Semantic Space for Document Similarity Test

Kemal Ade Sekarwati
Information System
Gunadarma Unversity
Depok, Indonesia
ade@staff.gunadarma.ac.id

Lintang Yuniar Banowosari
Information Management
Gunadarma University
Depok, Indonesia
lintang@staff.gunadarma.ac.id

I Made Wiryana
Information System
Gunadarma Unversity
Depok, Indonesia
mwiryana@staff.gunadarma.ac.id

Djati Kerami
Information Technology
Gunadarma Unversity
Depok, Indonesia
djati@staff.gunadarma.ac.id

***Abstract* — the study of Indonesian language document similarity test has been conducted by several researchers which is done by using algorithm Karp Rabin and string matching. Documents used in previous studies using the same type of document. Studies that tested different types of document similarity using semantic space are limited. This study examined the similarity of Indonesian language documents in the field of education using semantic space including automated essay grading and final report.**

The system developed is to measure the similarity existing document with other documents that have been stored in an internal database. Similarity calculation results in the form of a document which compared the percentage of similarity. The data to test the approach are scientific writing of computer science or information technology student of Gunadarma University. Document similarity algorithm testing done with the steps: (i) the input documents, (ii) pre-processing, (iii) the calculation of the term frequency-inverse document frequency (tf-idf), (iv) the calculation of latent semantic indexing (LSI), and (v) document similarity calculation using three types of calculation that cosine similarity resemblance, similarity dice, and Jaccard similarity.

The numbers of documents to be tested are 30 documents. Results of testing performed to compute the similarity of documents on a combination of three types of documents. Tested combination of documents, which are: (i) 30 pairs of the same document, (ii) five pairs of similar documents, and (iii) 30 pairs of documents that are not the same. The test results for the same document to produce a percentage of 100% similarity to similarity measurement using the cosine similarity, dice similarity, and Jaccard similarity. Tests on similar documents to produce a percentage of 83.69% -100% similarity to the cosine similarity, similarity dice produces 14.44% -100%, and Jaccard similarity yield of 8.52% -100%. Tests on a document that does not produce the same percentage of 28.63% -97.40% similarity to the cosine similarity, similarity dice produces 9.55% -39.02%, and Jaccard similarity yield 0.75% -21.41%. This study has shown that the algorithm can be used to test the similarity of documents.

Keywords—document; similarity; tf-idf; latent semantic indexing;similarity testing

I. INTRODUCTION

Document similarity problem has existed for a long time but with the advancement of information technology problem becomes worse. This is because there are a lot of electronic versions of the materials available to everyone. Web is an important and common source for document similarity. Some programs such as Turnitin document similarity detection [Lukashenko, Romans., Vita Graudina, and Grundspenkis, Janis., 2007] was developed to address this problem. To determine whether an article is copied from electronic sources or other web, document similarity detection should calculate the similarities between the two articles. It is often difficult to detect accurately document similarity after the article content is modified. For example, it is possible to simply replace a word with a synonym (eg "Program" with the "Software") and change the entire structure of the sentence.

Research on Indonesian language document similarity measurements in the field of education have already existed, which was from [Ana Kurniawati 2010], she used string matching method with the new algorithm which is an algorithm to determine the structure of sentences and calculate document similarity with synonyms factor. The object is a document used in Indonesian language. In her research, the data used was abstracts from scientific writing of student in computer science and news articles taken on-line or internet media. The study of news articles can only be performed for one page, to the next page is not successfully performed.

Indonesian-language document similarity detection is needed, especially in academic environments such as schools, universities and institutions to check the assignment or the results of research and analysis to it. This study focused on measuring similarity Indonesian language documents using Latent Semantic Analysis (LSA). The system developed is to measure the similarity between two documents is inputted into the system. Documents to be compared in the form of a text file consisting of letters and numbers, does not include images, tables, and formulas. The comparison process is done by using the term frequency-inverse document frequency (tf-idf) that calculates the weight of a frequency of occurrence of the word in the document. Tf-idf calculation results in the form of a matrix value. The value of the emergence of a matrix is the number of words in each document being compared. Document similarity calculation is taken from the value of the resulting matrix.

Referring to previous research opportunities that need to be developed that is document similarity measurements Indonesian language by using the LSA method, so the problem statement are: How to develop an algorithm to detect Indonesian language document similarity using semantic space? How to build architecture to detect Indonesian language documents similarity using semantic space? How to build a system to test the Indonesian-language document similarity using semantic space?

Document similarity detection is performed using scientific writing documents of computer science and or information technology area. Scientific writing documents stored in clear text and stored in a single folder on a local storage media. In this study, we do not discuss basic word searching and word stemming.

This template, modified in MS Word 2007 and saved as a "Word 97-2003 Document" for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

II. Related work

A. Selecting a Template (Heading 2)

Measurement of similarity between words is a fundamental part of the text similarity which is then used as the main stage for the sentences similarity, paragraphs similarity and documents similarity. Words can be said to be similar lexical and semantic. Words said to be similar in lexical if it has a similar sequence of characters. Words that are semantically similar if these words had the same thing are used in the same way, in the same context, and one word is a type of other words. The Lexical similarity algorithm used String-Based whereas the semantic similarity algorithm consists of Corpus-Based and Knowledge-Based [Gooma, Wael H., and A. Aly Fahmy, 2013].

Research to measure the similarity of documents already exist, such as is done by Saul Schleimer 2003 [Schleimer, S., et al, 2003] using fingerprint. Hari Bagus Firdaus 2003 [Hari Bagus Firdaus, 2003] using Karp Rabin algorithm in his research. Pavarti Iyer 2005 [Iyer, P., and Sing, A., 2005] conducted the research by using keywords or keyword similarity. Ana Kurniawati 2010 [Ana Kurniawati 2010] using the method of string matching to develop the new algorithm which is an algorithm to determine the structure of sentences and calculate document similarity with synonyms factor. The object used is a document in Indonesian language.

There is some software that is used to detect the similarity of documents, including [Zaka, Bilal., 2009]:

a. Turnitin

Turnitin is a product of iParadigms. Turnitin is a web-based service that detects and processes the document by distance. Users upload documents suspected to database system then the system makes a complete fingerprint documents and store them. Source database composed of archives are indexed internet, books, journals contained in the database ProQuest, and the documents sent to the Turnitin database.

b. Eve2 (Essay Verification Engine)

Eve2 working on the client side and uses the internet search mechanism to find content that has some similarities to the suspect documents. The result is a report on the identification of similarities was found on the World Wide Web.

c. CopyCatch

Copycatch device-based client used to compare documents locally. Copycatch consists of two versions of the gold version and the version of the campus, the difference lies in the number of local sources. CopyCatch also provides a web version that has the ability to detect similarities in the internet by using the Google API.

d. Wcopyfind

Wcopyfind software is open source device to detect a sentence or phrase that is stored in a local document storage. Then this product was developed ability to search on the internet by using the Google API.

e. GPSP (Glatt Plagiarism Screening Program)

Software GPSP works locally. GPSP detection software is based on the type and pattern of writing. Faculty has students submit a work suspected of being plagiarized to the Glatt Plagiarism Screening Program, which is free standing, non-Web-based software. The program replaces every fifth word of the suspected paper with a standard size blank, and the student is then prompted to supply the missing words (Glatt, 2007). The number of correct responses, the amount of time intervening between responses, and various other factors are considered in calculating a plagiarism probability index.

f. MOSS (Measure of Software Similarity)

Software MOSS receive a set of documents and returning it in the form of an HTML page that shows the parts of a document pair are very similar.

g. Jplag

Jplag is an Internet service that is used to detect a similarity between the program source codes. Users upload files to be compared and report the results of its identification system. Jplag analyze the syntax and structure of the programming language. Each device has an attribute that specifies the application. Two main attributes that exist on these devices is a type of operation on the text and the type of operation in the corpus [Lukashenko, Romans., Et al, 2007]. Based on the attribute type of operation on the text, the device is divided into two groups: a device that operates on unstructured text (free) and devices that operate on structured text (source code). While this type of operation in the corpus are grouped into three parts: the device can only operate intra corpally, extra corpally, and devices that can operate on both the intra and extra corpally corpally. A device that only operates intra corpally that lies the source and copies of documents were in the corpus. A device that only operates corpally extra copies of the documents that the location was in the corpus, while the source of the document is outside the corpus. While the device is only operated intra and extra corpally corpally namely the location of sources and copies of documents inside and outside the corpus.

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the file "MSW_USltr_format".

B. *Maintaining the Integrity of the Specifications*

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. METHODOLOGY

Before the document is read by the system, the document must go through the stages of pre-processing and document conversion. Document conversion is made by changing the file format Print Document Format (PDF) format to a text file (.txt), while the pre-processing stage is done is do the removal of symbols and punctuation marks other than the dot contained in the contents of the document are detected. Results of the documents entered represented in the form of a matrix. Each matrix cell contains the number of words that appear in a document. The value of a matrix is the number the emergence of words in each document being compared. Document similarity calculation is taken from the value of the resulting matrix.

Similarity measurement process consists of four processes, namely:

1. The pre-processing
2. The calculation process of Term Frequency-Inverse Document Frequency (TF-IDF)
3. The calculation process of Latent Semantic Indexing (LSI)
4. The similarity calculation process

Pre-processing process consists of five processes, namely:

1. The process of create a list of sentences (text list)
2. The process of create a list of words (word list)
3. The process of converting uppercase to lowercase
4. The process of removing the words contained in the stop list that prepositions and conjunctions
5. The process of removing punctuation other than point (punctuation removal).

If a word or term appears in the document, then the value is a nonzero vector (non-zero). To calculate these values, is using a term known as the weight of term (term weight). One of the famous formulas is weighted term frequency-inverse document frequency (tf-idf). Definition of terms, depending on the problem, it can be said, or keywords or phrases that are longer. If the term is the word, the dimension a vector is the number of words in the vocabulary (the number of different words that appear in the corpus). Vector operations can be used to compare the documents with the query. The following formula is the formula of tf-idf:

Weight vector d document is:

$$V_d = [(w_{1,d}, w_{2,d}, \dots, w_{N,d})] \qquad (1)$$

$$w_{t,d} = tf_{t,d} \, . \, log \frac{|D|}{|\{d' \in D | t \in d'\}|} \qquad (2)$$

Note:

1. $tf_{t,d}$ = Term-frequency of the term t in the document d (local parameter)

2. $log\frac{|D|}{|\{d'\in D|t\in d'\}|}$ = Inverse document frequency (global parameter)

3. $|D|$ = Total number of documents from the set of documents

4. $|\{d'\in D|t\in d'\}|$ = Number of documents that contain the term t.

At this stage, we do the conversion of the reference corpus to mm corpus, and then calculate the tf-idf corpus.

Corpus tf-idf generated in the previous step is converted into the form of a matrix, and then the matrix tf-idf converted into matrix form LSI. At this stage, conducts the process of create a semantic space. Semantic space is a matrix-word document created using Singular Value Decomposition (SVD) to reduce the dimensionality of the original document word matrix made from the corpus. The original parse SVD matrix X, into the product of three new matrices namely W, S, and P. S is a diagonal matrix containing the singular values. Singular value of X is the square root XTX eigenvalues are arranged in order of decreasing size [D. C. Lay, 1996].

The resulting semantic space can be used to find similarities between the two documents. To compare the two documents can be created vectors for each document, and use as a vector between the cosine similarity values. A document vector is found by adding the vector to every word that is found in the document. This means that if a word is missing from the semantic space, it will not affect the value of the document semantics. Broad corpus is very important to have since broad corpus includes certain subjects to get the best results. Each vector will have the same length that would equal the number of the documents in the semantic space.

At this stage, the process of comparing documents sentences examined by the phrase contained in other documents. Comparisons are made to the words contained in the first sentence in the document or are examined in comparison with the words contained in a sentence in the document 2. The results of the calculation of the percentage of similarity is in the form of a document similarity. In the present study tested the similarity measurement documents using the formula Dice's Similarity, Jaccard Similarity, and Cosine Similarity. The following variables are used in the calculation of document similarity with Dice and Jaccard's Similarity Similarity [Zhang, J., Yunchuan Sun, Wang Huilin, and Yanqing He., 2011]:

s_a = Sentence with the length of m (m ≥ 2)

s_b = Sentence with the length of n (n ≥ 2)

$s_a = w_{a1}w_{a2}w_{a3} \dots w_{am}((m \geq 2)$

$s_b = w_{b1}w_{b2}w_{b3} \dots w_{bn}((n \geq 2)$

Note:

1. $w_{ai}(i \in [1,m])$ and $w_{bj}(j \in [1,n])$ = word or separator at s_a and s_b.
2. $w(s_a)$ = group of word which consist of all words $w_{ai}(i \in [1,m])$
3. $w(s_b)$ = group of word which consist of all words $w_{bj}(j \in [1,n])$.

1. Dice's Similarity

$$Dice(s_a, s_b) = \frac{2|w(s_a)\cap w(s_b)|}{|w(s_a)|+|w(s_b)|} \quad (3)$$

Formula note:

s_a = Sentence with the length of m (m ≥ 2)

s_b = Sentence with the length of n (n ≥ 2)

$w(s_a)$ = group of word which consist of all words $w_{ai}(i \in [1,m])$

$w(s_b)$ = group of word which consist of all words $w_{bi}(j \in [1,n])$

2. *Jaccard Similarity*

$$Jaccard\,(s_a, s_b) = \frac{|w(s_a)\cap w(s_b)|}{|w(s_a)\cup w(s_b)|} \quad (4)$$

Formula note:

s_a = Sentence with the length of m (m ≥ 2)

s_b = Sentence with the length of n (n ≥ 2)

$w(s_a)$ = group of word which consist of all words $w_{ai}(i \in [1,m])$

$w(s_b)$ = group of word which consist of all words $w_{bi}(j \in [1,n])$

3. Cosine Similarity :

To calculate the similarity of sentences based on word vectors, vector words of sentences built earliest.

If the word on $w(s_a)$ and $w(s_b)$ as weights, s_a and s_b can be represented as bags of words. Vector of two sentences as follows:

$$v(s_a) = \{(w_1, w_{a1}), (w_2, w_{a2}), \dots, (w_{i+j}, w_{a(i+j)})\}$$

$$v(s_b) = \{(w_1, w_{b1}), (w_2, w_{b2}), \dots, (w_{i+j}, w_{b(i+j)})\}$$

$$Cosine\ (s_a, s_b) = \frac{\sum_{k=1}^{i+j} w_{ak} w_{ab}}{\sqrt{\sum_{k=1}^{i+j} w_{ak}^2}\sqrt{\sum_{k=1}^{i+j} w_{bk}^2}} \quad (5)$$

Cosine of two vectors can be obtained using the Euclidean dot product formula:

$$a \,.\, b = \|a\| \, \|b\| \, \cos\theta \quad (6)$$

There are two attributes of vectors, namely A and B, cosine similarity cos(0) is represented by using a dot product and magnitude:

$$similarity = \cos\theta = \frac{A \,.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \, x \, B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \, x \, \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (7)$$

IV. Result and Discusion

Stages of testing consisted of test preparation, test scenarios, the determination of documents for testing.

Data preparation is to collect data to be used for testing. The data used to test as many as 100 documents abstract of students scientific writing of Informatics Engineering Faculty of Industrial Technology Gunadarma University. The number of sentences contained in the documents that were tested ranged from 3-44 sentences, the number of words ranging between 66-336, the number of unique words ranged between 23-221, entropy ranged from 3.98 to 4.27 and the size of the document that is used between 1 - 3 Kilobytes (Kb).

Scenario testing is performed as follows: 1. Determine the documents being tested. Documents used to test as many as 101 documents consisting of 100 abstracts document of scientific writing and 1 (one) document scientific writing theme.

2. Perform document similarity testing. The test is performed by comparing the first document with the second document. The purpose of this test is whether the algorithm calculates similarity properly and as intended. Tests conducted on the similarity:

a. Baseline document: testing is done by comparing the abstract document with a document containing the repeated word "word".

b. Document transposition is test carried out on documents that sentence is moved or changed the position of its sentence as much as two and three sentences.

c. Similar documents: that the testing is done by using the document abstract compare to the abstract of the same document, but there are some words are replaced with its synonyms.

d. The same document type. In the same document examination consists of two types of tests are tests on the same abstract document and testing of different abstract document.

e. Different type's document: tests performed by using a document in the form of a collection of sentences with a documents of collection of words. Examples of documents in the form of a collection of sentences are abstract of scientific writing while the document of a collection of words that is the theme of scientific writing.

The test results of the first group of documents that the testing of the baseline document similarity percentage yield of 0% for all measurements, meaning that the two documents do not have similarities because both documents is to have different content. The test results transposition document group produces a percentage similarity with Cosine similarity to the transposition of two sentences and the three sentences has increased the percentage. Similar test results document group produces Cosine similarity percentage of similarity with almost all of them approaching 100%. The test results document groups for testing different types of the same abstract document similarity percentage yield of 100%, while for different abstract document similarity percentages varied produce. The test results showed that the use of topic modeling two different types of documents, namely the document in the form of a collection of sentences and a collection of documents that can be tested resemblance words. From these results it can be concluded that the algorithm used in this study can examine documents resemblance to different types of documents, namely the document in the form of a collection of sentences and documents in the form of a collection of words.

References

[1] http://www.kamusbesar.com/35527/semantik, tanggal akses : 04 Februari 2013.

[2] Aiken, A., 1994, *MOSS : A System For Detecting Software Plagiarism, Stanford University,* Tanggal Akses 1 Februari 2012, <http://theory.stanford.edu/~aiken/moss/>.

[3] Alsmadi, Izzat., and Zakaria Issa Saleh., 2012, *Documents Similarities Algorithms for Research Papers Authenticity*, ICCIT.

[4] Ana Kurniawati dan I Wayan Wicaksana., 2008, *Perbandingan Pendekatan Deteksi Plagiarism Dokumen Dalam Bahasa Inggris*, Seminar Ilmiah Nasional, Universitas Gunadarma, Jakarta.

[5] Ana Kurniawati, 2010, *Algoritma Mengukur Kemiripan Dokumen Berbahasa Indonesia Dengan Faktor Sinonim*, Disertasi, Universitas Gunadarma.

[6] Anzelmi, Daniele, Domenico Carlone, Fabio Rizzello.,et al., 2011, Plagiarism Detection Based On SCAM Algorithm, Proceedings Of The International MultiConference Of Engineers And Computer Scientists, Vol I, IMECS 2011, March 16-18.

[7] Alzahrani, Salha, Naomie Salim, and Ajith Abraham., 2011, *Understanding Plagiarism Linguistic Pattern, Textual Feature And Detection Methods,* IEEE Transactions On Systems, Man, And Cybernetics, Part C : Applications And Reviews, Volume : PP Issue : 99, page : 1-17.

[8] Androutsopoulos, I., and Malakasiotis, P., 2010, *A Survey of Paraphrasing And Textual Entailment Methods*, Journal Of Artificial Intelligence Research 38, 135-187.

[9] Agus Novanta, 2009, Pendeteksian Plagiarism Pada Dokumen Teks Dengan Menggunakan Algoritma Smith-Waterman, Medan.

[10] Badge, J., and Scott, J., 2009, *Dealing With Plagiarism In The Digital Age*, University Of Leicester, pp. 1-18.

[11] Brezovnik, Janez and Milan Ojtersek., 2011, *Texproc–A Natural Language Processing Framework and Its Use As Plagiarism Detection System,* International Journal Of Education And Information Technologies, Issue 3, Volume 5.

[12] Cedeño, Alberto Barrón., and Paolo Rosso., 2009, *On Automatic Plagiarism Detection Based on n-grams Comparison,* In Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ECIR 2009, LNCS 5478:696-700, Springer-Verlag, and Berlin Heidelberg.

[13] Ceska, Z., and Fox, C., 2009, *The Influence Of Text Pre-processing On Plagiarism Detection,* International Conference RANLP 2009 - Borovets, Bulgaria, pp. 55–59.

[14] Chang, J. S., and Chang, Y, 2004, Computer Assisted Language Learning Based On Corpora And Natural Language Processing : The Experience Of Project CANDLE, In Interactive Workshop On Language e-Learning, IWLeL, pp. 15-23.

[15] Chong, M., Specia, L., and Mitkov, R., 2010, *Using Natural Language Processing for Automatic Detection Of Plagiarism*, Proceedings Of The 4th International Plagiarism Conference (IPC-2010), Newcastle-upon-Tyne, UK.

[16] Clough, P., 2003, *Old And New Challenges In Automatic Plagiarism Detection*, National Plagiarism Advisory Service, February, pp. 391-407.

[17] Dhir, Amandeep., Gaurav Arora and Anuj Arora., 2005-2008, *Architectural Designing And Analysis Of Natural Language Plagiarism Detection Mechanism,* Journal Of Theoretical And Applied Information Technology.

[18] Elayed, Tamer., Jimmy Lin and Douglas W. Oard., 2008, *Pairwise Document Similarity in Large Collections with MapReduce*, Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 265–268, Columbus, Ohio, USA.

[19] Gunawan, W., Augustinus, R., dan Sembiring, K., 2005, *Penerapan Algoritma Edit Distance Pada Pendektesian Praktik Plagiat*, Bandung.

[20] Glatt Plagiarism Services (2007). http://plagiarism.com/. Accessed November 5, 2007

[21] Hari Bagus Firdaus, 2003, *Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp*, Jurnal Ilmu Komputer Dan Teknologi Informasi, Vol III No.2.

[22] Deepika, J., V. Archana, V. Bagyalakshmi, P. Preethi, and G.S. Mahalakshmi., 2011, *A Knowledge Based Approach to Detection of Idea Plagiarism in Online Research Publications*, International Journal on Internet and Distributed Computing Systems, Vol : 1 No. 2, pp 51-61.

[23] Grman, Jan., and Rudolf Ravas., 2011, Improved Implementation For Finding Text Similarities in Large Collections of Data, In Proceedings of PAN 2011.

[24] Kummar, J. Prasanna, and P. Govindarajulu., 2009, *Duplicate And Near Duplicate Documents Detection : A Review,* European Journal Of Scientific Research, ISSN 1450-216X, Vol. 32 No.4, pp.514-527.

[25] Leung, Chi-Hong and Yuen Yan Can., 2007, *A Natural Language Processing Approach To Automatic Plagiarism Detection,* Procedding Of The 8th ACM SIGinformation Conference On Information Technology Education SIGITE 07.

[26] Lukashenko, R., Graudina, V., and Grundspenkis, J., 2007, *Computer-Based Plagiarism Detection Methods And Tools : An Overview,* International Conference On Computer Systems And Technologies - *CompSysTech'07*.

[27] Maguitman, Ana G., Filipo Menczer, Heather Roinestad, and Alessandro Vespignani., 2005, Algorithmic Detection Of Semantic Similarity, Proceedings Of The 14th International Conference On World Wide Web, pp. 107-116.

[28] Pataki., M., 2003, *Plagiarism Detection And Document Chunking Methods*, ACM Vol xxx, Budapest, Hungary.

[29] Pera, Maria Soledad., and Yiu-kai Ng., 2010, *SimPaD : A Word-Similarity Sentence-Based Plagiarism Detection Tool on Web Documents*, In Journal on Web Intelligence and Agent Systems, IOS Press.

[30] Potthast, Martin., Benno Stein, Alberto Barron-Cedeno, and Paolo Rosso., 2010, *An Evaluation Framework For Plagiarism Detection,* Poster Volume, pp 997-1005, Beijing.

[31] Schleimer, S., Wilkerson, D. S., and Aiken, A., 2003, *Winnowing : Local Algorithms For Document Fingerprinting*, SIGMOD, San Diego, California.

[32] Sinta Agustina, 2008, Aplikasi Anti Plagiarisme Dengan Algoritma Karp-Rabin Pada Penulisan Ilmiah Universitas Gunadarma, Jakarta.

[33] Stamatatos, Efstathios., 2010, *Plagiarism Detection Based on Structural Information,* In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM'11.

[34] Runeson, P., Alexandersson, M., and Nyholm, O., 2007, *Detection Of Duplicate Defect Reports Using Natural Language Processing*, 29th International Conference On Software Engineering, ICSE'07, pp. 499-510.

[35] Terol, R., Martinez-Barco, P., and Palomar, M., 2006, *Applying NLP Techniques And Biomedical Resources To Medical Questions In QA Performance*, Lecture Notes In Computer Science, Vol. 4293, Springer, pp. 996-1006.

[36] Warin, Martin., 2004, *Using WordNet and Semantic Similarity to Disambiguate an Ontology*, Stockholms Universitet, Institutionen för Lingvistik.

[37] Weir, G. R. S., Gordon, M. A., and MacGregor, G., 2004, *Work In Progress – Technology In Plagiarism Detection And Management*, 34th ASEE/IEEE Frontiers in Education Conference, pages 18-19.