Implementation of Clustering and Similarity Analysis for Detecting Content Similarity in Student Final Projects

A A Ilham1*, A Bustamin1, I Aswad1, and F Armin1

1Department of Informatics, Faculty of Engineering, Hasanuddin University, Makassar, Indonesia

*Email: amil@unhas.ac.id

Abstract. To finish study, students are requested to submit final projects. In some universities, the final projects are not necessary to be submitted for publication. The final project reports are stored in a local database. As the number of final projects is growing in the local database, similar contents may exist among the documents. The commercial tools cannot be used to detect the content similarity since the documents are not published. This paper proposed a system to detect content similarity in documents that are stored in a local database. Considering the number of stored documents, this similar content detection system implements two step processes. First, clustering documents to find most related documents. Second, finding content similarity among the selected documents. The experiment results show that the system is successfully clustering documents and detecting content similarity by implementing TF-IDF and Cosine Similarity algorithms. This system is limited to proceed documents that are written in Bahasa.

1. Introduction

Similar contents among documents may indicated the documents contain plagiarism. Plagiarism is defined as taking of essays, opinions and so on from others and makes them appear like their own essays and opinions. The documents that have little content, the examination can be done manually without the help of the system. However, on documents that have thousands of lines and pages, it is certainly not possible. Many commercial tools exist, but they cannot be used to examine documents that are stored in a local database.

A number of algorithms have been used to detect content similarity in documents such as string-based detection technique, Vector Space Models (VSM), syntax and semantic based detection technique, structural-based detection technique and citation based detection technique [1]. The Rabin-Karp Algorithm has been used in [2]. The hash value generated from this algorithm is the benchmark of similarity between the documents being tested. However, this research still needs to be adjusted when choosing the K-gram value to be used in the tokenization process. The weakness of this algorithm is that the system cannot know the order in which documents appear in this case the time of publication. This algorithm only focuses on determining the similarity between documents being compared. The comparison of the level of document similarity using several methods such as K-Means Clustering, K-Nearest Neighbor and Shingling Algorithm is done in [3]. The purpose of this study is to compare two vectors of text documents that measure the degree of similarity. The best level of accuracy generated by the K-Nearest Neighbor algorithm is equal to 95%.

The Rabin-Karp algorithm is also applied in the Indonesian document similarity detection system. In addition to Rabin-Karp, the Confix-Stripping method is used to maximize the steps for searching basic
words in text processing. The system works well in detecting document similarities with an average processing time of 0.0123 seconds and an average similarity level of documents tested 89.1967%. Overall the level of accuracy shown by the system is 0.7 [4].

A number of commercial tools have been developed such as PlagAware, PlagScan, Check for Plagiarism, iThenticate, PlagiarismDetection.org, Academic Plagiarism, The Plagiarism Checker, Urkund, Doculoc and Turnitin [5]. PlagAware provides two main features namely tracing content theft and proof of authorship [6]. PlagScan provides a number of features such as database checking, Internet checking, publications checking, synonym and sentence structure checking [7]. CheckForPlagiarism.net implements fingerprint and document sources to protect documents from plagiarism. It provides features such as Internet checking, publications checking, synonym and sentence structure checking, and multiple document comparison [8].

2. Research Method
Considering the number of stored documents, our system implements two step processes. First, clustering documents to find 10 most related documents. Second, finding content similarity among the selected documents. The system design can be seen in Figure 1.

![Figure 1. System design](image)

2.1. Preprocessing
Preprocessing is the initial stage of text mining where the text data will be cleaned so that the text becomes more structured before entering the next stage for further processing. Continuous character sets (text) must be broken into more meaningful. This can be done on several different levels. Text preprocessing stages in this study consist of tokenizing, case folding, filtering, and stopword.
2.2. TF-IDF Weighting

Term Frequency (TF) is a simple weighting where the importance of a word is assumed to be proportional to the number of times the word appears in a document. Inverse Document Frequency (IDF) is a weighting that measures how important a word in a document is when viewed globally in all documents. The TF x IDF weighting value will be high if the TF value is large and the word observed is not found in many documents. The TF value is calculated using the following Equation (1):

\[ TF(d, t) = f(d, t) \] (1)

where \( f(d, t) \) is the number of times the word \( t \) appears in document \( d \). IDF considers the frequency of words in all documents. IDF weighting assumes that the weight of a word will be large if the word often appears in a document but not many documents contain the word. IDF values are calculated using the following Equation (2):

\[ IDF(t) = \log \left( \frac{N}{df(t)} \right) \] (2)

where \( df(t) \) is the number of documents that have the word \( t \). The results of previous studies show that TF x IDF weighting can improve performance better. The TF x IDF value is calculated using the following function [9].

\[ TFIDF(d, t) = TF (d, t) \times IDF(t) \] (3)

At Term Frequency (TF), there are several types of formulas used, i.e. [10]:

a. Binary TF: pays attention to whether or not a word exists in the document. If it exists, a value of one is given, if not, a zero value is given.

b. Raw TF: the TF value is based on the number of occurrences of a word in the document. For example, if it appears five times the word will be worth five.

c. TF logarithmic: used to avoid the dominance of documents that contain few words in the query but have a high frequency.

d. TF normalization: using a comparison between the frequencies of a word with the total number of words in a document.

2.3. Cosine Similarity

Cosine similarity is a method used to calculate the level of similarity between two objects. Cosine similarity is used to measure the closeness between two vectors. Cosine similarity is the result of the dot product of the two vectors that is normalized by dividing by Euclidean Distance between the two vectors. Equation (4) obtained is as follows [11]:

\[ \text{sim} (d_j, q) = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \] (4)

where:

\( d_j \) = document \( j \)

\( q \) = query document

\( \sum_{i=1}^{N} w_{i,j} \) = total weight of the word \( i \) in document \( j \)

\( \sum_{i=1}^{N} w_{i,q} \) = the weights number of word \( i \) in the query

The TF-IDF weighting process can be seen in Figure 2. The TF-IDF method combines two concepts for weight calculation, namely the frequency of occurrence of a word in a particular document and the inverse frequency of the document containing the word. The frequency with which words appear in a
given document indicates how important that word is in the document. The frequency of documents containing the word indicates how common the word is. So, the weight of the relationship between a word and a document will be high if the frequency of the word is high in the document and the overall frequency of the document containing the word is low in the document.

![Figure 2. Weighting process](image)

3. Implementation and Result
To illustrate how the system works, we compared a test document (Q) with four other documents (D2, D3, D4 and D5) as shown in Table 1. The test documents are written in Bahasa.

We perform preprocessing to all documents including tokenizing, case folding and filtering. In case folding, the contents of the document are converted into lowercase. Filtering process eliminates punctuation and words that are considered not important in Bahasa such as word "di", "dan", "untuk", etc. The output of the preprocessing can be seen in Table 2.
Table 1. Test document

| Doc Name | Content |
|-----------|---------|
| Q         | Fenomena plagiarisme yang lebih spesifik sering terjadi di dunia akademis, khususnya dilakukan oleh mahasiswa dalam menyelesaikan tugas kuliah maupun tugas akhir karena tersedianya fasilitas untuk menyalin suatu teks dan menaruh salinan teks tersebut dari satu dokumen ke dokumen lainnya. |
| D2        | Perkembangan teknologi memiliki dampak yang sangat signifikan dalam kehidupan sehari-hari, mulai dari kegiatan yang sederhana hingga kegiatan yang membutuhkan tingkat ketelitian yang tinggi. Kegiatan yang umum dilakukan oleh sebuah instansi adalah kegiatan pengarsipan dokumen, baik dokumen dalam bentuk fisik maupun elektronik. |
| D3        | Kegiatan menjiplak tugas sering dilakukan mahasiswa yang merupakan tindakan plagiat, banyak tugas yang terkumpul dan waktu yang terbatas membuat dosen sulit untuk memeriksa tugas satu per satu. Oleh karena itu, diperlukan suatu aplikasi yang dapat mendeteksi kemiripan dokumen teks. |
| D4        | Paper ini mendiskusikan tentang deteksi plagiarisasi dengan menggunakan metode string matching algoritma rabin-karp. Metode diimplementasikan dalam aplikasi berbasis web untuk mendeteksi plagiarisasi dengan cara menguji teks (huruf) yang ada pada dokumen abstraksi dari karya skripsi atau jurnal mahasiswa. |
| D5        | Kesamaan dokumen dapat digunakan untuk menjadi petunjuk dan contoh mencari informasi yang sama. Kemampuan mencari kesamaan ini dapat mengurangi waktu. Untuk menggambarkan tingkat kesamaan antara dokumen dapat diukur oleh Metode Cosine Similarity. Berdasarkan tingkat kesamaan dokumen dapat diklasifikasikan dengan menggunakan Algoritma Single Pass Clustering. |

Table 2. Preprocessing results

| Doc Name | Content |
|-----------|---------|
| Q         | Fenomena plagiarisme spesifik dunia akademis mahasiswa menyelesaikan tugas kuliah tugas tersedianya fasilitas menyalin teks menaruh salinan teks dokumen dokumen alasan penulis mencoba membangun sistem deteksi plagiarisme dokumen bahasa Indonesia metode vector space model |
| D2        | Perkembangan teknologi memiliki dampak signifikan kehidupan seharihari kegiatan sederhana kegiatan membutuhkan tingkat ketelitian kegiatan instansi kegiatan pengarsipan dokumen dokumen bentuk fisik elektronik |
| D3        | Kegiatan menjiplak tugas mahasiswa tindakan plagiat tugas terkumpul terbatas dosen sulit memeriksa tugas aplikasi mendeteksi kemiripan dokumen teks |
| D4        | Paper mendiskusikan deteksi plagiasi metode string matching algoritma rabinkarp metode diimplementasikan aplikasi berbasis web mendeteksi plagiasi menguji teks huruf dokumen abstraksi karya skripsi jurnal mahasiswa |
| D5        | Kesamaan dokumen petunjuk contoh mencari informasi kemampuan mencari kesamaan mengurangi menggambarkan tingkat kesamaan dokumen diukur metode cosine similarity berdasarkan tingkat kesamaan dokumen diklasifikasikan algoritma single pass clustering |

After the preprocessing, TF-DF is calculated by counting the appearance of each term in Q in other documents. The TF and DF values can be seen in Table 3.
Table 3. TF and DF results

| Term            | D2 | D3 | D4 | D5 | DF |
|-----------------|----|----|----|----|----|
| Fenomena        | 0  | 0  | 0  | 0  | 0  |
| plagiarisme     | 0  | 0  | 0  | 0  | 0  |
| spesifik        | 0  | 0  | 0  | 0  | 0  |
| dunia           | 0  | 0  | 0  | 0  | 0  |
| akademis        | 0  | 0  | 0  | 0  | 0  |
| mahasiswa       | 0  | 1  | 1  | 0  | 2  |
| menyelesaikan   | 0  | 0  | 0  | 0  | 0  |
| tugas           | 0  | 3  | 0  | 0  | 1  |
| kuliah          | 0  | 0  | 0  | 0  | 0  |
| tersedianya     | 0  | 0  | 0  | 0  | 0  |
| fasilitas       | 0  | 0  | 0  | 0  | 0  |
| menyalin        | 0  | 0  | 0  | 0  | 0  |
| teks            | 0  | 1  | 1  | 0  | 2  |
| menaruh         | 0  | 0  | 0  | 0  | 0  |
| salinan         | 0  | 0  | 0  | 0  | 0  |
| dokumen         | 2  | 1  | 1  | 3  | 4  |
| alasan          | 0  | 0  | 0  | 0  | 0  |
| penulis         | 0  | 0  | 0  | 0  | 0  |
| mencoba         | 0  | 0  | 0  | 0  | 0  |
| membangun       | 0  | 0  | 0  | 0  | 0  |
| sistem          | 0  | 0  | 0  | 0  | 0  |
| deteksi         | 0  | 0  | 1  | 0  | 1  |
| bahasa          | 0  | 0  | 0  | 0  | 0  |
| indonesia       | 0  | 0  | 0  | 0  | 0  |
| metode          | 0  | 0  | 2  | 1  | 2  |
| vector          | 0  | 0  | 0  | 0  | 0  |
| space           | 0  | 0  | 0  | 0  | 0  |

TF-DF values are used to calculate IDF using $\log(n/df)$ formula. TF x IDF values as can be seen in Table 4.

Table 4. IDF and TF-IDF calculation results

| Term     | IDF = log (n/df) | D2 | D3 | D4 | D5 |
|-----------|-----------------|----|----|----|----|
| fenomena  | 0               | 0  | 0  | 0  | 0  |
| plagiarisme | 0              | 0  | 0  | 0  | 0  |
| spesifik  | 0               | 0  | 0  | 0  | 0  |
| dunia     | 0               | 0  | 0  | 0  | 0  |
| akademis  | 0.301           | 0  | 0  | 0  | 0  |
| mahasiswa | 0.301           | 0  | 0  | 0  | 0  |
| menyelesaikan | 0.602 | 0  | 1.806 | 0  | 0  |
| tugas     | 0.602           | 0  | 1.806 | 0  | 0  |
| kuliah    | 0               | 0  | 0  | 0  | 0  |
| tersedianya | 0             | 0  | 0  | 0  | 0  |
| fasilitas | 0               | 0  | 0  | 0  | 0  |
| menyalin | 0               | 0  | 0  | 0  | 0  |
| teks      | 0.301           | 0  | 0  | 0  | 0  |
| menaruh   | 0               | 0  | 0  | 0  | 0  |
| salinan  | 0               | 0  | 0  | 0  | 0  |
The third EPI International Conference on Science and Engineering 2019 (EICSE2019)  
IOP Publishing  
IOP Conf. Series: Materials Science and Engineering 875 (2020) 012039  
doi:10.1088/1757-899X/875/1/012039

\[
TF \times IDF = \log \frac{n}{df}
\]

| Term            | IDF = log (n/df) | D2 | D3 | D4 | D5 |
|-----------------|------------------|----|----|----|----|
| dokumen         | 0                | 0  | 0  | 0  | 0  |
| alasan          | 0                | 0  | 0  | 0  | 0  |
| penulis         | 0                | 0  | 0  | 0  | 0  |
| mencoba         | 0                | 0  | 0  | 0  | 0  |
| membangun       | 0                | 0  | 0  | 0  | 0  |
| sistem          | 0                | 0  | 0  | 0  | 0  |
| deteksi         | 0.602            | 0  | 0  | 0.602 | 0 |
| bahasa          | 0                | 0  | 0  | 0  | 0  |
| indonesia       | 0                | 0  | 0  | 0  | 0  |
| metode          | 0.301            | 0  | 0  | 0.602 | 0.301 |
| vector          | 0                | 0  | 0  | 0  | 0  |
| space           | 0                | 0  | 0  | 0  | 0  |
| **Document Weight Value** | **0** | **2.408** | **1.806** | **0.301** |

As can be seen in Table 4, document D3 has the highest weighting value among other documents. This shows that document D3 has a high similarity with document Q. We implement Cosine Similarity algorithm to measure the level of similarity between document Q and D3, as shown in Table 5.

**Table 5. Cosine similarity value**

| Term            | nQ | nD3 | (nQ x nD3) | (nQ)^2 | (nD3)^2 |
|-----------------|----|-----|------------|--------|---------|
| fenomena        | 1  | 0   | 0          | 1      | 0       |
| plagiarisme     | 2  | 0   | 0          | 4      | 0       |
| spesifik        | 1  | 0   | 0          | 1      | 0       |
| dunia           | 1  | 0   | 0          | 1      | 0       |
| akademis        | 1  | 0   | 0          | 1      | 0       |
| mahasiswa       | 1  | 1   | 1          | 1      | 1       |
| menyelesaikan   | 1  | 0   | 0          | 1      | 0       |
| tugas           | 2  | 3   | 6          | 4      | 9       |
| kuliah          | 1  | 0   | 0          | 1      | 0       |
| tersedianya     | 1  | 0   | 0          | 1      | 0       |
| fasilitas       | 1  | 0   | 0          | 1      | 0       |
| menyalin        | 1  | 0   | 0          | 1      | 0       |
| teks            | 2  | 1   | 2          | 4      | 1       |
| menaruh         | 1  | 0   | 0          | 1      | 0       |
| salinan         | 1  | 0   | 0          | 1      | 0       |
| dokumen         | 3  | 1   | 3          | 9      | 1       |
| alasan          | 1  | 0   | 0          | 1      | 0       |
| penulis         | 1  | 0   | 0          | 1      | 0       |
| mencoba         | 1  | 0   | 0          | 1      | 0       |
| membangun       | 1  | 0   | 0          | 1      | 0       |
| sistem          | 1  | 0   | 0          | 1      | 0       |
| deteksi         | 1  | 0   | 0          | 1      | 0       |
| bahasa          | 1  | 0   | 0          | 1      | 0       |
| indonesia       | 1  | 0   | 0          | 1      | 0       |
Based on the Cosine Similarity formula in Equation (4), the result is:

\[
\text{Cosine Similarity} = \frac{12}{\sqrt{45} \times \sqrt{24}} = 0.365
\]

The level of similarity between document \( Q \) and document \( D3 \) is 0.365.

We developed a web-based interface and implemented the algorithms to find content similarity among documents in local database. The system can recognize the order in which documents appear, in this case the time of publication. Only documents that have older publication time are compared to the tested document. The web-based interface provides Add Document page, Administration page, Sorting Document page and Similarity page. Figure 3 shows Add Document page which is used to add a new document to be tested. Administration page as shown in Figure 4 is used to check similarity or delete a document. Figure 5 shows Sorting Document page which clustering most related document, and Figure 6 shows Similarity page which compare the tested document with the most related documents. Our functional testing shows that the system is able get the input, process the document and display the results as expected.
**Figure 3.** Add document page

**Figure 4.** Administration page

**Figure 5.** Sorting Document page
4. Conclusion
A system to detect content similarity in documents that were stored in local database was built in this study. Considering the number of stored documents, the system implemented two step processes by clustering documents to find most related documents, and then finding content similarity among the selected documents. The system was able to recognize the order in which documents appeared, in this case the time of publication. Only documents that had older publication time were compared to the tested document. The functional testing showed that the system was able get the input, processed the document and displayed the results as expected. The system was successfully clustering documents and detecting content similarity by implementing TF-IDF and Cosine Similarity algorithms. Future research can be developed using other weighting methods for large-scale documents.

Acknowledgment
This research is funded by Lab Based Education Grant 2019, Faculty of Engineering, Hasanuddin University.

References
[1] K. Vani and D. Gupta, “Study on Extrinsic Text Plagiarism Detection Techniques and Tools,” J. Eng. Sci. Technol. Rev., vol. 9, no. 3, pp. 99–105, 2016.
[2] R. E. Putri and A. P. Siahaan 2017 Examination of Document Similarity using Rabin-Karp Algorithm, International Journal of Recent Trends in Engineering and Research, vol. 03, no. 08, pp. 196-201.
[3] M. Oghbaie and M. M. Zanjireh 2018 Pairwise document similarity measure based on present term set Journal of Big Data, vol. 5, no. 52, pp. 1-23.
[4] D. D. Sinaga and S. Hansun 2018 Indonesian Text Document Similarity Detection System using Rabin-Karp and Confix-Stripping Algorithms, International Journal of Innovative Computing, Information and Control, vol. 14, no. 5, pp. 1893-1903.
[5] A. M. El Tahir Ali, H. M. Dahwa Abdulla, and V. Snášel, “Overview and comparison of plagiarism detection tools,” CEUR Workshop Proc., vol. 706, pp. 161–172, 2011.
[6] “PlagAware.” [Online]. Available: https://www.plagaware.com/. [Accessed: 09-Apr-2019].
[7] “PlagScan: Online Plagiarism Checking.” [Online]. Available: https://www.plagscan.com. [Accessed: 09-Apr-2019].
[8] “CheckForPlagiarism.net.” [Online]. Available: https://www.checkforplagiarism.net/. [Accessed: 09-Apr-2019].

[9] R. Nabil, L. Amine, A. Issam, B. L. EL Habib and L. El Houssine 2015 A New Approach that improves TF-IDF Weighting Measure, *International Journal of Information and Communication Technology*, vol. 5, no. 10, pp. 1-10.

[10] C. Breitinger, B. Gipp and S. Langer 2015 Research-paper recommender systems: a literature survey, *International Journal on Digital Libraries*, vol. 17, no. 4.

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