# IMPLEMENTATION SUPPORT VECTOR MACHINE (SVM) AND MAXIMAL MARGINAL IMPORTANCE (MMI) FOR MULTI DOCUMENT ARTICLES SUMMARIZATION IN INDONESIAN LANGUAGE

Novriana

Informatics Engineering – Indonesian Computer University
Dipati Ukur Street 114, Bandung
E-mail : Novriana@hotmail.com

**ABSTRACT**

Automatic text summarization is part of NLP (natural language processing) that allows an application can automatically summarize. In previous research, the method of support vector machine (SVM) and maximum marginal relevance (MMR) is used to doing automatic text summarization for multi document articles, which the SVM method used to process classification the summary candidate sentence and MMR methods of is done to do selection sentence summary results. The test results from the study obtained the best parameter values classification SVM with RBF kernel value is C = 4.0 and γ = 512.0. The f-measure performance value obtained of 0771. In another research maximal marginal importance (MMI) method which is a modification of the MMR method to doing automatic text summarization on a single document. From this research obtained the optimal threshold value for MMI method is 0.1. Therefore, in this study will be implemented MMI and SVM method in automatic text summarization to know how well the use of both methods.
In this research automatic text summarization is divided into four main stages: preprocessing, classification, feature extraction of sentences using SVM, and make a summary using MMI method.
From performance testing value, obtained a value of recall, the value of precision and f-measure of.

Keywords : automatic text summarization, support vector machine, maximal marginal importance, feature extraction, multi-document

## 1. INTRODUCTION

Automatic text summarization is part of NLP (natural language processing) that allows an application can automatically summarize. Text Automatic compaction can be done with a ranked method which was done as the process of selecting appropriate sentence on the text by giving the weighting value to the sentences so that sentences containing important information will occupy the top order with the highest weight value and value is chosen to be included as a candidate for summary sentences [1].

On the research [2] of compaction is performed automatic text to multi document by way of classifying sentences in each article into the relevant class (included in the summary) and a class of non-relevant (not included in the summary) using SVM classifier. Further selected n sentences from class positive based on the value relevance that can be either the value of the probability of occurrence of the relevant class or marginal relevance value generated through MMR. The research done by the best parameter values in the parameters C = 4.0 and γ = 512.0, f-measure and value obtained is amounting to 0771. Research [1] using the maximal marginal importance (MMI) method of which is a modification of the method of maximum marginal relevance (MMR), in the study said that the method of maximal marginal importance was chosen because it viewed the shortage at the maximum marginal relevance that is the query based methods which determine the level of importance of a sentence is by looking at the similarities between the sentence with a query that user entered. From this research obtained the optimal threshold value for method of MMI is 0.1.

Based from the description, in this research will do automatic text summarization using support vector machine (SVM) method and maximal marginal importance (MMI) to see the extent to which the method can produce a good summary.

## 2. ANALISYS MODEL AND IMPLEMENTATION

To be able to examine the implementation of the method used, then the system will be built will be divided into sub-sub system, where the process in generating summary will be described and identified into several stages and processes. The input of the system are the Indonesia-language news articles with similar topics. The output of the system is the summary already done extraction (compression rate). This system has two main processes, namely the process of training and testing process. Where in the process of training method of SVM model will look for klasifikas that

will be used in the process of testing. As for a general overview of the system that will be created can be seen in Figure 1. the following:

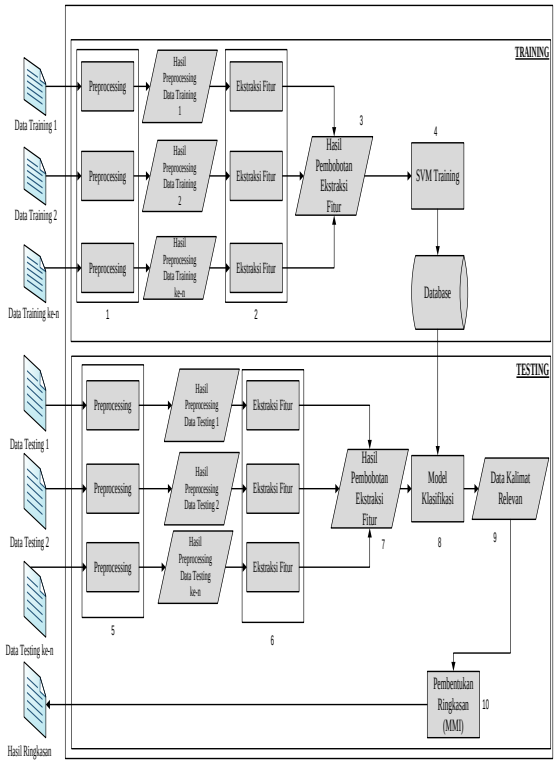



**Figure 1 An Overview Of The System**

## 2.1. Preprocessing

Preprocessing stage is to prepare the text into data that will be processed in the next stage. Text preprocessing is used on this research uses the process of formation of sentence, case folding, and tokenizing stopwords. Preprocessing stage can be seen in Figure 2.

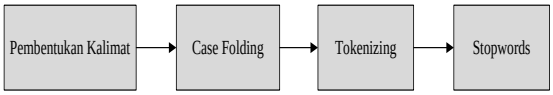



**Figure 2 Preprocessing Stage**

Sample text in the input data, which can be seen in table 1 below:

**Table 1 Examples Of Data Input**

| Jessica Wongso Divonis 20 Tahun Penjara |
|---|
| Majelis hakim Pengadilan Negeri Jakarta Pusat telah membacakan vonis atas kasus pembunuhan berencana Wayan Mirna Salihin. Terdakwa tunggal Jessica Kumala Wongso dijatuhi vonis 20 tahun penjara. "Menyatakan Jessica Kumala alias Jessica Kumala Wongso atau Jess telah terbukti sah dan meyakinkan bersalah melakukan pembunuhan berencana dan menjatuhkan pidana selama 20 tahun penjara dipotong masa tahanan dan menetapkan terdakwa tetap ditahan," ujar Ketua Majelis Hakim Kusworo di PN Jakarta Pusat, Kamis (27/10/2016). Vonis hakim ini sama dengan tuntutan jaksa penuntut umum (JPU). |

### 2.1.1. Formation of Sentence

The formation of sentences that is the process of forming a long string of text documents into a set of sentences. In the process to transform documents into sentences, use the function split (), with a dot ".", "?" question mark, and exclamation mark "!" as the delimiter string to cut the document [3]. After the formation of the sentence, then the input data in table 1 will be sentences that can be seen in table 2.

**Table 2 Result Of Formation Sentences**

| S | Kalimat |
|---|---|
| 1 | Majelis hakim Pengadilan Negeri Jakarta Pusat telah membacakan vonis atas kasus pembunuhan berencana Wayan Mirna Salihin. |
| 2 | Terdakwa tunggal Jessica Kumala Wongso dijatuhi vonis 20 tahun penjara. |
| 3 | “Menyatakan Jessica Kumala alias Jessica Kumala Wongso atau Jess telah terbukti sah dan meyakinkan bersalah melakukan pembunuhan berencana dan menjatuhkan pidana selama 20 tahun penjara dipotong masa tahanan dan menetapkan terdakwa tetap ditahan," ujar Ketua Majelis Hakim Kusworo di PN Jakarta Pusat, Kamis (27/10/2016). |
| 4 | Vonis hakim ini sama dengan tuntutan jaksa penuntut umum (JPU). |

### 2.1.2. Case Folding

Case folding is the process by which all text processing text converted into the same case. In this case all the modified text representation into small letters (lowercase) and accept only numbers and omitting the symbols. After case folding of

sentences results data in table 2, then the results from case folding can be seen in table 3 below:

**Table 3 The Results Of Case Folding**

| S | Kalimat |
|---|---|
| 1 | majelis hakim pengadilan negeri jakarta pusat telah membacakan vonis atas kasus pembunuhan berencana wayan mirna salihin |
| 2 | terdakwa tunggal jessica kumala wongso dijatuhi vonis 20 tahun penjara |
| 3 | menyatakan jessica kumala alias jessica kumala wongso atau jess telah terbukti sah dan meyakinkan bersalah melakukan pembunuhan berencana dan menjatuhkan pidana selama 20 tahun penjara dipotong masa tahanan dan menetapkan terdakwa tetap ditahan ujar ketua majelis hakim kusworo di pn jakarta pusat, kamis 27102016 |
| 4 | vonis hakim ini sama dengan tuntutan jaksa penuntut umum jpu |

### 2.1.3. Tokenizing

On the process of tokenizing a sentence results case folding must be broken down first into a smaller section in the form of a single word that has meanings or commonly called the token. After tokenizing data, then the results of case folding in table 3 can be seen in table 4 below:

**Table 4 The Result of Tokenizing**

| S | Kalimat | Token |
|---|---|---|
| 1 | majelis hakim pengadilan negeri jakarta pusat telah membacakan vonis atas kasus pembunuhan berencana wayan mirna salihin | "majelis", "hakim", "pengadilan", "negeri", "jakarta", "pusat", "telah", "membacakan", "vonis", "atas", "kasus", "pembunuhan", "berencana", "wayan", "mirna", "salihin". |
| 2 | terdakwa tunggal jessica kumala wongso dijatuhi vonis 20 tahun penjara | "terdakwa", "tunggal", "jessica", "kumala", "wongso", "dijatuhi", "vonis", "20", "tahun", "penjara" |
| 3 | menyatakan jessica kumala alias jessica kumala wongso atau jess telah terbukti sah dan meyakinkan bersalah melakukan pembunuhan berencana dan menjatuhkan pidana selama 20 tahun penjara dipotong masa tahanan dan menetapkan terdakwa tetap ditahan ujar ketua majelis hakim kusworo di pn jakarta pusat, kamis 27102016 | "menyatakan", "jessica", "kumala", "alias", "jessica", "kumala", "wongso", "atau", "jess", "telah", "terbukti", "sah", "dan", "meyakinkan", "bersalah", 'melakukan", "pembunuhan", "berencana", "dan", "menjatuhkan", "pidana", "selama", "20", "tahun", "penjara", "dipotong', "masa", 'tahanan", "dan", "menetapkan", "terdakwa", "tetap", "ditahan", "ujar", "ketua", "majelis", "hakim", "kusworo", "di", 'pn", "jakarta", "pusat", "kamis", "27102016" |
| 4 | vonis hakim ini sama dengan tuntutan jaksa penuntut umum jpu | "vonis", "hakim", "ini", "sama", "dengan", "tuntutan", "jaksa", "penuntut", "umum", "jpu" |

### 2.1.4. Stopwords

Text filtering aims to take words that can represent the contents of a document by way of disposing of words that are considered important not commonly referred to as stopword. Stopwords can be either conjunctions, preposition, interjection of words such as "and", "a", "and", "to", "wow", "as well as", "wow" and others [3]. In this study used a list of stopwords list for Indonesia [4]. After stopwords from the data in table 4, then results of stopwords can be seen in table 5 below.

**Table 5 The Result of Stopwords**

| S | Kalimat | Token |
|---|---|---|
| 1 | majelis hakim pengadilan negeri jakarta pusat telah membacakan vonis atas kasus pembunuhan berencana wayan mirna salihin | "majelis", "hakim", "pengadilan", "negeri", "jakarta", "pusat", "membacakan", "vonis", "pembunuhan", "berencana", "wayan", "mirna", "salihin". |
| 2 | terdakwa tunggal jessica kumala wongso dijatuhi vonis 20 tahun penjara | "terdakwa", "tunggal", "jessica", "kumala", "wongso", "dijatuhi", "vonis", "20", "tahun", "penjara" |
| 3 | menyatakan jessica kumala alias jessica kumala wongso atau jess telah terbukti sah dan meyakinkan bersalah melakukan pembunuhan berencana dan menjatuhkan pidana selama 20 tahun penjara dipotong masa tahanan dan menetapkan terdakwa tetap ditahan ujar ketua majelis hakim kusworo di pn jakarta pusat, kamis 27102016 | "jessica", "kumala", "alias", "jessica", "kumala", "wongso", "jess", "terbukti", "sah", "meyakinkan", "bersalah", "pembunuhan", "berencana", "menjatuhkan", "pidana", "20", "tahun", "penjara", "dipotong', 'tahanan", "menetapkan", "terdakwa", "ditahan", "ketua", "majelis", "hakim", "kusworo", 'pn", "jakarta", "pusat", "kamis", "27102016" |
| 4 | vonis hakim ini sama dengan tuntutan jaksa penuntut umum jpu | "vonis", "hakim", "tuntutan", "jaksa", "penuntut", "jpu" |

## 2.2. Feature Extraction

For using statistical methods, the phrase is represented as a vector features [5]. These features are the attributes that are used to represent the data that is used in the task. Keywords are used to extract the feature, where the feature was the result of extraction are assumed to provide information about the sentence [2]. In this study used 8 features, where these features have been used before on other research such as research [2], [5] and [6]. Eight features described below will be used for the process of classification using support vector machine (SVM).

### 2.2.1. The length of Sentences Feature ($F_1$)

This feature is used to calculate the length of the sentence, where this feature is calculated by dividing the number of words in a sentence against a number of words from the longest sentence in each article [2]. Features a long sentence is calculated using the following equation:

$$Scoref1(s_i) = \frac{Jumlah\ kata\ yang\ terdapat\ pada\ kalimat}{jumlah\ kata\ yang\ terdapat\ pada\ kalimat\ terpanjang} \quad (1)$$

### 2.2.2. Features Position Sentence ($F_2$)

This feature represents the first and last sentence in each article the results of preprocessing is an important sentence [5], then the value of the features for each of the first and last sentence in each article will be worth 1 and other sentence would be worth 0.

### 2.2.3. The features of numerical Data ($F_3$)

Usually the sentence that contains numerical data is an important sentence and the sentence are usually entered into the summary [2]. Features of numerical data calculated using the following equation:

$$Scoref3(s_i) = \frac{jumlah\ data\ numerik}{panjang\ kalimat(jumlah\ kata\ pada\ kalimat)} \quad (2)$$

### 2.2.4. The feature Words in sentences Thematik ($F_4$)

This feature calculates the relative occurrence of keywords in a sentence, usually a sentence which has a relatively good keywords is a summary sentence [2]. Features words of thematik calculated using the following equation:

$$Scoref4(s_i) = \frac{jumlah\ kata\ kunci\ pada\ kalimat}{panjang\ kalimat\ (jumlah\ kata\ pada\ kalimat)} \quad (3)$$

#### 2.2.5. The feature that resembles the title Sentence ($F_5$)

The sentence which resembles the title is a sentence that has a vocabulary of overlap between the sentence with the title of [2]. A feature that resembles the title phrase is calculated using the following equation:

$$Scoref5(s_i) = \frac{jumlah\ kata\ pada\ judul}{jumlah\ kata\ yang\ sama\ pada\ judul} \quad (4)$$

#### 2.2.6. Similarity of sentences with the Feature set of the other Sentence ($F_6$)

Similarity of sentences can be seen from the vocabulary of overlap between the sentence with the sentence of the other, to make the word is seen is just the keyword [2]. Features of similarity of sentences with a collection of other sentence is calculated using the following equation:

$$Scoref6(s_i) = \frac{kata\ kunci\ dalam\ kalimat\ \cap kata\ kunci\ dalam\ kalimat\ lain}{kata\ kunci\ dalam\ kalimat\ \cup kata\ kunci\ dalam\ kalimat\ lain} \quad (5)$$

#### 2.2.7. The lexical features of a bond With the previous sentence ($F_7$)

The bond between lexical sentence with the previous sentence is defined as a word (stem) that appears in the second sentence, the value will be 1 if the lexical relationship, would be worth 0 if it does not have.

#### 2.2.8. The lexical features of a bond with the sentence Afterwards ($F_8$)

Ikatan leksikal antara kalimat dengan kalimat sesudahnya didefinisikan sebagai kata (stem) yang muncul dalam kedua kalimat tersebut, nilai 1 apabila memiliki hubungan leksikal, dan akan bernilai 0 jika tidak punya.

### 2.3. Support Vector Machine (SVM)

A classification method of many developed and implemented is the support vector machine (SVM) [6]. The basic idea is to maximize the boundary hyperplane SVM (maximal margin hyperplane) [6]. In this technique, SVM will attempt to find the function of the separator (classifier) which is optimum that could separate the two sets of data from two distinct classes [7]. In mathematics, the SVM optimization problem formulations for linear classification in the primal space are:

$$min_{w,b} \frac{1}{2} \|w\|^2 \quad (6)$$

Subject to : $y_i(wx_i + b) \geq 1, i = 1, \dots, N$

Where is the input, the output is the value = 1 or-1. $w$ and $b$ is a parameter that will look for its value. In the $\frac{1}{2}\|w\|^2$ equation (2.11), minimize the $\frac{1}{\|w\|}$ destination function (objective function). or $w^T w$ maximize the quantity or with constraint:

$$y_i(wx_i + b) \geq 1 \quad (7)$$

When the output data $y_i = +1$ then become delimiters:

$$(wx_i + b) \geq 1 \quad (8)$$

Conversely, if $y_i = -1$ then the delimiter becomes:

$$(wx_i + b) \leq -1 \quad (9)$$

In case that isn't feasible (infeasible) where some of the data may not be classified correctly, his mathematical formulation becomes:

$$min_{w,b,t} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi i \quad (10)$$

Subject to: $y_i(wx_i + b) \geq 1, i = 1,2, \dots, N$

In this formulation SVM will attempt to minimize the error of classification stated in the presence of the slack variables. The use of the slack variable is to resolve cases of impropriety (infeasibility) of delimiters (constraints) in equation (7) gives way by applying the penalty C parameters. Vector w perpendicular to the separator functions:

$$wx + b = 0 \quad (11)$$

The constants b specify the location of the separator functions relative to the origin. Equation (10) programme is nonlinear. This can be seen from the square-shaped goal functions. To make this problem more easily and efficiently to complete, this problem can be transformed into the dual space. For it, equations (10) into the Lagrangian function:

$$max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \quad (12)$$

Subject to: $\sum_{i=1}^{N} \alpha_i y_i = 0, i = 1, .., n$

Equation (12) is a Quadratic Programming (QP) and delimiter (constraint) linear.

Many of the techniques of data mining or machine learning developed with the assumption kelinieran. Therefore if a classification showing the perceptron algorithm, ketidaklinieran can't handle it. The

kernel is one of the methods to address them [7]. By mapping any data to feature space, then equation (12) would be:

$$max \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \varphi(x_i)^T \varphi(x_j) \quad (13)$$

Subject to: $\sum_{i=1}^{N} \alpha_i y_i = 0, i = 1, .., n$

In the equation (13), the dot product between $\varphi(x_i)^T \varphi(x_j)$ can be represented $K(x_i, x_j)$ by where K is a kernel function. Next after SVM trained and found value $\alpha$ and b it will be obtained by the equation:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \quad (14)$$

The optimal separator function is:

$$g(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \quad (15)$$

Where $\alpha_i = 1, .., N$ is the optimal solution of the equations (13) and b was chosen so that for any $y_i f(x_i) = 1$. Data $x_i$ which $\alpha_i > 0$ is called support vector data and stating the necessary training to represent the optimal decision. To address the problem of nonlinier that often happens in the case of a real, workable method then the kernel. Kernel methods used in this study i.e. Radial Basis Function (RBF is) with the equation:

$$K(x, y) = \exp(-\gamma \|x - x_i\|^2) \quad (16)$$

### 2.4. Maximal Marginal Importance (MMI)

Maximal marginal importance (MMI) is a modification of the method of maximal marginal relevance (MMR) [1].

#### 2.4.1. Weighting the Unit or a feature of a sentence

To get the-the most informative text, usually awarding a score or value weighting terhdap text units that are considered essential. The weighting of the sentence is a sentence that is calculated from the value of the units text in it that indicates how important the sense of the sentence [1]. Sentence features used to make the process of the formation of sentences using the MMI is as follows:

#### 2.4.1.1. Word Sentence Score (WSS)

WSS is a formula granting weighting value that implements the methods of the TF-ISF (Term Frequency-Inverse Sentence Frequency). To calculate the value of the sutu words (terms) in a sentence the source text can be calculated using the following equation:

$$w_{ij} = tf_{ij} \times isf = tf(t_{ij}, s_i) \left[1 - \frac{\log(sf(t_{ij})+1}{\log(n+1)}\right] \quad (17)$$

Description:

$w_{ij}$ = weighting value/term a word $t_{ij}$ in a sentence the source text

$t_{ij}$ = suatu term pada kalimat teks sumber

$tf(t_{ij}, s_i)$= the frequency of occurrence of the term $t_{ij}$ in a sentence

$sf(t_{ij})$= the number of sentences that contain the term $t_{ij}$

$n$ = the number of whole sentences sentences in the source text

Having gained weight the value of term $t_{ij}$ in the $s_i$ sentence, then the calculation of the weighting value of sentence $s_i$ (WSS) with the following equation:

$$WSS(s_i) = 0.1 + \frac{\sum tj \in s_i w_{ij}}{HTFS}, \quad \text{the number of sentences containing } t_j \geq \frac{1}{2} LS \quad (18)$$

Description :

$WSS(s_i)$= weighting value of sentences $s_i$ in the source text based on term frequency.

LS = number/length of sentence summary

$t_j$ = term in the $s_i$ sentence qualified i.e. sentences in the source text which contains more than 1 or = LS.

$w_{ij}$ = weighted value of a term $t_{ij}$ in a sentence

HTFS = the greatest value $w_{ij}$ from the overall term of the sentence.

Weighting value 0.1 is the minimum value obtained a sentence if term-term in the sentence is not important.

#### 2.4.1.2. Keyword Feature

Ten term with $w_{ij}$ most of the rest of the sentences of the source text into keyword [1]. The sentence containing the keyword will be assigned a value of 0.1 /keyword.

#### 2.4.1.3. Nfriends Feature

Nfriends $(s_i, s_j)$ measure connectedness between sentences $s_i$ and $s_j$. $s_i$ $friends$ determined by calculating the value of the similarity between $s_i$ a sentence and the other in the source text. If the value $(s_i, s_w) \geq threshold$ of the similarity, then it

is friends. And for calculating the Nfriends between sentences $s_i$ and $s_j$ use the following equation:

$$Nfriends(s_i, s_j) = \frac{s_i(friends) \cap s_j(friends)}{|s_i(friends) \cup s_j(friends)|} \quad \text{where } i \neq j \qquad (19)$$

#### 2.4.1.4. Ngrams Feature

Ngrams $(s_i, s_j)$ is a method that calculates a value similarity between n sentences $s_i$ and $s_j$. Ngrams do separation per n words from a sentence. For calculating similarity between sentences $s_i$ and $s_j$ Ngrams used the following equation:

$$Ngrams(s_i, s_j) = \frac{s_i(ngrams) \cap s_j(ngrams)}{|s_i(ngrams) \cup s_j(ngrams)|} \quad , \text{ where } i \neq j \qquad (20)$$

#### 2.4.1.5. The Similarity to first sentence (sim_fsd)

This feature gives value to the sentences based on the similarity of that sentence with the first sentence in the source text. Because on the news articles first sentence is considered an important sentence.

$$simFSD(s_i) = sim(s_i, s_0) \qquad (21)$$

Where $s_0$ is the first sentence in the source text.

#### 2.4.1.6. Sentence Centrality (SC)

This feature is calculated based on three factors, namely the similarity, Nfriends features, and features Ngrams. The value of the similarity is calculated using equation 22:

$$sim(s_i, s_j) = \frac{\sum w_i \in s_i, s_j \, tf(w_i, s_i) tf(w_i, s_j) \left[1 - \frac{\log(sf(w_i)+1)}{\log n+1}\right]^2}{\sqrt{\sum w_i \in s_i (tf(w_i, s_i) \cdot \left[1 - \frac{\log(sf(w_i)+1)}{\log n+1}\right]^2} \times \sqrt{\sum w_i \in s_j tf(w_i, s_j) \cdot \left[1 - \frac{\log(sf(w_i)+1)}{\log n+1}\right]^2}} \qquad 22$$

Sentence centrality calculated using the following equation:

$$SC(s_i) = \frac{\sum_{j=1}^{n-1} sim(s_i, s_j) + \sum_{j=1}^{n-1} nfriends(s_i, s_j) + \sum_{j=1}^{n-1} ngrams(s_i, s_j)}{n-1}$$

, where $i \neq j$ and $sim(s_i, s_j) \geq 0 \qquad (23)$

Description :

$SC(s_i)$ = weighting value SC $s_i$ sentences in the source text

$sim(s_i, s_j)$ = the value of the similarity between sentences $s_i$ and $s_j$

$Nfriends(s_i, s_j)$ = the value of nfriends between sentences $s_i$ and $s_j$

$Ngrams(s_i, s_j)$ = the value of ngrams between sentences $s_i$ and $s_j$

0 = *threshold*

$n$ = the number of whole sentences sentences in the source text

#### 2.4.1.7. Tittle Help Sentence (THS)

THS is a formula for calculating the value of commonality ngrams between sentences with title text. THS is calculated using the following equation:

$$THSRS(s_i) = \frac{s_j(ngrams) \cap THS(s_j(ngrams))}{|s_i(ngrams) \cup THS(s_j(ngrams))|} \qquad (24)$$

#### 2.4.1.8. Tittle Help Sentence Relevance Sentence (THSRS)

This feature is used to calculate the value of commonality ngrams between sentences with sentences that have most large THS is calculated using the previous equation (24). THSRS calculated using the equation:

$$THSRS(s_i) = \frac{s_j(ngrams) \cap THS(s_j(ngrams))}{|s_i(ngrams) \cup THS(s_j(ngrams))|} \qquad (25)$$

Then the calculated value is also a $SS_{NG}$ sentence $s_i$ which is the average value of the value $THS(s_i)$ and $THSRS(s_i)$ use the following equation:

$$SS_NG_{(s_i)} = \frac{THS(s_i) + THSRS(s_i)}{2} \qquad (26)$$

### 2.4.2. Value of Importance and Relevance Sentence

Weighting value of importance is the value of the interests of a sentence [1]. The value of the importance a sentence is calculated using the following equation:

$$IMPR(s_i) = avg(WSS(s_i) + SC(s_i) + SS_{NG(s_i)} + simFSD(s_i) + kwrd(s_i)) \qquad (27)$$

Description :

$WSS(s_i)$= weighting value of sentences $s_i$ in the source text based on term frequency.

$SC(s_i)$ = weighting value SC $s_i$ sentences in the source text

$SS_NG_{(s_i)}$= weighting value of SS_NG sentences in the source text

$simFSD(s_i)$= weighting value of simFSD sentences $s_i$ in the source text

$kwrd(s_i)$= weighting value keyword sentences $s_i$ in the source text

Weighting value relevance is the value the connectedness of a sentence with other sentences in the source text. Calculated using the following equation:

$$Rel(s_i, s_j) = avg(Nfriends(s_i, s_j) + Ngrams(s_i, s_j) + sim(s_i, s_j))$$

(28)

Description:

$sim(s_i, s_j)$ = the value of the similarity between sentences $s_i$ and $s_j$

$Nfriends(s_i, s_j)$ = the value of nfriends between sentences $s_i$ and $s_j$

$Ngrams(s_i, s_j)$ = the value of ngrams between sentences $s_i$ and $s_j$

Prior to the calculation of the maximal marginal importance, first sought the value by using the following equation:

$$ScoreBT(s_i) = impr(s_i) + (1 - (1 - impr(s_i) \times friendsNo(s_i)))$$

(29)

Description:

$impr(s_i)$= the value of the importance of the sentence $s_i$

$friendsNo(s_i)$= the number of friends on sentence $s_i$

After obtained $impr(s_i)$ the value of the MMI value calculated using the sentence the following equation:

$$MMI(s_i) = Arg\ \max_{si \in cs \setminus ss}[(ScoreBT(s_i) - \beta(s_i)) - max_{sj \in ss}(Rel(s_i, s_j))]$$

(30)

Description:

$s_i$ = index to the sentence-i who is a member of the sentence summary

$s_j$ = index to the sentence-a sentence that is also a member of the summary.

CS = competitive phrase in 1 binary tree

SS = sentences that elected a sentence summary

$Rel(s_i, s_j)$= the value of rel between sentences $s_i$ and $s_j$

$\beta(s_i)$= the value of the penalty dinisialisasi 0.01 * level of sentence on a binary tree

## 3. TESTING

At the stage of testing in a test of 5 documents that are used as sample data input with the same topic. Then selected a number of n by the abstractor sentence based on compression of 30%. Sample input data to be tested can be seen in the following table:

**Table 6 Skenario Detail Pengujian**

| D | Nama File | Sumber | *Link* |
|---|---|---|---|
| 1 | Rio2016 | BBC Indonesia | http://www.bbc.com/indonesia/olahraga/2016/08/160817_live_badminton |
| 2 | Rio2016 1 | Liputan6 | http://bola.liputan6.com/read/2579629/tontowi-liliyana-rebut-emas-olimpiade-rio |
| 3 | Rio2016 2 | Okezone | http://sports.okezone.com/read/2016/08/18/40/1466341/liliyana-natsir-tontowi-ahmad-raih-emas-bagi-indonesia-di-olimpiade-rio-2016 |
| 4 | Rio2016 3 | Liga Olahraga | http://www.ligaolahraga.com/detail/20655/berita-badminton-kado-71-tahun-indonesia-merdeka-tontowi-liliyana-raih-emas-olimpiade-rio-2016/ |
| 5 | Rio2016 | Badminton | http://badmintonindonesia.org/ap |

| | | | |
|---|---|---|---|
| | 4 | Indonesia | p/information/newsDetail.aspx?/5297 |

Testing the value of the performance is done by calculating the value of the recall, precision, and f-measure. As for the value of ferformansi is calculated using the following equation:

$$recall = \frac{\sum jumlah\ kalimat\ yang\ sama}{\sum kalimat\ ringkasan\ manual} \quad (31)$$

$$precision = \frac{\sum jumlah\ kalimat\ yang\ sama}{\sum kalimat\ ringkasan\ sistem} \quad (32)$$

$$kalimat\ yang\ sama = kalimat\ ringkasan\ sistem \cap kalimat\ ringkasan\ manual \quad (33)$$

$$f - measure = 2 \times \frac{recall \times precision}{precision + recall} \quad (34)$$

From the testing that is performed, the obtained results summary of the system which can be seen in the following table:

**Table 7 The Summary Result From System**

| D | Jumlah Kalimat | Jumlah Ringkasan Sistem | Kalimat Ke- |
|---|---|---|---|
| 1 | 10 | 4 | 1,2,4,10 |
| 2 | 16 | 3 | 8,10,13 |
| 3 | 9 | 5 | 1,2,3,7,8 |
| 4 | 21 | 4 | 1,2,7,10 |
| 5 | 19 | 7 | 1,3,4,5,6,7,13 |
| | 75 | 23 | |

Summary results will be compared with the results of system summary manual created by the abstractor that can be seen in the following table:

**Table 8 The Summary Result From Abstractor**

| D | Jumlah Kalimat | Jumlah Ringkasan Manual | Kalimat Ke- |
|---|---|---|---|
| 1 | 10 | 2 | 1,9 |
| 2 | 16 | 2 | 8,12 |
| 3 | 9 | 6 | 2,3,4,6,7,8 |
| 4 | 21 | 6 | 1,2,3,7,10,13 |
| 5 | 19 | 7 | 1,2,4,5,13,18,19 |
| | 75 | 23 | |

So if the comparison is done on the system at the summary results table 4.2 performance value obtained will then be as follows:

| D | Recall | Precision | F-Measure |
|---|---|---|---|
| 1 | 50% | 25% | 33.33% |
| 2 | 50% | 33% | 39.76% |
| 3 | 67% | 80% | 72.92% |
| 4 | 67% | 100% | 80.23% |
| 5 | 42% | 42% | 42% |
| Rata-Rata | 55.2% | 56% | 53.65% |

## 4. CONCLUSION

Based on the results of the research that has been done, it can be noted that the summary produced by the automatic text peringkas this is not to say either. However, the resulting summary indicates that the automatic text peringkas built this generates sentences summary results that are less in accordance with the summary results of a man. The parameter C is chosen from this study is 1 and the value of 0.5. The value of performance obtained the values recall amounted to 55.2%, the value of precision of 56% and the value of f-measure of 53.65%.

## 5. BIBLIOGRAPHY


[1] I. N. Sari, P. K. Angelinna and L. H. Imam, "Analisis dan Implementasi Peringkas Teks Otomatis Menggunakan Metode Maximal Marginal Importance(MMI)," 2011.

[2] M. L. K. B. R. T. Fitriaman Deni, "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Pada Multi-Document Menggunakan Metode Support Vector Machines (SVM)".

[3] Z. A. R. K. Mustaqhfiri M, "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance," 2011.

[4] T. F. Z, A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.

[5] E. N. H. S. Jamhari Muhammad, "Pengklusteran Dokumen Teks Hasil Peringkas Dokumen Otomatis Yang Menggunakan Metode Seleksi Fitur Dan Latent Sematic Analysis (LSA)," *Jurnal Teknologi Informasi, Volume 10 Nomor 1*, 2014.

[6] P. Eko, Data Mining Konsep dan Aplikasi Menggunakan Matlab, Yogyakarta: Andi, 2007.

[7] S. B, Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis, 2007: Graha Ilmu,

Yogyakarta.

[8] Aristoteles, Pembobotan Fitur Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritme Genetika, 2011.

[9] Z. B. d. E. Winarko, "Analisis Fitur Kalimat untuk Peringkas Teks Otomatis pada Bahasa Indonesia," *ournals of Computing and Cybernetics Systems 5 (2);60-68,* 2011.