# Text Similarity Detection Between Documents Using Case Based Reasoning Method with Cosine Similarity Measure (Case Study SIMNG LPPM Universitas Sriwijaya)

Nabila Febriyanti [a,1], Dian Palupi Rini [b,2,*], Osvari Arsalan [b,3]

[a] Student at Informatics Engineering, Universitas Sriwijaya
[b] Lecturer, Department Informatics Engineering, Faculty of Computer Science, Sriwijaya University,
Jl. Srijaya Negara Bukit Besar, Palembang 30128, Indonesia
[1] research.nabilaf@gmail.com; [2*] dprini@unsri.ac.id; [3] osvari.arsalan@unsri.ac.id
* corresponding author

ARTICLE INFO

**Article history**
Received -
Revised -
Accepted -

**Keywords**
Case Based Reasoning
Text Similarity Detection
Cosine Similarity Measure
Vector Space Model

ABSTRACT

LPPM Universitas Sriwijaya is an institution that coordinates academic research and community service inside Universitas Sriwijaya. In carrying out the duty, LPPM assesses every proposal's originality which would be impossible to do manually in the future due to massive data growth. Thus, automatization for the proposal's originality check is needed. The Case Based Reasoning method is used in this research because it allows the system to reuse the information that has been obtained to find documents that are similar to the test document. In this study, the data is represented in the form of the Vector Space Model and uses Cosine Similarity to measure document to document similarity. The data is represented by giving weight for each part of the tested documents. In this study, four formulas from previous research will be used for term weighting then the final result will be compared. The process begins by extracting data, separating parts of the document, figuring the similarity value of the test document to the case base utilizing Cosine Similarity Measure, results filtering with a certain threshold, summarizing the calculation results, and finally preserving the results obtained to be reused in the next calculation. The results of this study indicate that the text-similarity detection between documents has been successfully carried out using the proposed method with the best sensitivity level and the fastest computation time achieved in configuration II.

## 1. Introduction

Author of [1] stated that technological developments support the rapid distribution of scientific work documents to the public. But on the other hand, it has an impact on increasing the possibility of plagiarism. The rapid growth of the distribution of documents makes it impossible to detect plagiarism manually [2, 3]. For academicians, the act of plagiarism has serious penalty such as a warning to the cancellation of a diploma or dismissal from the position currently being occupied [4].

In this study, the case study used was the Sriwijaya University Research and Community Service Institute (LPPM). To carry out the task of coordinating community service and research, LPPM builds a management information system that is used to collect data and distribute results related to grant proposals[1]. The assessment of the originality of the submitted work was initially carried out manually by the assessment team, but this has become difficult due to the rapid development of the number of submissions each period. Therefore, a system is needed that can automatically measure the originality of submitted documents with previous submissions.

[1] http://lppm.unsri.ac.id/2020/wp-content/uploads/2020/08/user-Guide-simng-v1.pdf

Researches related to the detection of literal text plagiarism between documents have been carried out using various methods, including string-based with the Rabin-Karp algorithm in [5, 6] and vector-based with Vector Space Model (VSM) using Jaccard Coefficient and Cosine Similarity measure in [7] and using hybrid method in [8].

Plagiarism detection with the Rabin-Karp algorithm has a weakness in dealing with the problem of the same hash value in words [9] and requires a longer computation time than the Levenshtein Distance algorithm [10]. Cosine Similarity works better than Jaccard Coefficient in testing using VSM trigrams because Jaccard Coefficient is less able to work well in giving more weight to unique terms [7]. In research [8] the data is represented into the Vector Space Model and then combines TF-IDF for word weighting, Cosine Similarity and word occurrence probability for similarity measurements in the case of plagiarism detection in text.

In research [11] explained that Case Based Reasoning is used because it allows the system to reuse information that has been obtained to find documents that are similar to one main document. In this study, the data is represented in the Vector Space Model. In this study, the data is represented in the Vector Space Model.

Based on these studies, this research will apply the Case Based Reasoning method with Cosine Similarity Measure to detect the similarity of text between research grant application documents in the SIMNG LPPM Sriwijaya University and compare the weights used in several previous studies in this case.

## 2. Theoretical Basis

### a. Plagiarism

According to Kamus Besar Bahasa Indonesia, plagiarism is defined as an act of plagiarism that violates copyright. According to [4] the act of plagiarism is defined as the act of using all forms of information belonging to others without mentioning the source properly and correctly. Referring to a survey conducted by [3], based on the taxonomy, plagiarism is divided into two, namely literal plagiarism and intelligent plagiarism. While the task of plagiarism is divided into two, namely extrinsic plagiarism and intrinsic plagiarism [3, 12, 13]. The task of plagiarism that is the focus of this research is extrinsic plagiarism, namely the detection of plagiarism by comparing the suspected document with one or more comparative document data by focusing on the type of literal plagiarism detection where the perpetrator is purely copying all, part of, or reconstructing another person's ideas or writings. others without proper citation of the original author [3]. This study also stated that the results of the similar parts in a plagiarism detection system are collected and can be used as information to humans to determine the final result, namely plagiarism or not plagiarism.

### b. Case Based Reasoning

Case Based Reasoning (CBR) is a problem-solving method that uses similar experiences as the basis for drawing conclusions or solutions to a problem. Textual Case Based Reasoning is used to solve cases using the CBR method on textual data [14]. Compared to relying on general knowledge, CBR is able to utilize specific knowledge of problems that have been previously solved and CBR allows a system to carry out continuous learning using experience or new knowledge available in subsequent problem solving [11, 15, 16]. The CBR cycle can be seen in Fig. 1.

Fig. 1. Describe the main cycles of the CBR, namely retrieval, reuse, revise and retain [14, 15]. In this cycle, the new problem first enters the retrieve stage where cases similar to new problems from the case base are found so that the solution can be adapted from the information that has been stored. Furthermore, the results of the cases that have been found enter the reuse stage where information from the collection is reused, either used directly or adapted according to the needs of new cases. Furthermore, the results of case resolution are evaluated in the review stage and finally, the retain stage where the case base is updated with new information for further searches.

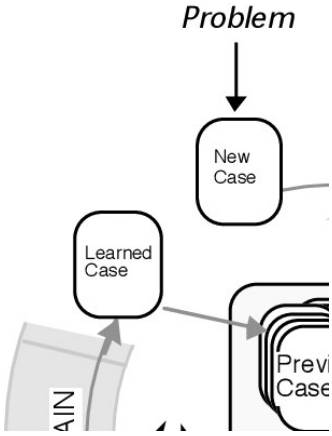


**Fig. 1.**CBR Cycle

### c. Text Pre-Processing

Pre-processing aims to reduce features by reducing vocabulary size and increasing accuracy in feature representation. The processes involved in text pre-processing vary depending on the text data being studied. In general, there are four main processes carried out, namely tokenization, removing stopwords, case folding and stemming [17].

1. Cleaning, which is the stage where all characters, numbers, white spaces and other language characters that are not relevant to the original language are removed [18].
2. Case folding or changing the case of letters is a procedure to equalize all types of letters either into all uppercase (capital) or all lowercase letters [19].
3. Tokenization is the procedure of breaking all text into words, phrases or other fractional forms called tokens. Tokenization is part of segmenting a text. Generally, segmentation separates the alphabet or alphanumeric characters based on non-alphanumeric characters, such as punctuation marks or spaces [17].
4. Stopword Removal is a procedure to remove words that are considered less meaningful or generally appear in large numbers [20]. Indonesian stopwords that are generally used are the results of research by [21].
5. Stemming is a procedure to find the root word of a word by removing the affix from a derived word [19]. The most influential and used research until now for Indonesian stemming is research by [22].

### d. Vector Space Model and TF-IDF Weighting

Vector Space Model is a way to represent text data from a document into a vector form whose attributes are derived from mathematical word calculations [7, 23, 24], one of which is TF-IDF weighting. TF-IDF is the value of the weight of a word by considering the degree of importance of a word in a collection of documents. Term Frequency (TF) is the frequency value of one word appearing in a document, while Inverse Document Frequency (IDF) is a value that measures the degree of importance of the word itself based on the number of words in one document and the whole [25].

The TF-IDF weights used in this study were derived from four different studies listed in (1) to (4).

$$W_{t,d} = TF_{t,d} \times IDF_{t,d} = TF_{t,d} \times \left(\log\left(\frac{N}{df_t}\right)\right) \quad (1)$$

$$W_{t,d} = TF_{t,d} \times IDF_{t,d} = TF_{t,d} \times \left(1 + \log\left(\frac{N}{df_t}\right)\right) \quad (2)$$

$$W_{t,d} = TF_{t,d} \times IDF_{t,d} = TF_{t,d} \times \left(\log\left(\frac{N}{1 + df_t}\right)\right) \quad (3)$$

$$W_{t,d} = TF_{t,d} \times IDF_{t,d} = TF_{t,d} \times \left(\ln\left(\frac{N}{df_t}\right) + 1\right) \quad (4)$$

Where $W_{t,d}$ is the weight of word $t$ in document $d$, $TF_{t,d}$ is the occurrence of word $t$ in document $d$, $IDF_{t,d}$ is IDF value of word $t$ in document $d$, $N$ is the number of all documents compared and finally $df_t$ is word $t$ occurrence out of all documents.

Equation (1) is the TF-IDF weighting formula used in research [26], while (2) is the TF-IDF weighting formula used in research [7], (3) used in research [27] and (4) used in research [8].

**e. Cosine Similarity Measure**

Cosine Similarity is a vector-based similarity measurement model that is widely used in information retrieval and text mining. This approach compares two strings that have been transformed into vectors. The cosine similarity equation can be seen in (5) [26].

$$cosine\ similarity = \cos\theta = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|\|\vec{B}\|} = \frac{(a_1 \times b_1) + (a_2 \times b_2) + \cdots + (a_n \times b_n)}{\sqrt{{a_1}^2 + {a_2}^2 + \cdots + {a_n}^2}.\sqrt{{b_1}^2 + {b_2}^2 + \cdots + {b_n}^2}} \quad (5)$$

Where $A$ and $B$ are two vectors to be compared with the elements $a_1$ to $a_n$ and $b_1$ to $b_n$, respectively.

## 3. Methodology

The research was carried out in stages according to the framework in Figure 2.

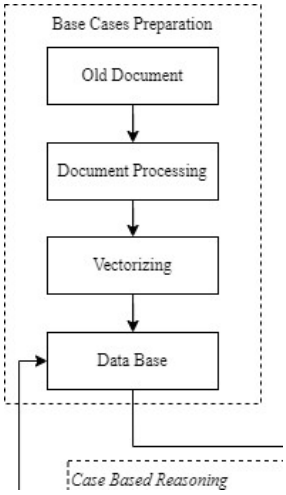


**Fig. 2.**Framework Used in This Research

Each stage in Fig. 2 is discussed in more detail below.

**a. Data Collection**

Data on grant proposals with the extension *.pdf* totaling 60 documents were collected through the SIMNG LPPM Sriwijaya University application, with 50 documents as case base documents and 10 test documents. The proposals that will be used are selected in accordance with the writing systematics provided by the LPPM Sriwijaya University. In this study, a document is separated into defined document parts. The sub-chapters that will be used in detecting the similarity of text between documents are as follows:

1. Judul (Jdl) / Title
2. Rangkuman (Rkm) / Summary
3. Latar Belakang (LB) / Introduction

4. Kajian Teori (KT) / Theoritical Study
5. Metode Penelitian (MP) / Research Method

A snippet of the list of grant proposals used in this study is contained in Table 1.

**Table 1.** Snippets of the list of proposals used in research

| No. | Grant Proposal Title |
|---|---|
| 1 | Penataan Permukiman Ekologis pada Lahan Basah Tepian Sungai Musi, Palembang |
| 2 | Kajian Efek Sinergis Dari Bawang Putih Dan Belimbing Wuluh Sebagai Kandidat Obat Untuk Penyakit Ikan |
| 3 | Penapisan Fitokimia Melalui Metode Ekstraksi Berbeda Pada Tanaman Air Sebagai Potensi Obat Penyakit Ikan |

**b. Document Processing**

In this stage, in general, what will be done is extracting text from the document, conducting pre-processing consisting of cleaning, case folding, splitting the document into subchapters, tokenization with space separators, stop word removal and stemming. The final result of this stage is the basic word tokens which are separated by part of the document.

**c. Text Similariy Detection between Documents using CBR**

Before detecting the similarity of text between documents using the CBR method, the data is first transformed in the form of words (terms) into vector form using (1) to (4). The result of this transformation is the weight of each word ready to be calculated.

In the retrieval stage, each new document section is compared with all the data in the case base. The parts are compared and the similarity value is calculated using cosine similarity according to (5). Furthermore, in the reuse stage, the retrieval results are filtered into M candidates with a specified threshold. Referring to research [8], threshold = 0.2 will be used as the limit for candidate selection. The results of this screening will be used in the revision stage to select the top candidates and combine all the information in each section as a conclusion to the search and calculations carried out. The results of the final conclusion enter the retain stage where at this stage the calculation of new documents is entered into the case basis to be reused as material for further case searches.

**d. Retrieval of Results and Analysis**

The results in each test of all test documents will be collected for later analysis which will be discussed in the next chapter. Evaluation looks at the results given from each configuration. In addition, the computation time for each configuration scenario will be calculated and compared.

## 4. Result and Discussion

The results that will be discussed in this study in general are the results of the measure of similarity, the results in each stage of the Case Based Reasoning method and the computational time seen from each test document and configuration used. The summary graph of the results is loaded in Fig. 3.

The calculation of the cosine similarity value is carried out to see the similarity of two documents that have previously been represented in vector form. The cosine similarity value has a range from 0 to 1. The higher the cosine similarity value, the more similar the two documents. The lowest cosine similarity value obtained in testing all documents in all configurations is 0, meaning that in each document test there are parts that are not at all similar to certain parts of the document. This can happen if there are no words in common in the two documents being compared. In this case, there are two possible causes for the resulting cosine similarity = 0, namely the size of the document is too small or the two documents being compared have a much different size. The smaller the size of the document compared, the less likely the two documents have the same vocabulary.

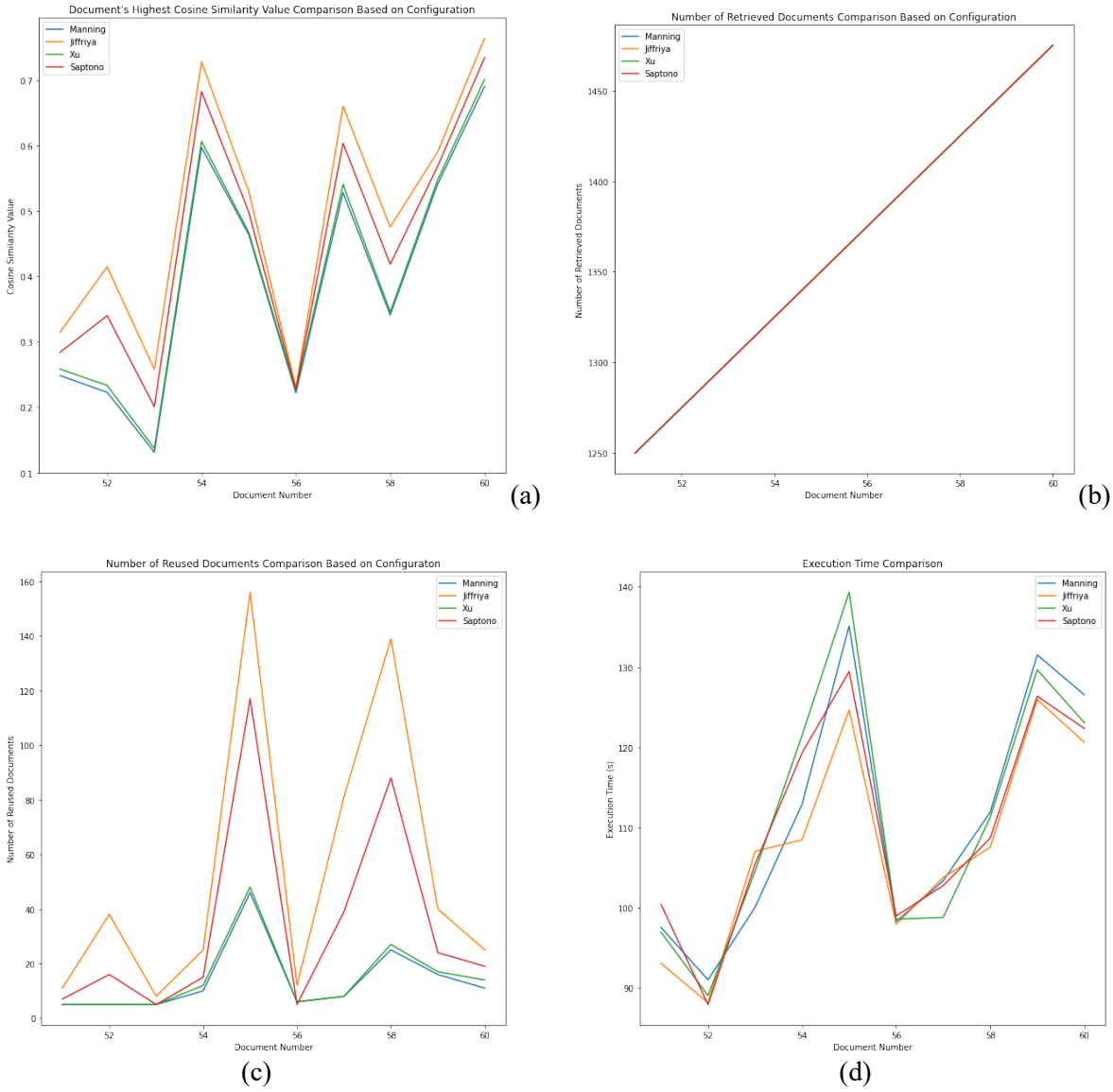


**Fig. 3.**Comparison of Computational Results of Each Test Document Based on Configuration (a) Comparison of the Number of Retrieved Documents Parts, (b) Comparison of the Highest Cosine Similarity Values between Documents, (c) Comparison of the Number of Reused Documents Parts, (d) Comparison of Computing Time

In Fig. 3 (a) it can be seen that the cosine similarity value generated by each configuration in the test document is different. This shows that the weighting has an effect on the resulting cosine similarity value. In (2) and (4) the resulting IDF value becomes larger due to the addition of the value after the logarithm calculation. A larger IDF value then makes the weight value as a product of the frequency of word occurrences even greater. The largest cosine similarity value in each test document resulted in a weighted configuration in [7] while the highest cosine similarity value and the lowest among other configurations was a test with a weighted configuration in [26]. The difference in cosine similarity values that are influenced by this weighting can also be followed by several differences in the final results of the similar documents produced. However, most of the test results show the final results of the same similar documents with different cosine similarity values.

The Case Based Reasoning method used in this study divides the work into four main stages, namely retrieval, reuse, revise and retain. The first step is retrieval, which is collecting all parts of the case base document and measuring its similarity with the test document section. The comparison of the number of documents used in the retrieval stage for each test document can be seen in Fig. 3 (b). In the visualization, it can be seen that each configuration uses the same number of document

parts in the retrieval stage, which means that the number of document parts always increases as several consecutive tests run. This proves that the case based reasoning method can actually reuse the information that has been added from the previous calculation. Case based reasoning stores the results of each final calculation for later inclusion in the next calculation or search.

The next stage is reuse. At this stage, parts of the document will be filtered based on the threshold value that has been defined previously. The comparison of the number of parts of the document generated in the reuse stage based on the configuration can be seen in Fig. 3(c).

Based on the visualization, it can be seen that the number of document parts produced in the reuse stage varies with each document and configuration. This depends on the value of the cosine similarity that has been calculated in the previous step. The more the number of parts of the document that is generated, it means that the more parts of the document that are similar above a predetermined threshold. This discussion relates to the analysis of the results of the previous cosine similarity calculation, where each configuration will have a different value and cause the sensitivity to similar documents to be different. A higher level of sensitivity to document similarity in this case can be more advantageous because it allows more data to be obtained to serve as a basis for drawing conclusions.

Based on the results of this study, the reuse stage can only take one part of the document due to two reasons, namely if the cosine similarity value of all comparisons of the document part is below the threshold so that only the part of the document with the highest cosine similarity value is taken or if there is only one comparison of the part of the document that has the highest cosine similarity value higher than the threshold value. The configuration II with weighting in [7] filters out the most parts of the document in this stage because it produces a greater cosine similarity value compared to other configurations. On the other hand, the configuration I with internal weighting [26] filters at least part of the document to be sent to the next stage because the resulting cosine similarity value is lower than the other configurations.

The next stage is revise, where the results of document similarity detection are summarized by only taking the comparison results in each part of the document that has the highest cosine similarity value. Lastly, at the retain stage, the final result is then sent to the database to be stored and reused in the detection of similarity between documents.

The results of the comparison of computational time based on the configuration used are summarized in Fig. 3(d). Based on the visualization, the difference in computational time in each configuration is quite varied with the smallest difference being 1,043 seconds for the 56th document while the farthest difference in computational time is 14,685 seconds for the 55th document. It can be seen in the visualization that the test configuration uses internal weighting [7] in general has a shorter computation time compared to other test configurations, which is 7 out of a total of 10 test documents. While the test configuration using internal weighting [26] requires a longer computational time compared to other configurations with 5 out of 10 test documents processed in the longest time.

Based on the observations in this study, the difference in computational time required by each configuration can be influenced by several reasons, including the computational complexity based on the formula used and the operations that need to be performed in the calculations. In addition, a significant increase in computational time is also associated with the use of very large document sections such as the research methods section.

Initially, if the document sizes are relatively the same, the computation time will increase as the number of documents being compared increases. However, there can also be a significant decrease or increase if the size of the test document is too small or too large, respectively. For example, in document 52 and document 56, there is a decrease in computing time due to the smaller document size compared to other document sizes. Meanwhile, a significant increase in computation time occurred in document 55 due to the large document size. Calculations can take less time on smaller documents because they will generate fewer tokens, making comparisons of test documents one by one to all documents in the database faster. Vice versa, a larger document size causes a significant increase in computing time because more tokens will be generated so that the comparison of test documents one by one against documents in the database takes longer.

Based on the test results, the part of the document that has the highest similarity value in each test document is mostly part of the research method, which is 7 to 9 out of 10 documents in each configuration. This is because the discussion in research methods tends to be more easily detected similar due to the uniform discussion points in context. Examples of discussions in the research methods section, including discussions about how to collect data for research, how data processing is carried out, how the process is in research, frameworks in research and other related discussions. If this section is most similar to the other document research methods sections, this indicates that the steps involved in the two studies are uniform or similar. For example, test document 60 and test document 6. The two research proposal documents discuss legal aspects or legal perspectives on different issues. Therefore, the methods or approaches used in these two studies are likely to be very similar.

The possibility of a very high similarity value in the research methods section makes future researchers need to reconsider using this section because the similarities that occur in this section of the document are unavoidable and tend to always occur. In other words, the part of the research method document which tends to be larger in size than the other part of the document but does not provide significant information in decision making needs to be reconsidered its use in future research.

In addition, if the theoretical study sections of the two documents are very similar, it can be said that the two studies have the same or even the same concentration because the basic theoretical sources for the research are close or similar. For example in test document 55 and test document 15. Based on the title, these two research proposal documents discuss the development of textbooks (textbooks) but on different objects so that the theoretical basis used in these two studies is similar or tends to be the same.

## 5. Conclusion

Based on the results of the data and analysis that have been carried out, the detection of text similarity between the SIMNG LPPM grant application documents, Sriwijaya University was successfully carried out by measuring the similarity between documents using the Case Based Reasoning method which is able to reuse information that has been obtained from previous tests and the measure of Cosine Similarity that sees the number of word distributions and the degree of word importance used by the two parts of the comparison document. In addition, this study also compares the weighting formulas in previous studies with the results showing that the weighting has an effect on the resulting cosine similarity value and has an impact on the results in the detection stages of text similarity between documents.

The process of detecting text similarity between documents applying for research grants from SIMNG LPPM Sriwijaya University begins with extracting data, separating parts of documents, pre-processing text into ready-to-process data, carrying out word weighting, calculating the similarity value of test documents to the case base with Cosine Similarity Measure, filtering with a certain threshold, summarizing the calculation results and finally saving the results obtained so that they can be reused in the next calculation.

With the test document used, the lowest cosine similarity value generated in the calculation is 0, meaning that there is no similarity at all between the two parts of the comparison document. Meanwhile, the highest cosine similarity value produced in the calculation is 0.763 obtained in the 60 test document using the weighting configuration in [7]. The test results show that the highest level of similarity detection sensitivity and the shortest computation time is achieved in configuration II, which is using internal weighting [7]. On the other hand, the lowest sensitivity level and the longest computation time are achieved in configuration I, i.e. using weighting in[28].

Based on the research results obtained, it can be concluded that the software built and the use of the method have been implemented properly. Some improvements and research opportunities that can be used as references include:

1. Re-evaluate the selection of the document section to be used considering the results in this study to obtain a more effective and faster performance in terms of computing.
2. Combining the framework in the Case Based Reasoning method, namely retrieval, reuse, revise and retain with other algorithms based on the objectives of each stage in the CBR framework

## References

[1] I. M. I. Subroto and A. Selamat, "Plagiarism detection through internet using hybrid artificial neural network and support vectors machine," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 12, no. 1, pp. 209–218, 2014, doi: 10.12928/TELKOMNIKA.v12i1.648.

[2] P. Clough, "Plagiarism in natural and programming languages: an overview of current tools and technologies," *Finance*, no. July, pp. 1–31, 2000, [Online]. Available: http://www.dcs.shef.ac.uk/nlp/meter/Documents/reports/plagiarism/Plagiarism.pdf.

[3] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 2. pp. 133–149, 2012, doi: 10.1109/TSMCC.2011.2134847.

[4] Y. Yuliati, "Perlindungan Hukum Bagi Pencipta Berkaitan Dengan Plagiarisme Karya Ilmiah Di Indonesia," *Arena Huk.*, vol. 5, no. 1, pp. 54–64, 2012, doi: 10.21776/ub.arenahukum.2012.00501.7.

[5] W. G. S. Parwita, I. G. A. A. D. Indradewi, and I. N. S. W. Wijaya, "String Matching based Plagiarism Detection for," *2019 5th Int. Conf. New Media Stud.*, 2019.

[6] D. Leman, M. Rahman, F. Ikorasaki, B. S. Riza, and M. B. Akbbar, "Rabin Karp and Winnowing Algorithm for Statistics of Text Document Plagiarism Detection," 2019, doi: 10.1109/CITSM47753.2019.8965422.

[7] M. A. C. Jiffriya, M. A. C. A. Jahan, and R. G. Ragel, "Plagiarism detection on electronic text based assignments using vector space model," *2014 7th Int. Conf. Inf. Autom. Sustain. "Sharpening Futur. with Sustain. Technol. ICIAfS 2014*, 2014, doi: 10.1109/ICIAFS.2014.7069593.

[8] R. Saptono, H. Prasetyo, and A. Irawan, "Combination of cosine similarity method and conditional probability for plagiarism detection in the thesis documents vector space model," *J. Telecommun. Electron. Comput. Eng.*, vol. 10, no. 2–4, pp. 139–143, 2018.

[9] J. Priambodo, "Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp dengan Metode Rolling Hash," *J. Inform. Univ. Pamulang*, vol. 3, no. 1, p. 39, 2018, doi: 10.32493/informatika.v3i1.1518.

[10] A. H. Purba and Z. Situmorang, "Analisis Perbandingan Algoritma Rabin-Karp Dan Levenshtein Distance Dalam Menghitung Kemiripan Teks," *J. Tek. Inform. Unika St. Thomas*, vol. 02, pp. 24–32, 2017.

[11] M. Mihajlovic and N. Xiong, "Finding the most similar textual documents using Case-Based Reasoning," *arXiv*, 2019.

[12] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeno, and P. Rosso, "Overview of the 1st international competition on plagiarism detection," *CEUR Workshop Proc.*, vol. 502, pp. 1–9, 2009.

[13] Z. F. Alfikri and A. Purwarianti, "Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach (Naive Bayes and SVM)," *TELKOMNIKA Indones. J. Electr. Eng.*, vol. 12, no. 11, pp. 7884–7894, 2014, doi: 10.11591/telkomnika.v12i11.6652.

[14] M. M. Richter and R. O. Weber, *Case-Based Reasoning*. Springer International Publishing, 2013.

[15] A. Agnar and E. Plaza, "Case-Based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994, doi: 10.3233/AIC-1994-7104.

[16] A. Mubarak *et al.*, "Case-Based Reasoning Untuk Aplikasi Pemilihan Pestisida Hama Case-Based Reasoning for Web Based Selection of Rice Pesticides," vol. 3, no. 2, pp. 119–124, 2020, doi: 10.33387/jiko.

[17] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, 2014, doi: 10.1016/j.ipm.2013.08.006.

[18] B. Furlan and V. Batanovi, "Semantic similarity of short texts in languages with a de fi cient natural

language processing support," vol. 55, pp. 710–719, 2013, doi: 10.1016/j.dss.2013.02.002.

[19] R. Goyena and A. . Fallis, "Pengembangan Aplikasi Pendeteksi Plagiarisme Pada Dokumen Teks Menggunakan Algoritma Rabin-Karp," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[20] A. E. Budiman, "Analisis Pengaruh Teks Preprocessing Terhadap Deteksi Plagiarisme Pada Dokumen Tugas Akhir," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, pp. 475–488, 2020, doi: http://dx.doi.org/10.28932/jutisi.v6i3.2892 Ariel.

[21] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," *M.Sc. Thesis, Append. D*, vol. pp, pp. 39–46, 2003.

[22] M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi, and H. Williams, "Stemming Indonesian: A confix-stripping approach.," *ACM Trans. Asian Lang. Inf. Process.*, vol. 6, 2007.

[23] M. Zechner, M. Muhr, R. Kern, M. Granitzer, and K.-C. Graz, "External and Intrinsic Plagiarism Detection Using Vector Space Models," 2009.

[24] A. A. P. Ratna *et al.*, "Cross-language plagiarism detection system using latent semantic analysis and learning vector quantization," *Algorithms*, vol. 10, no. 2, 2017, doi: 10.3390/a10020069.

[25] A. Mishra and S. Vishwakarma, "Analysis of TF-IDF Model and its Variant for Document Retrieval," *Proc. - 2015 Int. Conf. Comput. Intell. Commun. Networks, CICN 2015*, pp. 772–776, 2016, doi: 10.1109/CICN.2015.157.

[26] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[27] L. Xu, S. Sun, and Q. Wang, "Text similarity algorithm based on semantic vector space model," 2016, pp. 1–4, doi: 10.1109/ICIS.2016.7550928.

[28] S. Reddy, D. Chen, and C. D. Manning, "CoQA: A conversational question answering challenge," *arXiv*, vol. 7, no. March, pp. 249–266, 2018, doi: 10.1162/tacl_a_00266.