Text Feature Weighting for Summarization of Documents in Bahasa Indonesia Using Genetic Algorithm

Aristoteles[1], Yeni Herdiyeni[2], Ahmad Ridha[3] and Julio Adisantoso[4]

**[1] Department of Computer Science, University of Lampung
Bandar Lampung, 35145, Indonesia**

**[2,3,4] Department of Computer Science, Bogor Agricultural University
Bogor, 16680, Indonesia**

Abstract

This paper aims to perform text feature weighting for summarization of documents in bahasa Indonesia using genetic algorithm. There are eleven text features, i.e, sentence position (f1), positive keywords in sentence (f2), negative keywords in sentence (f3), sentence centrality (f4), sentence resemblance to the title (f5), sentence inclusion of name entity (f6), sentence inclusion of numerical data (f7), sentence relative length (f8), bushy path of the node (f9), summation of similarities for each node (f10), and latent semantic feature (f11). We investigate the effect of the first ten sentence features on the summarization task. Then, we use latent semantic feature to increase the accuracy. All feature score functions are used to train a genetic algorithm model to obtain a suitable combination of feature weights. Evaluation of text summarization uses F-measure. The F-measure is directly related to the compression rate. The results showed that adding f11 increases the F-measure by 3.26% and 1.55% for compression ratio of 10% and 30%, respectively. On the other hand, it decreases the F-measure by 0.58% for compression ratio of 20%. Analysis of text feature weight showed that only using f2, f4, f5, and f11 can deliver a similar performance using all eleven features.

Keywords: *text summarization, genetic algorithm, latent semantic feature*

1. Introduction

Understanding the contents of a document via a text summarized version of the document requires a shorter time than reading the entire document, so that the summary text becomes very important. However, a summarization requires a lot of time and cost when the documents are numerous and long document. Therefore, automatic summarization required to overcome the problem of reading time and cost.

Text summarization is a process that produces documents 50% or less their original sizes [1] with the purpose of obtaining information in a short time [2]. According to [3, 4] to perform text summarization, certain parts such as chapter headings, bold text, and the beginning of a sentence are important. In addition, according to [3] phrases like "in this summary," "this conclusion", "this paper describes" are useful for identifying the important part of the text.

Criterion of text summarization can be based on summary extraction or abstraction [5]. Extraction technique the most important or informative text units of the text into the summary, while abstraction technique takes the essence of the source text to build a summary by creating new sentences that represent the essence of the source text in a different form [5].

Several methods to do automatic text summarization have been done, including the method that use techniques lexical chains [6] to obtain a text representation. Mitra et. al. [7] have created text summary using the techniques to generate extraction path Bushy paragraph. Yeh et. al. [8] have created a text summary using lantent semantic analysis (LSA), where the summary is based on the semantic sentence. Text summarization has also be done using genetic algorithms [9, 10, 11]. Genetic algorithm is used to find the optimal weights on the features of text sentences.

Weighting sentence is an important part in text summarization. Khalessizadeh et al. [10]; Fattah & Ren [11] have created a summary of the text using genetic algorithms to weight sentences. According to [6], genetic algorithms are more effective in determining the weight compared to using TFIDF technique.

Results of the research conducted by Fattah and Ren [11] showed that text summarization produced using genetic algorithms were better than mathematical regression techniques. Accuracy of genetic algorithm was 44.94%, where mathematical regression techniques had an accuracy of 43.92%.

Fattah and Ren [11] used 10 text features to create a summary with genetic algorithms, but they did not involve semantic relations between sentences. The semantic sentence is a sentence that characterizes the semantic relationships between words based on semantics. Sentence semantics can be determined using singular value decomposition technique (SVD). Therefore, this research needs to be done to make the summary text, involving ten features of text [11] and semantic features of text

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

sentences. Determination of optimal weight or relative importance to each text feature uses genetic algorithm.

In this paper we present text summarization optimization using genetic algorithms and analyze the result of adding sentence semantic using singular value decomposition technique. The result can be used to produce an optimal summary text, to summarize quickly and to save time to get the essence of the document.

2. Background

2.1 Text Features

We use ten text features based on research of Fattah and Ren [11] and semantic feature of text sentences using Singular Value Decomposition (SVD). In this section we explain those features.

Sentence Position (f$_1$)

Sentence position is a sentence location in a paragraph. We assumed that the first sentence of each paragraph is the most important sentence. Therefore, we sort the sentence based on its position. Assuming s is a sentence in the original document, X is the position of the sentence in paragraph, and N is the number of sentences in paragraph, f1 can be calculated as follows:

$$Score_{f_1}(s) = \frac{X}{N} \qquad (1)$$

Positive Keyword **(f$_2$)**

Positive keyword is the keyword that is frequently included in the summary. It can be calculated as follows:

$$Score_{f_2}(s) = \frac{1}{length(s)} \sum_{i=1}^{n} tf_i * P(s \in S|keyword_i) \qquad (2)$$

Assume s is a sentence in the summary. S is a sentence in the document, f_2 is positive keyword text features, n is the number of keywords in sentences, tf_i is the number of keywords that appears in the sentence.

$$P(s \in S|keyword_i) = \frac{P(keyword_i|s \in S)P(s \in S)}{P(keyword_i)}$$

$$P(keyword_i|s \in S) = \frac{(Sentence\ in\ summary, and\ contains\ keyword_i)}{(Sentence\ in\ summary)}$$

$$P(keyword_i|s \in S) = \frac{(Sentence\ in\ summary, and\ contains\ keyword_i)}{(Sentence\ in\ summary)}$$

$$P(s \in S) = \frac{(Sentence\ in\ training\ corpus, and\ also\ in\ summary)}{(Sentence\ in\ training\ corpus)}$$

$$P(keyword_i) = \frac{(Sentence\ in\ training\ corpus, and\ contains\ keyword_i)}{(Sentence\ in\ training\ corpus)}$$

$P(s \in S|keyword_i)$ is calculated from the training corpus (summaries manual) , tf_i , n, and sentence length are calculated using the phrase "s" at the testing stage.

Negative Keyword **(f$_3$)**

In contrast to f_3, the negative keyword is the keywords that unlikely occurs in the summary, and it can be calculated as follows:

$$Score_{f_3}(s) = \frac{1}{length(s)} \sum_{i=1}^{n} tf_i * P(s \notin S|keyword_i) \qquad (3)$$

Assume s is a sentence in the summary, S is a sentence in the document, f_3 is negative keyword feature in text, n is the number of keywords in sentences, tf_i is the number of keywords that appears in the sentence.

Sentence centrality (similarity with other sentences) (f$_4$)

Sentence centrality is the vocabulary overlap between this sentence and other sentences in the document. It is calculated as follows:

$$Score_{f_4}(s) = \left|\frac{Keywords\ in\ s \cap Keywords\ in\ other\ sentences}{Keywords\ in\ s \cup Keywords\ in\ other\ sentences}\right| \qquad (4)$$

Sentence Resemblance to the title (f$_5$)

Sentence resemblance to the title is the vocabulary overlap between this sentence and the document title. It is calculated as follows:

$$Score_{f_5}(s) = \left|\frac{Keywords\ in\ s \cap Keywords\ in\ title}{Keywords\ in\ s \cup Keywords\ in\ title}\right| \qquad (5)$$

Sentence inclusion of name entity (proper noun) (f$_6$)

Usually the sentence that contains more proper nouns is important and it is most probably included in the document summary. The score of f$_6$ is calculated as follows:

$$Score_{f_6}(s) = \frac{proper\ name\ in\ s}{length\ (s)} \qquad (6)$$

Sentence inclusion of numerical data (f$_7$)

The sentence that contains numerical data is an important and usually included in the document summary. The score of f$_7$ is calculated as follows:

$$Score_{f_7}(s) = \frac{numerical\ data\ in\ (s)}{length\ (s)} \qquad (7)$$

Sentence length (f$_8$)

This feature is employed to penalize sentences that are too short, since these sentences are not expected to belong to the summary. We use the relative length of the sentence, which is calculated as follows:

$$Score_{f_8}(s) = \frac{number\ of\ words\ in\ (s)}{unique\ word\ in\ the\ document} \qquad (8)$$

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

Bushy path of the node (sentence) (f_9)

The bushiness of a node (sentence) on a map is defined as the number of links connecting it to other nodes (sentences) on the map. Since a highly bushy node is linked to a number of other nodes, it has an overlapping vocabulary with several sentences and is likely to discuss topics covered in many other sentences [7]. The Bushy path is calculated as follows:

$$Score_{f_9}(s) = \#\, branch\ connected\ to\ the\ node \qquad (9)$$

Summation of similarities for each node (aggregate similarity) (f_{10})

Aggregate similarity measures the importance of a sentence. Instead of counting the number of links connecting a node (sentence) to other nodes (Bushy path), aggregate similarity sums the weights on the links. Aggregate similarity is calculated as follow:

$$Score_{f_{10}}(s) = \sum koneksi\ antar\ kalimat \qquad (10)$$

Semantic Sentence (f_{11})

The semantic sentence is a sentence that characterizes relationships between sentences that are based on semantics. Assume D is a document, t (| t | = M) is the number of words in D, and S (| S | = N) is the number of sentences in D. Word matrix can be seen in formula (11), with S_j is the sentence in the document and t_i is a term that appears in the document.

$$\mathrm{A} = \begin{array}{c|cccc} & S_1 & S_2 & \dots & S_N \\ \hline t_1 & w_{1,1} & w_{1,2} & \dots & w_{1,N} \\ t_2 & w_{2,1} & w_{2,2} & \dots & wa_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \dots \\ t_M & w_{M,1} & w_{2,1} & \dots & w_{M,N} \end{array} \qquad (11)$$

$w_{i,j}$ is defined in formula (12), and tf_i is the number of appearance of term in the sentence. SF_i is the number sentences that contain the term, while the $ISF_i = \log\left(\frac{N}{SF_i}\right)$ is a measure of discriminant of the term in the document, N is the number of sentences in one document .

$$w_{i,j} = tf_i \times ISF_i \qquad (12)$$

The sentence semantics is determined by using the SVD technique [8]. Singular Value Decomposition (SVD) is A = USV ^ T, with U is M × M matrix of left singular vectors, S is a diagonal matrix M × N singular values, and V is the N × N matrix of right singular vectors. The vector V represents the sentence, while the vector U represents the words that exist in a document. The vector S is eigenvector of matrix A. The scores of this feature can be applied to (13) by assuming s is a sentence.

$$Score_{f_{11}}(s) = matrix\ of\ similarity\ \ (s) \qquad (13)$$

2.2 Genetic Algorithm

The cycle process in genetic algorithm was first introduced by Goldberg [12]. This cycle comprises several parts, namely: the initial population, fitness evaluation, individual selection, crossover, mutation, and the new population.

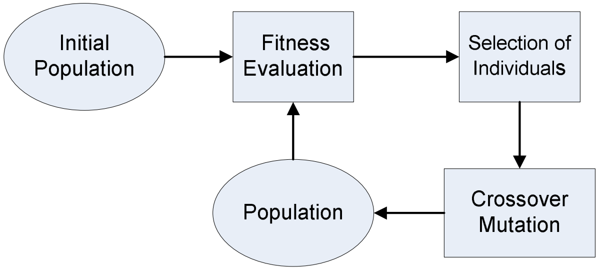

Fig. 1 The cycle process in genetic algorithm [12].

Initial population is a set of initial chromosomes which are randomly generated within a generation. The new population is a set of new chromosomes result of the selection, crossover and mutation. Process the number of population in genetic algorithm depends on the problem to be solved. Chromosome is a collection of genes that form a certain value, which is represented as a solution or an individual.

An individual or chromosome is evaluated based on a particular function as a measure of ability. Fitness function is a function used to measure the similarity or the optimal value of an individual. Fitness value is a value that states whether or not a solution. It will be referenced in achieving optimum value in the genetic algorithm.

Selection

Selection is a stage in the functioning of genetic algorithms to choose the best chromosome for crossover and mutation process [13] and get a good prospective parent. If an individual has a high fitness value is likely to be selected. If a chromosome has a small fitness value, then it will be replaced by new better chromosomes. Each chromosome in the pool selection will receive a chance of reproduction depends on the objective values of chromosomes towards the objective value of all chromosomes in the pool selection.

Crossover

Crossover is an important component in GA [14]. Crossover is the operator of the genetic algorithm involves two parents to form new chromosomes. Crossover produces new point in the search space that is ready to be tested. This operation is not always performed on all individuals exist. Individuals were randomly selected to be crossing the P_c between 0.6 and 0.95. If the crossover is not done, then the value will be derived from parent to child (offspring).

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

The principle of crossovers is to do genetic operations (exchange, arithmetic) in the corresponding genes from two parents to produce new individuals. Crossovers are performed on each individual with opportunities crossovers that have been determined. Figure 2 illustrates the process flow diagram crossovers.

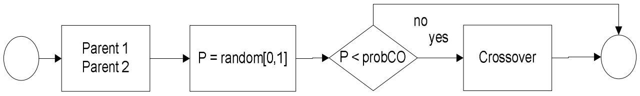

Fig. 2 Crossover flow chart.

Mutation

Mutation is supporting operators in genetic algorithms that act to change the structure of chromosomes. These changes cause the formation of a new chromosome which is genetically different from the previous chromosomes. Mutation is required to find the optimum solution, namely 1) restore the missing genes in the next generation, 2) create new genes which have never appeared in previous generations [14].

Mutation rate (p_m) is the ratio between the expected numbers of genes mutated in each generation with a total number of genes in the population. Probability mutations are used for running the program is usually low between 0001 and 0.2. If a low mutation rate is too low, then the smaller the rise of new genes. If the mutation is too high then many mutants that arise as a result many of the characteristics of the parent chromosomes are lost in the next generation so that the genetic algorithm will lose the recall or learn from the previous process [14]. Figure 3 illustrates a flow diagram of mutations.

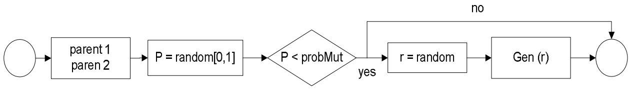

Fig. 3 Mutation flow chart.

3. Methodology

The research was done in three phases: a text document collection phase, training phase and testing phase. Figure 4 shows the stages of training and testing. We need the document text files in Indonesian language . We used 150 documents on national news. They come from the online news daily Kompas [15].

Training phase is divided into three main parts: a summary document, text features, and genetic algorithm modeling. At this stage the document summary, document manually summarized by three different people. The number of documents used as many as 100 documents Indonesian language news. The document summarized by compression (compression rate) by 30%, 20%, and 10%. Text feature extraction is an extraction process to get the text of the document. The results of the features of the text are a text extraction as sentence position (f1), positive keyword (f2), negative keywords (f3), sentence centrality (similarity with other sentences) (f4), sentence resemblance to the title (f5), sentence inclusion of name entity (proper noun) (f6), sentence inclusion of numerical data (f7), sentence length (f8), bushy path of the node (sentence) (f9), summation of similarities for each node (aggregate similarity) (f10), and semantic sentences (f11).

At this stage of modeling genetic algorithms, genetic algorithm serves as a search method for the optimal weighting on each text feature extraction. Stage summary and manual extraction of text features are used to calculate the fitness function that serves to evaluate the chromosomes. The process of genetic algorithm is shown in Figure 5. The process of genetic algorithm starts with the initial value of the population. Each contains a population of 1000 chromosomes. In Figure 6, a chromosome is represented as a weighted combination of all features in the form $(w_1, w_2, \ldots, w_{11})$.

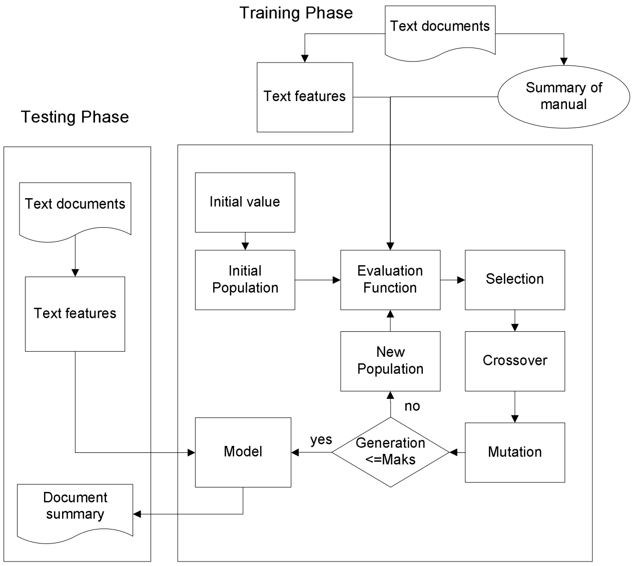

Fig. 4 Automatic text summarization.

Figure 6 is a representation of chromosomes in the text feature extraction weighting with weights w_1 on the extraction of text features (f1), w_2 the weight on the extraction of text features (f2), and so on. Weight $(w_1, w_2, \ldots, w_{11})$ value between 0 and 1 with the normalization of weights, so that total value of weight is 1.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

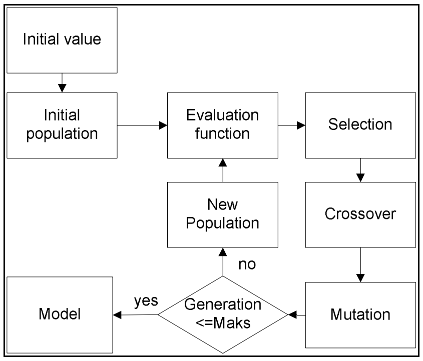

Fig. 5 Process of genetic algorithm.

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}

Fig. 6 Representation of chromosomes in text feature weighting.

The following stages of the process of genetic algorithm:

a. Randomly generated initial population of 1000 chromosomes, where each chromosome represents a weighted value text feature extraction. The weight or value that existed at chromosome applied to the formula (13) whose function is to get the score of each sentence.

b. Each chromosome is evaluated by the average F-measure, where precision and recall values obtained from slices of the summary made by machine and manual summaries. For each chromosome, F-measure process carried out at 100 documents. The average value of the F-measure can be seen in formula (14).

c. Minimum fitness function used to select chromosomes. Selection of chromosomes serves to select the chromosomes which will be selected for crossover, mutation and get a good prospective parent.

$$Score(s) = w_1 * Score_{f_1}(s) + w_2 * Score_{f_2}(s) + w_3 * Score_{f_3}(s) + w_4 * Score_{f_4}(s) + w_5 * Score_{f_5}(s) + w_6 * Score_{f_6}(s) + w_7 * Score_{f_7}(s) + w_8 * Score_{f_8}(s) + w_9 * Score_{f_9}(s) + w_{10} * Score_{f_{10}}(s) + w_{11} * Score_{f_{11}}(s) \quad (13)$$

d. We used a crossover probability is 0.88. Crossovers occur if the probability of chromosomal crossover smaller than the probability of crossovers that have been determined. The technique used in crossovers is one point. Mutation probability used is 0.2.

e. 250 generations of genetic algorithm applied to the process to get the weight of the optimal text feature extraction.

Here is the calculation of the F-measure, precision, and recall according to [16] :

$$F_{\beta=1} = \frac{(\beta^2+1)PR}{\beta^2 P+R} = \frac{2PR}{(P+R)};\ P = \frac{|S\cap T|}{|S|};\ R = \frac{|S\cap T|}{|T|} \quad (14)$$

β is the weight of the precision (P) and recall (R) if β <1 the emphasis on precision and if β> 1 the emphasis on the recall. F-measure values between 0 and 1, when the value of β = 1. Assume that S is a text summary of the results of the machine (the score function of the training documents) and T is a summary of the manual.

Testing phase using 50 documents (documents used at this stage different from the documents used in the training phase). The next process is the extraction of text features. This process is similar to that done in the text feature extraction stage of training. The process of summarizing text automatically based on models that have been created in the training stage. This model is represented as weight $(w_1, w_2, \dots, w_{11})$ on features text a stable or optimal. The combination weights $(w_1, w_2, \dots, w_{11})$ applied to the function score for each sentence and can be seen in formula (13). This function is used to integrate all the features of the text. Selection of the sentence serves to generate a summary. Therefore, the entire sentence ordered by the value calculated from the formula (13), and the number of sentences that set the top-score are ordered using the compression rate (CR) 10%, 20%, and 30%.

4. Result

Tests were performed by a total of five trials for each CR 10%, CR 20%, and 30% CR. The result of the F-measure was calculated based on the average of all test documents. At this stage, the tests were performed on the model of the best chromosome $(w_1, w_2, \dots, w_{11})$ on CR 10%, CR 20%, and 30% CR. Based on Figure 7, F-measure does not increase significantly in each compression rate.

On CR 30% showed the highest accuracy rate compared with the results of an accuracy of 10% CR, and CR 20%. This indicates that the compression rate is high that cause the similarity value system summary with manual summaries higher also.

Table 1 show about CR 30%. From this table we knew that w_5 (sentence resemblance to the title), w_4 (sentence centrality), w_2 (positive keyword), w_{11} (semantic sentence) have high weight compared to the other weight's features.

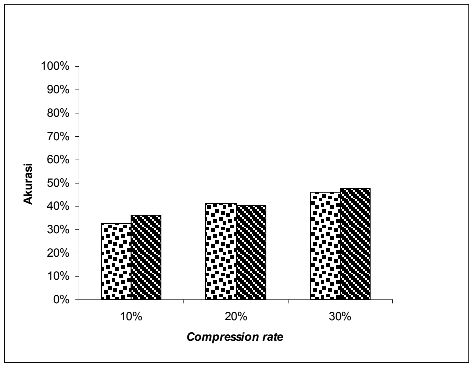

Fig. 7 Comparison testing of the F-Measure 'ten features text' (▒) and 'eleven features text' (▧) on CR 10%, CR 20%, and CR 30%.

Table 1 Illustration of the weight on chromosome model CR 30%

Weight	Experiment					Total number of weight
	1	2	3	4	5	
w_1	4	2	4	1	8	19
w_2	5	5	8	5	7	30
w_3	0	1	0	0	0	1
w_4	9	9	3	9	9	39
w_5	10	10	9	10	10	49
w_6	7	6	2	6	3	24
w_7	8	7	1	7	5	28
w_8	2	4	7	2	1	16
w_9	1	3	10	4	2	20
w_{10}	3	0	6	8	4	21
w_{11}	6	8	5	3	6	28

5. Conclusions

Genetic algorithms can be used as a determinant of the optimal weights on the text features. The feature (f5) "sentence resemblance to the title" is very important in summarizing text, the feature (f3) "negative keyword" can be ignored in text summarization. The features (f2) "positive keywords", (f4) "sentence centrality (similarity with other sentences)", f(5) "sentence resemblance to the title", f(11) "sentence semantics" represent the eleven text features to summarize text. The computing time for text features (f2, f4, f5, F11) are shorter than the computational time eleven text features. The compression rate has a positive correlation to the high accuracy. The addition of semantic sentence using the SVD is able to replace other features.

Acknowledgments

The author would like to thank Bogor Agricultural University for their educational and Ministry of National Education for financial support.

References

[1] Radev D, Hovy E, McKeown K. 2002. *Introduction to the special issue on text summarization. Computer linguist.* 28(4).

[2] Blake C, Pratt W, Rules B, Fiturs F. 2001. *A semantic approach to selecting fiturs from text*. ICDM. 59–66.

[3] Edmundson H. 1969. *New methods in automatic extracting. Journal of ACM*. 16(2):264-285.

[4] Hovy E, Lin C. 1997. *Automatic text summarization in summarist. ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*. 18-24.

[5] Jezek K, Steinberger J. 2008. *Automatic text summarization* (*The state of the art 2007 and new challenges*). Vaclav Snasel (Ed) : Znalosti. 1-12.

[6] Barzilay R, Elhadad M. 1997. *Using lexical chains for text summarization. Proceedings of Intelligent Scalable Text Summarization Workshop* (ISTS'97). ACL.

[7] Mitra M, Singhal A, dan Buckley C. 1997. *Automatic text summarization by paragraph extraction*. Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. *39-46*.

[8] Yeh J, Ke H, Yang W, Meng I. 2005. *Text summarization using a trainable summarizer and latent semantic analysis*. Information Processing & Management. 41(1). 75-95.

[9] Silla CN, Pappa GL, Freitas, Kaestner CA. 2004. *Automatic text summarization with genetic algorithm-based attribute selection*. 9th Ibero-America Conference on AL, Lecture Notes in Computer Science. 305-314.

[11] Fattah MA, Ren F. 2008. *Automatic text summarization*. Proceeding of Word Academic of Science, Engineering and Technology. ISSN 1307-6884.

[10] Khalessizadeh SM, Zaefarian R, Nasseri SH, Ardil E. 2006. *Genetic Mining : Using genetic for topic based on Concept Distribution*.World Academy of Science, Engineering and Technology 13. 144-147.

[12] Goldberg DE. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley Longman, Inc.

[13] Cox E. 2005. *Fuzzy modeling and genetic algorithm for data mining and exploration*. USA: Morgan Kauftman Publisher.

[14] Gen M, Cheng R. 1997. *Genetic algorithm and engineering design*. John Wiley & Sons, Inc. Canada.

[15] Ridha A. 2002. *Pengindeksan otomatis dengan istilah tunggal untuk dokumen berbahasa indonesia* [senior thesis]. Bogor. Ilmu Komputer, Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor.

[16] Baeza-Yates R, Ribeiro-Neto B. 1999. *Modern information retrieval*. ACM Press New York. Addison-Wesley.

Aristoteles is B.Sc in Computer Science (University of Padjadjaran, Indonesia, 2004), M.Sc in Computer Science (Bogor Agricultural University, Indonesia, 2011). Since 2006 the author active as a lecturer in the Department of Computer Science, University of Lampung, Indonesia.

Yeni Herdiyeni is B.Sc in Computer Science (Bogor Agricultural University, Indonesia, 1999), M.Sc in Computer Science (University of Indonesia, Indonesia, 2005), Doctor in Computer Science (University of Indonesia, Indonesia, 2010). Since 2000 the author active as a lecturer in Department of Computer Science Bogor Agricultural University, Indonesia.

Ahmad Ridha is B.Sc in Computer Science (Bogor Agricultural University, Indonesia, 2002), M.Sc in Computer Science (King Fahd University of Petroleum & Minerals (KFUPM), Arab Saudi, 2008). Since 2005 the author active as a lecturer in Department of Computer Science Bogor Agricultural University, Indonesia. He is a member of the IEEE and the IEEE Computer Society.

Julio Adisantoso is active as a lecturer in Department of Computer Science Bogor Agricultural University, Indonesia.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.