IMPLEMENTATION OF METHOD OF GENERALIZED VECTOR SPACE MODEL (GVSM) BY LESK ALGORITHM ON INFORMATION RETRIEVAL SYSTEM

Abdurrahman Auliya Fatahillah[1], Ednawati Rainarli, S.Si., M.Si.[2]

[1,2] Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : thanxs.rahman@gmail.com[1], irene_edna@yahoo.com[2]

ABSTRAK

By the time, searching system is getting needed by people who wants relevant data inside the information source. Then, Information Retrieval System is built in order to find data and information within the information source. Generalized Vector Space Model (GVSM) is one of information retrieval systems and also built to seek the relevant data within the information source.

Sometimes, what we search is irrelevant with what we want. It is because of certain problem like ambiguous searching keyword. One of solutions to optimalize this problem is by using lesk algorithm. Lesk algorithm is an algorithm which is used to find the meaning of one word that linked with the main keyword. In other word, it separates unsubmitted word in the query but included in the searching keyword.

A research shows that adding lesk algorithm in the information retrieval system can be implemented and worked well. Lesk algorithm can work as it needed on the information retrieval system to decrease – even vanishing – ambiguity. Based on the study of 30 documents, the value of recall and precision without using an algorithm lesk for 94.736% and 78.26%, while using an algorithm lesk of 100% and 82.608%

Keywords : Information Retrieval System, Generalized Vector Space Model, Algoritma Lesk, Keyword, Query.

1. INTRODUCTION

Information Retrieval System is built in order to find relevant data or information in the information source. *Generalized Vector Space Model* (GVSM) is one of Information Retrieval Systems. Information retrieval system is close with representation, saving, and file access system, and the target file must be relevant with the needed information as written in the query.

In several cases, what we search is irrelevant with what we need. It is because of several causes like ambiguous searching keyword[13]. One of solutions to optimalize this problem is by using lesk algorithm. Lesk algorithm is an algorithm which is used to find the meaning of one word that linked with the main keyword. In other word, it separates unsubmitted word in the query but included in the searching keyword.

This research shows the effectiveness of lesk algorithm in searching ambiguous word or sentence. The purpose of this research is to be the final assignment. The topic of this paper is "IMPLEMENTATION OF METHOD OF GENERALIZED VECTOR SPACE MODEL (GVSM) BY LESK ALGORITHM ON INFORMATION RETRIEVAL SYSTEM"\

1.1 *Information Retrieval* (IR)

Information Retrieval System works in order to find data in unstructuralized data source in a computer[5]. The purpose of this system is to fulfill user's required information by retrieving all maybe-relevant documents, which is in the same time, retrieving irrelevant document as less as possible. A good Information Retrieval System can quickly determine the result. The aim then is how to arrange the result in order from the most relevant to lesser relevant.

Objectives to be met is how to put together a document which has been obtained from documents has become the featured sequences that have a high relevance to a lower level of relevancy. Preparation of the document referred to as the ranking of the document.

1.2 *Generalized Vector Space Model* (GVSM)

Generalized Vector Space Model (GVSM) is an advance system of *Vector Space Model* (VSM) which is works by adding extra information aside of terms, in representating document[6]. Information Retrieval System in GVSM representating document with vector similarity in all available documents.

Wong (1985) gives GVSM as alternative in Information Retrieving System.Short description of GVSM also written by Carbonell. Carbonell defines

the terms in VSM are linearly independent, and it avoids assumptions with using documents as basic of vector in terms. In *Dual Space*, a document is represented by vector which is its dimension refers to a document.

GVSM algorithm works by using vector space concept. Input from user and documents are transcripted into vectors. After that, those vectors is processed and the result is the reference of query relevance towards document.

There are several steps to get the result of input query.

1. Deleting preposition and conjunction
2. Using stemmer in documents and query
3. Determining miterm to define word frequency pattern

$M_1 = (0,0,0)$
$M_2 = (1,0,0)$
.....
$M_n = (0,0,0)$

4. Counting the appearing word frequency in a document referring to the query
5. Counting index term

$$\overrightarrow{Ki} = \frac{\sum_{\forall r.gi(Mr)=1} Ci,r.\overrightarrow{Mr}}{\sqrt{\sum_{\forall r.gi(Mr)=1} Ci,r^2}} \quad \ldots. (1)$$

Where :
$\overrightarrow{Ki}$: term index ke-i
$\overrightarrow{Mr}$: Orthogonal vector corresponding minterm pattern applicable
Ci,r : Factors correlation between the index of the i-th term with minterm r
While the correlation factor as follows:

$$C_{i,r} = \sum_{dj|gi(\overrightarrow{dj})=gi(Mr)} W_{i,j} \quad \ldots. (2)$$

Where :
$W_{i,j}$: Document vector-j
$gi(Mr)$: The weight of the index term in minterm Mr Ki

6. Change the document and query vectors

$$\overrightarrow{d_j} = \sum_{i=1}^{n} W_{i,j} \times \overrightarrow{K_t} \quad \ldots. (3)$$

$$\vec{q} = \sum_{i=1}^{n} q_i \times \overrightarrow{K_i} \quad \ldots. (4)$$

Where:
$\overrightarrow{d_j}$: Document vector-j
$\vec{q}$: *Query* vector
$W_{i,j}$: term weight i to document j
q_i : term weight for i
n : the number of index terms

7. Sort the documents by similarity, by calculating the vector multiplication

$$Sim(\overrightarrow{d_j}.\vec{q}) = \frac{\overrightarrow{d_j}.\vec{q}}{|\overrightarrow{d_j}|\,|\vec{q}|} \ldots. (5)$$

Where :
$\overrightarrow{d_j}$: Document vector-j
$\vec{q}$: Query vector

1.3 Algoritma Lesk

Lesk algorithm is an algorithm which function is to decrease ambiguity. Lesk algorithm is one of implementations of unsupervised approaching method. This system works by "intuition" that ambiguous word can be used to refer to the same topic in a search. Lesk algorithm have *relatedness* function which is turnback input query.

Lesk algorithm is an implementation of the approach or the unsupervised neural network is used to remove the ambiguity in the meaning of the word. This algorithm is based on the intuition that the ambiguous words that are contained together in a sentence, is used to refer to the same subject and meaning associated with that topic is defined in the dictionary by using the same words. Lesk algorithm has the function of relatedness (connectedness) will return the number of words that overlap between the definitions of the two words being entered.

1.4 *Preprocessing*

Preprocessing is a process in GVSM which is having two phases; first is reading text by using *indexing*, *tokenizing*, *filtration*, *stop word removal*, and *stemming*, second is counting relevantness in preprocessed documents with user's query, this process also counting frequeny of word in the document.

Query entered by the user and the document is translated into vectors then the vectors are subjected to multiplication operations and the results become a reference point in determining the relevancy of user input (query) against the document.

1.4.1 *Indexing*

Indexing is searching relevant information by stacking word referring to a page of document. By using indexing, user can easily find information they want. The elements of indexing is index term required from text inside a document. Another element is term weighting as ranking system of relevant document which is has a same term.

The index is the language used in a conventional book to find information based on words or terms that refer to a page. Using the index, information seekers can easily find the information he wants. In the information retrieval system, the index is later used to represent information in a document.

Elements of the index is the term index (index terms) arising from the text that is broken down in a document. Other elements are weighted term (term weighting) as the relevant criteria for determining the rank of a document that has the same term.

Baeza-Yates and Ribeiro-Neto [7] describes the process of making the index from a text document, known as the process of text analysis (automatic text analysis) through several stages:

a. The removal process digits, hyphens, punctuation and uniformity of the letters are used.
b. Screening involves the removal of the word instead of a keyword which is called filtering or stopword removal.
c. Enforced affix the word, either prefix or suffix the word. Disappearance additive is known by the word stemming.
d. Selection of terms to define a word or stem (group of words) which will be used as an element of the index.
e. The formation of such groups category term structured equation of words used for the expansion of basic query provided by the user retrieval of information with other appropriate term.

Indexing can be done in two ways: manual and automatic. Ideally, to get the perfect term index an indexing is done manually (conventional). However, according to the Salton [7] systems search and text analysis did not produce a fully automated retrieval performance is worse than conventional systems that use manual document indexing and search formulation manual.

1.4.2 *Tokenizing*

Tokenizing is “dismembering” sentence into parts. Before the index compared with the document, tokenization is done beforehand, ie chopping the sentence into its parts. For example, "he called Rahman", after the phrase entered the token task is to break down into its parts into a "he", "name", "rahman".

Seen from the example above contained the phrase "he called Rahman" then the tokenization process is done by breaking the word in the sentence into three fractions which he named, and Rahman.

1.4.3 *Filtering*

In a document there are many words that are not keywords within documents or extra words just to connect the word, for example, is conjunctive and there are also signs read. In the process of indexing is done the process to eliminate words to reduce the indexing process early and reduce the words and punctuation that will not directly related to the keywords.

In addition to indexing process mengungari the process is done so that the application of document similarity calculation with the documents sought are kesesuaikan due to reduced noise conjunctions and punctuation if not removed will be entered into the calculation.

1.4.4 *Stemming*

Stemming an integral part in Retrieval System (Information Retrieval System). Algorithms Nazief & Adriani as stemming algorithms for Indonesian-language text that has the capability percentage of accuracy (precision) better than other algorithms. This algorithm is needed and decisive in the process of preprocessing in the document Indonesia.

Stemming is one of the ways used to improve the performance Retrieval System (Information Retrieval System) by means of transforming the words in a text document to form the basic word. Stemming algorithms for language differs from stemming algorithms for other languages. For example, the English language has a different morphology with Indonesian so that stemming algorithms for the two languages are different. The process of stemming the Indonesian-language text more complicated / complex because there are variations affixes that must be removed to get the word basic (root word) of a word. In general, basic words in Indonesian consists of a combination of:

Prefix 1 Prefix 2 + stopwords + base + 3 + Suffix Suffix 2 + Suffix 1

Algorithms Nazief & Adriani created by Bobby Nazief and Mirna Adriani has the following stages:

1. First find words that will diistem dictionary basic words. If found, the word is the root word is assumed, then the algorithm stops.
2. Inflection Suffixes ("that", "-kah", "me", "thy" or "him") removed. If the form of particles ("that", "-kah", "-tah" or "-pun") then this step is repeated to remove possesive pronouns ("me," "thy," or "him") , If there are.
3. Clear Derivation Suffixes ("-i", "late" or "-kan"). If the word found in a dictionary, then the algorithm stops. If not then to step 3a

a. If the "-an" has been removed and the last letter of the word is "-k", hence the "-k" also deleted. If the word is found in the dictionary then the algorithm stops. If not found then go to step 3b.
b. Suffix is removed ("-i", "late" or "-kan") is returned, go to step 4.

4. Clear Derivation Prefix. If in step 3 the suffix is removed then go to step 4a, if not go to step 4b.

a. Check the table prefix-suffix combination that is not permitted. If found then the algorithm stops, otherwise
b. For i = 1 to 3, specify the type of the prefix and then remove the prefix. If the root word has not been found steps 5, if it is then the algorithm stops. Note: if the same prefix with a prefix of the first algorithm stops.

5. Recoding.
6. If all steps have been completed, but that does not work then the initial word is assumed to be the root word, the process is complete.

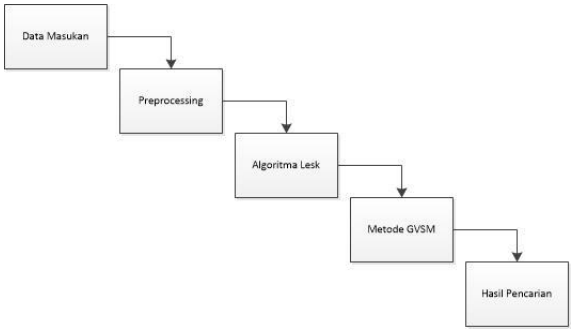
Type prefix is determined through the following steps:

1. If the prefix is: "be", "in-", or "a" then type the prefix in a row is the "in-", "in-", or "a".
2. If the prefix is "neighbor", "him", "some" or "PE" then it takes an additional process to determine the type of the prefix.
3. If the first two characters instead of "was", "in-", "a", "neighbor", "some", "him", or "PE" then stop.

If the type of the prefix is "none" then stop. If the type of the prefix is not "none" then remove the prefix if found.

2. RESEARCH CONTENT

This part shows the analysis of method in implementation of GVSM by using lesk algorithm in Information Retrieving System. The process can be observed in Picture 2.1.



Picture 2.1. Main System Process

2.1 Input Data

There are two kinds of input, first is query in Bahasa results based on the Ministry of Education and Culture 27 August 1975 Number 0196/U/1975[5]. second is by documents inside a computer then using text extraction with library on .net, that is *Microsoft.Office.Interop.Word*.

Example, there is a query (Q document 1 (D1), document 2 (D2), the document 3 (D3), the document 4 (D4), the document 5 (D5) as if:

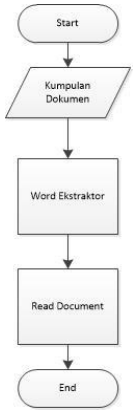
Q : Faktor kepala cabang dalam mempengaruhi kinerja karyawan
D1: UNIKOM_AI KARTINI_BABIII
D2: UNIKOM_FERY TRI LAKSANA_BAB2
D3: UNIKOM_Fujiutama_Bab 2
D4: UNIKOM_Putri Famawati_Abstrak
D5: UNIKOM_Wupi Ocktavia K_Bab 5

2.2 Preprocessing

At this stage, the data that has been entered will be done preprocessing which consists of reading text (* .doc) with tokenizing, filtration, and algorithms stemming lesk.

1. *Reading text*

At this stage, reading text using multi-threaded methods to improve system speed reading documents in the same way. Here are the steps to make reading text on the document can be seen in Picture 3.2. below this:



Picture 2.2. Flowchart *Reading Text*

2. *Case Folding*

In this process of checking the capitals that are in each sentence. If found the capital letters, it will be lowercase, that is, change to lowercase. Here are the steps necessary to perform case folding on the document and query can be seen in Picture 2.3. below this:



Picture 2.3. Flowchart *Case Folding*

In this case, the query is converted to lowercase become "faktor pemimpin dalam mempengaruhi kinerja karyawan".

3. *Tokenizing*

In this process the removal of punctuation and numbers. After the process, the document is broken down into tokens by cutting into a word (term). Here are the steps to perform tokenizing the document and query can be seen in Picture 2.4. below this:



Picture 2.4. Flowchart *Tokenizing*

In this case, the query is divided into six parts contained in Table 2.1.

Table 2.1. *Tokenizing Query* Results

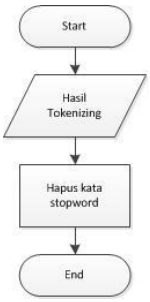
faktor
kepala
cabang
dalam
mempengaruhi
kinerja
karyawan

4. *Filtering*

Filtering process is a process of removing the words are not important are the results of tokenizing. To perform filtering can use the stoplist or word list or stopword. The data will be compared with the results of tokenizing a dictionary, if in the dictionary then the word will be deleted. The remaining words are the important words. For more details filtering process steps are as follows:

1. The word tokenizing process results compared with word filtering / stopword.
2. If the data is the same as the word tokenizing result in stopword table will be deleted.
3. If it is not the same as the table 2.1. said filtering / stopword then the word will be saved.

Here are the steps to perform tokenizing the document and query can be seen in Picture 2.5. below this:



Picture 2.5. Flowchart *Filtering*

In this case, the word "the" include both stopword the word "in" is deleted. Table 3.2. shows changes in query results stopword.

Tabel 2.2. query results *stopword*

faktor
kepala
cabang
mempengaruhi
kinerja
karyawan

5. *Stemming*

After the filtering process, documents and queries are entered into the process of stemming. Stemming process that removes some of the front and rear so that the words be a basis. The author uses Stemming Algorithm Indonesian Nazief and Adriani. For basic word author took from Big Indonesian Dictionary (KBBI).

In this case, there is the word "mempengaruhi" that has affix mem- and -i into effect. Table 3.3. shows the changes that have been in stemming the word.

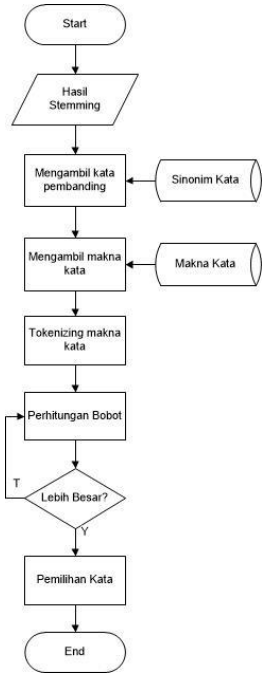
Table 3.3. results stemming

faktor
kepala
cabang
pengaruh
kinerja
karyawan

2.3 Lesk Algorithm

After preprocessing process, then the next stage to optimize the keywords (queries) so unambiguous that uses algorithms lesk. Lesk algorithmic process which compares the meaning of words in comparison with the meaning of the word input (query) to find the right words synonymous with the query. The whole meaning of the word took on a large dictionary Indonesian website and to said comparator is taken from Indonesian synonyms website. For more details, stemming process steps are as follows:

1. Picking stemming query result
2. Determine the synonym of a query which will be a benchmark
3. Taking the meaning of words from the query and said comparator
4. Conducting the process of tokenizing on the meaning of the query and said comparator
5. Calculate the weight of said comparison by comparing the meanings of words with the meaning of the word query comparison
6. Choosing the comparison is based on the weight of the greatest Here are the steps to make the process lesk on query algorithms can be seen in Picture 2.6. below this:



Picture 2.6. *Flowchart* Lesk Algorithm

In this case there are six queries that will be compared with the comparison words. Lesk algorithmic process can be seen in Table 2.4.

Table 2.4. algorithms lesk

Kata (*query*)	Makna	Kata Pembanding	Makna	Bobot
Kepala	bagian tubuh yang di atas leher (pada manusia, beberapa jenis hewan merupakan tempat otak, pusat jaringan saraf, dan beberapa pusat indra)	akal	daya pikir, jalan cara melakukan sesuatu, daya upaya, ikhtiar	1
	pemimpin, ketua (kantor, pekerjaan, perkumpulan)	pemimpin	orang yang memimpin	2

Based on the calculation algorithm lesk, said (query) "kepala" has two said comparator is "akal" which has a weight of 0 and a "pemimpin" who has a weight of 2, then said comparator taken as a result of the calculation algorithm lesk is a leader because it has value greater weight. Results from lesk algorithm will be added to the query so that more optimal results. Table 2.5. is the result of the calculation algorithm lesk

Table 2.5. lesk algorithm results

aspek
pemimpin
filial
akibat
prestasi
buruh

2.4 *Generalized Vector Space Model* (GVSM)

There are several steps or processes to obtain the results of the query is entered, called an algorithm Generalized Vector Space Model [6]:

1. Throw prepositions and conjunctions.
2. Using Stemming the document and query, the application used to eliminate affixes (prefixes, suffixes). Example: handsome: handsome, error: wrong.
3. Determine minterm to determine possible patterns of word frequency. Long minterm is based on a lot of words that is inputted to the query. Then converted into orthogonal vectors according to minterm pattern emerging.
4. Calculate the number or frequency of occurrence of the word in the document that match the query
5. Calculate the index term
6. Change the document and query into a vector
7. Sort the documents by similarity, by calculating the vector

2.4.1 Generalized Vector Space Model (GVSM) by Using Lesk Algorithm

Table 2.6. The results of calculations GVSM by using lesk algorithm

Document	Similarity Weights
D1	0.999702951479197
D2	0.986850140318568
D3	0.913581007337747
D4	0
D5	0

Based on the results of similarity between documents by querying it can be concluded that the sequence of documents relevant to the query is:

1. Document 1 (D1) = 0.999702951479197
2. Document 2 (D2) = 0.986850140318568
3. Document 3 (D3) = 0.913581007337747
4. Document 4 (D4) = 0
5. Document 5 (D5) = 0

Since the value similiaritas document 2 is larger than the value other then similiaritas documents ($Sim(\overrightarrow{d_1}.\vec{q}) > Sim(\overrightarrow{d_2}.\vec{q}) > Sim(\overrightarrow{d_3}.\vec{q}) > Sim(\overrightarrow{d_4}.\vec{q}) > Sim(\overrightarrow{d_5}.\vec{q})$). Based on the case can be concluded that the Generalized Vector Space Model (GVSM) calculates the correlation between queries and documents by counting all term used orthogonal

vectors to calculate the Index term and after that every term in the document generalized to vector orthogonal by multiplying the result of index term to term document and query, then each of the charged vector multiplication operation and the results become a reference point in determining the relevance of input (query) against the document.

2.4.2 *Generalized Vector Space Model* (GVSM) without Lesk Algorithm

Table 2.7. The results of calculations without lesk algorithm

Document	Similarity Weights
D1	0.816474545780421
D2	0.996230230720323
D3	0.653918516245756
D4	0
D5	0

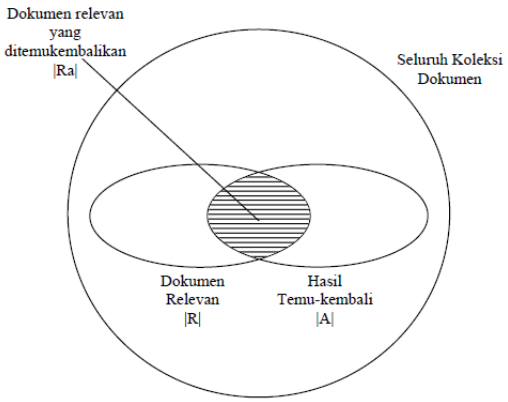
Based on the results of similarity between documents by querying it can be concluded that the sequence of documents relevant to the query are:

1. Document 2 (D2) = 0.996230230720323
2. Document 1 (D1) = 0.816474545780421
3. Document 3 (D3) = 0.653918516245756
4. Document 4 (D4) = 0
5. Document 5 (D5) = 0

Since the value similiaritas document 2 is larger than the value other then similiaritas documents ($Sim(\vec{d_2}.\vec{q}) > Sim(\vec{d_1}.\vec{q}) > Sim(\vec{d_3}.\vec{q}) > Sim(\vec{d_4}.\vec{q}) > Sim(\vec{d_5}.\vec{q})$). Based on the case can be concluded that the Generalized Vector Space Model (GVSM) calculates the correlation between queries and documents by counting all term used orthogonal vectors to calculate the Index term and after that every term in the document generalized to vector orthogonal by multiplying the result of index term to term document and query, then each of the charged vector multiplication operation and the results become a reference point in determining the relevance of input (query) against document.

2.5 *Recall* and *Precission* Test

Evaluation from Information Retrieval System is conducted by two main parameters, recall and precision. Recall is ratio between relevant document inside a system, and precision is ratio of relevant document which is successfully retrieved from all documents.



Picture 2.7. *Recall* dan *Precision*

Based on Picture 2.7. recall and precision can be expressed as following:

$$Recall = \frac{|Ra|}{|R|}$$

$$Precision = \frac{|Ra|}{|A|} \quad \text{..... (6)}$$

To test the precision and accuracy of applications, testing is done by calculating the recall and precission starting from documents that were 10 to 69.

Table 2.8. Results Recall and Precision

No.	Document Value	Using Lesk Algorithm		Without Lesk Algorithm	
		Recall (%)	Precission (%)	Recall (%)	Precission (%)
1.	10	100	60	100	60
2.	20	100	65	100	65
3.	30	100	80	79.166	63.333
4.	40	100	82.5	96.96	80
5.	50	100	94	95.744	90
6.	60	100	85	98.039	83.333
7.	69	100	82.608	94.736	78.26

3. CONCLUSION

According to the research, adding lesk algorithm in Information Retrieval System can be implemented and can remove ambiguous words in query. Lesk algorithm can work in order as input of Information Retrieval System.

The value generated by the algorithm lesk influence on Similarity Weights. Based on the test in the previous chapter, using algorithms lesk Similarity Weights for. Not only Similarity Weights were increased, but the value of recall and precission increased, too. In the folder that contains 30 Document produce recall value and precission without using algorithms lesk for 94 736% and 78.26%, while the use of algorithms lesk of 100% and 82 608%. Lesk seen that the algorithm can improve the accuracy of search Document well.

Based on the results of research, analysis, design through to the manufacture of implementation methods Generalized Vector Space Model (GVSM) using an algorithm lesk on Retrieval System (Information Retrieval System), it can be given the following advice:

1. Stages of preprocessing such as tokenizing, filtering, stemming, and the addition of a relatively long lesk algorithm is expected to be optimized back.
2. Completeness of databases such as WordNet meenggunakan Indonesian as a reference in eliminate ambiguity, but to WordNet Indonesian until now exist and are still being developed.
3. Taking the meaning of words and synonyms that can be done automatically by the system
4. For further development, can be equipped with an ambiguous word in the form of an ambiguous phrase, word berimbuhan ambiguous.
5. The test can be performed on the Document collection more.

BIBLIOGRAPHY

[1] Zainal A. Hasibuan , Yofi Andri : Penerapan Berbagai Teknik Sistem Temu-Kembali Informasi Berbasis Hiperteks, Fakultas Ilmu Komputer, Universitas Indonesia
[2] Chairul Furqon, Konsep Informasi (diakses tanggal 5/10/2015)
[3] http://widuri.raharja.info/index.php?title=BAB_II_ROOSTER_(KKP) diakses tgl 5/10/2015 pukul 20.26
[4] Sahirul Alim Tri Bawono, *Information Retrieval* Meningkatkan Pencarian Data yang Relevan, Universitas Gadjah Mada.
[5] Jasman Pardede, Mira Musrini Barmawi, Wildan Denny Pramonor "IMPLEMENTASI METODE GENERALIZED VECTOR SPACE MODEL PADA APLIKASI INFORMATION RETRIEVAL", Institut Teknologi Nasional Bandung 2013
[6] Baeza, Ricardo, B. Ribeiro. 1999. *Modern Information Retrieval*. ACM Press. United States of America. 1999.
[7] Nazief, B. d. Approach to Stemming Algorithm. *Confix-Stripping*.
[8] Salton, G. (1989). *Automatic Text Processing,The Transformation, Analysis, andRetrieval of Information by Computer*. United States of America: Addison – Wesly Publishing Company,Inc. All rights reserved.Nadirman, S, 2006,
[9] Sistem Temu-Kembali Informasi dengan Metode Vector Space Model pada Pencarian File Document Berbasis Teks, Yogyakarta, Universitas Gadjah Mada.
[10] Jasman Pardede, IMPLEMENTASI MULTITHREADING UNTUK MENINGKATKAN KINERJA INFORMATION RETRIEVAL DENGAN METODE GVSM", Jurnal Sistem Komputer Vol. 4 No. 1, Mei 2014
[11] Abdul Rouf, Pengujian Perangkat Lunak Dengan Menggunakan Metode White box dan Black box, Sistem Informasi – STMIK HIMSYA Semarang
[12] SQL Server Official Site (www.microsoft.com/sql/default.asp) diakses tanggal 28/11/2015
[13] Stanjeev Banerjee and Ted Pedersen, "An Adapted Lesk Algorithm For Word Sense Disambiguition Using WordNet", University of Minnesota, Duluth, MN 55812 USA