat *cost-sensitive* jika pemangkasan dan parameter berhenti diawal ditetapkan secara proporsional dengan jumlah lebih dari *over-sampling* yang dilakukan (Drummond & Holte, 2003).

Maka dari itu *decision tree* dengan kasus *class imbalance* diperlukan metode yang dapat mengatasi masalah tersebut untuk meningkatkan kinerja klasifikasi *decision tree* agar dapat menghasilkan kinerja yang lebih baik. Algoritma *boosting* adalah algoritma iteratif yang memberikan bobot yang berbeda

pada distribusi training data pada setiap iterasi. Setiap iterasi *boosting* menambahkan bobot pada contoh-contoh kesalahan klasifikasi dan menurunkan bobot pada contoh klasifikasi yang benar, sehingga secara efektif dapat merubah distribusi pada data training (Kotsiantis, Kanellopoulos, & Pintelas, 2006, hal. 25-36). Metode *Boosting* (AdaBoost) yang diusulkan (Kotsiantis & Pintelas, 2009, hal. 123) dengan *selective costing ensemble* dapat menjadi solusi yang lebih efektif untuk masalah *class imbalance* dan memungkin meningkatkan identifikasi dari *class* minoritas yang sulit serta menjaga kemampuan klasifikasi dari *class* mayoritas. Karena adaboost metode pembelejaran *ensamble* yang dapat mengurangi varian, hal ini terjadi karena efek bias rata-rata *ensamble* untuk mengurangi varian dari satu set pengkalsifikasian. Bias dapat dicirikan sebagai ukuran kemampuan untuk menjeneralisasi benar untuk satu set tes. Selain itu pendekatan untuk mengatasi masalah tersebut dapat dilakukan dengan beberapa metode (Weiss, McCarthy, & Zabar, 2007) yaitu *over-sampling*, *under-sampling*, dan *cost-sensitive*.

Pada penelitian ini yang akan dilakukan adalah penerapan adaBoost untuk penyelesaian *class imbalance* pada *decision tree* untuk penentuan kelulusan, sehingga dapat menghasilkan kinerja yang baik pada dataset yang tidak seimbang.

2. PENELITIAN TERKAIT

Penelitian tentang prediksi kelulusan telah banyak dilakukan dan telah dipublikasikan. Untuk melakukan penelitian ini perlu ada kajian terhadap penelitian yang terkait sebelumnya agar dapat mengetahui metode apa saja yang digunakan, data seperti apa yang diproses, dan model seperti apa yang dihasilkan.

Undavia, Dolia, dan Shah (2013) dalam penelitiannya pada lembaga pendidikan tinggi yang mengalami rendahnya persentase dari hasil penempatan dan minat mahasiswa menggunakan *decision tree* pada sistem pendukung keputusan untuk memprediksi pasca kelulusan bagi mahasiswa berdasarkan prestasi akademik pada data historis. Hasil akurasi sebesar 67,1875% dan Kappa yang diperoleh 0,1896. Jumlah Confusion Matrix 86 (73+13) record klasifikasi benar untuk MBA dan MCA, sedangkan 42 (32+10) record diklasifikasikan salah untuk kedua Program PG.

Ramesh, Parkavi, dan Ramar (2013) dalam penelitiannya mengidentifikasi faktor-faktor yang mempengaruhi prestasi siswa dalam ujian akhir dan mencari tahu algoritma mana yang cocok untuk memprediksi *grade* dari siswa sehingga dapat memberikan secara tepat waktu dan memberikan peringatan bagi siswa mereka yang beresiko. Hasil yang diperoleh dari pengujian hipotesis menunjukan bahwa jenis sekolah tidak mempengaruhi prestasi siswa dan kedudukan orang tua memiliki peran utama dalam memprediksi *grade*. Algoritma klasifikasi terbaik yang digunakan dalam studi tersebut yaitu Multilayer Perception. Pencarian atribut terbaik dengan menggunakan metode "Ranker" dan mendapatkan 10 atribut teratas yang dipilih dari 27 atribut, sebanyak 500 record diambil untuk dijadikan analisis. Hasil akurasi ternyata Multilayer Perception memiliki akurasi yang terbaik sebesar 72,38%.

Marquez-Vera, Cano, Romero, dan Ventura (2013) dalam penelitiannya memprediksi kegagalan siswa di Sekolah dengan mengukur faktor yang dapat mempengaruhi rendahnya prestasi siswa dan sifat dari dataset yang tidak seimbang (*imbalance*). Algoritma yang diusulkan yaitu *genetic programming* dan pendekatan data mining yang berbeda. Data yang digunakan sekitar 670 siswa tingkat atas dari sekolah Zacatecas – Meksiko.

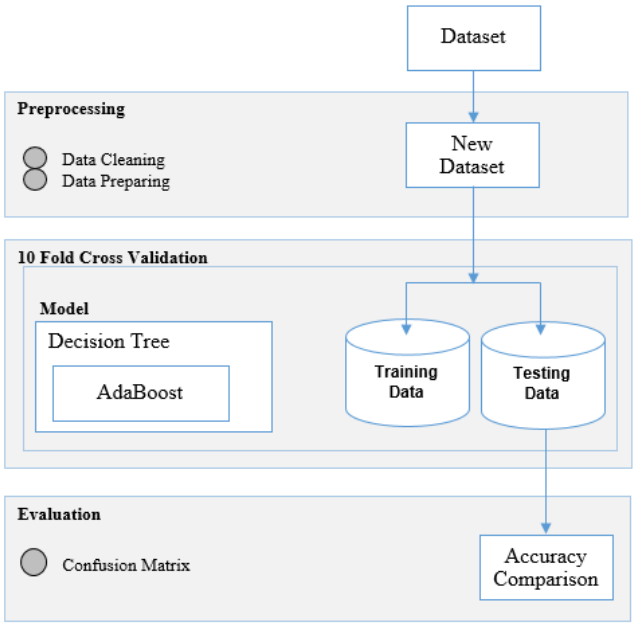
Sebagian besar kasus data yang telah dilakukan untuk klasifikasi siswa gagal atau dropout adalah tidak seimbang, yang berarti bahwa hanya sebagian kecil siswa gagal dan sebagian besar lulus. Metode yang diusulkan mampu memecahkan masalah dalam dunia pendidikan, seperti menggunakan klasifikasi *cost-sensitive* dan *resampling* dari dataset asli.

Thammasiri, Delen, Meesad, dan Kasap (2014) dalam penelitiannya membandingkan perbedaan teknik data yang tidak seimbang untuk meningkatkan akurasi prediksi dalam *class* minoritas tetapi tetap mempertahankan kepuasan kinerja klasifikasi secara keseluruhan. Secara khusus Thammasiri menggunakan tiga metode pengujian *balancing* yaitu teknik *over-sampling*, *under-sampling* dan *synthetic minority over-sampling* (SMOTE) beserta empat metode klasifikasi yang populer (LR, DT, NN, dan SVM). Hasil penelitian menunjukan bahwa SVM dengan SMOTE mencapai kinerja terbaik dengan tingkat akurasi mencapai 90.24% pada sampel *10-fold holdout*.

Pada penelitian ini berbeda dengan penelitian yang telah ada, untuk klasifikasi menggunakan decision tree, sedangkan untuk mengatasi masalah ketidakseimbangan kelas menggunakan metode adaboost untuk menyelesaikan masalah data atau kelas yang sifatnya tidak seimbang (*imbalance*).

3. METODE YANG DIUSULKAN

Metode yang diusulkan dalam penelitian ini yaitu untuk meningkatkan kinerja algoritma decision tree dengan metode adaboost yang dapat menangani ketidakseimbangan kelas pada klasifikasi dataset kelulusan. Sedangkan untuk validasi menggunakan *10-fold cross validation*. Hasil pengukuran dengan analisa menggunakan t-Test. Model kerangka pemikiran metode yang diusulkan ditunjukan pada Gambar 1.



Gambar 1. Kerangka Pemikiran Model yang diusulkan

Pada Gambar 1 dalam pengolahan awal, data yang sudah didapat dibersihkan dan pilah, sehingga menjadi sebuah dataset baru untuk *training* dan testing dari atribut yang sudah ditentukan. Setelah itu dimasukan kedalam *classifier* dengan

metode algoritma decision tree, kemudian dataset yang tidak seimbang diselesaikan oleh *boosting* (adaBoost). Pada dasarnya, metode *boosting* dapat meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi dari suatu model, tetapi hasil klasifikasi atau prediksi yang dipilih adalah model yang memiliki nilai bobot paling besar. Jadi, setiap model yang dibangkitkan memiliki atribut berupa nilai bobot. Dataset yang telah seimbang akan divalidasi dengan menggunakan *10-fold cross validation*. Hasil dari validasi akan menghasilkan data yang diukur yaitu AUC dan *Accuracy*.

Zhou dan Yu (2009) menjelaskan teknik pembobotan pada algoritma adaboost sebagai berikut:

Input:
Dataset (D_t) = (x_1, y_1), …, (x_m,y_m);
Weak Lern (*L*)
T menyatakan jumlah iterasi
Proses:
Inisialisasi nilai bobot
$D_1(i) = \frac{1}{m}$ untuk i = 1, …, *m*
for *t* = 1, … , *T*;
Pengujian terhadap distribusi D_t
$h_t = L(D_t)$
hitung *error* dari $\epsilon_t = Pr_{x\sim D_t, y}\, I[h_t(x) \neq y]$
if $\epsilon_t > 0.5$ then break
menentukan bobot dari h_t
$\alpha t = \frac{1}{2} ln\left(\frac{1-\epsilon t}{\epsilon t}\right)$;
$D_{t+1}(i) = \frac{D_t(i)}{Z_i}$ x $\begin{cases}\exp(-\alpha_t)\ if\ h_t(x_i) = y_i \\ \exp(\alpha_t)\ if\ h_t(x_i) \neq y_i\end{cases}$

Update distribusi, dimana *Zt* sebuah faktor normalisasi yang mengaktifkan D_{t+1} menjadi distribusi

$$\frac{Dt(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

end for
Output:
H(x) = $sign(\sum_{t=1}^{T} \propto_t\ h_t(x))$

Pada tahap evaluasi dilakukan dengan memperoleh hasil AUC, *Accuracy*, kemudian validasi dilakukan dengan pengujian *Root Mean Square Error* (RMSE) dengan uji t-Test untuk mengetahui apakah ada perbedaan antara metode decision tree dengan decision tree dan adaboost.

Evaluasi terhadap model mengukur akurasi dengan *confusion matrix* yang menitikberatkan pada *class* secara umum, sedangkan untuk AUC menggunakan ROC Curve dan proses dengan menggunakan *10 fold cross validation*.

Pengukuran akurasi dengan *confusion matrix* dapat dilihat pada Tabel 1.

Tabel 1. Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Formulasi perhitungan yang digunakan (Gorunescu, 2011) adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$$

$$Sensitivity = TP\ rate = \frac{TP}{TP+FN}$$

$$Specificity = TN\ rate = \frac{TN}{TN+FP}$$

$$FP\ rate = \frac{FP}{FP+TN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F\text{-}Measure = \frac{2RP}{R+P}$$

$$G\text{-}Mean = \sqrt{Sensitivity * specificity}$$

Evaluasi dengan *F-Measure*, rata-rata harmonik dari dua angka cenderung lebih dekat dengan lebih kecil dari dua, oleh karena itu nilai *F-Measure* yang tinggi dapat memastikan bahwa kedua *recall* (*sensitivity*) dan presisi yang cukup tinggi. Jika hanya kinerja kelas positif dianggap sebagai dua langkah penting yaitu *TP rate* dan *Positive Predictive Value* (*PP value*). *PP value* didefinisikan sebagai presisi yang menunjukan presentasi objek yang relevan yang didefinisikan untuk retrieval. Dalam pencarian informasi *TP rate* didefinisikan sebagai *recall* yang menunjukan presentasi dari objek yang diambil itu adalah relevan. Rata-rata harmonik adalah gabungan dari ukuran presisi dan *recall*.

Evaluasi dengan *Receiver Operating Character Curve* (*ROC Curve*), secara teknis menggambarkan graifk dua dimensi, dimana tingkat *True Positive* (*TP*) terletak pada garis sumbu *Y*, sedangkan untuk *False Positive* (*FP*) terletak pada garis sumbu *X*. dengan demikian *ROC* menggambarkan *trade-off* antara *TP* dan *FP*. Pencatatan dalam *ROC* dinyatakan dalam sebuah klausa yaitu semakin rendah titik kekiri (0.0), maka dianyatakan sebagai klasifikasi prediksi mendekati/menjadi negatif, sedangkan semakin keatas titik kekanan (1.1), maka dinyatakan sebagai klasifikasi prediksi mendekati/menjadi positif. Titik dengan nilai 1 dinyatakan sebagai tingkat *True Positif* (*TP*), sedangkan titik dengan nilai 0 dinyatakan sebagai tingkat *False Positive* (*FP*). Pada titik (0.1) merupakan klasifikasi prediksi adalah sempurna karena semua kasus baik positif maupun negatif dinyakatakan dengan benar (*True*). Sedangkan untuk (1.0) klasifikasi prediksi semuanya dinyatakan sebagai tidak benar (*False*).

Dalam pengklasifikasian keakuratan dari tes diagnostik menggunakan *Area Under Curve* (*AUC*) (Gorunescu, 2011, hal. 325-326) sebuah sistem nilai yang disajikan pada Tabel 2.

Tabel 2. Nilai AUC dan Keterangan

AUC	Keterangan
0.90 - 1.00	*excellent classification*
0.80 - 0.90	*good classification*
0.70 - 0.80	*fair classification*
0.60 - 0.70	*poor classification*
< 0.60	*failure*

Evaluasi dengan *Root mean square error* (RMSE) adalah sebuah metode konvensional yang digunakan untuk menghitung rata-rata ukuran dari sebuah deviasi dan disebut juga sebagai perbedaan antara nilai aktual dengan nilai prediksi (Barreto & Howland, 2006). RMSE adalah estimasi dari sebuah sampel peta dan referensi poin (Congalton & Green, 2009). Jadi, *root mean square error* (RMSE) atau disebut juga sebagai root *mean square deviation* (RMSD) suatu ukuran yang digunakan dari perbedaan antara nilai-nilai yang diprediksi oleh sebuah model dengan nilai-nilai aktual.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i' - y_i)^2}{n}}$$

Keterangan formulasi RMSE:
y' : nilai aktual
y : nilai prediksi
n : jumlah sampel data
i : terasi

4. HASIL EKSPERIMEN

Eksperimen dilakukan dengan menggunakan sebuah platform komputer berbasis Intel Core i3-3220 @3.30GHz (4 CPUs), RAM 4GB, dan sistem operasi Microsoft Windows 8.1 Profesional 64-bit. Sedangkan lingkungan pengembangan aplikasi dengan bahasa pemrograman Java Netbeans IDE 8.0 dan Rapidminer 6.0, untuk analisis hasil menggunakan aplikasi Excel Data Anlysis.

Data kelulusan mahasiswa yang bisa digunakan sebagai dataset yaitu dengan jumlah mahasiswa 429 record dengan status lulus. Dari jumlah tersebut memiliki 15 atribut dengan status klasifikasi tepat waktu sebesar 119 record (27,74%) dan tidak tepat waktu atau terlambat sebesar 310 record (72,26%). Karakterisitik dari dataset universitas pamulang dapat dilihat pada Tabel 3.

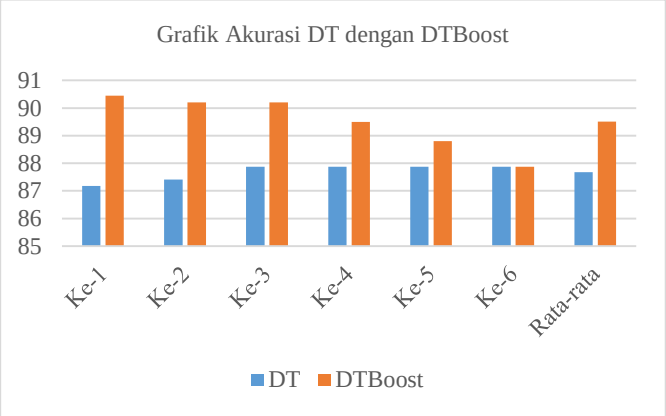
Tabel 3. Karakterisitik Dataset Universitas Pamulang

No	Atribut	Tipe Data	Keterangan
1	NIM	Nominal	Nomor Induk Mahasiswa (ID)
2	Prodi	Nominal	Program Studi
3	JK	Nominal	Jenis Kelamin
4	Shift	Nominal	Shift waktu perkuliahan
5	Usia	Numeric	Usia Mahasiswa
6	IPS-1	Numeric	Indek Prestasi Semester – 1
7	IPS-2	Numeric	Indek Prestasi Semester – 2
8	IPS-3	Numeric	Indek Prestasi Semester – 3
9	IPS-4	Numeric	Indek Prestasi Semester – 4
10	IPS-5	Numeric	Indek Prestasi Semester – 5
11	IPS-6	Numeric	Indek Prestasi Semester – 6
12	IPS-7	Numeric	Indek Prestasi Semester – 7
13	IPS-8	Numeric	Indek Prestasi Semester – 8
14	IPK	Numeric	Indek Prestasi Kumulatif
15	Status	Nominal	Tepat Waktu atau Tidak Tepat Waktu (Label)

Dalam eksperimen ini dilakukan dengan nilai parameter decision tree: *criterion*, *minimal size for split*, *minimal leaf size*, *minmal gain*, *maximal depth*, *confidence* dan *split vaildation* dengan parameter *split*, *split ration*, *stratified sampling* untuk mendapatkan nilai AUC, *Accuracy*, dan RMSE. Hasil eksperimen dapat dilihat pada Tabel 4 sampai dengan Tabel 8.

Tabel 4. Perbandingan Akurasi

Eksperimen	DT	DTBoost
Ke-1	87,18	90,45
Ke-2	87,41	90,21
Ke-3	87,87	90,21
Ke-4	87,87	89,50
Ke-5	87,87	88,80
Ke-6	87,87	87,87
Rata-rata	**87,68**	**89,51**



Gambar 2. Grafik Perbandingan Akurasi

Tabel 5. Confusion Matrix Model DT

	True Terlambat	True Tepat	Class Precision
Pred. Terlambat	274	19	93.52%
Pred. Tepat	36	100	73,75%
Class Recall	88,39%	84,03%	

$$Accuracy = \frac{(TN+TP)}{(TN+FN+TP+FP)}$$

$$Accuracy = \frac{(274+100)}{(274+19+100+36)}$$

$$Accuracy = 0{,}8718 = 87{,}18\%$$

Dari jumlah data sebanyak 429 klasifikasi kelas dengan status terlambat sebesar 310 record dan status tepat sebesar 119 record. Data diprediksi yang sesuai dengan status terlambat sejumlah 274, data yang diprediksi terlambat tetapi kenyataannya tepat sejumlah 19, data yang diprediksi tepat tetapi kenyataannya terlambat sejumlah 36, dan sedangkan data yang diprediksi tepat dan sesuai sejumlah 100.

Tabel 6. Confusion Matrix Eksperimen DTBoost

	True Terlambat	True Tepat	Class Precision
Pred. Terlambat	291	22	92,97%
Pred. Tepat	19	97	83,62%
Class Recall	93,87%	81,51%	

$$Accuracy = \frac{(TN+TP)}{(TN+FN+TP+FP)}$$

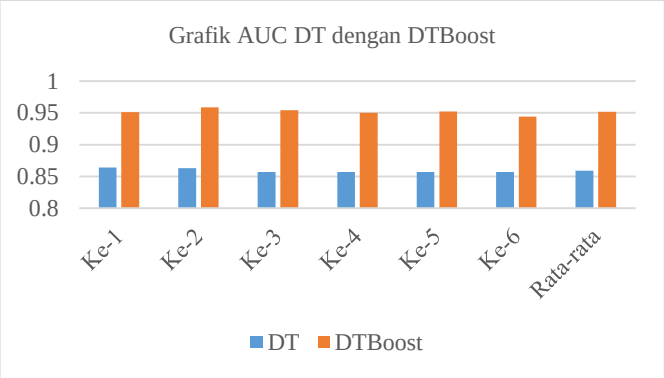
$$Accuracy = \frac{(291+97)}{(291+22+97+19)}$$

$$Accuracy = 0{,}9045 = 90{,}45\%$$

Dari jumlah data sebanyak 429 klasifikasi kelas dengan status terlambat sebesar 310 record dan status tepat sebesar 119 record. Data diprediksi yang sesuai dengan status terlambat sejumlah 291, data yang diprediksi terlambat tetapi kenyataannya tepat sejumlah 22, data yang diprediksi tepat tetapi kenyataannya terlambat sejumlah 19, dan sedangkan data yang diprediksi tepat dan sesuai sejumlah 97.

Tabel 7. Perbandingan Area Under Curve (AUC)

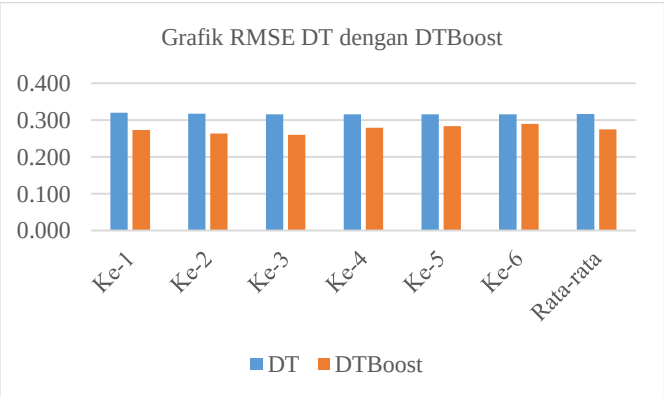
Eksperimen	DT	DTBoost
Ke-1	0,864	0,951
Ke-2	0,863	0,959
Ke-3	0,857	0,954
Ke-4	0,857	0,950
Ke-5	0,857	0,952
Ke-6	0,857	0,944
Rata-Rata	**0,859**	**0,952**



Gambar 3. Grafik Perbandingan AUC

Tabel 8. Perbandingan RMSE

Eksperimen	DT	DTBoost
Ke-1	0,320	0,273
Ke-2	0,318	0,264
Ke-3	0,316	0,260
Ke-4	0,316	0,279
Ke-5	0,316	0,284
Ke-6	0,316	0,290
Rata-rata	**0,317**	**0,275**



Gambar 4. Grafik RMSE DT dengan DTBoost

Pada penelitian ini dilakukan pengujian hipotesis dengan uji *paired sample t-Test* decision tree (DT) dengan decision tree dan adaboost (DTBoost). T-Test adalah hubungan antara variabel respon dengan varibel prediktor (Larose, 2007). Hipotesis nol (H_0) menyatakan bahwa tidak ada hubungan yang linier antara varibel-variabel, sedangkan hipotesis alternatif (Ha) menyatakan bahwa adanya hubungan antara variabel-variabel. H_0 merupakan tidak ada perbedaan antara DT dan DTBoost, Ha merupakan ada perbedaan antara DT dan DTBoost. Pada *paired sample t-Test* dengan *root mean square error* (RMSE) yang terdiri dari variabel DT dan variabel DTBoost dapat dilihat pada Tabel 9.

Tabel 9. *Paired Samples Test* RMSE DT dengan DTBoost

	DT	*DTBoost*
Mean	0,317	0,275
Variance	2,8E-06	0,0001344
Observations	6	6
Pearson Correlation	-0,309294787	
Hypothesized Mean Difference	0	
df	5	
t Stat	8,42249	
P	**0,000193471**	

Pada Tabel 9 menunjukan hasil dari *paired sample test* RMSE DT dengan DTBoost, bahwa untuk nilai uji t memiliki aturan apabila *P-value* < 0,05 terdapat perbedaan pada taraf signifikan yaitu 5%, dan apabila *P-value* > 0,05, maka tidak ada perbedaan antara sebelum dan sesudah. Hasil yang didapat dari nilai uji t untuk *P-value* sebesar 0,000193471 < 0,05 yang berarti bahwa H_0 ditolak atau Ha diterima, adanya perbedaan yang signifikan antara DT dan DTBoost.

Metode decision tree dengan adaboost menghasilkan tingkat akurasi yang lebih baik dibandingkan dengan menggunakan decision tree versi standar. Hal tersebut seperti dikatakan oleh Quinlan, bahwa adaboost dapat memberikan keuntungan, lebih efektif dan akurat dalam pengklasifikasian (Quinlan, Bagging, Boosting, and C4.5, 1996).

5. KESIMPULAN

Penelitian dengan menerapkan metode adaboost untuk penyelesaian ketidakseimbangan kelas (*class imbalance*) pada penentuan kelulusan mahasiswa dengan metode decision tree, eksperimen telah dilakukan untuk mendapatkan sebuah model arsitektur yang optimal dan mendapatkan hasil estimasi yang akurat. Hasil pengujian diatas dapat disimpulkan bahwa metode adaboost sebagai metode boosting terbukti efektif dalam penyelesaian ketidakseimbangan kelas pada penentuan kelulusan mahasiswa dengan metode decision tree.

REFERENSI

Barreto, H., & Howland, F. M. (2006). *Introductory Econometrics: Using Monte Carlo Simulation with Microsoft ExceIntroductory Econometrics: Using Monte Carlo Simulation with Microsoft Excel.* New York: Cambridge University Press.

Capparuccia, R., Leone, R. D., & Marchitto, E. (2007). Integrating support vector machines and neural networks. *Neural Networks*, 590-597.

Chawla, N. V., Cieslak, D. A., Hall, L. O., & Joshi, A. (2008). Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 225-252.

Cieslak, D. A., Hoens, T. R., Chawla, N. V., & Kegelmeyer, W. P. (2012). Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 136-158.

Congalton, R. G., & Green, K. (2009). *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices, Second Edition (Mapping Science).* Boca Raton: CRC Press.

Drummond, C., & Holte, R. C. (2003). C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling. *Institute for Information Technology*,

National Research Council (pp. 1-8). Canada, Ottawa, Ontario: Department of Computing Science, University of Alberta.

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques.* Verlag Berlin Heidelberg: Springer.

Hulse, J. V., & Khoshgoftaar, T. (2009). Knowledge discovery from imbalanced and noisy data. *Elsevier*, 1513-1542.

Karamouzis, S. T., & Vrettos, A. (2008). An Artificial Neural Network for Predicting Student Graduation Outcomes. *WCECS (World Congress on Engineering and Computer Science)*, 991-994.

Kotsiantis, S. B., & Pintelas, P. E. (2009). Selective costing ensemble for handling imbalanced data sets. *International Journal of Hybrid Intelligent Systems*, 123-133.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 25-36.

Larose, D. T. (2007). *Data Mining Methods and Models.* Hoboken, New Jersey: A John Wley & Sons, Inc Publication.

Marquez-Vera, C., Cano, A., Romero, C., & Ventura, S. (2013). Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied Intelligence*, 315-330.

Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. F. (2013). Predicting Student Performance: An Application Of Data Mining Methods With The Educational Web-Based System Lon-Capa. *IEEE (Institute of Electrical and Electronics Engineers)*.

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

Quinlan, J. R. (1996). Bagging, Boosting, and C4.5. *AAAI'96 Proceedings of the thirteenth national conference on Artificial intelligence - Volume 1* (pp. 725-730). Australia: ACM Digital Library.

Ramesh, V., Parkavi, P., & Ramar, K. (2013). Predicting Student Performance: A Statistical and Data Mining Approach. *International Journal of Computer Applications*, 35-39.

Sun, Y., Kamel, M. S., Wong, A. K., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition Society*, 3358–3378.

Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications: An International Journal*, 321-330 .

Undavia, J. N., Dolia, P. M., & Shah, N. P. (2013). Prediction of Graduate Students for Master Degree based on Their Past Performance using Decision Tree in Weka Environment. *International Journal of Computer Applications*.

Weiss, G. M., McCarthy, K., & Zabar, B. (2007). Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs? *DMIN*, 35-41.

Zhang, H., & Wang, Z. (2011). A Normal Distribution-Based Over-Sampling Approach to Imbalanced Data Classification. *Advanced Data Mining and Applications - 7th International Conference* (pp. 83-96). Beijing, China: Springer.

Zhou, Z.-H., & Yu, Y. (2009). *The Top Ten Algorithms in Data Mining.* (X. Wu, & V. Kumar, Eds.) Chapman & Hall/CRC.

BIOGRAFI PENULIS

Achmad Bisri. Memperoleh gelar Sarjana Komputer (S.Kom) dibidang Teknik Informatika dari STMIK Banten Jaya, Serang-Banten, gelar Magister Komputer (M.Kom) dibidang Software Engineering (Rekayasa Perangakat Lunak) dari STMIK Eresha, Jakarta. Dia saat ini sebagai staf pengajar di Universitas Pamulang (Unpam), Tangerang Selatan. Minat Penelitiannya saat ini meliputi software engineering (rekayasa perangkat lunak) dan mechine learning.

Romi Satria Wahono. Memperoleh gelar B.Eng. dan M.Eng. di bidang ilmu komputer dari Saitama University, Jepang dan gelar Ph.D. di bidang Software Engineering dari Universiti Teknikal Malaysia Melaka. Saat ini sebagai pengajar dan peneliti pada program Pascasarjana Ilmu Komputer di Universitas Dian Nuswantoro, Indonesia. Juga merupakan pendiri dan CEO PT Brainmatics Cipta Informatika, sebuah perusahaan pengembangan perangkat lunak di Indonesia. Minat penelitiannya saat ini meliputi rekayasa perangkat lunak dan machine learning. Anggota profesional dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Algoritma Cluster Dinamik untuk Optimasi Cluster pada Algoritma K-Means dalam Pemetaan Nasabah Potensial

Widiarina
Magister Ilmu Komputer, STMIK Nusa Mandiri
Email:widiarina11@gmail.com

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

Abstract: Pelanggan merupakan salah satu sumber keuntungan perusahaan. Pemahaman yang baik tentang pelanggan sangat penting dilakukan untuk mengetahui nilai potensial pelanggan. Saat ini pelaksanaan CRM (*Customer Relationship Management*) dapat membantu dalam pemahaman nilai pelanggan. Segmentasi pelanggan adalah salah satu metode yang digunakan untuk pemetaan pelanggan. Nilai potensial pelanggan dapat diukur menggunakan metode RFM (*Recency, Frequency, Monetary*). Algoritma *K-means* salah satu metode yang banyak digunakan untuk segmentasi pelanggan. *K-means* banyak dipakai karena algoritma nya mudah dan sederhana, tetapi algoritma ini memiliki kekurangan yaitu sensitifitas pada partisi awal jumlah cluster(k). Untuk menyelesaikan masalah sensitifitas partisi awal jumlah cluster pada algoritma *K-means*, maka diusulkan algoritma cluster dinamik untuk menetapkan jumlah cluster(k). Hasil percobaan menunjukan bahwa algoritma cluster dinamik pada *K-means*, dapat menghasilkan kualitas cluster yang lebih optimal.

Keywords: segmentasi pelanggan, RFM, K-Means, quality Cluster

1 PENDAHULUAN

Pelanggan menduduki posisi penting dalam pengembangan strategi bisnis, pelanggan juga merupakan salah satu sumber keuntungan dalam perusahaan. Untuk itu diperlukan suatu pemahamaan yang baik tentang pelanggan. Pemahaman yang baik terhadap pelanggan dapat digunakan perusahaan untuk berinvestasi pelanggan yang potensial. Masalah yang sering dihadapi adalah kesulitan dalam menganalisa nilai pelanggan (Xing, 2010). Banyak pemasar mengalami kesulitan untuk mengidentifikasi pelanggan atau nasabah yang tepat (Chai & Chan, 2008), hal tersebut dapat mengakibatkan perusahan kehilangan nasabah potensial dan tentunya akan sangat merugikan perusahaan.

Segmentasi pelanggan adalah metode yang populer yang digunakan untuk memilih pelanggan atau nasabah yang tepat untuk memulai promosi (Chai & Chan, 2008). Dengan segmentasi nasabah berdasarkan prilakunya, kita dapat menargetkan tindakan mereka dengan lebih baik. Seperti peluncuran produk yang disesuaikan, target pemasaran dan untuk memenuhi harapan pelanggan (Balaji & Srivatsa, 2012). Namun untuk menganalisa data pelanggan atau nasabah dalam jumlah besar memerlukan tenaga dan waktu yang banyak.

Beberapa algoritma segmentasi telah digunakan dalam segmentasi pelanggan temasuk metode *Self Organizing Map* (Chan, 2005), dan *K-means* (Prasad & Malik, 2011). Algortima *K -means* adalah algoritma *clustering* yang paling populer digunakan karena memiliki kelebihan yaitu algoritmanya sederhana, mudah di implementasikan. Dan merupakan salah satu metode yang cukup efisien dalam hal kompleksitas nya *O(nkt)* (Aggarwal & Aggarwal, 2012). Salah satu kekurangan algoritma *K-means* yaitu mempunyai masalah sensitifitas terhadap penentuan partisi awal jumlah *cluster*(k) dan solusi akhir menyatu pada *local minima*. Penentuan partisi jumlah *cluster*(k) sangat penting bagi algoritma *K-means,* tetapi tidak ada ketentuan yang berlaku untuk menentukan berapa jumlah *cluster*(k) yang akan dibentuk (Zhang & Fang, 2013). Pada prakteknya penentuan partisi awal jumlah *cluster* sangat sulit, karna penentuan partisi awal jumlah k yang berbeda dapat menghasilkan kelompok *cluster* yang berbeda pula.

Pada penelitian ini kami mengusulkan algoritma *cluster* dinamik pada algoritma *K-means* dalam menetapkan jumlah cluster(k) agar dapat menghasilkan kualitas *cluster* yang optimal sehingga memberikan hasil pemetaan nasabah potensial lebih baik dan tepat.

2 PENELITIAN TERKAIT

K-means merupakan suatu algoritma pengklasteran yang cukup sederhana yang mempartisi dataset kedalam beberapa klaster *k*. Algoritmanya cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan (Wu & Kumar, 2009). Prinsip utama dari teknik ini adalah menyusun *k* buah partisi/pusat massa (*centroid*)/rata-rata (*mean*) dari sekumpulan data. Algoritma *K-means* dimulai dengan pembentukan partisi klaster di awal kemudian secara iteratif partisi klaster ini diperbaiki hingga tidak terjadi perubahan yang signifikan pada partisi klaster (Witten, Eibe, & Hall, 2011).

Beberapa penelitian yang telah dilakukan mengenai masalah sensitifitas inisialisasi jumlah *cluster*(k), dan algoritma yang digunakan. Penelitian yang dilakukan oleh (Deelers & Auwatanamongkol, 2007) mengusulkan sebuah algoritma partisi data untuk menghitung awal pusat kluster. Partisi data mencoba membagi ruang data kedalam sel kecil atau kelompok, mana yang jarak intercluster sebesar mungkin dan jarak intracluster sekecil mungkin. Sel dipartisi satu persatu sampai jumlah sel sama dengan jumlah kluster (k) yang telah ditetapkan, dan pusat-pusat sel k menjadi awal pusat kluster untuk *K-means.* Hasil percobaan menunjukan bahwa algoritma partisi data bekerja lebih baik dibandingkan dengan inisialisasi pusat kluster secara acak dari sebagian kasus eksperimental dan dapat mengurangi waktu *running* algoritma *K-means* untuk dataset yang besar. (Yi et al., 2010), mengusulkan sebuah algoritma partisi data untuk memperbaiki awal pusat *cluster* yaitu Algoritma awal pusat *cluster* berbasis kepadatan (*density*). Algoritma ini menggunakan fungsi gaussian untuk memenuhi konsistensi global fitur *clustering*.

Algoritma yang diusulkan memilih titik kepadatan terbesar sebagai titik pusat awal pertama dari dataset, kemudian menentukan pusat awal kedua menggunakan metode yang sama dari dataset sehingga menghapus titik pertama dan tetangganya. Proses ini berlanjut sampai set *M* awal berisi *k* poin. Hasil percobaan menunjukan bahwa algoritma yang diusulkan sangat meningkatkan kualitas dan stabilitas algoritma *K-means.* (Zhang & Fang, 2013) melakukan penelitian dalam perbaikan algoritma K-means untuk mengoptimalkan inisialisasi pusat *cluster*. Dengan menemukan satu set data yang mencermikan karakteristik distribusi data sebagai pusat awal *cluster* untuk mendukung pembagian data ke batas yang terbaik. Hasil percobaan didapatkan hasil akurasi algoritma perbaikan *K-means* meningkat secara signifikan dibandingkan dengan algoritma *K-means* tradisional, dan algoritma yang diusulkan menunjukan bahwa hasil setiap *cluster* lebih kompak.

Pada penelitian ini kami mengkombinasikan algoritma *cluster* dinamik dengan algoritma *K-Means* untuk menghasilkan kualitas *cluster* yang optimal dalam segmentasi nasabah potensial.

3 METODE YANG DIUSULKAN

Algoritma *cluster* dinamik pada algoritma *K-means* dinamik dapat dilihat pada Gambar 1. Algoritma yang diusulkan mencari jumlah *cluster* yang dijalankan berdasarkan kualitas *cluster* keluaran. Diawal cara kerja sama dengan algoritma *k-means,* diakhir akan dilakukan perhitungan intra dan inter cluster, jika jarak intra lebih kecil dan jika jarak intra lebih besar, maka algoritma menghitung *cluster* baru dengan menambahkan *counter* k dengan satu atau *k=k+1* disetiap iterasi sampai memenuhi batas validitas kualitas *cluster* yang berkualitas (M & Hareesha, 2012).

Berikut tahapan Algoritma *K-means*:

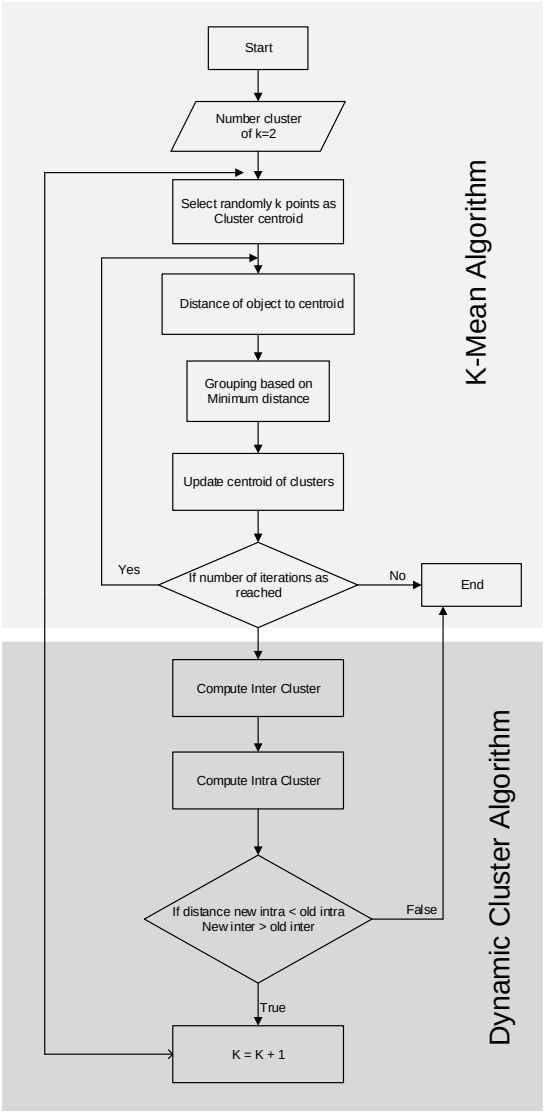
1. Membuat partisi sejumlah k dari segmentasi yang akan dibentuk.
2. Pilih secara acak k point untuk dijadikan pusat *cluster*
3. Menghitung jarak data yang lain dengan pusat *cluster*
4. Mengisi setiap obyek dalam dataset kedalam segmen terdekat.
5. Kalkulasi ulang setiap segmentasi yang terbentuk
6. Ulangi langkah hingga data di dalam segmentasi tidak berubah

Istilah inter adalah minimum jarak antar pusat *cluster*, inter digunakan untuk mengukur pemisahan antar *cluster*, yang didefinisikan sebagai:

$$inter = min\{\|m_k - m_{kk}\|\} \quad \forall\, k = 1,2,\ldots,K-1 \; dan \; k = k+1, \ldots, K \quad \text{..... (1)}$$

Istilah intra digunakan untuk mengukur kekompakan dari suatu kelompok. Standar deviasi digunakan untuk memeriksa kedekatan titik data setiap *cluster*, dan dihitung sebagai:

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - X_m)^2} \quad \text{..... (2)}$$



Gambar 1. Algoritma *Cluster* Dinamik pada *K-Means*

4 HASIL PENELITIAN

Dalam melakukan proses perhitungan, pada penelitian ini digunakan komputer dengan spesifikasi komputer Pentium(R) Dual-Core 2.00GHz, RAM 2 GB, Sistem Operasi Windows 7, 32 bit , dan program aplikasi menggunakan *software Matlab.* Dalam penelitian ini, data diperoleh dari data nasabah pengguna jasa penerimaan transaksi kartu pada suatu perusahaan perbankan. Untuk membantu pemilihan atribut, data yang digunakan adalah data nasabah dan data transaksi nasabah selama 6 bulan kebelakang, data awal terdiri dari bulan Juli sampai Desember 2012 dengan jumlah data sebanyak 1002 *record*. Data yang sudah terkumpul akan diolah melalui beberapa tahap pengolahan awal data (*preparation data*). Tahapan pengolahan data yang dilakukan yaitu validasi data, transformasi data, dan seleksi atribut. Contoh data yang sudah di transformasi data dapat dilihat pada Tabel 1. Atribut yang digunakan yaitu :*Last transaction(R), cust age(F),* dan *rate amount(M)* ketiga atribut tersebut dipilih berdasarkan faktor *Recency Frequency Monetary.*

Table 1. Data Transformasi

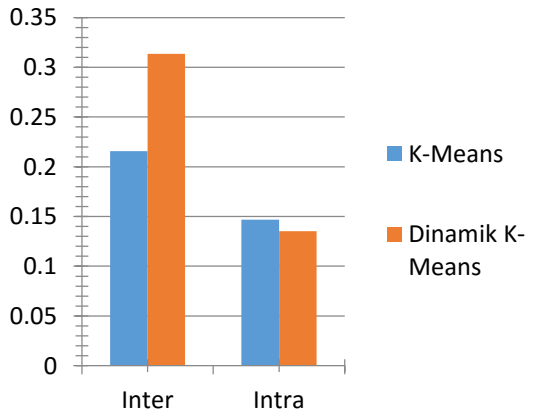
FIELD	Rec 1	Rec 2	Rec 3	Rec 4	Rec 5
ITEM DEBIT1	4	1	19	99	4
AMOUNT DEBIT1	3	1	16	99	4
ITEM PREPADE1	6	1	7	99	1
AMOUNT PREPADE1	3	1	6	99	1
ITEM CREDIT1	1	1	53	99	23
AMOUNT CREDIT1	1	1	40	99	58
ITEM TOTAL1	4	1	21	99	6
AMOUNT TOTAL1	2	1	18	99	10
LAST TRANSACTION	1	99	80	1	1
CUST AGE	99	87	47	8	1
RATE AMOUNT	2	1	3	99	9

Hasil segmentasi yang terbentuk akan dievaluasi menggunakan *Davies-Bouldin(DB) Index* dan *purity*. *Davies-Bouldin Index* adalah fungsi rasio dari jumlah antara *cluster scatter* sampai dengan *cluster separation* (Maulik & Bandyopadhyay, 2002). *Davies-Bouldin Index* merupakan metode validasi *cluster* dari hasil *clustering*. Pendekatan pengukuran DBI yaitu memaksimalkan jarak *inter cluster* serta meminimalkan jarak *intra cluster*. Nilai purity adalah kesesuaian antara *cluster* dengan *cluster* ideal, semakin besar nilai purity (semakin mendekati 1), semakin baik kualitas *cluster* (Yi, Qiao, Yang, & Xu, 2010). Semakin kecil nilai DB *Index* menunjukan skema *cluster* yang paling optimal. Semakin besar nilai *purity* (mendekati 1) semakin baik kualitas *cluster*.

Dari hasil percobaan yang dilakukan, algoritma *K-means* dengan *cluster* dinamik dapat menghasilkan kualitas *cluster* yang lebih baik dibandingkan dengan *K-means* tradisional. Tabel 2 menunjukan bahwa algoritma *K-means* dengan algoritma *cluster* dinamik menghasilkan jarak *inter* yang lebih besar. Dan jarak *intra* yang lebih kecil dibandingkan dengan jarak *inter* dan *intra* yang dihasilkan oleh algoritma *K-means* tradisional. Dengan peningkatan akurasi yang cukup signifikan, grafik dapat dilihat pada Gambar 2.

Tabel 2. Hasil Percobaan

Algoritma	Cluster(k)	Inter	Intra
K-Means	3	0,2157	0,1469
Dinamik K-Means	4	0,3135	0,1352



Gambar 2. Grafik Hasil Percobaan

Akhirnya, hasil evaluasi *cluster* menunjukan bahwa algoritma *K-means* dengan algoritma *cluster* dinamik memperoleh nilai DBI lebih kecil dibandingkan dengan algoritma *K-means* tradisional dengan nilai DBI sebesar 0.4313, dan nilai purity lebih besar yaitu 0.7647. Hasil evaluasi cluster dapat dilihat pada Tabel 3. Nilai DBI yang lebih kecil dan nilai purity yang lebih besar, dengan demikian menunjukan bahwa skema *cluster* lebih optimal.

Tabel 3. Nilai pengujian *Davies-Bouldin Index* dan *purity*

Algoritma	Cluster(k)	DBI	Purity
K-Means	3	0,6810	0,5294
Dinamik K-Means	4	0,4313	0,7647

Segmen yang terbentuk berdasarkan faktor *recency(R)*, *frequency(F)* dan *monetary(M)*. Semakin besar nilai R menunjukan bahwa nasabah sering melakukan transaksi, semakin besar nilai F menunjukan bahwa nasabah tersebut setia terhadap terhadap produk yang digunakan, dan semakin besar nilai M, menunjukan bahwa nilai transaksi yang dibayarkan semakin besar.

1. *Segmen 1*: 331 nasabah, memiliki rata rata nilai yang cukup besar dari ketiga faktor. Maka dapat digolongkan sebagai nasabah yang cukup potensial
2. *Segmen 2*: 44 nasabah, segmen ini tergolong sebagai nasabah yang tidak potensial, karna hanya faktor *recency* saja yang sangat tinggi, sedangkan kedua faktor lainnya sangat kecil.
3. *Segmen 3*: 93 nasabah, dengan nilai dari ketiga faktor yang besar, maka segmen ini tergolong sebagai nasabah yang potensial.
4. *Segmen 4*: 191 nasabah, memiliki nilai yang sangat besar dari ketiga faktor, sehingga segmen ini tergolong sebagai nasabah yang sangat potensial.

5 KESIMPULAN

Dari penelitian yang dilakukan, *K-means* dengan algoritma *cluster* dinamik, terbukti dapat meningkatkan akurasi model yang terbentuk. Peningkatan kualitas model dapat dilihat dari peningkatan akurasi yang cukup signifikan. Pengukuran *validity cluster* dengan menggunakan *Davies-Bouldin Index* (DBI) dan *purity*, membuktikan bahwa *K-means* dengan algoritma *cluster* dinamik menghasilkan kualitas *cluster* yang lebih optimal yang ditunjukan dengan nilai DBI yang lebih kecil dibandingkan dengan *K-means* tradisional, dan *purity* untuk *K-means* dengan algoritma *cluster* dinamik yang lebih besar dibandingkan dengan *K-means* tradisional. Nilai DBI yang lebih kecil mendekati 0 dan *purity* yang lebih besar mendekati 1, menunjukan skema *cluster* yang optimal.

Meskipun model yang diusulkan sudah memberikan hasil yang lebih baik, namun untuk penelitian selanjutnya dapat menerapkan algoritma Dinamik K-means untuk dataset yang lebih beragam dan berbeda, dan pengurangan waktu komputasi untuk dataset yang besar.

REFERENSI

Aggarwal, N., & Aggarwal, K. (2012). Comparative Analysis of k-means and Enhanced K-means clustering algorithm for data mining, *International Journal of Scientific & Eninnering Research*, 3(3).

Balaji, S., & Srivatsa, S. K. (2012). Customer Segmentation for Decision Support using Clustering and Association Rule based approaches, *International Journal of Computer Science & Engineering Technology*, 3(11), 525–529.

Chai, C., & Chan, H. (2008). Intelligent value-based customer segmentation method for campaign management : A case study of automobile retailer, Expert System with Application, 34, 2754–2762.

Chan, C. H. (2005). Online Auction Customer Segmentation Using a Neural Network Model, *International Journal of Applied Science and Engineering* 101–109.

Deelers, S., & Auwatanamongkol, S. (2007). Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance, World Academy of Science, Engineering and Technology, 26(December), 323–328.

M, A. S. B., & Hareesha, K. S. (2012). Dynamic Clustering of Data with Modified K-Means Algorithm, 27(Icicn), 221–225.

Maulik, U., & Bandyopadhyay, S. (2002). Performance Evaluation of Some Clustering Algorithms and Validity Indices, *IEEE Transaction On Pattern Analysis And Machine Intelligence*, 24(12), 1650–1654.

Prasad, P. (2011). Generating Customer Profiles for Retail Stores Using Clustering Techniques, *International Journal on Computer Science and Enginnering*, 3(6), 2506–2510.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning and Tool*. Burlington: Morgan Kaufmann Publisher.

Wu, Xindong & Kumar, Vipin. (2009). *The Top Ten Algorithms in Data Mining*. London: CRC Press.

Xing, B. I. (2010). The Evaluation of Customer Potential Value Based on Prediction and Cluster Analysis, *International Conference on Management Science & Engineering 17th*, Melbourne, Australia 613–618.

Yi, B., Qiao, H., Yang, F., & Xu, C. (2010). An Improved Initialization Center Algorithm for K-Means Clustering. 2010 *International Conference on Computational Intelligence and Software Engineering, IEEE* (1), 1–4.

Zhang, C., & Fang, Z. (2013). An Improved K-means Clustering Algorithm Traditional K-mean Algorithm, Journal of Information & Computational Science, 1, 193–199

BIOGRAFI PENULIS

Widiarina. 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 masing-masing di Ilmu Komputer dari Saitama University, Jepang, dan Ph.D dalam Rekayasa Perangkat Lunak dari Universiti Teknikal Malaysia Melaka. Pengajar di Pascasarjana Ilmu Komputer, Universitas Dian Nuswantoro, Indonesia. Juga pendiri dan CEO PT Brainmatics, sebuah perusahaan pengembangan perangkat lunak di Indonesia. Minat penelitiannya saat ini meliputi rekayasa perangkat lunak dan machine learning. Anggota profesional dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.

Penerapan Metode Bagging untuk Mengurangi Data *Noise* pada Neural Network untuk Estimasi Kuat Tekan Beton

Tyas Setiyorini
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
Email: tyas_setiyorini@yahoo.com

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

Abstract: Beton adalah bahan yang diolah yang terdiri dari semen, agregat kasar, agregat halus, air dan bahan tambahan lainnya. Kuat tekan beton sangat bergantung pada karakteristik dan komposisi bahan-bahan pembentuk beton. Metode neural network memiliki kinerja yang baik dalam mengatasi masalah data nonlinear, namun neural network memiliki keterbatasan dalam mengatasi data *noise* yang tinggi. Untuk menyelesaikan masalah tersebut diperlukan metode bagging untuk mengurangi data *noise* pada metode neural network. Beberapa eksperimen dilakukan untuk mendapatkan arsitektur yang optimal dan menghasilkan estimasi yang akurat. Hasil eksperiman dari delapan kombinasi parameter penelitian pada dataset *concrete compressive strength* dengan neural network didapatkan nilai rata-rata RMSE adalah 0,069 dan nilai RMSE terkecil adalah 0,057, sedangkan dengan menggunakan neural network dan bagging didapatkan nilai rata-rata RMSE adalah 0,062 dan nilai RMSE terkecil adalah 0,055. Hasil eksperimen dari delapan kombinasi parameter penelitian pada dataset *slump* dengan neural network didapatkan nilai rata-rata RMSE adalah 0,020 dan nilai RMSE terkecil adalah 0,011 sedangkan dengan neural network dan bagging didapatkan nilai rata-rata RMSE adalah 0,016 dan nilai RMSE terkecil adalah 0,010. Maka dapat disimpulkan estimasi kuat tekan beton dengan menggunakan metode bagging dan neural network lebih akurat dibanding dengan metode individual neural network.

Keywords: estimasi, kuat tekan beton, neural network, bagging

1 PENDAHULUAN

Dewasa ini kata beton sudah tidak asing lagi, baik di masyarakat umum maupun di kalangan para *engineer*. Kelebihan sifat beton dibanding bahan lain adalah: 1). Mampu memikul beban yang berat, 2). Dapat dibentuk sesuai dengan kebutuhan konstruksi 3). Tahan terhadap temperatur yang tinggi 4). Biaya pemeliharaan yang kecil (Mulyono, 2004).

Beton adalah bahan yang diperoleh dengan mencampurkan semen hidrolik (*portland cement*), agregat kasar, agregat halus, air dan bahan tambahan (Mulyono, 2004). Sebagian besar bahan pembuat beton adalah bahan lokal (kecuali semen atau bahan tambahan kimia lain), sehingga sangat menguntungkan secara ekonomi. Namun pembuatan beton akan menjadi mahal jika perencanaannya tidak memahami karakteristik bahan-bahan pembentuk beton yang harus disesuaikan dengan perilaku struktur yang akan dibuat. Dalam ilmu sipil, memprediksi sifat mekanik bahan konstruksi adalah tugas penelitian yang penting (Chou & Pham, 2013). Sifat dan karakteristik bahan pembentuk beton akan mempengaruhi mutu beton (Mulyono, 2004).

Mutu beton yang baik adalah jika beton tersebut memiliki kuat tekan tinggi (antara 20–50 Mpa, pada umur 28 hari). Dengan kata lain dapat diasumsikan bahwa mutu beton ditinjau dari kuat tekannya saja (Tjokrodimuljo, 1996). Kuat tekan beton akan berubah sesuai dengan bertambahnya umur beton tersebut dengan umur 3, 7, 14, 21, 28, 90, dan 365 hari. (PBI, 1971). Secara umum, para ahli laboratorium melakukan *mix design* dengan aturan standar tertentu yang dilakukan secara manual dengan melihat tabel dan grafik referensi dan keadaan lapangan, tetapi cara tersebut sangat tidak efisien dan tidak menjamin akurasi. Untuk menjamin tingkat akurasi dalam memprediksi kuat tekan beton sampai saat ini telah banyak penelitian yang dilakukan dengan berbagai macam metode komputasi dengan berbagai jenis dataset kuat tekan beton, dalam cabang ilmu komputer yang disebut *data mining*.

Dari beberapa penelitian yang telah dilakukan disimpulkan bahwa metode individual yang paling baik adalah neural network. Hubungan antara komponen dan komposisi bahan pembentuk beton dengan kuat tekan beton bersifat sangat nonlinear (Chou & Pham, 2013; Erdal, 2013). Neural network memiliki keunggulan dalam memprediksi hasil dikotomis atau membuat keputusan diagnostik dibandingkan dengan model linear regression, termasuk kemampuan untuk mendeteksi hubungan kompleks yang bersifat nonlinear antara faktor prediksi dan hasil prediksi (Alshihri, Azmy, & El-Bisy, 2009; Chen, Zhang, Xu, Chen, & Zhang, 2012). Neural network menjadi alat yang sangat ampuh untuk memecahkan banyak masalah teknik sipil, khususnya dalam situasi, dimana data mungkin rumit atau dalam jumlah yang cukup besar (Nazari & Pacheco Torgal, 2013). Meskipun metode neural network telah mampu membuktikan dalam menangani masalah nonlinear, neural network masih memiliki beberapa kelemahan.

Seperti banyak penelitian lainnya, prediksi kuat tekan beton menderita efek negatif yaitu *noise* dari data pelatihan, hal ini dapat dapat mempengaruhi akurasi prediksi (Erdal, Karakurt, & Namli, 2013). *Noise* adalah data yang berisi nilai-nilai yang salah atau anomali, yang biasanya disebut juga *outlier*. Penyebab lain yang mungkin dari *noise* yang harus dicari dalam perangkat rusak adalah pengukuran data, perekaman dan transmisi. Itu adalah adanya data yang dinyatakan dalam satuan pengukuran heterogen, sehingga menyebabkan anomali dan ketidakakuratan (Vercellis, 2009). Dataset kuat tekan beton mengandung *noise* yang tinggi, hal ini dapat dilihat dari penyebaran data (varians) yang tidak merata atau meluas (heterogen). *Noise* yang tinggi pada dataset kuat tekan beton mengganggu proses estimasi, sehingga menyebabkan estimasi yang kurang akurat.

Neural network memiliki kemampuan untuk model yang kompleks dengan berbagai masalah nonlinier, namun kelemahan utama dari neural network adalah ketidakstabilan

mereka, terutama dalam kondisi noise dan dataset yang terbatas (Dimopoulos, Tsiros, Serelis, & Chronopoulou, 2004). Neural network telah sangat berhasil dalam sejumlah aplikasi pemrosesan sinyal, namun keterbatasan fundamental dan kesulitan yang melekat yaitu ketika menggunakan neural network untuk pengolahan *noise* yang tinggi dan sinyal ukuran sampel yang kecil (Giles & Lawrence, 2001). Untuk mengatasi kelemahan neural network dalam mengatasi data *noise* yang tinggi dibutuhkan metode gabungan dengan metode lain untuk memecahkan masalah data *noise* agar mendapatkan prediksi yang lebih akurat dibandingkan dengan metode individual.

Breiman (1996) menganggap bagging sebagai teknik pengurangan varians (*noise*) untuk metode dasar seperti decision tree atau neural network (Breiman, (1996). Bagging dikenal sangat efektif bila pengklasifikasi tidak stabil, yaitu ketika penturbing set pembelajaran dapat menyebabkan perubahan yang signifikan dalam perilaku klasifikasi, karena bagging meningkatkan kinerja generalisasi dengan cara mengurangi varians (*noise*) dengan tetap menjaga atau hanya sedikit meningkatkan bias (Breiman, 1996). Menurut Wange et al dalam Erdal et al, bagging mampu mengurangi pengaruh *noise* (Erdal et al., 2013). Modifikasi algoritma hybrid bagging, mampu menyediakan kecepatan komputasi, perbaikan tambahan dalam akurasi, dan ketahanan untuk vektor respon *noise* (Culp, Michailidis, & Johnson, 2011). Bagging sering memiliki akurasi secara signifikan besar, dan lebih kuat terhadap efek *noise* dan overfitting dari data pelatihan asli (Han, Kamber, & Pei, 2012). Bagging juga baik diterapkan untuk skema pembelajaran untuk prediksi numerik (Witten, Frank, & Hall, 2011). Bagging adalah algoritma yang tepat untuk mengurangi data *noise* pada neural network, serta baik diterapkan pada dataset kuat tekan beton yang memiliki atribut dan label yang bersifat numerik.

2 PENELITIAN TERKAIT

Dataset kuat tekan beton merupakan dataset yang kompleks, sangat nonlinear dan mengandung *noise* yang tinggi. Neural network adalah alat yang baik untuk model sistem nonlinear, namun masih kurang mampu mengatasi *noise*. Penelitian yang dilakukan oleh Alshihri (Alshihri et al., 2009) dalam peningkatan akurasi prediksi metode neural network untuk memprediksi kekuatan tekan beton ringan dengan dua model yang digunakan yaitu, Feed-forward Back Propagation (BP) dan Cascade Correlation (CC). Hasil disimpulkan bahwa CC mampu mengurangi *noise* pada neural network, serta menunjukkan hasil yang sedikit akurat dan mampu belajar dengan sangat cepat dibandingkan dengan prosedur backpropagation.

CC masih menunjukkan hasil yang sedikit akurat dan belum mampu mengurangi noise yang tinggi pada neural network, oleh karena itu dilakukan penelitan oleh Erdal et al (Erdal et al., 2013) dengan menggunakan metode gabungan *(ensemble)* neural network, discrete wavelet transform dan gradien boosting. Hasil penelitian menunjukkan nilai RMSE yang menurun dibanding dengan metode individual neural network.

Pendekatan model *ensemble* kembali dilakukan oleh Erdal (Erdal, 2013) dengan metode decision tree. Decision tree mudah dipengaruhi oleh data *noise*. Hasil penelitian ini menunjukkan metode *ensemble* bagging mampu mengurangi *noise* pada decision tree.

Dari permasalahan pada penelitian-penelitian di atas disimpulkan bahwa dataset kuat tekan beton merupakan data kompleks yang bersifat nonlinear dan memiliki data *noise* yang tinggi. Berdasarkan analisa bahwa metode gabungan dua metode atau lebih *(ensemble)* menunjukkan hasil yang lebih akurat dibanding metode individual. Neural netwok yang ampuh mengatasi masalah data nonlinear namun kurang mampu mengatasi data *noise* yang tinggi, sedangkan bagging mampu mengurangi data *noise*. Oleh karena itu pada penelitian ini diusulkan menggunakan metode *ensemble* dengan menggabungkan metode bagging untuk mengurangi *noise* pada neural network dengan kombinasi *adjustment* parameter yang berbeda-beda.

3 PENGUMPULAN DATA

Dalam penelitian ini dikumpulkan dua dataset yaitu dataset *concrete compressive strength* dan dataset *slump* seperti pada Tabel 1 dan Tabel 2 yang menampilkan unit, nilai minimal, maximal, mean, varians dan standard deviation dari variabel input dan variabel target (*class*). Dari model input ini menunjukkan seberapa tinggi tingkat *noise* pada dataset terlihat dari besarnya nilai varians atau perbandingan nilai standard deviation yang lebih besar dari nilai mean.

Tabel 1. Dataset *Concrete Compressive Strength*

Input	Unit	Min	Max	Mean	Varians	Standard Deviation
Cement	kg/m³	102,0	540,0	281,2	10911,14	104,46
Blast furnace slag	kg/m³	0,0	359,4	73,9	7436,86	86,24
Fly ash	kg/m³	0,0	200,1	54,2	4091,57	63,97
Water	kg/m³	121,8	247,0	181,6	455,62	21,35
Superplasticizer	kg/m³	0,0	32,2	6,2	35,65	5,97
Coarse aggregate	kg/m³	801,0	1145,0	972,9	6039,79	77,72
Fine aggregate	kg/m³	594,0	992,0	773,6	6421,86	80,14
Age	day	1	365	46	3986,56	63,14
Concrete compressive strength (class)	MPa	2,332	82,599	35,8	278,81	16,70

Tabel 2. Dataset *Slump*

Input	Unit	Min	Max	Mean	Varians	Standard Deviation
Cement	kg/m³	137,00	374,00	229,9	6161,21	78,49
Slag	kg/m³	0,00	193,00	78,0	3620,09	60,17
Fly ash	kg/m³	0,00	260,00	149,0	7225,41	85,00
Water	kg/m³	160,00	240,00	197,2	404,40	20,11
SP	kg/m³	2,00	22,00	8,6	9,76	3,12
Coarse aggregate	kg/m³	708,00	1049,90	884,0	7737,18	87,96
Fine aggregate	kg/m³	640,60	902,00	739,6	3973,27	63,03
Slump	cm	0,00	29,00	18	75,83	8,71
Flow	cm	20,00	78,00	49,6	305,65	17,48
Compressive strength (28 day) (class)	MPa	17,19	58,53	36,1	229,43	7,84

4. PENGOLAHAN DATA AWAL (*PREPROCESSING*)

Merupakan tindak lanjut dari pengumpulan data, dengan melakukan normalisasi data. Normalisasi data dilakukan sesuai fungsi aktivasi yang digunakan, dalam penelitian ini digunakan fungsi *binary sigmoid*, data harus dinormalisasikan dalam *range* 0 sampai 1, tapi akan lebih baik jika ditransformasikan ke interval yang lebih kecil, misal pada interval [0.1,0.9] (Jong Jek Siang, 2009). Maka, pada data sinoptik yang ada dilakukan *transform* data dengan interval [0.1,0.9], dengan rumus sebagai berikut:

$$x^1 = \frac{0{,}8\ (x-a)}{b-a} + 0{,}1 \qquad (1)$$

Keterangan : x^1 = nilai transform
x = nilai asli
a = nilai minimal
b = nilai maximal

5 METODE YANG DIUSULKAN

5.1 Baging

Bagging adalah singkatan dari *bootstrap aggregating*, menggunakan sub-dataset (*bootstrap*) untuk menghasilkan set pelatihan L (*learning*), L melatih dasar belajar menggunakan prosedur pembelajaran yang tidak stabil, dan kemudian, selama pengujian, mengambil rata-rata (Breiman 1996). Bagging baik digunakan untuk klasifikasi dan regresi. Dalam kasus regresi, untuk menjadi lebih kuat, seseorang dapat mengambil rata-rata ketika menggabungkan prediksi. Bagging adalah (Alypadin, 2010) sebuah algoritma pembelajaran yang stabil pada perubahan kecil dalam *training* set menyebabkan perbedaan besar dalam peserta didik yang dihasilkan, yaitu algoritma belajar pada data yang memiliki varians tinggi (*noise*). Bagging mampu meningkatkan akurasi secara signifikan lebih besar dibanding model individual, dan lebih kuat terhadap efek *noise* dan *overfitting* dari data pelatihan asli. (Han et al., 2012; Culp et al., 2011).

Algoritma Bagging (Brieman, 1996):

Perulangan for b = 1, 2, . . ., B

1. Buat sampel *boostrap* $\{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \dots, (X_n^*, Y_n^*)\}$ dengan penggantian secara acak dari data *training* $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ mencocokkan dengan *classifier* C_b dinyalakan pada sampel yang sesuai *bootstrap*.
2. Output *classifier* akhir:

$$C(x) = B^{-1} \sum_{b=1}^{B} C_b(x) \qquad (2)$$

Karya (Breiman, 1994) pada Kim & Kang melaporkan bahwa bagging dapat meningkatkan kinerja dengan penggabungan (*ensemble*) algoritma seperti Decision Tree (DT), Neural Network (NN), dan Support Vector Machine (SVM) (M. Kim & Kang, 2012).

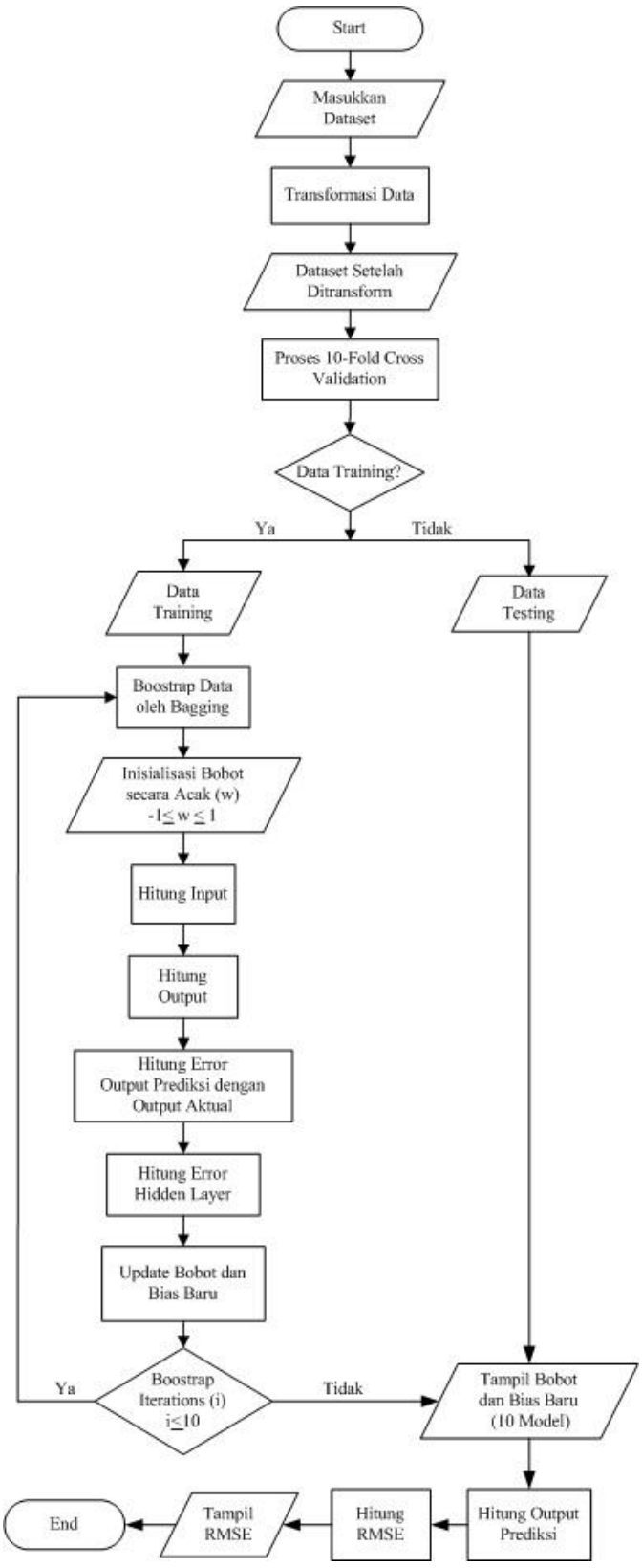
Dataset dengan *noise* yang tinggi menyebabkan kesalahan dalam generalisasi pengklasifikasian, sehingga dibutuhkan algoritma yang tepat untuk digabungkan (*ensemble*) dengan neural network agar akurasi prediksi dapat meningkat.

5.2 Neural Network

Neural Network (NN) atau biasa juga disebut Jaringan Saraf Tiruan (JST) merupakan upaya untuk meniru fungsi otak manusia. Otak manusia diyakini terdiri dari jutaan unit pengolahan kecil yang bekerja secara paralel, yang disebut neuron. Neuron saling terhubung satu sama lain melalui koneksi neuron. Setiap individu neuron mengambil input dari satu set neuron. Hal ini kemudian memproses input tersebut dan melewati output untuk satu set neuron. Output dikumpulkan oleh neuron lain untuk diproses lebih lanjut (Shukla, Tiwari, & Kala, 2010).

Pada penelitian ini menggunakan algoritma backporpagation. Algoritma backpropagation bekerja melalui proses secara iteratif menggunakan data *training*, membandingkan nilai prediksi dari jaringan dengan setiap data yang terdapat pada data *training* (Han et al., 2012).

5.3 Neural Network dan Bagging



Gambar 1. Penggabungan Algoritma Bagging dan Neural Network

Gambar 1 menggambarkan metode yang diusulkan dalam penelitian ini yaitu metode bagging pada neural network. Pada pengolahan dataset awal, dataset ditransformasi ke dalam range 0 sampai 1. Kemudian dataset dibagi dengan metode 10-fold cross validation yaitu dibagi menjadi data *testing* dan data *training*. Kemudian data *training* dibagi lagi oleh bagging

menjadi sub dataset (*boostrap*) sebanyak 10 perulangan (*iterations*). Masing-masing *boostrap* data kemudian diproses dengan metode neural network. Langkah awal neural network yaitu memberikan inisialisasi bobot awal untuk *input layer*, *hidden layer*, dan bias secara acak. Simpul bias terdiri dari dua, yaitu pada input layer yang terhubung dengan simpul-simpul pada *hidden layer*, dan *hidden layer* yang terhubung pada *output layer*. Setelah semua nilai awal dinisialisasi, kemudian dihitung *input*, *output* dan *error*. Selanjutnya membangkitkan *output* untuk simpul menggunakan fungsi aktifasi *sigmoid*. Selanjutnya dihitung nilai *error output* prediksi dengan *output* aktual, selanjutnya dibalik ke layer sebelumnya (backpropagation) untuk menghitung *error* pada *hidden layer*. Proses neural network tersebut akan terus berulang sebanyak 10 perulangan *boostrap*. Setelah perulangan selesai semua hasil model *boostrap* dihitung hingga menghasilkan 10 model. Selanjutnya dihitung output prediksi rata-rata 10 model tersebut. Kemudian dihitung *error* rata-rata selisih antara *output* prediksi dengan *output* aktual yaitu *Root Mean Square Error* (RMSE).

6 HASIL DAN PEMBAHASAN

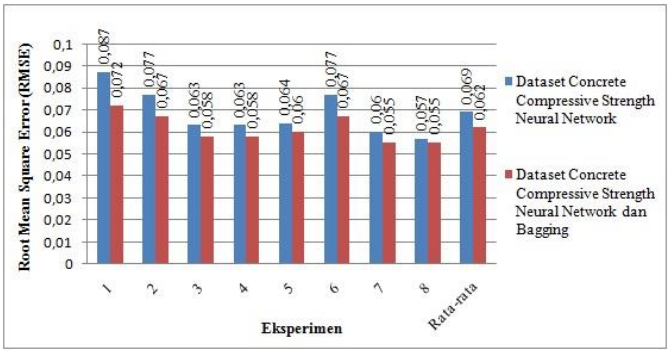
Penelitian yang dilakukan menggunakan komputer dengan spesifikasi CPU Intel Core i5 1.6GHz, RAM 8GB, dan sistem operasi Microsoft Windows 7 Professional 64-bit. Aplikasi yang digunakan adalah RapidMiner 5.2. Penelitian ini menggunakan dua dataset yaitu dataset *concrete compressive strength* dataset *slump*. Dataset ini didapat dari UCI Machine Learning Repository.

Setelah eksperimen yang dilakukan dengan neural network dan neural network dan bagging, kemudian dikomparasi hasil RMSE pada metode neural network dengan neural network dan bagging dari 8 eksperimen pada dataset *concrete compressive strength* dan dataset *slump.*

Pada Tabel 3 dan Gambar 2 dari 8 eksperimen dan rata-rata keseluruhan eksperimen pada dataset *concrete compressive strength* secara konsisten menunjukkan penurunan nilai RMSE yang signifikan antara neural network dengan neural network dan bagging.

Tabel 3. Hasil Eksperimen Neural Network dengan Neural Network dan Bagging (Dataset *Concrete Compressive Strength*)

Parameter Neural Network					Parameter Bagging	RMSE	
Training Cycles	*Learning Rate*	*Momentum*	*Hidden Layer 1*	*Hidden Layer 2*	*Iterations*	NN	NN+Bagging
500	0,3	0,2	2	-	10	0,087	0,072
500	0,3	0,2	6	-	10	0,077	0,067
500	0,1	0,1	6	-	10	0,063	0,058
500	0,1	0,2	6	-	10	0,063	0,058
300	0,1	0,2	6	-	10	0,064	0,060
450	0,3	0,2	6	-	10	0,077	0,067
1000	0,1	0,2	12	10	10	0,060	0,055
1000	0,1	0,2	13	9	10	0,057	0,012
Rata-rata						**0,069**	**0,062**



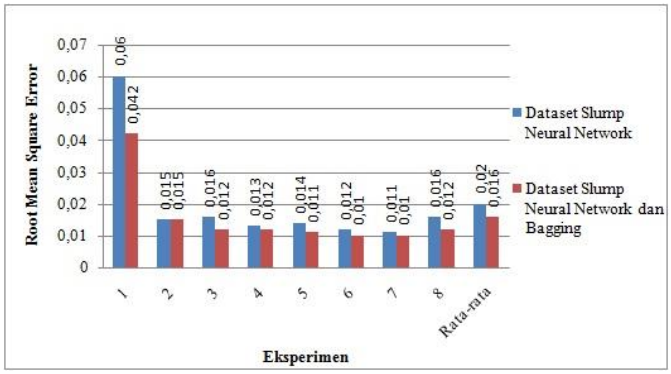
Gambar 2. Grafik Perbedaan Metode Neural Network dengan Neural Network dan Bagging (Dataset *Concrete Compressive Strength*)

Pada Tabel 4 dan Gambar 3 dari delapan eksperimen dan rata-rata keseluruhan eksperimen pada dataset *slump* juga secara konsisten menunjukkan penurunan nilai RMSE antara neural network dengan neural network dan bagging.

Penurunan nilai RMSE yang signifikan dan konsisten dari enam belas eksperimen pada dataset *concrete compressive strength* yang ditunjukkan pada Tabel 3 dan Gambar 2 serta dataset *slump* yang ditunjukkan pada Tabel 4 dan Gambar 3, menunjukkan bahwa penggunaan neural network dan bagging memiliki nilai RMSE lebih kecil dibanding dengan penggunaan neural network saja. Dari hasil pengujian tersebut menunjukkan bahwa bagging mampu mengurangi data *noise* pada neural network, sehingga menghasilkan kinerja atau tingkat akurasi estimasi kuat tekan beton yang lebih baik dibanding dengan menggunakan metode individual neural network. Merujuk pada penelitian yang telah dilakukan sebelumnya oleh (Kim & Kang, 2010) dalam mengkomparasi hasil *error rate* neural network dengan neural network dan bagging pada 10 eksperimen yang berbeda, hasil penelitian tersebut juga menunjukkan penurunan *nilai error rate* yang konsisten dan signifikan, sehingga menunjukkan bahwa metode neural network dan bagging memiliki kinerja yang lebih baik dibanding neural network saja.

Tabel 4. Hasil Eksperimen Neural Network dengan Neural Network dan Bagging (Dataset *Slump*)

Parameter Neural Network					Parameter Bagging	RMSE	
Training Cycles	*Learning Rate*	*Momentum*	*Hidden Layer 1*	*Hidden Layer 2*	*Iterations*	NN	NN+Bagging
500	0,3	0,2	2	-	10	0,060	0,042
500	0,1	0,5	6	-	10	0,015	0,015
1000	0,3	0,2	6	-	10	0,016	0,012
1000	0,1	0,5	6	-	10	0,013	0,012
3000	0,3	0,2	6	-	10	0,014	0,011
3000	0,1	0,5	6	-	10	0,012	0,010
3000	0,1	0,5	14	-	20	0,011	0,010
3000	0,1	0,5	14	1	10	0,016	0,012
Rata-rata						**0,020**	**0,016**



Gambar 3. Grafik Perbedaan Metode Neural Network dengan Neural Network dan Bagging (Dataset *Slump*)

Setelah diuji distribusi data dengan menggunakan SPSS, hasil uji distribusi data hasil eksperimen neural network dengan neural network dan bagging pada dataset *concrete compressive strength* seperti pada Tabel 5 dan hasil uji distribusi data hasil eksperimen neural network dengan neural network dan bagging pada dataset *slump* seperti pada Tabel 6 menunjukkan bahwa nilai sifnifikansi (Asymp.Sig > 0.05) maka data tersebut adalah berdistribusi normal.

Tabel 5. Hasil Uji Distribusi Data Hasil Eksperimen Neural Network dengan Neural Network dan Bagging pada Dataset *Concrete Compressive Strength*

One-Sample Kolmogorov-Smirnov Test			
		X1	X2
N		8	8
Normal Parameters	Mean	.06850	.06150
	Std. Deviation	.010502	.006374
Most Extreme Differences	Absolute	.291	.218
	Positive	.291	.218
	Negative	-.166	-.182
Kolmogorov-Smirnov Z		.823	.618
Asymp. Sig. (2-tailed)		.508	.840

Tabel 6. Hasil Uji Distribusi Data Hasil Eksperimen Neural Network dengan Neural Network dan Bagging pada Dataset *Slump*

One-Sample Kolmogorov-Smirnov Test			
		X1	X2
N		8	8
Normal Parameters	Mean	.01963	.01550
	Std. Deviation	.016414	.010823
Most Extreme Differences	Absolute	.462	.393
	Positive	.462	.393
	Negative	-.300	-.306
Kolmogorov-Smirnov Z		1.308	1.113
Asymp. Sig. (2-tailed)		.065	.168

Untuk menjamin evaluasi hipotesis penelitian ini, dibutuhkan pengujian dengan metode statistik untuk menguji hubungan antara penggunaan metode neural network dengan neural network dan bagging, apakah terdapat hubungan di antara keduanya. Dikarenakan data hasil eksperimen berdistribusi normal maka pengujian hipotesis menggunakan metode t-Test (Dem, 2006). Metode ini termasuk yang paling umum dalam metode statistik tradisional, yaitu t-Test (Maimon, 2010). Ada atau tidaknya perbedaan antara dua model membutuhkan pengujian, salah satunya dengan uji t-Test (Larose, 2007), dengan melihat nilai P. Jika nilai P < 0,05 maka menunjukkan hipotesis nol ditolak atau disebut hipotesis alternatif (Sumanto, 2014). Hipotesis nol menyatakan tidak ada pengaruh atau perbedaan antara dua buah variabel, sedangkan hipotesis alternatif menyatakan adanya pengaruh atau perbedaan antara dua buah variabel (Sumanto, 2014).

Pada Tabel 7 menampilkan t-Test untuk hasil RMSE pada dataset *concrete compressive strength* menunjukkan hipotesis nol ditolak (hipotesis alternatif) yaitu dengan nilai P < 0,05 yaitu 0,0004. Pada Tabel 8 t-Test untuk hasil RMSE pada dataset *slump*, juga menunjukkan hipotesis nol ditolak (hipotesis alternatif) yaitu dengan nilai P < 0,05 yaitu 0,0259.

Hasil t-Test dengan hipotesis nol ditolak (hipotesis alternatif) tersebut menunjukkan bahwa antara penggunaan metode neural network dengan neural network dan bagging menunjukkan adanya pengaruh atau perbedaan yang sifnifikan. Neural network dan bagging menghasilkan kinerja atau tingkat akurasi yang lebih baik dibanding dengan menggunakan metode neural network saja. Hal tersebut seperti dikatakan pada penelitian Kim & Kang (Kim & Kang, 2010) bagging secara konsisten menunjukkan mampu meningkatkan akurasi prediksi pada neural network, hal ini berarti bahwa dua metode gabungan (*ensemble*) bagging dan neural network yang diusulkan dapat menjadi alat yang efektif untuk meningkatkan kinerja neural network.

Tabel 7. *Paired Two-tailed t-Test* dengan Metode Neural Network dan Bagging (Dataset *Concrete Compressive Strength*)

	Variable 1	*Variable 2*
Mean	0,0685	0,0615
Variance	9,6 E-05	0,00003525
Observations	9	9
Pearson Correlation	0,99231096	
Hypothesized Mean Difference	0	
df	8	
t Stat	5,25	
P(T<=t) one-tail	**0,00038692**	
t Critical one-tail	1,85954803	
P(T<=t) two-tail	0,00077383	
t Critical two-tail	2,30600413	

Tabel 8. *Paired Two-tailed t-Test* dengan Metode Neural Network dan Bagging (Dataset *Slump*)

	Variable 1	*Variable 2*
Mean	0,019625	0,0155
Variance	0,000235734	0,0001025
Observations	9	9
Pearson Correlation	0,993526309	
Hypothesized Mean Difference	0	
df	8	
t Stat	2,283872233	
P(T<=t) one-tail	**0,025878775**	
t Critical one-tail	1,859548033	
P(T<=t) two-tail	0,05175755	
t Critical two-tail	2,306004133	

7 KESIMPULAN

Hasil eksperiman dari delapan kombinasi parameter penelitian pada dataset *concrete compressive strength* dengan neural network didapatkan nilai rata-rata RMSE adalah 0,069 dan nilai RMSE terkecil adalah 0,057, sedangkan dengan menggunakan neural network dan bagging didapatkan nilai rata-rata RMSE adalah 0,062 dan nilai RMSE terkecil adalah 0,055. Hasil eksperimen dari delapan kombinasi parameter penelitian pada dataset *slump* dengan neural network didapatkan nilai rata-rata RMSE adalah 0,020 dan nilai RMSE terkecil adalah 0,011 sedangkan dengan neural network dan bagging didapatkan nilai rata-rata RMSE adalah 0,016 dan nilai RMSE terkecil adalah 0,010.

Dari hasil pengujian di atas maka dapat disimpulkan bahwa bagging mampu mengurangi data *noise* pada neural network, sehingga menghasilkan kinerja atau tingkat akurasi estimasi kuat tekan beton yang lebih baik dibanding dengan menggunakan metode individual neural network.

REFERENSI

Alshihri, M. M., Azmy, A. M., & El-Bisy, M. S. (2009). Neural networks for predicting compressive strength of structural light weight concrete. Construction and Building Materials, 23(6), 2214–2219.

Breiman, L. (1996). Bagging predictors. Machine Learning, 26(2), 123–140.

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.

Chou, J.-S., & Pham, A.-D. (2013). Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. Construction and Building Materials, 49, 554–563.

Culp, M., Michailidis, G., & Johnson, K. (2011). On Adaptive Regularization Methods in Boosting. Journal of Computational and Graphical Statistics, 20(4), 937–955.

Dem, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets, 7, 1-30

Dimopoulos, L. F., Tsiros, L. X., Serelis, K., & Chronopoulou, A. (2004). Combining Neural Network Models to Predict Spatial Patterns of Airborne Pollutant Accumulation in Soils around an Industrial Point Emission Source. Journal of the Air & Waste Management Association, 54(12), 1506–1515.

Erdal, H. I. (2013). Two-level and hybrid ensembles of decision trees for high performance concrete compressive strength prediction. Engineering Applications of Artificial Intelligence, 26(7), 1689–1697.

Erdal, H. I., Karakurt, O., & Namli, E. (2013). High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform. Engineering Applications of Artificial Intelligence, 26(4), 1246–1254.

Han, J., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Techniques. San Fransisco: Morgan Kauffman.

Jong Jek Siang. (2009). Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB. Yogyakarta : Andi Yogyakarta.

Kim, M., & Kang, D. (2012). Expert Systems with Applications Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction. Expert Systems With Applications, 39(10), 9308–9314.

Kim, M.-J., & Kang, D.-K. (2010). Ensemble with neural networks for bankruptcy prediction. Expert Systems with Applications, 37(4), 3373–3379.

Larose, D. T. (2007). Data Mining Methods and Model. New Jersey: John Willey & Sons, Inc.

Maimon, Oded&Rokach, Lior. (2010). Data Mining and Knowledge Discovey Handbook. New York: Springer

Mulyono, Tri. (2004). Teknologi Beton. Yogyakarta: Andi Publishing.

Nazari, A., & Pacheco Torgal, F. (2013). Predicting compressive strength of different geopolymers by artificial neural networks. Ceramics International, 39(3), 2247–2257.

Shukla, A., Tiwari, R., & Kala, R. (2010). Real Life Applications of Soft Computing. New York: CRC Press.

Sumanto. (2014). Statistika Deskriptif. Yogyakarta: Center of Academic Publishing Service.

Tjokrodimuljo, Kardiyono. (1996). Teknologi Beton. Yogyakarta: Nafiri.

Vercellis, Carlo (2009). Business Intelligent: Data Mining and Optimization for Decision Making. Southern Gate, Chichester, West Sussex: John Willey & Sons, Ltd.

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning and Tools. Burlington: Morgan Kaufmann Publisher.

BIOGRAFI PENULIS

Tyas Setiyorini. Menempuh pendidikan S1 Sistem Informasi di STMIK Swadharma, Jakarta, S2 Magister Sistem Informasi di STMIK Nusa Mandiri Jakarta. Saat ini menjadi dosen di Akademi Bina Sarana Informatika. Minat penelitian saat ini adalah 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.

Penerapan Gravitational Search Algorithm untuk Optimasi Klasterisasi Fuzzy C-Means

Ali Mulyanto
Program Studi Teknik Informatika STMIK Eresha
Email: aliemulyanto@gmail.com

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

Abstract: Klasterisasi fuzzy merupakan masalah penting yang merupakan subjek penelitian aktif dalam beberapa aplikasi dunia nyata. Algoritma fuzzy c-means (FCM) merupakan salah satu teknik pengelompokan fuzzy yang paling populer karena efisien, dan mudah diimplementasikan. Namun, FCM sangat mudah terjebak pada kondisi local minimum. Gravitational search algorithm (GSA) merupakan salah satu metode heuristik yang efektif untuk menemukan solusi optimal terdekat. GSA digabungkan ke FCM untuk menemukan pusat klaster yang optimal dengan meminimalkan fungsi objektif FCM. Hasil penelitian menunjukkan bahwa metode yang diusulkan gravitational search algorithm fuzzy c-means (GSA-FCM) dapat menunjukkan hasil yang lebih optimal daripada algoritma FCM.

Keywords: klasterisasi fuzzy, Fuzzy c-means, gravitational search algorithm

1. PENDAHULUAN

Klasifikasi dan klasterisasi merupakan dua bidang garapan yang paling sering ditemui untuk mengekstrak pengetahuan. Analisis klaster merupakan metode *unsupervised learning* dalam analisis data yang digunakan untuk membuat penilaian awal dari struktur data, untuk menemukan struktur yang tersembunyi dalam dataset, dan untuk mengekstrak informasi (Azar, El-Said, & Hassanien, 2013). Sebagai metode *unsupervised learning*, tujuan dari analisis klaster adalah untuk menemukan fitur struktur data melalui metode pemartisan.

Klasterisasi merupakan proses mengelompokkan objek atau pola yang bertujuan untuk menempatkan objek data ke dalam satu himpunan atau kelompok yang saling berhubungan (disebut klaster) sehingga objek dalam setiap klaster memiliki kemiripan satu sama lain. Dengan demikian unsur-unsur dalam klaster memiliki derajat kesamaan yang besar daripada kesamaan data tersebut dengan data dalam kelompok lain (Izakian & Abraham, 2011). Dalam beberapa tahun terakhir, klasterisasi telah banyak diterapkan di berbagai bidang seperti pattern recognition, machine learning, data mining (Izakian & Abraham, 2011), analisis data statistik, dan segementasi citra (Taherdangkoo & Bagheri, 2013). Teknik klasterisasi yang paling populer adalah metode hirarki dan metode partisi (de Carvalho, Lechevallier, & de Melo, 2012).

Klasterisasi hirarki menemukan urutan partisi yang diawali dari satu data tunggal yang dianggap sebagai sebuah kelompok, dua atau lebih kelompok kecil data bergabung menjadi sebuah kelompok besar dan begitu seterusnya sampai semua data dapat bergabung menjadi sebuah kelompok *singleton* (Pimentel & de Souza, 2013). Metode ini dapat diklasifikasikan lebih lanjut ke dalam metode agglomerative dan metode devisive (Azar et al., 2013). Metode agglomerative (de Carvalho et al., 2012) menghasilkan urutan partisi bersarang dimulai dengan pengelompokan terendah di mana setiap item data berada dalam klaster yang unik dan berakhir dengan pengelompokan dimana semua item data berada dalam klaster yang sama. Sedangkan metode devisive dimulai dengan semua item data dalam satu klaster dan melakukan prosedur pemisahan sampai kriteria berhenti terpenuhi atau setelah mendapatkan partisi klaster tunggal.

Klasterisasi partisi secara langsung membagi dataset ke beberapa klaster yang tetap menggunakan fungsi objektif yang sesuai. Keuntungan dari metode partisi adalah kemampuannya untuk memanipulasi dataset dalam jumlah yang besar. Pimentel & de Souza (2013) mengelompokkan klasterisasi partisi ke dalam *hard partition* dan *fuzzy partition.* Dalam metode pengelompokan *hard partition*, setiap objek dari kumpulan data harus ditugaskan secara tepat pada satu klaster. Kelemahan utama dari teknik pengelompokan *hard partition* (Azar et al., 2013) adalah bahwa dengan metode ini mungkin akan kehilangan beberapa informasi penting yang mengarah pada pengelompokan tersebut. Sedangkan pengelompokan f*uzzy partition* didasarkan pada gagasan dari keanggotaan parsial dari masing-masing pola dalam sebuah klaster tertentu. Hal ini memberikan fleksibilitas untuk menyatakan bahwa titik data memiliki lebih dari satu klaster pada waktu yang sama dan derajat keanggotaannya jauh lebih halus dari model data. Selain menetapkan titik data ke dalam klaster, menurut (Azar et al. (2013) derajat keanggotaan juga bisa mengungkapkan ambiguitas titik data yang dimiliki sebuah klaster.

Ada beberapa metode yang digunakan untuk klasterisasi (Oliveira & Pedrycz, 2007) yaitu k-means, possibilistic c-means (PCM) dan fuzzy c-means (FCM). K-means merupakan salah satu teknik klasterisasi terkenal untuk *hard partition*. Algoritma k-means adalah algoritma pengelompokkan partisi yang efisiensi dalam mengelompokkan dataset yang besar. Namun menurut (Bai, Liang, & Dang, 2011), penggunaan algoritma k-means terbatas hanya pada data numerik. Klasterisasi PCM merupakan salah satu metode klasifikasi yang kuat terhadap *noise* atau data terisolasi (Hamasuna, Endo, & Miyamoto, 2009). Namun algoritma klasterisasi PCM (Ji, Sun, & Xia, 2011) mengorbankan stabilitas algoritma dan terlalu sentitif terhadap inisialisasi klaster.

2. PENELITIAN TERKAIT

FCM merupakan salah satu metode pengelompokkan yang paling terkenal (Wu, 2012) (Maimon & Rokach, 2010), paling banyak digunakan (Zhao, Jiao, & Liu, 2013). Algoritma FCM juga memiliki karakteristik yang kuat untuk ambiguitas dan dapat menyimpan informasi lebih banyak daripada metode *hard c-means* (Yong Zhang, Huang, Ji, & Xie, 2011). Namun algoritma FCM dalam pencarian klaster yang optimal

didasarkan pada fungsi objektif, sehingga mudah terjebak pada kondisi dimana nilai yang dihasilkan bukan nilai terendah dari himpunan solusi atau disebut *local minimum* (Dong, Dong, Zhou, Yin, & Hou, 2009).

Untuk memecahkan masalah pada algoritma FCM, para peneliti telah berhasil menerapkan algoritma evaluasi untuk meningkatkan kinerja FCM seperti ant colony optimization (Kanade & Hall, 2007). Kanade dan Hall (2007) memperkenalkan suatu algoritma dengan konsep peningkatan laju penguapan feromon data yang lebih dekat ke pusat klaster dan mencapai respon yang jauh lebih baik daripada algoritma sebelumnya. Evolutionary Programming Fuzzy C-Means (EPFCM) dimanfaatkan untuk mengoptimalkan fungsi objektif pada FCM (Dong et al., 2009). Algoritma artificial bee colony yang diusulkan oleh (Karaboga & Ozturk, 2010) meniru perilaku lebah madu dalam mencari makanan untuk mengatasi masalah seleksi acak pada pusat kalster awal pada FCM. Fuzzy Particle Swarm Optimization (FPSO) (Izakian & Abraham, 2011) juga telah digunakan untuk mengatasi masalah seleksi acak di titik pusat FCM.

Gravitational search algorithm (GSA) merupakan salah satu metode optimisasi *heuristik* yang efektif dalam analisa klaster yang diusulkan untuk memecahkan masalah pada FCM. Motivasi penggunaan GSA didasarkan pada kesuksesan para peneliti dalam memecahkan masalah optimasi. Kelebihan GSA menurut (Rashedi, Nezamabadi-pour, & Saryazdi, 2009) adalah kemampuan menemukan hasil yang lebih optimal dari algoritma optimasi yang lain. Kelebihan lain dari GSA (Kumar, Chhabra, & Kumar, 2014) terletak pada penggunaan memori yang lebih kecil dari algoritma optimasi lainnya, serta posisi agen yang ikut berpartisipasi dalam memperbarui iterasi. Pendekatan GSA yang diusulkan oleh (Rashedi, Nezamabadi-pour, & Saryazdi, 2011) telah digunakan untuk memecahkan masalah estimasi parameter untuk *infinite impulse response* (IIR) dan menfilter rasional *non-linier*. Hal ini menunjukkan bahwa GSA dapat memecahkan masalah yang kompleks dan hasil penentiannya menunjukkan bahwa kinerja GSA sebanding dengan algoritma genetic algorithm dan particle swarm optimization. Pendekatan GSA juga diusulkan oleh (Hatamlou, Abdullah, & Nezamabadi-pour, 2012) yang telah digunakan untuk mencari ruang masalah dalam menemukan solusi optimal yang terdekat untuk memecahkan masalah pada algoritma k-means.

Pada penelitian ini, GSA akan diterapkan untuk mengoptimalkan fungsi objektif pada FCM.

3. METODE YANG DIUSULKAN

Metode yang diusulkan untuk mengatasi masalah *local minimum* adalah algoritma berbasis populasi, dimana beberapa calon solusi untuk masalah pengelompokkan diciptakan secara acak. Masing-masing solusi kandidat yang juga disebut massa (agen), menemukan pusat klaster. Setelah membuat solusi acak untuk masalah klasterisasi, calon solusi berinteraksi sebagai massa dalam semesta melalui hukum gravitasi Newton. Dengan cara ini, calon agen yang juga memiliki massa yang besar, menarik massa lain dan menjadi agen untuk menemukan solusi yang lebih baik. Jumlah massa untuk setiap agen akan dihitung dengan fungsi objektif calon agen tersebut. Solusi akan ditemukan ketika agen memiliki nilai fungsi objektif terkecil dan memiliki massa yang besar.

Kontribusi utama dari metode yang diusulkan adalah memanfaatkan algoritma GSA untuk mengoptimalkan fungsi objektif sehingga kinerja pengelompokkan dapat dicapai. Tujuan ini dapat dicapai dengan memperkenalkan sekelompok fungsi objektif yang direpresentasikan sebagai agen dalam algoritma *gravitational search algorithm*, dimana dimensi agen ditentukan oleh jumlah klaster. Para agen bergerak melalui ruang pencarian menggunakan aturan algoritma dan proses ini terus berlanjut sampai kriteria konvergensi terpenuhi.

Tahapan dari metode yang diusulkan ditunjukkan pada Gambar 1 dan dijabarkan dalam algoritma sebagai berikut:

1. Menyiapkan dataset kemudian melakukan reduksi data dengan tujuan untuk menghilangkan atribut yang tidak diperlukan dalam proses algoritma.
2. Inisialisasi jumlah klaster, nilai *epoch*, nilai maksimal iterasi, nilai gravitasi konstanta, nilai best, nilai worst, Nilai *mass*, nilai *force*, nilai *acceleration*, dan nilai *velocity*.
3. Inisialisasi derajat keanggotaan data pada pusat klaster.
4. Menghitung fungsi *fitness*. *Fitness* merupakan ukuran kinerja individu agar tetap bertahan hidup dalam lingkungannya. Dalam GSA, fungsi *fitness* adalah fungsi objektif dari masalah yang akan dioptimasi. Fungsi *fitness* yang digunakan dalam penelitian ini mengadopsi rumus fungsi objektif yang terdapat pada FCM untuk meminimumkan jarak antara data dengan titik pusat klaster. Fungsi *fitness* ditunjukkan pada persamaan (1).

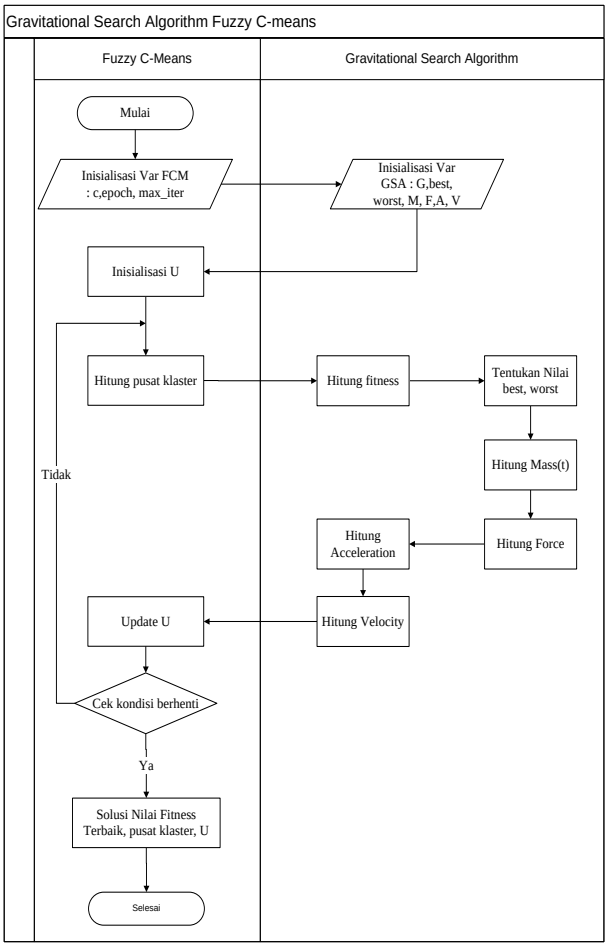
$$J = \sum_{i=1}^{n}\sum_{k=1}^{c}([\sum_{j=1}^{m}(X_{ij} - C_{kj})^2](\mu_{ik})^w) \quad (1)$$

dimana $||X_{ij} - C_{kj}||$ menentukan jarak antara objek X_i dan pusat klaster V_k, n adalah banyaknya item data, c adalah banyaknya klaster yang terbentuk, m adalah dimensi pusat klaster, w adalah nilai pembobotan *fuzzyness*, X adalah item data, V adalah pusat klaster, dan μ adalah derajat keanggotaan data pada pusat klaster. Fungsi jarak yang digunakan dalam penelitian ini adalah *Euclidean distance* yang ditunjukkan pada persamaan (2).

$$\|X_i - C_j\| = \sqrt{\sum_{j=1}^{c}(X_i - C_j)^2} \quad (2)$$

dimana c adalah jumlah klaster, X adalah data, i adalah banyaknya data dan j adalah jumlah klaster. Nilai dari fungsi *fitness* yang dihasilkan kemudian diseleksi untuk menentukan agen terbaik yang memiliki nilai terkecil sebagai calon akhir solusi dengan persamaan (3), dan juga menentukan salah satu agen terburuk dengan persamaan (4). Agen terbaik maupun agen terburuk digunakan untuk menghitung nilai tiap massa.

$$best(t) = min\ \{fitness(t)\}\ ,j \in \{1, 2, ..., S\} \quad (3)$$
$$worst(t) = max\ \{fitness(t)\}\ ,j \in \{1, 2, ..., S\} \quad (4)$$



Gambar 1. Flowchart Algoritma yang Diusulkan

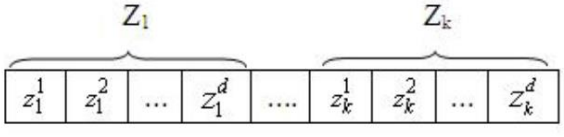
5. Menghitung massa agen menggunakan persamaan (5) dan persamaan (6) berdasarkan fungsi *fitness*, *best* dan *worst*.

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, i = 1,2,..,S \quad (5)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad (6)$$

dimana $M_i(t)$ dan $fit_i(t)$ mewakili nilai massa dan nilai *fitness* pada agen ke-*i* di iterasi ke-*t*, *N* adalah banyaknya data.

6. Menerapkan hukum gravitasi Newton ke dalam algoritma. Hukum gravitasi Newton menjelaskan bahwa setiap partikel di alam semesta menarik setiap partikel lain dengan kekuatan yang berbanding lurus dengan produk dari massa partikel dan berbanding terbalik dengan kuadrat dari jarak antar partikel (Rashedi et al., 2009). Untuk menerapkan hukum gravitasi Newton dalam klaster analisis, digunakan *array* untuk mengkodekan fungsi objektif.



Gambar 2. Contoh Kandidat Solusi Dipetakan Dalam Array Satu Dimensi

Setiap solusi kandidat di dalam populasi terdiri dari *array* satu dimensi seperti ditunjukkan pada Gambar 2 dengan panjang *d* x *k* yang digunakan untuk menampung semua kandidat solusi, dimana *d* adalah dimensi objek data dan *k* adalah jumlah dari kelompok klaster yang diinginkan. Dalam penelitian ini, hukum gravitasi Newton untuk klasterisasi ditunjukkan dengan persamaan (7).

$$F_i^d(t) = \sum_{j \in kbest, j \neq i} \frac{rand_j G(t) \left(M_j(t) M_i(t)\right)}{\left(R_{ij}(t) + \varepsilon\right)\left(x_j^d(t) - x_i^d(t)\right)} \quad (7)$$

dimana $rand_j$ adalah bilangan acak dalam rentang nilai 0 sampai 1, ε adalah nilai kecil untuk menghindari pembagian bilangan nol, R_{ij} adalah jarak *eucliden* antara dua agen *i* dan agen *j*, *kbest* adalah adalah himpunan pertama agen *k* dengan nilai *fitness* terbaik dan massa terbesar, dan *G* adalah nilai konstanta gravitasi yang awalnya diberi nilai 1 dan nilai menurun sampai nilai nol pada iterasi terakhir.

Persamaan (7) digunakan untuk menghitung hasil dari semua gaya yang bekerja pada massa (agen) yang dipilih oleh semua partikel lainnya. Sedangkan persamaan (8) digunakan untuk menghitung percepatan agen.

$$a_i(t) = \frac{F_i(t)}{M_i(t)}, i=1,2,..s \quad (8)$$

dimana F_i dan M_i adalah gaya gravitasi dan massa pada iterasi ke-*i* di dalam iterasi ke-*t*

7. Update kecepatan agen (*velocity*) dengan persamaan (9) dan kemudian memperbarui derajat keanggotaan data dengan pusat klaster (10)

$$v_i^d(t+1) = rand_i \text{ x } v_i^d(t) + a_i^d(t) \quad (9)$$

$$U_i^d(t+1) = U_i^d - v_i^d(t+1) \quad (10)$$

dimana $rand_i$ adalah bilangan acak dalam rentang nilai 0 sampai 1, v_i^d adalah *velocity* ke-*i* pada dimensi ke-*d* dalam iterasi ke-*t*, a_i^d adalah *acceleration* ke-*i* dimensi ke-*d* dalam terasi ke-*t*, *dan* U_i^d adalah derajat keanggotaan data ke-*i* pada dimensi ke-*d*.

3.1 Evaluasi Model yang Diusulkan

Evaluasi tingkat keakuratan klaster terdiri dari dua (Maimon & Rokach, 2010) yaitu evaluasi yang hanya menggunakan nilai-nilai keanggotaan μ_{ij} dari partisi fuzzy data dan yang kedua evaluasi yang melibatkan kedua matriks *U* dan kumpulan data itu sendiri. Validasi algoritma yang menggunakan nilai-nilai keanggotaan μ_{ij} terdiri dari dua yaitu partition coefficient dan classification entropy, sedangkan yang melibatkan kedua matriks U dan kumpulan data yaitu Xie-Beni index. Pada penelitian ini, evaluasi tingkat keakuratan klaster menggunakan partition coefficient, classification entropy dan Xie-Beni Index. Ketiga evaluasi tingkat keakuratan klaster tersebut merupakan evaluasi keakuratan klaster yang paling banyak digunakan (Yunjie, Zhang & Wang, 2008).

3.1.1 Partition Coefficient (PC)

Partition coefficient (PC) mengukur *overlapping* antar klaster. PC diusulkan oleh Bezdeck di dalam (Maimon & Rokach, 2010) dengan persamaan (11)

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}(\mu_{ij})^2 \quad (11)$$

dimana *N* adalah banyaknya data dalam dataset. Semakin dekat nilai Partition coefficient ke 0, berarti *crisp clustering*. Nilai indeks yang mendekati batas atas 1 menunjukkan tidak adanya struktur pengelompokkan dalam kumpulan data atau ketidakmampuan algoritma tersebut.

3.1.2 Classification Entropy (CE)

Classification entropy mengukur *fuzziness* klaster (Azar et al., 2013). CE mirip dengan PC dimana evaluasi hanya menggunakan matriks keanggotaan saja. CE dihitung menggunakan persamaan (12).

$$CE(c) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}.Log(\mu_{ij}) \quad (12)$$

dimana *N* adalah banyaknya data dalam dataset, *c* adalah jumlah klaster. Ketika teknik klasterisasi dievaluasi, semakin dekat indeks PC dengan nilai 1 dan semakin dekat indeks CE dengan nilai 0, maka semakin baik pengelompokkan tersebut (Azar et al., 2013).

3.1.3 Xie-Beni Index (XB)

Xie-Beni Index diusulkan oleh Xie dan Beni di dalam (Maimon & Rokach, 2010). Xie mengukur keseluruhan kekompakan rata-rata pemisahan data antar klaster. Validasi Xie-Beni Index ditunjukkan dengan persamaan (13).

$$V_{xb} = \frac{\sum_{i=1}^{c}\sum_{=1}^{N}\mu_{ij}^2\left|\left|x_j - v_i\right|\right|^2}{N * min_{i\neq j}\left|\left|x_i - v_j\right|\right|^2} \quad (13)$$

dimana *c* adalah jumlah klaster, *N* adalah banyaknya item data pada dataset, μ adalah derajat keanggotaan data pada pusat klaster, *x* adalah item data, *v* adalah pusat klaster.

Dalam persamaan (13), pembilang adalah jumlah dari kekompakan tiap klaster fuzzy dan penyebut adalah pemisahan minimal antara cluster fuzzy. Partisi fuzzy optimal diperoleh dengan meminimalkan nilai XB. Jika nilai XB yang dihasilkan semakin kecil, maka hasil klaster dinilai semakin baik (Azar et al., 2013).

4. HASIL EKSPERIMEN

Dalam eksperimen ini digunakan komputer dengan spesifikasi prosesor Intel B940 2.00 GHz, memori 2 GB dan sistem operasi Windows 7 Ultimate 32 bit serta pemrograman matematika untuk menguji alg. Dataset untuk pengujian metode gravitational search algorithm fuzzy c-means (GSA-FCM) menggunakan enam dataset seperti yang ditunjukkan pada Tabel 1 yang dapat diunduh secara bebas pada laman (http://archive.ics.uci.edu/ml/datasets).

Tabel 1. Dataset untuk Pengujian

Nama Dataset	Jumlah Atribut	Jumlah Sampel Data
Iris	5	150
Wine	13	178
Glass	10	214
Contraceptive Method Choice (CMC)	9	1.473
Seeds	7	210
Perfume Data	2	560

Nilai parameter untuk jumlah klaster diatur pada nilai 3, untuk *epoch* diatur pada nilai 10^{-5}, max_iter pada nilai 100. Dari hasil pengujian menunjukkan bahwa algoritma GSA-FCM mampu keluar dari kondisi *local minimum* yaitu dengan menghasilkan fungsi objektif yang lebih optimal dari algoritma FCM seperti yang ditunjukkan pada Tabel 2.

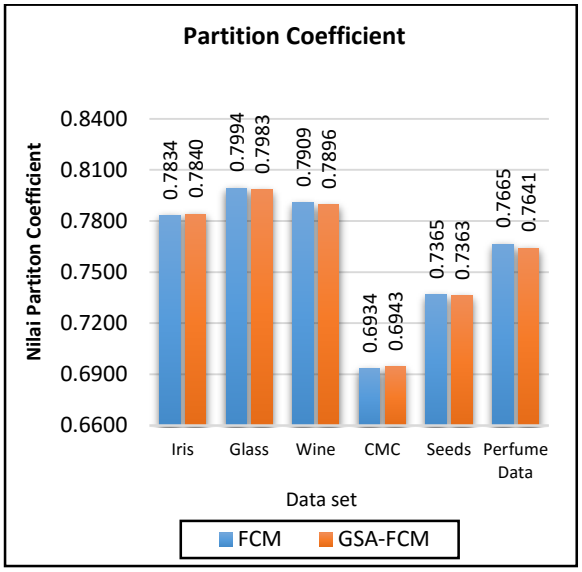
Tabel 2. Komparasi Algoritma FCM dengan GSA-FCM

No	Dataset	FCM		GSA-FCM	
		Iterasi	Fungsi Objektif	Iterasi	Fungsi Objektif
1	Iris	24	6.058,6900	19	**6.058,5599**
2	Wine	40	69.393,0820	55	**69.090,4074**
3	Glass	44	1.796.125,9370	50	**1.792.180,9158**
4	CMC	24	**18.137,3141**	65	18.142,8910
5	Seeds	16	**441,2240**	9	443,8684
6	Perfume Data	35	6.954,6250	23	**6.947,7642**

Tabel 3 dan Gambar 3 menunjukkan validasi hasil klasterisasi untuk dataset iris, glass, wine, CMC, seeds dan perfume data dengan menggunakan pengukuran partition coefficient. Validasi hasil klasterisasi pada data set iris dan CMC menunjukkan nilai yang lebih mendekati pada nilai satu. Hal ini menunjukkan bahwa algoritma yang diusulkan memiliki kinerja yang lebih baik dari algoritma FCM. Sedangkan validasi hasil klasterisasi pada data set glass, wine, seeds dan perfume data menghasilkan nilai yang lebih mendekati nilai nol. Hal ini menunjukkan bahwa kinerja algoritma FCM pada ketiga data set tersebut memiliki kinerja yang lebih baik dari algoritma yang diusulkan.

Tabel 3. Validasi Hasil Klasterisasi dengan Partition Coefficient

Dataset	**Partition Coefficient**	
	FCM	**GSA-FCM**
Iris	0,7834	**0,7840**
Glass	**0,7994**	0,7983
Wine	**0,7909**	0,7896
CMC	0,6934	**0,6943**
Seeds	**0,7365**	0,7363
Perfume Data	**0,7665**	0,7641

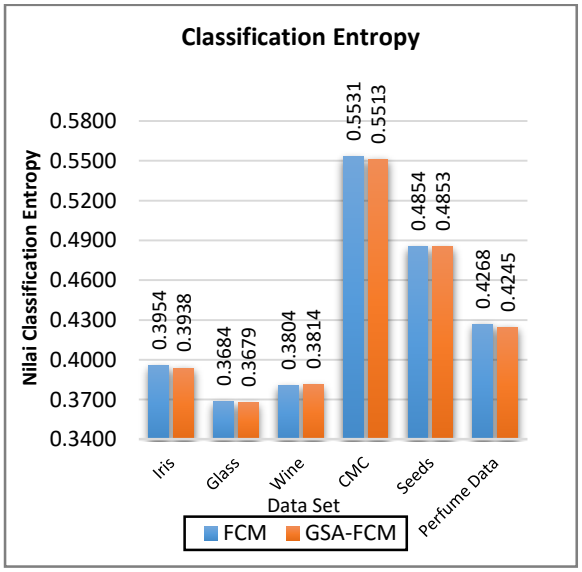


Gambar 3. Pengukuran Validasi Klaster dengan PC

Tabel 4 dan Gambar 4 menunjukkan validasi hasil klasterisasi untuk dataset iris, glass, wine, CMC, seeds, perfume data dengan menggunakan pengukuran classification entropy. Dari enam data set yang diuji, lima dataset yaitu iris, glass, CMC, seeds dan perfume data menghasilkan nilai yang lebih mendekati nol. Hal ini menunjukkan bahwa kinerja algoritma yang diusulkan lebih baik dari algoritma FCM.

Tabel 4. Validasi Hasil Klasterisasi dengan Classification Entropy

Dataset	Classification Entropy	
	FCM	GSA-FCM
Iris	0,3954	**0,3938**
Glass	0,3684	**0,3679**
Wine	**0,3804**	0,3814
CMC	0,5531	**0,5513**
Seeds	0,4854	**0,4853**
Perfume Data	0,4268	**0,4245**

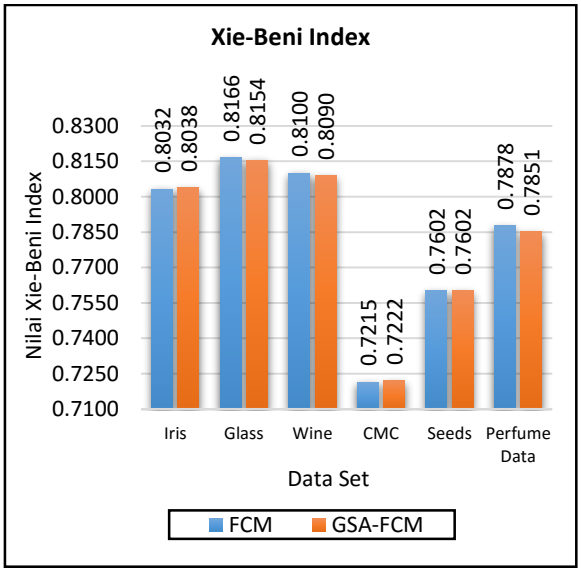


Gambar 4. Pengukuran Validasi Klaster dengan CE

Tabel 5 dan Gambar 5 menunjukkan evaluasi hasil klasterisasi menggunakan Xie-Beni Index. Dari hasil pengujian untuk dataset glass, wine dan perfume data menunjukkan kinerja algoritma yang diusulkan lebih baik dari algoritma FCM. Sedangkan pada data set iris dan CMC menunjukkan kinerja algoritma FCM masih lebih baik dibandingkan algoritma yang diusulkan. Sedangkan pada data set seeds, kinerja kedua algoritma menunjukkan kinerja yang seimbang.

Tabel 5 Validasi Hasil Klasterisasi dengan Xie-Beni Index

Dataset	Xie-Beni Index	
	FCM	GSA-FCM
Iris	**0,8032**	0,8038
Glass	0,8166	**0,8154**
Wine	0,8100	**0,8090**
CMC	**0,7215**	0,7222
Seeds	0,7602	0,7602
Perfume Data	0,7878	**0,7851**



Gambar 5. Pengukuran Validasi Klaster dengan XB

Secara keseluruhan, validasi hasil klasterisasi menggunakan PC, CE dan XB menunjukkan bahwa algoritma yang diusulkan lebih baik dari algoritma FCM.

5. KESIMPULAN

Dari penelitian yang dilakukan, fuzzy c-means dengan optimasi gravitational search algorithm terbukti memiliki kinerja yang lebih baik dari algoritma FCM. Pengujian algoritma dilakukan pada data set iris, wine, glass, CMC, seeds dan perfume data. Empat data set yaitu iris, wine, glass dan perfume data menghasilkan fungsi objektif yang lebih optimal dari algoritma FCM. Hal ini membuktikan bahwa algoritma yang diusulkan dapat keluar dari kondisi *local minimum*.

Evaluasi validitas klaster dengan menggunakan Partition Coeficient (PC), Classification Entropy (CE) dan Xie-Beni Index, juga membuktikan bahwa gravitational search algorithm fuzzy c-mans (GSAFCM) mampu menghasilkan kualitas klaster yang lebih optimal. Hal ini dibuktikan dengan nilai CE dan Xie-Beni Index yang lebih mendekati ke nilai nol. Sedangkan pada nilai PC, hasil klasterisasi dari algoritma yang diusulkan lebih optimal pada dataset iris dan CMC.

Meskipun model yang diusulkan sudah memberikan hasil yang lebih baik, namun untuk penelitian selanjutnya dapat dilakukan:

1. Dalam eksperimen dengan data set yang besar, algoritma GSAFCM mengalami konvergensi terlalu dini, sehingga masih perlu ditambahkan operator baru untuk lebih mengeksplorasi solusi.
2. Penentuan nilai konstan pada parameter *G* dalam algoritma GSA-FCM yang kurang tepat dapat mempengaruhi hasil klasterisasi, sehingga diperlukan suatu metode baru agar penentuan nilai *G* lebih akurat.

REFERENSI

Alata, M., Molhim, M., Ramini, A., & Arabia, S. (2013). Using GA for Optimization of the Fuzzy C-Means Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, *5*(3), 695–701.

Azar, A. T., El-Said, S. A., & Hassanien, A. E. (2013). Fuzzy and hard clustering analysis for thyroid disease. *Computer Methods and Programs in Biomedicine*, *111*(1), 1–16.

Bai, L., Liang, J., & Dang, C. (2011). An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. *Knowledge-Based Systems*, *24*(6), 785–795.

Dawson, C. W. (2009). *Projects in Computing and Information Systems*.

De Carvalho, F. D. a. T., Lechevallier, Y., & de Melo, F. M. (2012). Partitioning hard clustering algorithms based on multiple dissimilarity matrices. *Pattern Recognition*, *45*(1), 447–464.

Dong, H., Dong, Y., Zhou, C., Yin, G., & Hou, W. (2009). A fuzzy clustering algorithm based on evolutionary programming. *Expert Systems With Applications*, *36*(9), 11792–11800.

Hamasuna, Y., Endo, Y., & Miyamoto, S. (2009). On tolerant fuzzy c-means clustering and tolerant possibilistic clustering. *Soft Computing*, *14*(5), 487–494.

Hatamlou, A., Abdullah, S., & Nezamabadi-pour, H. (2011). Application of Gravitational Search Algorithm, 337–346.

Hatamlou, A., Abdullah, S., & Nezamabadi-pour, H. (2012). A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm and Evolutionary Computation*, *6*, 47–52.

Izakian, H., & Abraham, A. (2011). Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. *Expert Systems With Applications*, *38*(3), 1835–1838.

Ji, Z., Sun, Q., & Xia, D. (2011). Computerized Medical Imaging and Graphics A modified possibilistic fuzzy c -means clustering algorithm for bias field estimation and segmentation of brain MR image. *Computerized Medical Imaging and Graphics*, *35*(5), 383–397.

Kanade, P., & Hall, L. (2007). Fuzzy ants and clustering. *Systems, Man and Cybernetics, Part A: …*, *37*(5), 758–769.

Karaboga, D., & Ozturk, C. (2010). Fuzzy clustering with artificial bee colony algorithm. *Scientific Research and Essays*, *5*(14), 1899–1902.

Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Automatic cluster evolution using gravitational search algorithm and its application on image segmentation. *Engineering Applications of Artificial Intelligence*, *29*, 93–103.

Maimon, O., & Rokach, L. (2010). Data Mining and Knowledge Discovery Handbook. *Zhurnal Eksperimental'noi I Teoreticheskoi Fiziki*.

Oliveira, J. V. De, & Pedrycz, W. (2007). *Advances in Fuzzy Clustering and its Applications*. (J. Valente de Oliveira & W. Pedrycz, Eds.). Chichester, UK: John Wiley & Sons, Ltd.

Pimentel, B. a., & de Souza, R. M. C. R. (2013). A multivariate fuzzy c-means method. *Applied Soft Computing*, *13*(4), 1592–1607.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences*, *179*(13), 2232–2248.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2011). Filter modeling using gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, *24*(1), 117–122.

Ross, T. J. (2004). *Fuzzy Logic with Engineering Application*. Wiley.

Sabzekar, M., & Naghibzadeh, M. (2013). Fuzzy c-means improvement using relaxed constraints support vector machines. *Applied Soft Computing*, *13*(2), 881–890. d

Satapathy, S., & Patnaik, S. (2011). Data clustering using modified fuzzy-PSO (MFPSO). *Multi-Disciplinary Trends in Artificial Intelligence*, *7080*, 136–146.

Taherdangkoo, M., & Bagheri, M. H. (2013). A powerful hybrid clustering method based on modified stem cells and Fuzzy C-means algorithms. *Engineering Applications of Artificial Intelligence*, *26*(5-6), 1493–1502.

Wu, K. (2012). Analysis of parameter selections for fuzzy c -means. *Pattern Recognition*, *45*(1), 407–415.

Zhang, Y., Huang, D., Ji, M., & Xie, F. (2011). Image segmentation using PSO and PCM with Mahalanobis distance. *Expert Systems with Applications*, *38*(7), 9036–9040.

Zhang, Y., & Wang, W. (2008). A cluster validity index for fuzzy clustering, *178*, 1205–1218.

Zhao, F., Jiao, L., & Liu, H. (2013). Kernel generalized fuzzy c-means clustering with spatial information for image segmentation. *Digital Signal Processing*, *23*(1), 184–199.

BIOGRAFI PENULIS

Ali Mulyanto. Memperoleh gelar S.Kom dari Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Muhammadiyah Jakarta dan M.Kom dari program pasca sarjana program studi Teknik Informatika STMIK Eresha (d/a STTBI Benarif). Pernah bekerja sebagai ketua Program Studi Manajemen Informatika, kemudian sebagai ketua Program Studi Teknik Informatika di STMIK Cikarang. Saat ini bekerja sebagai dosen dan wakil ketua I bidang akademik di STMIK Cikarang.

Romi Satria Wahono. Mendapatkan gelar B.Eng and M.Eng di bidang ilmu komputer dari Saitama University, Jepang, dan gelar Ph.D di bidang software engineering dari Universiti Teknikal Malaysia Melaka. Saat ini menjadi pengajar dan peneliti pada Program Pascasarjana Ilmu Komputer di Univeristas Dian Nuswantoro. Selain itu juga sebagai founder dan CEO PT Brainmatics Cipta Informatika., sebuah perusahaan pengembang perangkat lunak di Indonesia. Tertarik pada penelitian di bidang software engineering dan machine learning. Professional member dari ACM, PMI dan IEEE Computer Society.

Penerapan Metode Distance Transform Pada Linear Discriminant Analysis Untuk Kemunculan Kulit Pada Deteksi Kulit

Muryan Awaludin
Pascasarjana Teknik Informatika, STMIK Eresha
Email: muryan_awaludin@yahoo.co.id

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

Abstract: Deteksi kulit memainkan peranan penting dalam berbagai aplikasi pengolah citra, mulai dari deteksi wajah, pelacakan wajah, penyaringan konten pornografi, berdasarkan sistem pencarian citra dan berbagai domain interaksi manusia dan komputer. Pendekatan informasi warna dapat mendeteksi warna kulit dengan baik menggunakan *skin probability map* (SPM) dengan aturan bayes. Namun SPM memiliki permasalahan dalam mendeteksi tekstur kulit. *Linear discriminant analysis* (LDA) merupakan algoritma ekstraksi fitur, dalam deteksi kulit digunakan untuk mengekstrak fitur tekstur kulit yang dapat menangani masalah SPM. Namun LDA memiliki permasalahan apabila digunakan untuk mengekstrak fitur tekstur kulit pada kernel yang berbeda. *Distance transform* (DT) merupakan algoritma untuk menghitung jarak citra biner pada setiap pikel gambar dan fitur poin terdekatnya, DT merupakan algoritma yang dapat mengatasi masalah pada LDA. Kombinasi algoritma SPM, LDA dan DT diusulkan untuk memperbaiki performa dari kemunculan kulit pada deteksi kulit. Dataset pada metode yang diusulkan menggunakan IBTD dataset. Hasil dari metode yang diusulkan bahwa metode yang diusulkan menunjukan peningkatan akurasi deteksi kesalahan yang signifikan pada SPM dan LDA.

Keywords: deteksi kulit, skin probability map, linear discriminant analysis, distance transform

1 PENDAHULUAN

Deteksi kulit memainkan peranan penting dalam berbagai aplikasi pengolah citra mulai dari deteksi wajah, pelacakan wajah, penyaringan pornografi, berdasarkan sistem pencarian citra dan berbagai domain interaksi manusia dan komputer (Kakumanu, Makrogiannis, & Bourbakis, 2007) (Lee, Kuo, Chung, & Chen, 2007) (Jie, Xufeng, Yitan, & Zhonglong, 2008). Deteksi kulit pada umumnya mentransformasikan piksel kulit ke ruang warna yang tepat, kemudian mengelompokan piksel kulit tersebut apakah termasuk kulit atau *non*-kulit (Tan, Chan, Yogarajah, & Condell, 2012).

Deteksi kulit manusia pada ruang warna adalah kunci utama dalam tahap proses aplikasi pengolahan citra (Guerrero-Curieses et al., 2009). Sebagian besar penelitian pada deteksi kulit menggunakan model berdasarkan warna kulit diusulkan untuk beragam warna kulit (Amjad, Griffiths, & Patwary, 2012). Beberapa ruang warna seperti RGB, normalisasi RGB, HSV, YcbCr merupakan ruang warna untuk menggambarkan warna kulit (Cheng, Feng, Weng, & Lee, 2012).

Secara umum, deteksi warna kulit mengandalkan pemodelan statistik kulit berdasarkan aturan dari pendekatan deteksi kulit (Kawulok, Kawulok, & Smolka, 2011), melalui sebuah survey yang membandingkan berbagai pendekatan deteksi kulit berdasarkan warna yang disajikan (Kakumanu et al., 2007).

SPM adalah metode deteksi kulit yang banyak digunakan, tetapi SPM mempunyai kelemahan sulit mendeteksi warna terutama gambar yang menyerupai tekstur kulit manusia (Jiang, Yao, & Jiang, 2007). Algoritma *linear discriminant analysis* telah sukses digunakan banyak aplikasi seperti pengenalan karakter tulisan tangan, pengenalan wajah, pengambilan gambar dan sebagainya, namun untuk masalah *multi-class* terbukti tidak optimal (Yao, Lu, Li, Xu, & Han, 2014**).** LDA baik digunakan untuk menentukan sebuah kelas vector yang diperkirakan pada ruang fitur, namun LDA mempunyai masalah jika digunakan untuk mengukur sampel yang lebih besar (Lu, Zou, & Wang, 2012).

Algoritma distance transform (DT) merupakan algoritma yang bagus untuk berbagai aplikasi seperti pengolahan citra, computer vision, pengenalan pola, analisis bentuk dan geometri komputasi (Arcelli, di Baja, & Serino, 2011), Distance transform (DT) dari citra biner akan menghitung jarak diantara setiap piksel citra dan fitur titik terdekatnya (J. Wang & Yagi, 2013).

Pada penelitian ini, kami mengembangkan skema penggabungan dari analisa spasial dengan keunggulan pemodelan kulit adaptif DT pada pencahayaan dan probabilitas kulit untuk mengatasi batas daerah kulit dan *non*-kulit yang kabur dengan ukuran kernel yang berbeda. Dan untuk mencapai ketepatan batas wilayah dari kulit dan *non*-kulit serta tepatnya informasi warna piksel menggunakan algoritma *linear discriminant analysis* (LDA) dan *skin probability map* (SPM). Dari integrasi metode diatas diharapkan dapat digunakan pada deteksi kulit dan mengurangi *false positive* yang menghasilkan tingkat akurasi yang tinggi untuk meminimalkan jarak antara kulit dan *non*-kulit dari ukuran kernel yang berbeda.

Pada penelitian ini akan disusun sebagai berikut. Pada bagian 2 akan dijelaskan tetang penelitian terkait. Bagian 3 metode yang diusulkan. Membandingkan hasil eksperimen metode yang diusulkan dengan metode lain dibahas pada bagian 4. Terakhir, penelitian yang kami lakukan ini diringkas dalam bagian terakhir.

2 PENELITIAN TERKAIT

SPM digunakan untuk menyaring warna dengan rendahnya penerimaan nilai ambang batas piksel warna yang diterapkan dalam ruang warna RGB. Kemudian, fitur tekstur diekstrak menggunakan *Gabor wavelets* dari sebuah citra warna masukan dikonversi ke *grayscale* (Jiang et al., 2007).

Respon yang diperoleh tergantung pada nilai ambang batas piksel warna, yang menghasilkan nilai piksel biner.

Tujuan menerapkan *Gabor wavelets* adalah untuk mengurangi *false positive rate* (FPR) berdasarkan penyaringan daerah kulit dan *non*-kulit yaitu dengan nilai piksel yang besar pada fitur tekstur (Yahya, Tan, & Hu, 2013). Tekstur yang tidak mirip kulit, tidak diklasifikasikan sebagai kulit pada penyaringan piksel warna kulit. Akhirnya, daerah kulit yang tumbuh menggunakan segmentasi *watershed* dengan penanda wilayah didefinisikan dengan baik untuk memanfaatkan informasi warna kulit.

Hasil yang ditunjukan bahwa metode tersebut dapat mengurangi *false positive rate* (dari 20,1% menjadi 4,2%) dan meningkatkan *true positive rate* (dari 92,7% menjadi 94,8%) yang dilakukan dengan penyaringan warna untuk dataset yang berisi 600 gambar (Bouzerdoum, 2003). Namun, dari keterangan diatas tidak memberikan nilai ambang batas piksel warna yang berbeda untuk diterapkan pada setiap gambar.

Algoritma *Artificial Neural Network* (ANN) pada deteksi kulit digunakan untuk mengestimasi kepadatan *non*-parametrik kelas kulit dan *non*-kulit (Taqa & Jalab, 2010). Umumnya, analisa tekstur terhadap citra masukan membantu mengurangi jumlah kesalahan klasifikasi piksel pada deteksi piksel warna kulit. Namun, daerah kekasaran kulit dan *non*-kulit dapat bervariasi antara gambar, sehingga penerapan dari algoritma segmentasi berdasarkan tekstur sulit untuk generalisasi dataset yang nyata.

Analisa SPM untuk segmentasi kulit dikendalikan oleh difusi (Ruiz-del-Solar & Verschae, 2004). Kelemahan dari metode ini adalah performa dalam kasus batas daerah kulit dan *non*-kulit yang kabur, karena proses difusi tidak berhenti jika transisi antara piksel kulit dan *non*-kulit halus.

LDA pada deteksi kulit dimanfaatkan untuk informasi tekstur kulit pada setiap citra masukan yang terdeteksi diekstrak pada fitur tekstur kulit yang paling diskriminatif (Kawulok et al., 2011). Setelah itu, seluruh gambar diproyeksikan keruang *discriminative textural features* (DTF). Pada percobaan penelitian tersebut menegaskan bahwa pentingnya menggunakan informasi tekstur dan menunjukan bahwa metode tersebut secara signifikan meningkatkan hasil pewarnaan yang diperoleh, meskipun domain DFT bagus untuk propagasi warna, namun ketepatannya terbatas pada batas daerah kulit dan *non*-kulit karena ukuran yang berbeda.

Pada penelitian yang dilakukan (Michal Kawulok, Jolanta Kawuloky, Jakub Nalepa, 2013) memperkenalkan pengembangan skema penggabungan dari analisa spasial dengan keunggulan pemodelan kulit adaptif menggunakan *distance transform* (DT) dan memecahkan masalah LDA yaitu terbatas pada batas wilayah karena ukuran kernel besar.

Dari penelitian yang berkaitan diatas masalah akurasi pada warna dan fitur tekstur sangat penting untuk meningkatkan sistem deteksi kulit dimana disebabkan oleh pengaruh pencahayaan, *background*, dan *real life* dataset. Oleh karena itu pada penelitian ini, kami menggabungan dari beberapa algoritma diatas, sistem kemunculan kulit akan diusulkan dimana didalamnya terdapat kombinasi tiga algoritma yaitu SPM yang digunakan untuk informasi warna piksel kulit. LDA digunakan untuk informasi fitur tekstur kulit dan DT yang digunakan untuk mengatasi masalah batas wilayah kulit dengan ukuran kernel yang lebih besar pada deteksi kulit.

3 METODE YANG DIUSULKAN

Kami mengusulkan sebuah metode yang disebut SPMLDA+DT, singkatnya untuk informasi piksel warna kulit dan informasi fitur tekstur kulit dengan SPM dan LDA kemudian untuk digunakan untuk kernel yang berbeda pada deteksi kulit menggunakan DT, untuk mencapai kinerja yang lebih baik pada deteksi kulit. Gambar 1 merupakan aktifitas diagram metode yang disusukan SPMLDA+DT.

SPM diperoleh menggunakan pemodelan kulit bayes. Menurut (Clair L. Alston, 2013) inti dari *bayes rule* (aturan bayes) adalah bagaimana caranya untuk mendapatkan nilai probabilitas hipotesis C_s benar jika diberikan evidence *v*, untuk mengetahui $P(C_s|v)$. Bahwa nilai probabilitas yang diberikan piksel termasuk kelas kulit dihitung dengan menggunakan aturan bayes:

$$P(Cs|v) = \frac{P(v|\mathrm{Cs})P(\mathrm{Cs})}{P(v|Cs)P(Cs) + P(v|Cns)P(Cns)}$$

Dimana v adalah piksel warna, probabilitas apriori $P(C_s)$ adalah probabilitas kulit dan $P(C_{ns})$ adalah probabilitas *non*-kulit dapat diperkirakan berdasarkan jumlah piksel di kedua kelas, tetapi sangat sering diasumsikan bahwa warna kulit dan non-kulit adalah sama $P(C_s) = P(C_{ns}) = 0.5$ (Kawulok, Kawulok, & Nalepa, 2013).

Tujuan menggunakan SPM yaitu untuk mendapatkan informasi warna piksel kulit menggunakan ruang warna YCbCr. Ruang YCbCr dipilih karena alasan berikut (Powar, 2011):

1. Gambar bitmap menggunakan ruang warna RGB sebagai warna gambar. Namun penelitian medis membuktikan bahwa mata manusia memiliki sensitivitas yang berbeda untuk warna dan kecerahan. Sehingga menggunakan transformasi RGB ke YCbCr.
2. Komponen pencahayaan (Y) dari YCbCr merupakan warna independen, sehingga dapat diadopsi untuk memecahkan masalah variasi pencahayaan dan mudah untuk digunakan.
3. Menurut (Hsu, Member, & Abdel-mottaleb, 2002) pengelompokan warna kulit lebih bagus menggunakan ruang warna YCbCr daripada ruang warna lain.
4. YCbCr memiliki adanya tumpang tindih paling sedikit antara kulit dan data non-kulit di bawah berbagai kondisi pencahayaan. YCbCr secara luas digunakan dalam standar kompresi video (misalnya, MPEG dan JPEG).
5. YCbCr adalah salah satu dari dua ruang warna utama yang digunakan untuk mewakili komponen video digital.
6. Perbedaan antara YCbCr dan RGB adalah bahwa YCbCr merupakan warna kecerahan dan dua sinyal warna yang berbeda, sedangkan RGB merupakan warna seperti merah, hijau dan biru.

LDA merupakan teknik yang efektif dan banyak digunakan untuk pengurangan dimensi dan ekstraksi fitur tekstur (Kim, Stenger, Kittler, & Cipolla, 2010), selain itu juga untuk menemukan sedikitnya subruang baru yang memberikan pemisahan terbaik antara kelas yang berbeda dalam input data (Fekry, Elsadek, Ali, & Ziedan, 2011).

Untuk menemukan subruang didefinisikan oleh arah yang paling diskriminatif dalam pelatihan himpunan vektor M-dimensi diklasifikasikan ke dalam kelas K. Analisis ini dilakukan, pertama dengan menghitung dua matriks kovarians: intra-class scatter matrix:

$$\mathrm{Sw} = \sum_{i=1}^{K} \sum_{u_k \in K_i} (u_k - \mu_i)(u_k - \mu_i)^T$$

Dan *inter-class scatter matrix:*

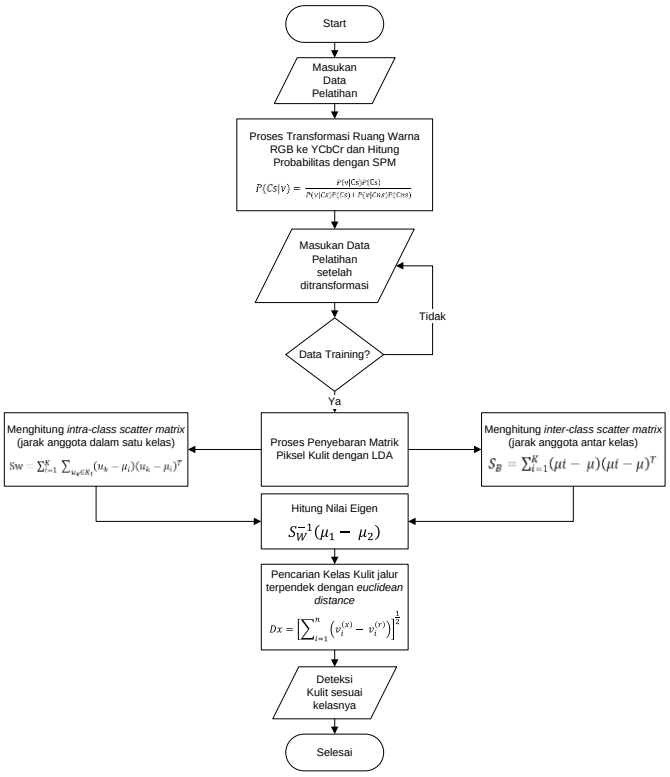
$$S_B = \sum_{i=1}^{K} (\mu i - \mu)(\mu i - \mu)^T$$

Dimana K adalah jumlah kelas, μ adalah vektor rata-rata data pelatihan dan μ_i adalah vektor rata-rata kelas i (disebut K_i), i adalah kelas, u_k adalah gambar ke-k dan T adalah vektor dimensi.

Tujuan menggunakan DT untuk penghitung jarak citra biner antara setiap pikel gambar dan fitur poin terdekatnya (J. Wang & Yagi, 2013). Peta kemungkinan kemunculan kulit diperoleh berdasarkan *euclidean distance (D)* (Lagerstrom & Buckley, 2012) atau jarak kesamaan vector kulit dan *non*-kulit dihitung dalam ruang kemunculan kulit pada setiap piksel x dari referensi piksel r, euclidean distance dapat didefinisikan sebagai:

$$Dx = \left[\sum_{i=1}^{n} \left(v_i^{(x)} - v_i^{(r)}\right)\right]^{\frac{1}{2}}$$

Dimana $v_i^{(x)}$ adalah dimensi i dari vektor jarak tranform kemunculan kulit diperoleh untuk pixel x, i adalah dimensi, n adalah jumlah dimensi, x adalah bobot vektor kelas pertama, r adalah bobot vektor kelas kedua. Referensi piksel ditentukan sebagai piksel dari nilai probabilitas maksimal dalam SPM mengalami pengurangan dengan menggunakan besarnya kernel. Semakin kecil skor $D(x,r)$ maka semakin mirip kedua vector fitur yang dicocokkan. Sebaliknya semakin besar skor $D(x,r)$, maka akan semakin berbeda pada kedua vector fitur.



Gambar 1. Diagram Aktifitas dari Metode SPMLDA+DT

4 HASIL EKSPERIMEN

Percobaan dilakukan dengan menggunakan platform komputer berbasis Intel Core i3 2,3 GHz CPU, 4 GB RAM, dan Microsoft Windows 7 Professional 64-bit dengan sistem operasi SP1. Software yang digunakan untuk pengembangan pada penelitian kami adalah MATLAB R2009a.

Data set yang digunakan adalah dataset IBTD. Dataset ini dapat diperoleh melalui situs http://lbmedia.ece.ucsb.edu/resources/dataset/ibtd.zip. Seperti pada Gambar 2, data yang berupa gambar ini memiliki ekstensi *.JPEG.

Gambar 2. Contoh Dataset IBTD yang Digunakan dalam Penelitian

Salah satu metode untuk menentukan nilai atribut menggunakan informasi matrik, yaitu dengan menggunakan standar deviasi untuk menentukan keputusan yang obyektif (Y.-M. Wang & Luo, 2010). Standar deviasi dan rata-rata deviasi banyak diusulkan untuk menentukan bobot vektor yang optimal secara objektif dengan asumsi bobot atribut sudah diketahui (Xu & Da, 2010). Oleh karena itu, pada penelitian ini untuk mengukur tingkat kesalahan deteksi kulit salah satunya menggunakan standar deviasi untuk mengetahui keragaman suatu kelompok data kulit dan *non*-kulit.

Hasil eksperimen dilakukan dengan berbagai macam kondisi pencahayaan, latar belakang, perbedaan etnis dan perbedaan kernel dengan menggunakan dataset IBTD. Dalam pengujian SPM menggunkan semua dataset untuk mendapatkan nilai minimal (proyeksi kelas pertama) dan maksimal (proyeksi kelas kedua) agar mendapatkan nilai standar deviasi seperti pada Tabel 1.

Tabel 1. Hasil Eksperimen Model SPM

Nilai Kemunculan Kulit dengan SPM						
Training Data	**Proyeksi Kelas Pertama**		**Std. Deviasi Kelas Pertama**	**Proyeksi Kelas Kedua**		**Std. Deviasi Kelas Kedua**
	Kelas Kulit	**Kelas non-Kulit**		**Kelas Kulit**	**Kelas non-Kulit**	
111	113.432	74.396	**27.603**	187.793	145.405	**29.972**
222	112.923	77.640	**24.949**	184.333	146.635	**26.657**
333	107.357	77.372	**21.203**	185.694	151.450	**24.214**
444	108.011	78.500	**20.868**	185.885	150.896	**24.741**
555	107.649	78.198	**20.825**	186.085	150.905	**24.876**

Hasil ekperimen pada Tabel 1 yang terdapat 555 data dari dataset IBTD, merupakan nilai rata-rata antara proyeksi kelas pertama dan kelas kedua mempunyai jarak antar rerata yaitu standar deviasi lebih besar kelas kedua dari kelas pertama.

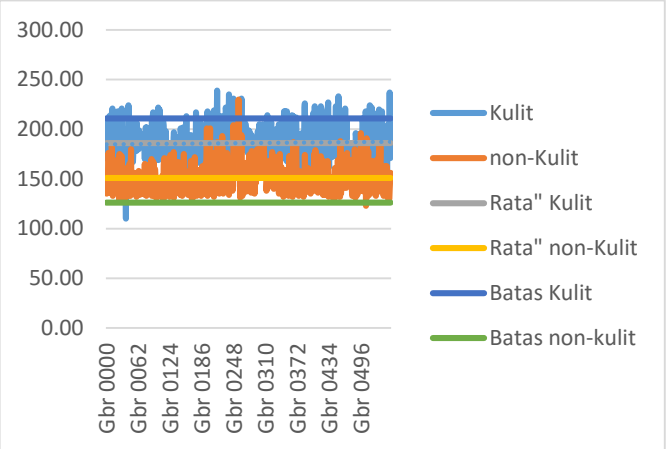
Pada Tabel 2 merupakan hasil deteksi *error* menggunakan metode SPM.

Tabel 2. Hasil Deteksi Error untuk Deteksi Kulit dengan SPM

Jumlah Dataset	DR	FNR	FPR
555	92,97%	7,03%	39%

Dari perhitungan seluruh dataset yang digunakan, nilai 92,97% merupakan bagian dari piksel kulit yang benar diklasifikasikan sebagai kulit dihitung dengan (*Recall)* η_{tp}= *TP/(FN+TP).* Nilai 7,03% merupakan bagian pengelompokan piksel kulit sebagai background dihitung dengan δ_{fn}= *FN/(FN+TP)* dan nilai 39% merupakan bagian pengelompokan piksel background sebagai kulit dihitung dengan δ_{fp}= *FP/(FP+TN).*

Pada Gambar 3 menunjukan grafik standar deviasi penyebaran warna kulit dan *non*-kulit pada deteksi kulit dengan SPM.



Gambar 3. Standar Deviasi Penyebaran Warna Kulit dan *non*-Kulit dengan SPM

Hasil ekperimen terdapat 555 data dari database IBTD, merupakan nilai jarak piksel kulit dan non-kulit dari proyeksi kelas pertama dan kelas kedua hasil komparasi antara SPM dan DT. Nilai standar deviasi untuk menentukan bobot vektor yang optimal secara objektif dengan asumsi bobot atribut sudah diketahui (Xu & Da, 2010), seperti yang ditunjukan pada Tabel 3.

Tabel 3. Hasil Eksperimen Model SPM+DT

Nilai Piksel Kemunculan Kulit dan *non*-Kulit dengan SPM+DT			
Training Data	**Proyeksi Kemunculan Kulit**		**Std. Deviasi**
	Kelas Kulit	**Kelas *non*-Kulit**	
111	46.10	70.96	17.58
222	43.25	69.26	18.39
333	43.04	74.02	21.91
444	42.98	72.61	20.95
555	40.74	72.88	22.73

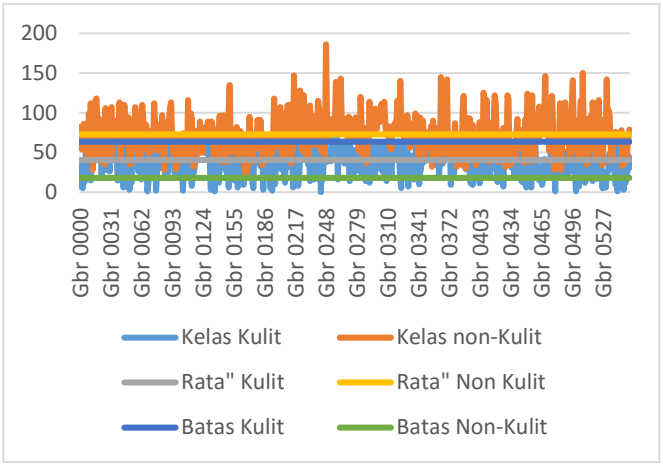
Hasil ekperimen pada Tabel 3 yang terdapat 555 data dari dataset IBTD, merupakan nilai jarak piksel kulit dan non-kulit dari proyeksi kelas pertama dan kelas kedua hasil komparasi antara SPM dan DT. Hasil deteksi error komparasi antara SPM+ DT seperti ditunjukan pada Tabel 4.

Tabel 4. Tabel Hasil Deteksi Error Model SPM+DT

Jumlah Dataset	DR	FNR	FPR
555	94%	6,13%	34%

Dari perhitungan seluruh dataset yang digunakan, nilai 94% merupakan bagian dari piksel kulit yang benar diklasifikasikan sebagai kulit dihitung dengan (*Recall)* η_{tp}= *TP/(FN+TP).* Nilai 6,13% merupakan bagian pengelompokan piksel kulit sebagai background dihitung dengan δ_{fn}= *FN/(FN+TP)* dan nilai 34% merupakan bagian pengelompokan piksel background sebagai kulit dihitung dengan δ_{fp}= *FP/(FP+TN).*

Gambar 4 menunjukan grafik standar deviasi penyebaran warna kulit dan *non*-kulit pada deteksi kulit dengan SPM+ DT.



Gambar 4. Standar Deviasi Penyebaran Warna Kulit dan non-Kulit dengan SPM+DT

Hasil eksperimen SPM+LDA pertama dilakukan dengan SPM untuk mencari nilai probabilitas dari kulit dan *non*-kulit menggunakan ruang warna Cb dan Cr (Powar, 2011) (Aibinu, Shafie, & Salami, 2012) (Sanchez-Cuevas, Aguilar-Ponce, & Tecpanecatl-Xihuitl, 2013) (Zaidan et al., 2014).

Kemudian dengan menggunakan metode LDA untuk matriks kovarian antar kelas (between-class covariance matrix) (Yao et al., 2014), sekaligus meminimumkan matriks kovarian dalam kelas (within-class covariance matrix), agar anggota di dalam kelas lebih terkumpul penyebarannya dan pada akhirnya dapat meningkatkan keberhasilan pengenalan.

Dalam pengujian SPM+LDA menggunakan semua dataset untuk mendapatkan nilai minimal (proyeksi kelas pertama) dan maksimal (proyeksi kelas kedua) agar mendapatkan nilai standar deviasi seperti pada Tabel 5.

Tabel 5. Hasil Eksperimen dengan SPM+LDA

Nilai Kemunculan Kulit dengan SPM+LDA						
Training Data	**Proyeksi Kelas Pertama**		**Std. Deviasi**	**Proyeksi Kelas Kedua**		**Std. Deviasi**
	Kelas Kulit	**Kelas non-Kulit**		**Kelas Kulit**	**Kelas non-Kulit**	
111	262.571	313.511	**36.020**	194.246	231.931	**26.647**
222	258.846	308.613	**35.190**	197.664	236.732	**27.626**
333	256.425	305.571	**34.752**	202.242	241.658	**27.871**
444	257.064	306.260	**34.787**	202.460	241.648	**27.710**
555	257.020	305.981	**34.621**	202.263	241.972	**28.078**

Hasil ekperimen pada Tabel yang terdapat 555 data dari dataset IBTD, merupakan nilai rata-rata antara proyeksi kelas

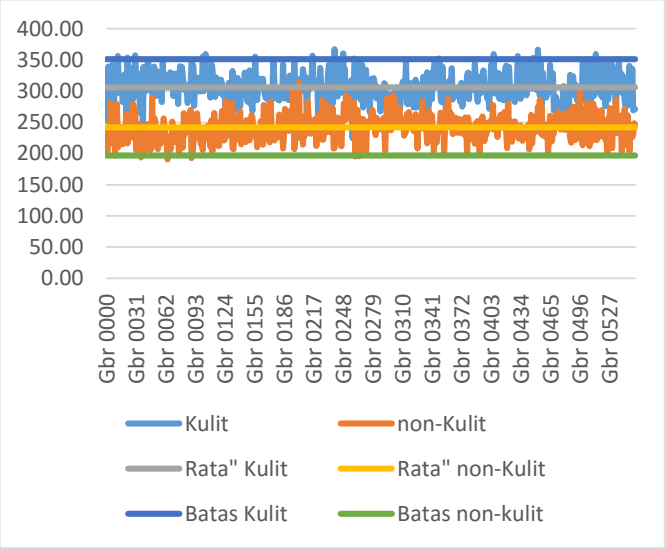
pertama dan kelas kedua mempunyai jarak antar rerata yaitu standar deviasi lebih besar kelas pertama dari kelas kedua. Pada Tabel 6. merupakan hasil deteksi error komparasi antara SPM dan LDA.

Tabel 6. Hasil Deteksi Error Model SPM+LDA

Jumlah Dataset	DR	FNR	FPR
555	96,76%	3,24%	18%

Dari perhitungan seluruh dataset yang digunakan, nilai 96,76% merupakan bagian dari piksel kulit yang benar diklasifikasikan sebagai kulit dihitung dengan (*Recall)* η_{tp}= *TP/(FN+TP).* Nilai 3,24% merupakan bagian pengelompokan piksel kulit sebagai background dihitung dengan δ_{fn}= *FN/(FN+TP)* dan nilai 18% merupakan bagian pengelompokan piksel background sebagai kulit dihitung dengan δ_{fp}= *FP/(FP+TN).*

Gambar 5. menunjukan grafik standar deviasi penyebaran warna kulit dan *non*-kulit pada deteksi kulit dengan SPM dan LDA.



Gambar 5. Standar Deviasi Penyebaran Warna Kulit dan *non*-Kulit Model SPM+LDA

Hasil ekperimen terdapat 555 data dari database IBTD, merupakan nilai jarak piksel kulit dan non-kulit dari proyeksi kelas pertama dan kelas kedua hasil komparasi antara SPM dan LDA, yang diproyeksikan kemunculan nilai piksel kulit dan non-kulit kedalam DT. Seperti yang ditunjukan pada Tabel 7.

Tabel 7. Hasil Eksperimen Model SPMLDA+DT

Nilai Piksel Kemunculan Kulit dan *non*-Kulit Model SPMLDA+DT			
Training Data	Proyeksi Kemunculan Kulit		Std. Deviasi
	Kelas Kulit	Kelas non-Kulit	
111	50.97	37.64	9.43
222	49.96	38.49	8.11
333	49.51	39	7.23
444	49.51	39.28	7.24
555	49.39	39.28	7.15

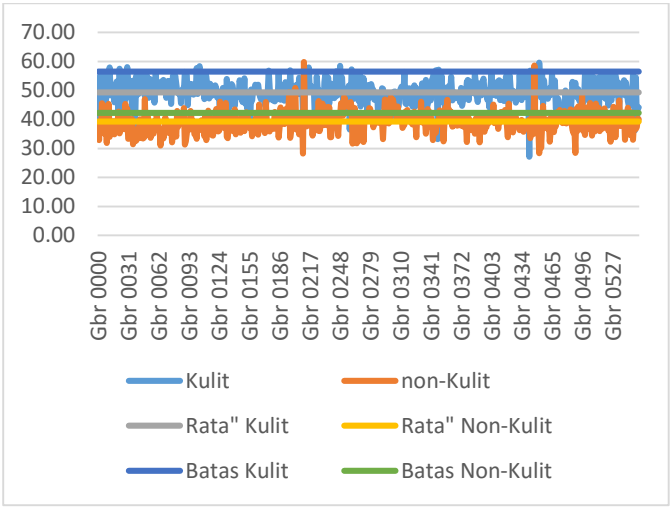
Dari hasil eksperimen dengan 555 data dari dataset IBTD, pada hasil yang ditunjukan Tabel 7. Merupakan proyeksi kemunculan kulit model SPMLDA+DT. Pada Tabel 8. merupakan hasil deteksi error model SPMLDA+DT.

Tabel 8. Tabel Hasil Deteksi Error Model SPMLDA+DT

Jumlah Dataset	DT	FNR	FPR
555	97,12%	2,88%	16%

Dari perhitungan seluruh dataset yang digunakan, nilai 97,12% merupakan bagian dari piksel kulit yang benar diklasifikasikan sebagai kulit dihitung dengan (*Recall)* η_{tp}= *TP/(FN+TP).* Nilai 2,88% merupakan bagian pengelompokan piksel kulit sebagai background dihitung dengan δ_{fn}= *FN/(FN+TP)* dan nilai 16% merupakan bagian pengelompokan piksel background sebagai kulit dihitung dengan δ_{fp}= *FP/(FP+TN)*

Gambar 6. menunjukan grafik standar deviasi penyebaran warna kulit dan *non*-kulit pada deteksi kulit model SPMLDA+DT.



Gambar 6. Standar Deviasi Penyebaran Warna Kulit Model SPMLDA+DT

Pada Tabel 9. menunjukan perbandingan *false positive rate* (FPR) yaitu bagian pengelompokan piksel background sebagai kulit dan *false negative rate* (FNR) yaitu bagian pengelompokan piksel kulit sebagai background (Jones & Rehg, 2002) dengan menggunakan dataset IBTD.

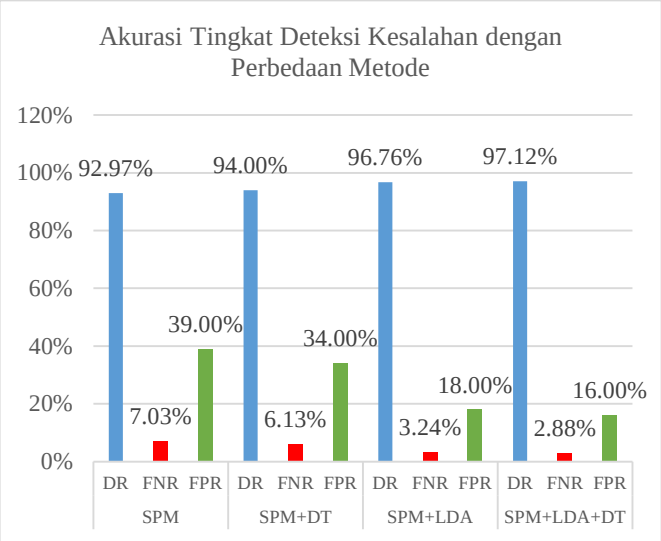
Tabel 9. Perbandingan FNR dan FPR Menggunakan Dataset IBTD

Metode	DR	FNR	FPR
SPM (Jiang et al., 2007)	92,97%	7,03%	39%
SPM+DT (Michal Kawulok, Jolanta Kawuloky, Jakub Nalepa, 2013)	94%	6,13%	34%
Metode yang diusulkan	97,12%	2,88%	16%

Hasil eksperimen dengan menerapkan metode *LDA pada SPM* didapatkan nilai FNR adalah 3,24% dan hasil ini menunjukan bahwa metode SPM+LDA lebih baik daripada hanya menggunakan *SPM* yang memiliki nilai FNR sebesar 7,03% dan *SPM+DT* yang menghasilkan nilai FNR sebesar 6,13%.

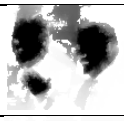
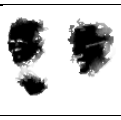
Hasil metode *SPMLDA+DT* yang diusulkan didapatkan nilai FNR sebesar 2,88% dan hasil ini menunjukan bahwa metode yang diusulkan lebih baik daripada menggunakan metode *SPM* (FNR=7,03%), *SPM+DT* (FNR=6,13%), maupun *SPM+LDA* (FNR=3,24%).

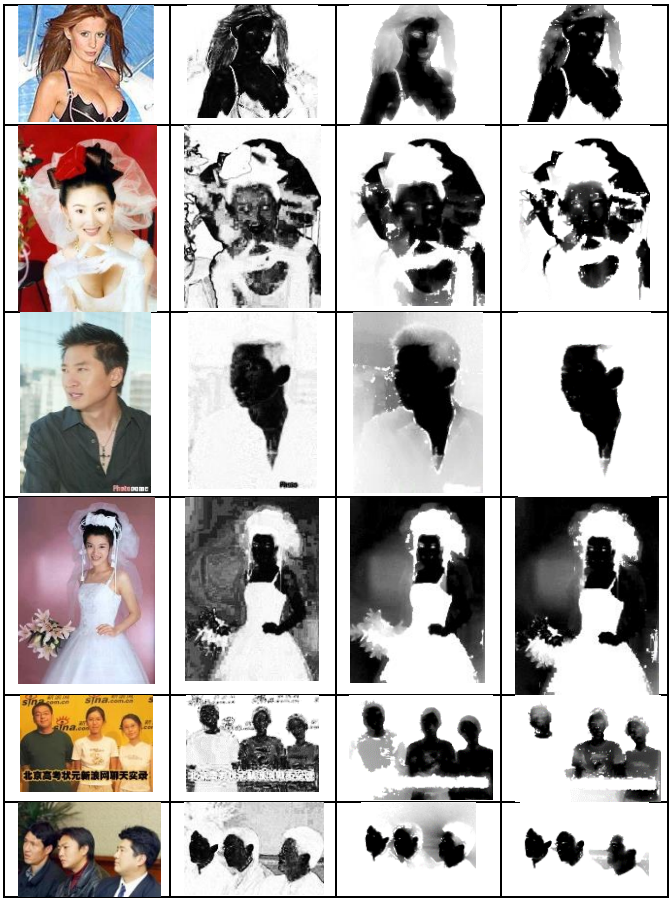
Hasil FNR dan FPR yang ditunjukan pada tabel 9. menunjukan bahwa metode yang diusulkan lebih baik dari metode yang lain seperti yang ditunjukan pada Gambar 7.



Gambar 7. Diagram Persentase dari Perbedaan Metode

Hasil pengujian aplikasi deteksi kulit dengan perbedaan metode ditunjukan pada Gambar 8, dengan model terbaik SPM LDA+DT.

Gambar Asli	Metode SPM	Metode SPM+DT	Metode yang diusulkan
			
			
			
			
			



Gambar 8. Aplikasi Deteksi Kulit dengan Perbedaan Metode

5 KESIMPULAN

Teknik kombinasi algoritma skin probability map, linear discriminant analysis dan distance transform disulkan untuk memperbaiki kinerja dari deteksi kulit. Skin probability map digunakan untuk mencari informasi warna kulit, linear disriminant anlaysis digunakan untuk informasi fitur tekstur kulit, sedangkan distance transfrom untuk mencari jarak terpendek antara kulit dan non-kulit sekaligus mengatasi jika digunakan pada kernel yang berbeda. Dataset yang digunakan disulkan menggunakan IBTD dataset. Hasil eksperimen menunjukan bahwa metode yang diusulkan menghasilkan tingkat *error* deteksi yang kecil yaitu sebesar 2,88%. Oleh karena itu, kami berkesimpulan bahwa metode yang diusulkan memberikan perbaikan kinerja pada skin probability map dan linear discriminant analysis.

REFERENSI

Aibinu, a. M., Shafie, A. a., & Salami, M. J. E. (2012). Performance Analysis of ANN based YCbCr Skin Detection Algorithm. *Procedia Engineering*, *41*(Iris), 1183–1189.

Amjad, a., Griffiths, A., & Patwary, M. N. (2012). Multiple face detection algorithm using colour skin modelling. *IET Image Processing*, *6*(8), 1093–1101.

Arcelli, C., di Baja, G. S., & Serino, L. (2011). Distance-driven skeletonization in voxel images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(4), 709–20.

Bouzerdoum, S. lam P. D. C. A. (2003). Adaptive Skin Segmentation In Color Images. *Proceedings of IEEE ICASSP*, 353–356.

Cheng, Y., Feng, Z., Weng, F., & Lee, C. (2012). Enhancing Model-based Skin Color Detection : from Low-level rgb Features to High-level Discriminative Binary-class Features School of Ece. *ICASSP IEEE*, 1401–1404.

Clair L. Alston, K. L. M. and A. N. P. (2013). *Case studies in bayesian statistics*. (A. L.Alston, Clair; L.Mengersen, Kerrie; N.Pettitt, Ed.). Wikey.

Fekry, S., Elsadek, A., Ali, H. F., & Ziedan, I. E. (2011). High Precision Face Detection and Recognition based on Fusion of Discernment Techniques. *ICGST International Journal on Graphics Vision and Image Processing (gvip)*, *11*(2), 31–39.

Guerrero-Curieses, a, Rojo-Álvarez, J. L., Conde-Pardo, P., Landesa-Vázquez, I., Ramos-López, J., & Alba-Castro, J. L. (2009). On the Performance of Kernel Methods for Skin Color Segmentation. *EURASIP Journal on Advances in Signal Processing*, *2009*(1), 856039.

Hsu, R., Member, S., & Abdel-mottaleb, M. (2002). Face Detection in Color Images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(5), 1–23.

Jiang, Z., Yao, M., & Jiang, W. (2007). Skin Detection Using Color, Texture and Space Information. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, 366–370.

Jie, Y., Xufeng, L., Yitan, Z., & Zhonglong, Z. (2008). A face detection and recognition system in color image series. *Mathematics and Computers in Simulation*, *77*(5-6), 531–539.

Kakumanu, P., Makrogiannis, S., & Bourbakis, N. (2007). A survey of skin-color modeling and detection methods. *Pattern Recognition*, *40*(3), 1106–1122.

Kawulok, M., Kawulok, J., & Nalepa, J. (2013). Spatial-based skin detection using discriminative skin-presence features. *Pattern Recognition Letters*.

Kawulok, M., Kawulok, J., & Smolka, B. (2011). Image colorization using discriminative textural features. *IAPR Conference on Machine Vision And Aoolications*, 198–201.

Kim, T.-K., Stenger, B., Kittler, J., & Cipolla, R. (2010). Incremental Linear Discriminant Analysis Using Sufficient Spanning Sets and Its Applications. *International Journal of Computer Vision*, *91*(2), 216–232.

Lagerstrom, R., & Buckley, M. (2012). An attribute weighted distance transform. *Pattern Recognition Letters*, *33*(16), 2141–2147.

Lee, J.-S., Kuo, Y.-M., Chung, P.-C., & Chen, E.-L. (2007). Naked image detection based on adaptive and extensible skin color model. *Pattern Recognition*, *40*(8), 2261–2270.

Lu, G.-F., Zou, J., & Wang, Y. (2012). Incremental complete LDA for face recognition. *Pattern Recognition*, *45*(7), 2510–2521.

Michal Kawulok, Jolanta Kawuloky, Jakub Nalepa, M. P. (2013). Skin Detection Using Spatial Analysis With Adaptive Seed. *ICIP IEEE*, *978-1-4799*, 3720–3724.

Powar, V. (2011). Skin Detection in YCbCr Color Space. *International Journal of Computer Applications in Technology*, 1–4.

Ruiz-del-Solar, J., & Verschae, R. (2004). Skin detection using neighborhood information. *Automatic Face and Gesture …*.

Sanchez-Cuevas, M. C., Aguilar-Ponce, R. M., & Tecpanecatl-Xihuitl, J. L. (2013). A Comparison of Color Models for Color Face Segmentation. *Procedia Technology*, *7*(444), 134–141.

Tan, W. R., Chan, C. S., Yogarajah, P., & Condell, J. (2012). Human Skin Detection. *IEEE Transaction on Industrial Informatics*, *8*(1), 138–147.

Taqa, A. Y., & Jalab, H. A. (2010). Increasing the reliability of skin detectors. *Academic Journals*, *5*(17), 2480–2490.

Wang, J., & Yagi, Y. (2013). Shape priors extraction and application for geodesic distance transforms in images and videos. *Pattern Recognition Letters*, *34*(12), 1386–1393. doi:10.1016/j.patrec.2013.04.008

Wang, Y.-M., & Luo, Y. (2010). Integration of correlations with standard deviations for determining attribute weights in multiple attribute decision making. *Mathematical and Computer Modelling*, *51*(1-2), 1–12.

Xu, Y.-J., & Da, Q.-L. (2010). Standard and mean deviation methods for linguistic group decision making and their applications. *Expert Systems with Applications*, *37*(8), 5905–5912.

Yahya, A. A., Tan, J., & Hu, M. (2013). A Novel Model of Image Segmentation Based onWatershed Algorithm. *Advances in Multimedia*, *2013*.

Yao, C., Lu, Z., Li, J., Xu, Y., & Han, J. (2014). A subset method for improving Linear Discriminant Analysis. *Neurocomputing*, *138*, 310–315.

Zaidan, a. a., Ahmad, N. N., Abdul Karim, H., Larbani, M., Zaidan, B. B., & Sali, A. (2014). Image skin segmentation based on multi-agent learning Bayesian and neural network. *Engineering Applications of Artificial Intelligence*, *32*, 136–150.

BIOGRAFI PENULIS

Muryan Awaludin. Memperoleh gelar S.Kom dari Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI) Jakarta dan M.Kom dari program pasca sarjana program studi Teknik Informatika STMIK Eresha (d/a STTBI Benarif). Saat ini bekerja sebagai dosen di STIKOM CKI Jakarta. Minat penelitiannya saat ini meliputi pengolahan citra dan computer vision.

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.

Komparasi Algoritma Klasifikasi Machine Learning Dan *Feature Selection* pada Analisis Sentimen Review Film

Vinita Chandani
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Email: vinita.chandani@gmail.com

Romi Satria Wahono, Purwanto
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Email: romi@brainmatics.com, purwanto@dsn.dinus.ac.id

Abstract: Analisis sentimen adalah proses yang bertujuan untuk menentukan isi dari dataset yang berbentuk teks bersifat positif, negatif atau netral. Saat ini, pendapat khalayak umum menjadi sumber yang penting dalam pengambilan keputusan seseorang akan suatu produk. Algoritma klasifikasi seperti Naïve Bayes (NB), Support Vector Machine (SVM), dan Artificial Neural Network (ANN) diusulkan oleh banyak peneliti untuk digunakan pada analisis sentimen review film. Namun, klasifikasi sentimen teks mempunyai masalah pada banyaknya atribut yang digunakan pada sebuah dataset. *Feature selection* dapat digunakan untuk mengurangi atribut yang kurang relevan pada dataset. Beberapa algoritma *feature selection* yang digunakan adalah information gain, chi square, forward selection dan backward elimination. Hasil komparasi algoritma, SVM mendapatkan hasil yang terbaik dengan *accuracy* 81.10% dan AUC 0.904. Hasil dari komparasi *feature selection*, information gain mendapatkan hasil yang paling baik dengan *average accuracy* 84.57% dan *average* AUC 0.899. Hasil integrasi algoritma klasifikasi terbaik dan algoritma *feature selection* terbaik menghasilkan *accuracy* 81.50% dan AUC 0.929. Hasil ini mengalami kenaikan jika dibandingkan hasil eksperimen yang menggunakan SVM tanpa *feature selection*. Hasil dari pengujian algoritma *feature selection* terbaik untuk setiap algoritma klasifikasi adalah information gain mendapatkan hasil terbaik untuk digunakan pada algoritma NB, SVM dan ANN.

Keywords: analisis sentimen, klasifikasi, feature selection, support vector machine, artificial neural network, naïve bayes, information gain, chi square, forward selection, backward eliminations

1 PENDAHULUAN

Analisis sentimen adalah proses yang bertujuan untuk memenentukan isi dari dataset yang berbentuk teks (dokumen, kalimat, paragraf, dll) bersifat positif, negatif atau netral (Kontopoulos, Berberidis, Dergiades, & Bassiliades, 2013). Analisis sentimen merupakan bidang penelitian yang cukup popular, karena dapat memberikan keuntungan untuk berbagai aspek, mulai dari prediksi penjualan (Yang Liu, Huang, An, & Yu, 2007), politik (Park, Ko, Kim, Liu, & Song, 2011), dan pengambilan keputusan para investor (Dergiades, 2012).

Saat ini, pendapat khalayak umum telah menjadi salah satu sumber yang begitu penting dalam berbagai produk di jejaring sosial (C.-L. Liu, Hsaio, Lee, Lu, & Jou, 2012). Demikian juga dalam industri film (Tsou & Ma, 2011). Popularitas internet mendorong orang untuk mencari pendapat pengguna dari internet sebelum membeli produk atau melihat situs film (C.-L. Liu et al., 2012). Pendapat orang-orang dapat mengurangi ketidakpastian terhadap suatu produk tertentu dan membantu konsumen menyimpulkan kualitas suatu produk tertentu (Koh, Hu, & Clemons, 2010).

Banyak situs yang menyediakan review tentang suatu produk yang dapat mencerminkan pendapat pengguna (C.-L. Liu et al., 2012). Salah satu contohnya adalah situs *Internet Movie Database* (IMDb). IMDb adalah situs yang berhubungan dengan film dan produksi film. Informasi yang diberikan IMDb sangat lengkap. Siapa saja aktor/aktris yang main di film itu, sinopsis singkat dari film, link untuk trailer film, tanggal rilis untuk beberapa negara dan review dari user-user yang lain. Ketika seseorang ingin membeli atau menonton suatu film, komentar-komentar orang lain dan peringkat film biasanya mempengaruhi perilaku pembelian mereka.

Algoritma klasifikasi sentimen seperti naïve bayes (NB) (Kang, Yoo, & Han, 2012), artificial neural network (ANN) (Moraes, Valiati, & Gavião Neto, 2013) (Zhu, Xu, & Wang, 2010), support vector machine (SVM) (Moraes et al., 2013) (S Tan & Zhang, 2008) diusulkan oleh banyak peneliti (Koncz & Paralic, 2011) untuk analisis sentimen *review restaurant* (Kang et al., 2012), dokumen (Moraes et al., 2013) (S Tan & Zhang, 2008), dan teks (Zhu et al., 2010). ANN mempunyai kelebihan dalam hal kemampuan untuk generalisasi, yang bergantung pada seberapa baik ANN meminimalkan resiko empiris namun ANN mempunyai kelemahan dimana menggunakan data pelatihan cukup besar (Vapnik, 1999). SVM mempunyai kelebihan yaitu bisa diterapkan untuk data yang berdimensi tinggi, tetapi SVM sulit untuk digunakan untuk data dengan jumlah yang besar (Nugroho, Witarto, & Handoko, 2003). NB mempunyai kelebihan mudah diimplementasikan, performance NB lebih baik. Pengklasifikasian pada NB didasarkan pada probabilitas bersyarat dari fitur salah satu kelas setelah fitur seleksi menggunakan algoritma yang ada (W. Zhang & Gao, 2011).

Beberapa peneliti telah melakukan komparasi menggunakan beberapa algoritma pada beberapa dataset. Penelitian yang dilakukan oleh B. Pang et al (Pang, Lee, Rd, & Jose, 2002) membandingkan algoritma NB, maximum entropy dan SVM. Didapatkan hasil yang terbaik adalah SVM. Rodrigo Moraes et al (Moraes et al., 2013) membandingkan antara ANN, SVM dan NB. Didapatkan hasil yang terbaik adalah ANN. Ziqiong Zhang et al (Z. Zhang, Ye, Zhang, & Li, 2011) membandingkan antara SVM dan NB dan NB merupakan hasil yang terbaik. Songbo Tan et al (S Tan & Zhang, 2008) membandingkan NB, centroid classifier, k-nearest neighbor (KNN), winnow classifier dan SVM merupakan hasil yang terbaik. Dataset yang digunakan dalam penelitian di atas berbeda-beda. Penelitian yang dilakukan oleh B. Pang et all (Pang & Lee, 2002) menggunakan dataset review film. Rodrigo Moraes et al (Moraes et al., 2013) menggunakan

dataset review film, *Global Positioning System* (GPS), buku dan kamera. Ziqiong Zhang (Z. Zhang et al., 2011) et al menggunakan dataset *review restaurant*, dan Songbo Tan (Songbo Tan & Wang, 2011) et al menggunakan dataset dokumen berbahasa cina.

Salah satu masalah pada klasifikasi sentimen teks adalah banyaknya atribut yang digunakan pada sebuah dataset (Wang, Li, Song, Wei, & Li, 2011). Pada umumnya, atribut dari klasifikasi sentimen teks sangat besar, dan jika semua atribut tersebut digunakan, maka akan mengurangi kinerja dari *classifier* (Wang, Li, Zhao, & Zhang, 2013). Atribut yang banyak membuat *accuracy* menjadi rendah. Untuk mendapatkan *accuracy* yang lebih baik, atribut yang ada harus dipilih dengan algoritma yang tepat (Xu, Peng, & Cheng, 2012).

Feature selection merupakan bagian penting untuk mengoptimalkan kinerja dari *classifier* (Wang et al., 2011). *Feature selection* dapat didasarkan pada pengurangan ruang fitur yang besar, misalnya dengan mengeliminasi atribut yang kurang relevan (Koncz & Paralic, 2011). Penggunaan algoritma *feature selection* yang tepat dapat meningkatkan *accuracy* (Xu et al., 2012) (Forman, 2000). Algoritma *feature selection* dapat dibedakan menjadi dua tipe, yaitu *filter* dan *wrapper* (Yuanning Liu et al., 2011). Contoh dari tipe *filter* adalah *information gain* (IG), *chi-square*, dan *log likelihood ratio*. Contoh dari tipe *wrapper* adalah *forward selection* dan *backward elimination* (Vercellis, 2009). Hasil *precision* dari tipe *wrapper* lebih tinggi daripada tipe *filter*, tetapi hasil ini tercapai dengan tingkat kompleksitas yang besar. Masalah kompleksitas yang tinggi juga dapat menimbulkan masalah (Koncz & Paralic, 2011).

Yang dan Perdersen (Yang & Pedersen, 1997) membandingkan lima algoritma *feature selection* pada klasifikasi dokumen. Lima algoritma tersebut adalah *document frequency*, IG, *chi-square*, *term strength* dan *mutual information*. Hasil penelitian mereka menunjukkan bahwa IG dan *chi-square* paling efisien. Forman (Forman, 2000) membandingkan 12 algoritma *feature selection* pada 229 klasifikasi teks menjadi dua kategori. Hasil penelitian menunjukkan IG dan *chi-square* mendapatkan hasil yang lebih baik dibandingkan metode *Bi-Normal Separation* yang diusulkan peneliti. Tan dan Zang (S Tan & Zhang, 2008) menggunakan algoritma *feature selection* untuk analisis sentimen dokumen berbahasa Cina. Hasil yang didapat IG mendapatkan yang paling baik.

Dari semua hasil penelitian yang sudah dilakukan belum ditemukan model yang paling tepat untuk analisis sentimen. Maka dari itu penulis akan melakukan komparasi terhadap beberapa algoritma klasifikasi (NB, SVM dan ANN), komparasi terhadap beberapa algoritma *feature selection* (IG, *chi-square*, *forward selection*, *backward elimination*) dan melakukan integrasi dari hasil komparasi algoritma klasifikasi dan algoritma *feature selection* yang terbaik pada dataset review film.

2 PENELITIAN TERKAIT

Salah satu masalah pada klasifikasi sentiment teks adalah data yang berdimensi tinggi sehingga menyebabkan banyaknya atribut yang kurang relevan. Jika semua atribut tersebut digunakan, maka akan mengurangi kinerja dari sebuah *classifier* (Wang et al., 2013). Atribut yang banyak membuat *accuracy* menjadi rendah. Untuk mendapatkan *accuracy* yang lebih baik, atribut yang ada harus dipilih dengan algoritma yang tepat (Xu et al., 2012). *Feature selection* merupakan bagian penting untuk mengoptimalkan kinerja dari *classifier* (Wang et al., 2011). *Feature selection* dapat digunakan untuk mengeliminasi atribut yang kurang relevan (Koncz & Paralic, 2011).

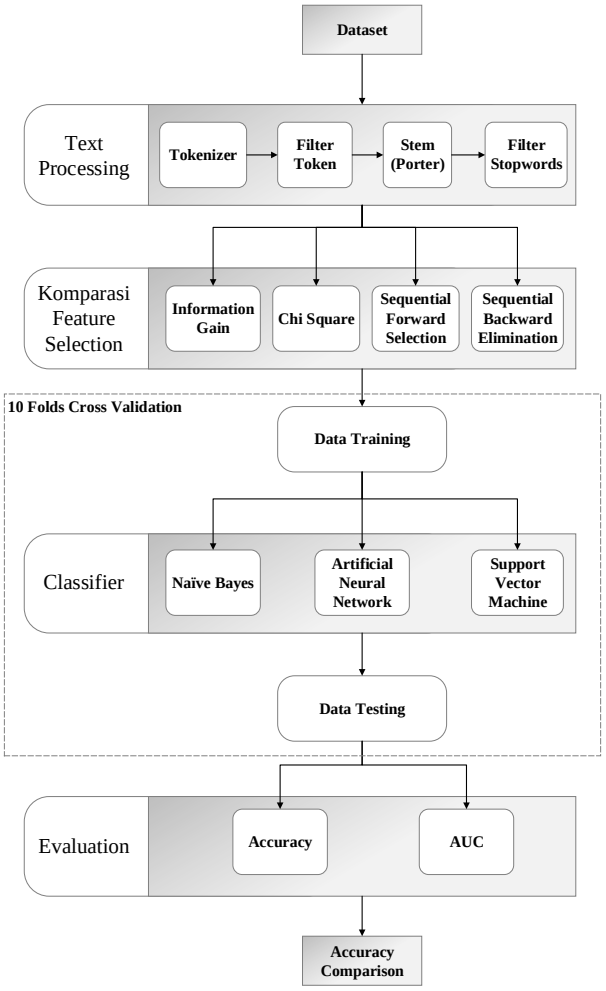
Beberapa peneliti telah mengkomparasi beberapa algoritma klasifikasi dan algoritma *feature selection* untuk mendapatkan hasil yang terbaik. Penelitian yang dilakukan oleh Peter Koncz dan Jan Paralic (Koncz & Paralic, 2011) menggunakan SVM untuk algoritma klasifikasinya dan algoritma *feature selection* n-grams+ *document frequency* dibandingkan dengan Information Gain (IG). Hasil yang diadapatkan IG lebih baik daripada algoritma yang diusulkan. Rodrigo Moraes, Joao Francisco Valiati, Wilson P (Moraes et al., 2013) mengkomparasi algoritma klasifikasi SVM, Naïve Bayes (NB) dan Artificial Neural Network (ANN). *Feature selection* yang digunakan adalah *expert knowledge*, *minimum frequency*, IG, *chi-square*. Hasil yang tebaik untuk algoritma klasifikasi adalah ANN dan untuk *feature selection* terbaik adalah IG. Zhu Jian, Xu Chen dan Wang Han Shi (Zhu et al., 2010) mengkomparasi algoritma klasifikasi *individual model (i-model)* berbasis ANN dibandingkan dengan hidden markov model dan SVM. *Feature selection* yang digunakan adalah *odd ratio*. Hasil algoritma klasifikasi yang terbaik adalah *i-model based on ANN*. Songbo Tan dan Jin Zhang (S Tan & Zhang, 2008) mengkomparasi lima algoritma klasifikasi (*centroid classifier*, *K-nearest neighbor*, *winnow classifier*, NB dan SVM), empat algoritma *feature selection* (Mutual Information, IG, *chi-square* dan *Document Frequency*). Hasil eksperimen menunjukan bahwa IG mendapatkan hasil yang terbaik untuk *feature selection* dan algoritma SVM mendapatkan hasil yang terbaik untuk klasifikasi sentimen.

3 METODE YANG DIUSULKAN

Peneliti mengusulkan untuk mengkomparasi tiga algoritma klasfiikasi (SMV, NB dan ANN) dan mengkomparasi empat algoritma *feature selection* (IG, Chi Square, Forward Selection dan Backward Elimination). Gambar 1 memnunjukan komparasi algoritma klasifikasi dan *feature selection* yang diusulkan.

Sebelum dilakukan komparasi, dataset dilakukan *text processing* terlebih dahulu. *Text processing* bertujuan untuk mempersiapkan dokumen teks yang tidak terstruktur menjadi data terstruktur yang siap digunakan untuk proses selanjutnya. Tahapan *text processing* meliputi:

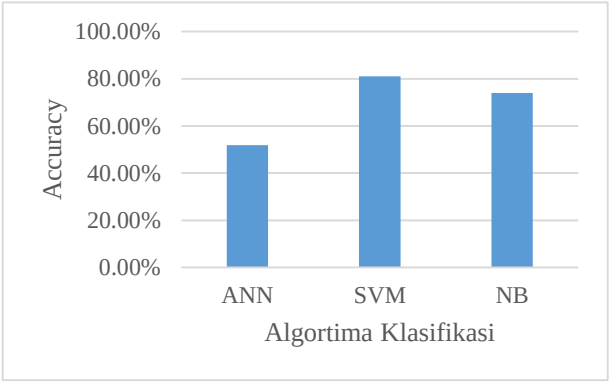
1. Tokenize merupakan proses untuk memisah-misahkan kata. Potongan kata tersebut disebut dengan token atau *term* (Manning, Raghavan, & Schutze, n.d.).
2. Filter Token merupakan proses mengambil kata-kata penting dari hasil token (Langgeni, Baizal, & W, 2010).
3. Stem yaitu proses pengubahan bentuk kata menjadi kata dasar. Metode pengubahan bentuk kata menjadi kata dasar ini menyesuaikan struktur bahasa yang digunakan dalam proses stemming (Langgeni et al., 2010).
4. Filter stopwords adalah proses menghilangkan kata-kata yang sering muncul namun tidak memiliki pengaruh apapun dalam ekstraksi sentimen suatu review. Kata yang termasuk seperti kata penunjuk waktu, kata tanya (Langgeni et al., 2010).



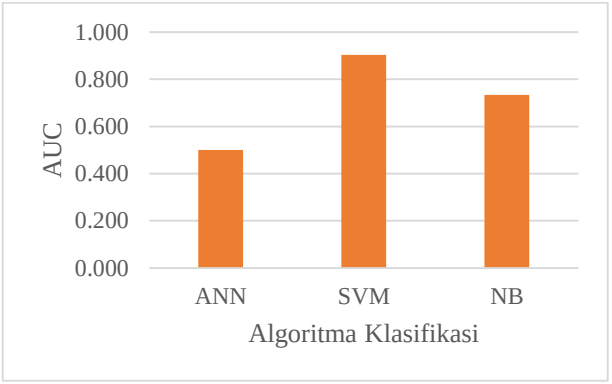
Gambar 1. Komparasi Algoritma Klasifikasi dan *Feature Selection*

Tabel 1. Komparasi Accuracy dan AUC Algoritma Klasifikasi

	Accuracy	AUC
ANN	51.80%	0.500
SVM	81.10%	0.904
NB	74.00%	0.734



Gambar 2. Komparasi accuracy algoritma klasifikasi



Gambar 3. Komparasi AUC algoritma klasifikasi

4 HASIL PENELITIAN

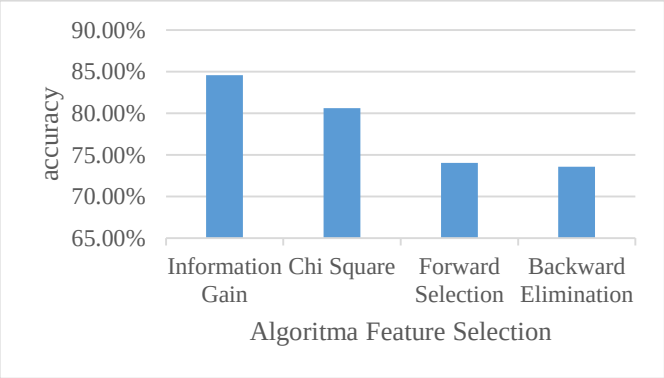
Penelitian yang dilakukan menggunakan komputer dengan spesifikasi CPU Intel Core i5 1.6GHz, RAM 8GB, dan sistem operasi Microsoft Windows 7 Professional 64-bit. Apliasi yang digunakan adalah RapidMiner 5.2. Data penelitian ini menggunakan *Data Movie Review Polarity Dataset V2.0* (Pang & Lee, 2002) yang diperoleh dari data movie review yang digunakan oleh Pang and Lee. Data ini dapat diambil di situs http://www.cs.cornell.edu/people/pabo/movie-review-data/. Data ini diambil dari situs IMDb. Data yang digunakan dalam penelitian terdiri dari 1000 review film, berisi 500 review positif dan 500 review negatif.

Tabel 5 merupakan rangkuman hasil komparasi algoritma klasifikasi. Berdasarkan Tabel 1, Gambar 2 dan Gambar 3 didapat hasil terbaik adalah SVM dengan *accuracy* = 81.10% dan AUC = 0.904. Hal ini mengkonfirmasi pada penelitian yang dilakukan oleh Songbo Tan (S Tan & Zhang, 2008) dalam mengkomparasi algoritma klasifikasi, dan SVM mendapatkan nilai yang paling baik. Klasifikasi pada analisis sentimen sangat tergantung pada data yang diuji. Untuk pengujian data IMDB review film, SVM merupakan algoritma yang paling baik.

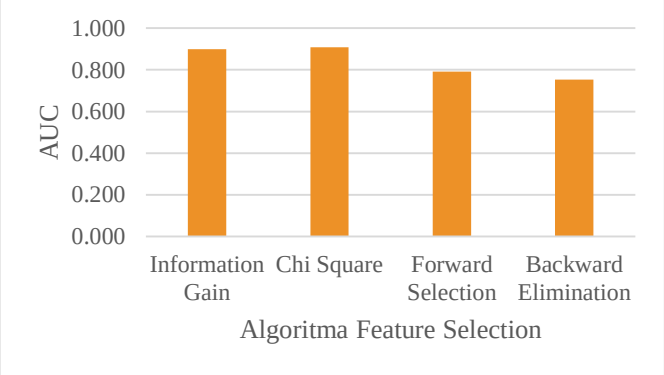
SVM menghasilkan nilai *accuracy* dan AUC terbaik dibanding ANN dan NB. Hal ini mengkonfirmasi pada penelitian yang dilakukan oleh Songbo Tan (S Tan & Zhang, 2008) dalam mengkomparasi algoritma klasifikasi, dan SVM mendapatkan nilai yang paling baik. Klasifikasi pada analisis sentimen sangat tergantung pada data yang diuji. Untuk pengujian data IMDB review film, SVM merupakan algoritma yang paling baik.

Tabel 2. Komparasi Accuracy dan AUC Algoritma *Feature Selection*

	Information Gain		**Chi Square**		**Forward Selection**		**Backward Elimination**	
	Top K (K=200)		**Top K (K=100)**					
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
ANN	91.40%	0.914	79.60%	0.900	75.50%	0.781	70.20%	0.724
SVM	81.50%	0.929	80.80%	0.853	67.67%	0.698	79.25%	0.844
NB	80.80%	0.853	80.30%	0.867	79.00%	0.807	71.25%	0.689
AVERAGE	84.57%	0.899	80.23%	0.873	74.06%	0.762	73.57%	0.752



Gambar 4. Grafik Komparasi Accuracy Algoritma Feature Selection



Gambar 5. Grafik Komparasi AUC Algoritma Feature Selection

Tabel 2 merupakan tabel komparasi *feature selection* terbaik. Data dari Tabel 2 diambil berdasarkan *average* (rata-rata) dari masing-masing parameter algoritma *feature selection.* Dari hasil *average* tersebut, diambil nilai *average* yang paling baik, dan kemudian dirangkumkan seperti pada Tabel 2. Berdasarkan Tabel 2 didapatkan hasil algoritma *feature selection* terbaik adalah information gain. Hal ini mengkonfirmasi pada penelitian yang dilakukan oleh Peter Koncz (Koncz & Paralic, 2011), Rodrigo Moraes (Moraes et al., 2013), dan Songbo Tan (S Tan & Zhang, 2008) yang juga menghasilkan information gain sebagai algoritma *feature selection* yang terbaik.

5 KESIMPULAN

Hasil dari komparasi algoritma klasifikasi antara Support Vector Machine (SVM), Naïve Bayes (NB) dan Artificial Neural Network (ANN) didapatkan SVM dengan hasil terbaik dengan nilai *accuracy* = 81.10% dan nilai AUC = 0.904. Hasil dari komparasi algoritma *feature selection* antara information gain, chi square, forward selection, backward elimination didapatkan information gain pada parameter top k dengan nilai k = 200 sebagai hasil terbaik, dengan nilai *accuracy average* adalah 84.57% dan nilai AUC = 0.899.

REFERENCES

Dergiades, T. (2012). Do investors' sentiment dynamics affect stock returns? Evidence from the US economy. *Economics Letters*, *116*(3), 404–407. doi:10.1016/j.econlet.2012.04.018

Forman, G. (2000). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, *3*, 1289–1305. doi:10.1162/153244303322753670

Kang, H., Yoo, S. J., & Han, D. (2012). Senti lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, *39*(5), 6000–6010. doi:10.1016/j.eswa.2011.11.107

Koh, N. S., Hu, N., & Clemons, E. K. (2010). Do online reviews reflect a product's true perceived quality? An investigation of online movie reviews across cultures. *Electronic Commerce Research and Applications*, *9*(5), 374–385. doi:10.1016/j.elerap.2010.04.001

Koncz, P., & Paralic, J. (2011). An approach to feature selection for sentiment analysis. In *2011 15th IEEE International Conference on Intelligent Engineering Systems* (pp. 357–362). IEEE. doi:10.1109/INES.2011.5954773

Kontopoulos, E., Berberidis, C., Dergiades, T., & Bassiliades, N. (2013). Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, *40*(10), 4065–4074. doi:10.1016/j.eswa.2013.01.001

Langgeni, D. P., Baizal, Z. K. A., & W, Y. F. A. (2010). Clustering Artikel Berita Berbahasa Indonesia, *2010*(semnasIF), 1–10.

Liu, C.-L., Hsaio, W.-H., Lee, C.-H., Lu, G.-C., & Jou, E. (2012). Movie Rating and Review Summarization in Mobile Environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(3), 397–407. doi:10.1109/TSMCC.2011.2136334

Liu, Y., Huang, X., An, A., & Yu, X. (2007). ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07* (p. 607). New York, New York, USA: ACM Press. doi:10.1145/1277741.1277845

Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An Improved Particle Swarm Optimization for Feature Selection. *Journal of Bionic Engineering*, *8*(2), 191–200. doi:10.1016/S1672-6529(11)60020-6

Manning, C. D., Raghavan, P., & Schutze, H. (n.d.). Introduction to Information Retrieval.

Moraes, R., Valiati, J. F., & Gavião Neto, W. P. (2013). Document Level Sentiment Classification: an Empirical Comparison

between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621–633. doi:10.1016/j.eswa.2012.07.059

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika. *IlmuKomputer.Com*.

Pang, B., & Lee, L. (2002). A Sentimental Education : Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *Association for Computational Linguistics*.

Pang, B., Lee, L., Rd, H., & Jose, S. (2002). Thumbs up ? Sentiment Classification using Machine Learning Techniques. *Association for Computational Linguistics*, *10*(July), 79–86.

Park, S., Ko, M., Kim, J., Liu, Y., & Song, J. (2011). The Politics of Comments : Predicting Political Orientation of News Stories with Commenters ' Sentiment Patterns.

Tan, S., & Wang, Y. (2011). Weighted SCL model for adaptation of sentiment classification. *Expert Systems with Applications*, *38*(8), 10524–10531. doi:10.1016/j.eswa.2011.02.106

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications*, *34*(4), 2622–2629. doi:10.1016/j.eswa.2007.05.028

Tsou, B. K., & Ma, M. (2011). Aspect Based Opinion Polling from Customer Reviews. *IEEE Transactions on Affective Computing*, *2*(1), 37–49. doi:10.1109/T-AFFC.2011.2

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, *10*(5), 988–99. doi:10.1109/72.788640

Vercellis, C. (2009). *Business Intelligence: Data Mining and Optomization for Decision Making*. John Wiley and Sons.

Wang, S., Li, D., Song, X., Wei, Y., & Li, H. (2011). A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. *Expert Systems with Applications*, *38*(7), 8696–8702. doi:10.1016/j.eswa.2011.01.077

Wang, S., Li, D., Zhao, L., & Zhang, J. (2013). Sample cutting method for imbalanced text sentiment classification based on BRC. *Knowledge-Based Systems*, *37*, 451–461. doi:10.1016/j.knosys.2012.09.003

Xu, T., Peng, Q., & Cheng, Y. (2012). Identifying the semantic orientation of terms using S-HAL for sentiment analysis. *Knowledge-Based Systems*, *35*, 279–289. doi:10.1016/j.knosys.2012.04.011

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning*, *20*(15), 412–420.

Zhang, W., & Gao, F. (2011). An Improvement to Naive Bayes for Text Classification. *Advanced in Control Engineeringand Information Science*, *15*, 2160–2164. doi:10.1016/j.proeng.2011.08.404

Zhang, Z., Ye, Q., Zhang, Z., & Li, Y. (2011). Sentiment classification of Internet restaurant reviews written in Cantonese. *Expert Systems with Applications*, *38*(6), 7674–7682. doi:10.1016/j.eswa.2010.12.147

Zhu, J., Xu, C., & Wang, H. (2010). Sentiment classification using the theory of ANNs. *The Journal of China Universities of Posts and Telecommunications*, *17*(July), 58–62. doi:10.1016/S1005-8885(09)60606-3

BIOGRAFI PENULIS

Vinita Chandani. Lahir pada tanggal 11 November 1990 di Tegal, Jawa Tengah. Memperoleh gelar Sarjana Komputer (S.Kom) dari fakultas Teknik Informatika, Universitas Aki Semarang pada tahun 2011. Serta memperoleh gelar M.Kom dari Fakultas Ilmu Komputer, Universitas Dian Nuswantoro pada tahun 2014.

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.

Purwanto. Menyelesaikan pendidikan S1 di Universitas Diponegoro Semarang, S2 di STMIK Benarif Indonesia dan S3 di Universitas Multimedia Malaysia. Saat ini menjadi dosen pascasarjana Magister Teknik Informatika di Universitas Dian Nuswantoro. Minat penelitian saat ini adalah data mining, machine learning, soft computing, artificial intelligence, decision support system.