 ISSN 2714-7533

TF-IDF Implementation for Similarity Checker on The Final Project Title

Dwiny Meidelfi [a,1,*], Yulherniwati [a], Indri Rahmayuni [a], Taufik Hidayat [a], Dikky Chandra [b]

[a] Department of Information Technology, Politeknik Negeri Padang, West Sumatera, Indonesia
[b] Department of Electronics Engineering, Politeknik Negeri Padang, West Sumatera, Indonesia

[1] dwinymeidelfi@pnp.ac.id
* corresponding author

ARTICLE INFO

Article history
Received February 10, 2021
Revised March 7, 2021
Accepted April 10, 2021

Keywords
Cosine Similarity
TF-IDF
Final project
Software enginering technology
Politeknik Negeri Padang

ABSTRACT

Students of the Software Engineering Technology Study Program, Department of Information Technology, Politeknik Negeri Padang, is required to compile a final project to complete their study period. In the implementation of the final project, several parties have involved such as the KBK team whose job is to check whether the proposed title is appropriate or not. The main issue undertaken by KBK is whether the title submitted has been used or not. The method used by KBK in checking the availability of titles was by looking at the titles of the final projects that have been submitted by previous students. The examination process carried out by the KBK, it took a long time. By utilizing the Cosine Similarity algorithm and TF-IDF, it is expected that it will make it easier for KBK to check the availability of final project titles. Cosine similarity is a method used to calculate the degree of similarity between 2 or more documents. While the TF-IDF Algorithm is a method used to weight a word in a document. The object of testing in this study was the title of the student's final project. The process of calculating the level of similarity of documents started from the preprocessing stage, then proceeds with weighting using TF-IDF and calculating the level of similarity used the Cosine Similarity algorithm. The final result found that system could calculate the degree of similarity of the title of the student's final project. From the results of testing the process of calculating the degree of similarity of titles using the cosine similarity algorithm can be undertaken quickly.

1. Introduction

The development of information and communication technology provides ease of accessing and looking for information through the internet. The development of information technology does not merely has a positive impact, but also has a negative impact. Plagiarism or also called plagiarism is the act of taking other people's work, whether in the form of ideas, opinions, and ideas belonging to other people and then making it as your own [1], [2]. Plagiarism can be undertaken by anyone everywhere [3][4]. Such as the act of plagiarism committed by students when making a lecture assignment who accidentally takes someone else's writing or ideas and forgets to provide or include the source. This action can not only occur during the making of college assignments, but can also occur when making the final project. The final project is a mandatory course that must be completed by every Undergraduate student of Software Engineering Study Program Applied to end its study period at Politeknik Negeri Padang and obtain an applied bachelor's degree [5]. In the management

of the final project, there are procedures that must be met and determined by each study program. Currently, the implementation of the final project at the Software Engineering Applied Undergraduate Study Program has been computerized. In the process of submitting the title of the final project, students are asked to fill out a form via google form. This is still inefficient and still provides opportunities for students to commit plagiarism. Plagiarism can occur because Google Forms does not provide a feature to detect pre-existing titles. To overcome this issue, the KBK team must check the existing titles one by one, even it has taken a long time. Dealing with these problems, it is necessary to build an Information System for Submission Title of Final Project that can be used optimally and in accordance with existing business processes, thus, it can meet user needs, especially related to plagiarism and the final project management process for students at the Software Engineering Applied in Undergraduate Study Program.

Information System of Submission Title for Final Project can detect the degree of similarity of the titles submitted by students using the Cosine Similarity and TF-IDF algorithms. Cosine Similarity is a method used to compare the level of similarity between two documents [6], [7]. However, the TF-IDF algorithm is the steps or method used to calculate the weight of a word (term) to the document. This method is known to be efficient, easy and has accurate results [8], [9].

2. Material and Method

The aim of this study was to find out the degree of similarity in the title of the final project submitted by students of the Applied Bachelor of Software Engineering Technology study program by applying the Cosine Similarity and TF-IDF algorithms. The following is a study that uses the Cosine similarity Algorithm. The implementation of the Cosine Similarity Algorithm for Detection of Content Similarities in Research and Community Service Information Systems [10]. This study utilized the Cosine Similarity algorithm to check the level of similarity in the content of research proposals and community service which submitted. The Cosine Similarity algorithm was used to calculate the abstract similarity between the proposed proposal and the existing proposal. The findings of this study were applications that facilitated for LPPM Pamulang University to manage data on research activities and community service. The application made was also equipped with a proposal substance checking feature to detect the level of similarity, so that the substance of the proposal will be more varied. The application of the Cosine Similarity algorithm was also used in the research of Eka [11]. This study has been focused on essay assessment. The purpose of this study was to save time on exams and make it easier for lecturers to assess student answers. The result of this study indicated that CBT website had an automatic assessment feature through the Cosine Similarity algorithm. The way of the system works was to compare student answers with the answer key using the Cosine Similarity algorithm. Research conducted by Loura Yasni also applied the Cosine Similarity and TF-IDF algorithms to determine the advisor of the final project supervisor [12].

In this study, the variable parameters used were lecturer data, student data, lecturers' areas of expertise, final assignment titles and abstracts that have been mentored by lecturers, final project titles, final project topics and student final project abstracts. The steps taken were preprocessing, TF-IDF weighting and calculation of Cosine Similarity. The TF-IDF algorithm is used to weight each word from the preprocessing results. While the Cosine Similarity algorithm is used to calculate the level of similarity between two objects. Based on the test results, it can be concluded that the system created has a fairly good performance, namely the results of precision and recall tests have an average performance of 0.74 and 1. The application of the Cosine Similarity algorithm is also used for the document archive system at Sultan Agung Islamic University [13]. In this study, Cosine Similarity was used to assist in looking for archives. The Cosine Similarity algorithm is used to compare the searched word with the title and content contained in the archive. The purpose of this study was to

make it easier for the General Administration Bureau of Sultan Agung Islamic University Semarang in managing data, and reducing the occurrence of data loss or data corruption. The final result of this study indicated that an archival information system has been equipped with an archive search feature. This system searched for digital archives by calculating the weight of the text of the document title and the contents of the document and then it will be weighted and matched with the data in the database. Based on the test results, it can be concluded that the cosine similarity algorithm is able to find documents with a high level of similarity so that it can find relevant documents, this is evidenced by the performance measurement of the Cosine Similarity Algorithm showing a precision of 88.8% and a recall of 76.1%. Research conducted by Zainal Abidin also applies the Cosine Similarity algorithm and TF-IDF to detect damage for operating system [14]. In this study, the cosine similarity algorithm was used to compare the sentences typed by the user with the knowledge base stored in the database. The findings of this study were a website that aimed to facilitate users in overcoming problems experienced when using the operating system. From this research, it can be concluded that the cosine similarity algorithm had quite good accuracy of 70%.

Research related to Cosine Similarity does not merely discuss the benefits or use of the Cosine Similarity algorithm, but also discusses the comparison of the Cosine Similarity algorithm with other algorithms. Dealing with previous research, Ahmad Dzul Fikri compared the Dice Similarity Method with Cosine Similarity Using Query Expansion in Searching for Ayatul Ahkam in Indonesian Translation of the Qur'an with an accuracy of 85% [15]. Based on other research, namely the Comparison of Cosine Similarity and Jaccard Similarity Methods for Automatic Assessment of Short Answers conducted by Uswatun Hasanah, the Cosine Similarity algorithm got the highest score of 0.62 [16]. Ogie Nurdiana also conducted a comparison of the Cosine Similarity Method with the Jaccard Similarity Method in the Search Application for the Alqur'an Translation in Indonesian. The Cosine Similarity algorithm obtained the highest score, namely 47%, while the Jaccard algorithm got 21% [17].

The Cosine Similarity method is a method for calculating the level of similarity between two or more objects or documents [18]. This calculation uses two objects (D1 and D2) to calculate the similarity between documents expressed in a vector using keywords. The advantage of Cosine Similarity is that the process of calculating the level of similarity between documents can be done quickly [13]. Cosine Similarity is part of text mining. Text mining is the process of mining data in the form of text where the data source is usually obtained from documents. The purpose of text mining is to find words that can represent the contents of the document so that an analysis of the relationship between documents can be carried out [19]. Here is the formula for Cosine Similarity [6] [20]

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

Description:
A = vector A
B = vector B
A.B = dot product between vector A and vector B
|A| = the length of vector A
|B| = the length of vector B
|A| |B| = cross product between |A| and |B|

The TF-IDF method is a way to weight the relationship of a word (term) contained in a document [21]. The method of TF-IDF is a method that combines two concepts for calculating weights. Term frequency (TF) is the frequency with which a word appears in the document. While Inverse Document Frequency (IDF) is the inverse frequency of documents that contain the word [18]. TF-

IDF is also a method that is known to be efficient, easy and has accurate results [22]. Here is the formula for TF-IDF [6] :

$$tfidf_{t,d} = tf_{t,d}\ x\ idf_t \quad (2)$$

$$idf_t = \log(\frac{N}{df_t}) \quad (3)$$

Description:
d = document of-d
t = the word of-t from the keyword
tfidf = weight of the d document to the word of -t
tf = many words that must be found in the document
idf = Inversed Document Frequency
N = many Documents
df = many documents contain the word that we are looking for

Preprocessing is the process of preparing raw data before performing other processes [23], [24]. Usually preprocessing is undertaken by eliminating inappropriate data or converting it into a form that is easier for the system to process [25].

1) Case Folding

Case folding is the process of converting every capital letter into lowercase in a sentence [26]. Case folding is conducted because not all documents are consistent with the use of capital letters.

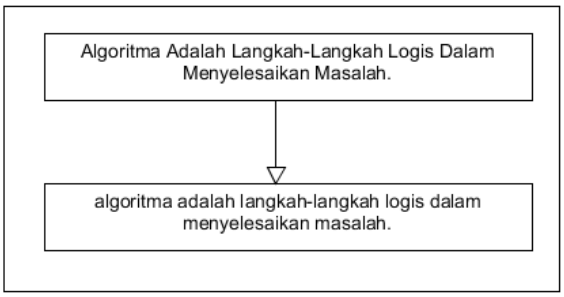

Fig. 1 Process of case folding

2) Tokenizing

Tokenizing is the process of cutting the input string in accordance with each word that composes it [27]. In tokenizing, some characters are also considered as word separators, such as whitespace, enter and period.

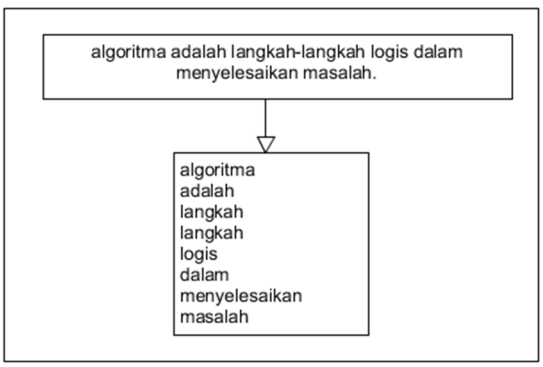

Fig. 2 Process of tokenizing

3) Filtering

Filtering is the step to remove unnecessary words [28][29]. Usually the words that are used as a stop list are stored in an array or text file. If the word that appears is the same as the word in the stop list, then the word will be removed from the document.

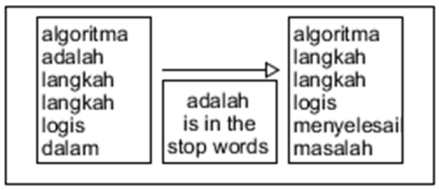

Fig. 3 The process of filtering

4) Stemming

Stemming is the process of converting words into basic words. The process carried out is like removing the suffix from each word.

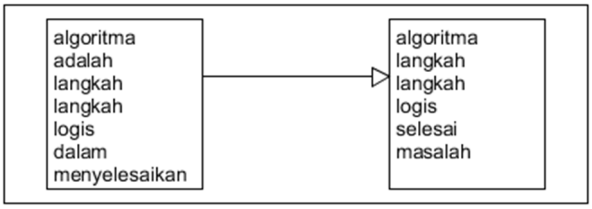

Fig. 4 The process of stemming

The following are the steps taken to calculate the degree of similarity between two documents:

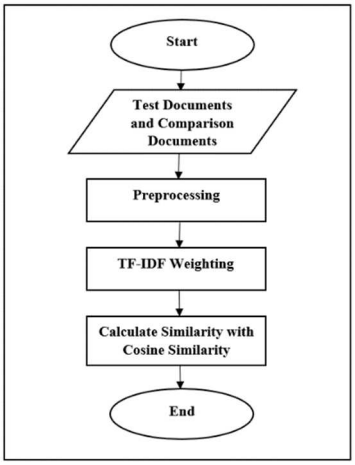

Fig. 5 Steps to calculate document similarity level [6]

1) Test Documents and Comparison Documents

The first step that needs to be undertaken is to determine the test documents and comparison documents. A test document is a document whose similarity level is measured. While the comparison document is a document used to measure the similarity of the test document. The test document used for this final project is the title of the final project of the TRPL Applied Undergraduate PS students. Whereas for the comparative documents consist of the titles of the final project students of the Informatics Management study program at the Politeknik Negeri Padang.

2) Preprocessing

Preprocessing consists of several stages, namely case folding, tokenizing, filtering, steaming. In this study, the steaming process was carried out by utilizing the Literature module for the PHP programming language.

3) The weighting of TF-IDF

This stage is the stage of calculating word weighting to calculate the frequency of occurrence of each word in the test document in each document in the dataset. The TF-IDF weighting stages can be seen in Figure 6.

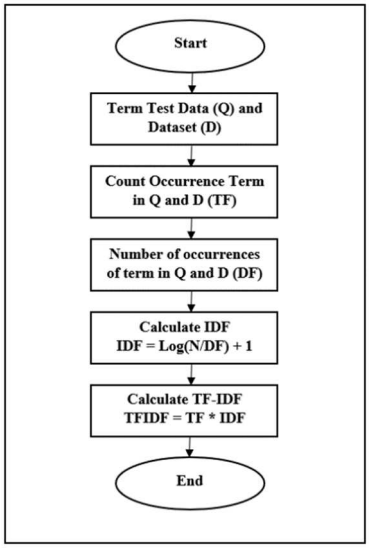

Fig. 6 The Step of Weighting TF-IDF [6]

4) Calculating Cosine Similarity

After getting the results of the TF-IDF weighting, the next step is to calculate the level of similarity between documents. To calculate the level of document similarity, it can use through Cosine Similarity. The stages of the Cosine Similarity calculation can be seen in Figure 7.

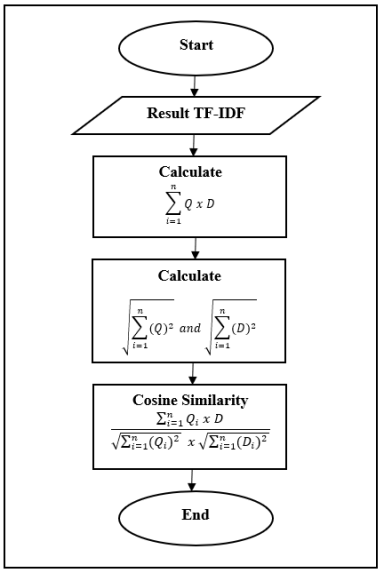

Fig. 7 The stage of calculating the similarity with cosine similarity [6]

The result indicates that web-based information system are equipped with features to calculate the degree of similarity between the proposed title and the existing title.

3. Result and discussion

3.1. The Process of Calculation

1) Determine the comparison document(Q) and test document)

Q = PERANCANGAN DAN IMPLEMENTASI SISTEM PENGAJUAN PROPOSAL TUGAS AKHIR (STUDI KASUS TEKNOLOGI INFORMASI PNP)
D = SISTEM PENDETEKSI KEMIRIPAN JUDUL PROYEK AKHIR MENGGUNAKAN ALGORITMA TF-IDF DAN COSINE SIMILIARITY PADA SISTEM PENGAJUAN JUDUL PROYEK AKHIR SARJANA TERAPAN TEKNOLOGI REYAKASA PERANGKAT LUNAK (STR TRPL)

2) Do Preprocessing

Q = perancangan dan implementasi sistem pengajuan proposal tugas akhir (studi kasus teknologi informasi pnp)

D = sistem pendeteksi kemiripan judul proyek akhir menggunakan algoritma tf-idf dan cosine similiarity pada sistem pengajuan judul proyek akhir sarjana terapan teknologi reyakasa perangkat lunak (str trpl)

3) Extract every word in both documents

From the two documents extracted the words contained in it, so then it obtains the following results:

Table 1. Word From Each Document

Category	Words
Document 1	Rancang
	implementasi
	sistem
	aju
	proposal
	tugas
	akhir
	studi
	kasus
	teknologi
	informasi
	pnp
Document 2	sistem
	deteksi
	mirip
	judul
	proyek
	akhir
	guna
	algoritma
	tfidf
	cosine

Category	Words
	similarity
	sistem
	aju
	judul
	proyek
	akhir
	sarjana
	terap
	teknologi
	rekayasa
	perangkat
	lunak
	str
	trpl

4) Count TF

TF or Term Frequency is the frequency with which a word appears in a document.

Table 2. Value Of TF From Every Word

Words	Q - TF	D - TF
ancang	1	0
implementasi	1	0
sistem	1	2
aju	1	1
proposal	1	0
tugas	1	0
akhir	1	2
studi	1	0
kasus	1	0
teknologi	1	1
informasi	1	0
pnp	1	0
deteksi	0	1
mirip	0	1
judul	0	2
proyek	0	2
guna	0	1
algoritma	0	1
tfidf	0	1
cosine	0	1
similarity	0	1
sarjana	0	1
terapan	0	1
rekayasa	0	1
perangkat	0	1

Words	Q - TF	D - TF
lunak	0	1
str	0	1
trpl	0	1

5) Count DF

DF is the number of occurrences of words in documents Q and D. It is due to the examination process is carried out on two documents, the number of DFs is a maximum of 2 and a minimum of 1.

Table 3. The Value Of DF From Every Word

Words	Q - TF	D - TF	DF
ancang	1	0	1
implementasi	1	0	1
sistem	1	2	2
aju	1	1	2
proposal	1	0	1
tugas	1	0	1
akhir	1	2	2
studi	1	0	1
kasus	1	0	1
teknologi	1	1	2
informasi	1	0	1
pnp	1	0	1
deteksi	0	1	1
mirip	0	1	1
judul	0	2	1
proyek	0	2	1
guna	0	1	1
algoritma	0	1	1
tfidf	0	1	1
cosine	0	1	1
similarity	0	1	1
sarjana	0	1	1
terapan	0	1	1
rekayasa	0	1	1
perangkat	0	1	1
lunak	0	1	1
str	0	1	1
trpl	0	1	1

6) Count IDF

For example, for the words “ancang” and “implementation”

$$IDF\ (\text{"}ancang\text{"}) = Log\left(\frac{2}{1}\right) + 1 = 1{,}301029996$$

$$IDF\ (\text{implementation}) = Log\left(\frac{2}{1}\right) + 1 = 1{,}301029996$$

7) Count TF-IDF

Further, calculate TF-IDF which is the product of TF and IDF

$TF - IDF$ ("ancang") = $1 * 1{,}301029996 = 1{,}301029996$

$TF - IDF$ ("implementation") = $1 * 1{,}301029996 = 1{,}301029996$

8) The process of calculating cosine similarity

The next step is to find out the Cosine Similarity value using the following formula (1). For the calculation of cosine similarity, it is undertaken by dividing it into several steps, namely finding DxQ, it is squared TF-IDF, calculating the value of cosine similarity.

9) Count DxQ

Finding DxQ and it can be undertaken by multiplying the TF-IDF scalars of each D against TF-IDF Q, then looking for the total. For example :

DxQ ("ancang", D *on* Q) = $TF - IDF(D,$"ancang")* TF-IDF(Q,"ancang") = $1{,}30103 * 0 = 0$

DxQ ("implementation", D *on* Q) = $TF - IDF$ (D, "implementation") $* TF -$ IDF (Q, "implementation") = *1,30103 * 0*

Table 4. The Calculation Result of DxQ

Words	Q - TF-IDF	D - TF-IDF	DxQ
ancang	1,301029996	0	0
implementasi	1,301029996	0	0
sistem	1	2	2
aju	1	1	1
proposal	1,301029996	0	0
tugas	1,301029996	0	0
akhir	1	2	2
studi	1,301029996	0	0
kasus	1,301029996	0	0
teknologi	1	1	1
informasi	1,301029996	0	0
pnp	1,301029996	0	0
deteksi	0	1,301029996	0
mirip	0	1,301029996	0
judul	0	2,602059991	0
proyek	0	2,602059991	0
guna	0	1,301029996	0
algoritma	0	1,301029996	0
tfidf	0	1,301029996	0
cosine	0	1,301029996	0
similarity	0	1,301029996	0
sarjana	0	1,301029996	0
terapan	0	1,301029996	0
rekayasa	0	1,301029996	0
perangkat	0	1,301029996	0
lunak	0	1,301029996	0
str	0	1,301029996	0
trpl	0	1,301029996	0
Total			**6**

10) Calculate the square of TF-IDF

Find the square of TF-IDF. Then find the total, and the total is the square root. As an example:
$Square\ TF-IDF$ ("ancang", Q) =1,301029996 ^ 2 = 1,69267905

$Square\ TF-IDF$ ("ancang", D) =0 ^ 2 = 0

Table 5. The Square Result of TF-IDF

Words	Q - TF-IDF ^ 2	D - TF-IDF ^ 2
ancang	1,69267905	0
implementasi	1,69267905	0
sistem	1	4
aju	1	1
proposal	1,69267905	0
tugas	1,69267905	0
akhir	1	4
studi	1,69267905	0
kasus	1,69267905	0
teknologi	1	1
informasi	1,69267905	0
pnp	1,69267905	0
deteksi	0	1,69267905
mirip	0	1,69267905
judul	0	6,770716198
proyek	0	6,770716198
guna	0	1,69267905
algoritma	0	1,69267905
tfidf	0	1,69267905
cosine	0	1,69267905
similarity	0	1,69267905
sarjana	0	1,69267905
terapan	0	1,69267905
rekayasa	0	1,69267905
perangkat	0	1,69267905
lunak	0	1,69267905
str	0	1,69267905
trpl	0	1,69267905
Total	**17,5414324**	**47,23893909**
Square	**4,188249324**	**6,873058933**

11) Count the value of Cosine Similarity

Find Cosine Similarity Results for each D of [total DxQ] for each D is divided by the product of [total square root of TF-IDF square] belonging to D and [total square root of TF-IDF square] it belongs to Q

$$Cosine\ Similarity\ (D) = 6\ \div (4{,}1882493 * 6{,}87305893) =\ 0{,}2084$$

Therefore, the degree of similarity between D and Q is 20,84%

Figure 8 shows the results of calculation of the cosine similarity and the percentage of similarity results from existing titles.

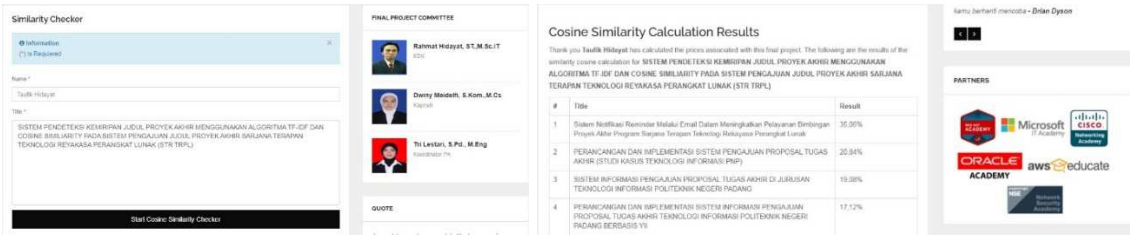

Fig. 8 Page of calculation results on cosine similarity

4. Conclusion

The method of Cosine Similarity and TF-IDF could be used to calculate the degree of similarity in the title of a student's final project. The number of test documents greatly affected the processing time for calculating the level of similarity between documents. The more test documents, the longer the process of calculating the level of similarity between documents.

References

[1] KBBI, "Kamus Besar Bahasa Indonesia (KBBI)," 2020.

[2] S. Awasthi, "Plagiarism and academic misconduct: A systematic review," *DESIDOC Journal of Library and Information Technology*, vol. 39, no. 2. 2019, doi: 10.14429/djlit.39.2.13622.

[3] "Plagiarism in higher education environment: causes and solutions," Rwandan J. Educ., vol. 4, no. 2, 2018.

[4] Foltýnek et al., "Testing of support tools for plagiarism detection," *Int. J. Educ. Technol. High. Educ.*, vol. 17, no. 1, 2020, doi: 10.1186/s41239-020-00192-4.

[5] D. Meidelfi, Yulherniwati, F. Sukma, D. Chandra, and A. H. Soleliza Jones, "The implementation of SAW and BORDA method to determine the eligibility of students' final project topic," *Int. J. Informatics Vis.*, vol. 5, no. 2, 2021, doi: 10.30630/joiv.5.1.447.

[6] M. Z. Naf'an, A. Burhanuddin, and A. Riyani, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," *J. Linguist. Komputasional*, vol. 2, no. 1, pp. 23–27, 2019, doi: 10.26418/jlk.v2i1.17.

[7] N. A. Rakhmawati, A. A. Firmansyah, P. M. Effendi, R. Abdillah, and T. A. Cahyono, "Auto Halal detection products based on euclidian distance and cosine similarity," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, 2018, doi: 10.18517/ijaseit.8.4-2.7083.

[8] R. T. Wahyuni, D. Prastiyanto, and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," *J. Tek. Elektro*, vol. 9, no. 1, pp. 18–23, 2017.

[9] B. Hashemzadeh and M. Abdolrazzagh-Nezhad, "Improving keyword extraction in multilingual texts," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 6, 2020, doi: 10.11591/ijece.v10i6.pp5909-5916.

[10] F. A. Nugroho, F. Septian, D. A. Pungkastyo, and J. Riyanto, "Penerapan Algoritma Cosine Similarity untuk Deteksi Kesamaan Konten pada Sistem Informasi Penelitian dan Pengabdian Kepada Masyarakat," *J. Inform. Univ. Pamulang*, vol. 5, no. 4, p. 529, 2021, doi: 10.32493/informatika.v5i4.7126.

[11] E. L. Amalia, A. J. Jumadi, I. A. Mashudi, and D. W. Wibowo, "Analisis Metode Cosine Similarity Pada Aplikasi Ujian Online Otomatis (Studi Kasus JTI POLINEMA)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, p. 343, 2021, doi: 10.25126/jtiik.2021824356.

[12] L. Yasni, I. M. I. Subroto, and S. F. C. Haviana, "Implementasi Cosine Similarity Matching Dalam Penentuan Dosen Pembimbing Tugas Akhir," *Transmisi*, vol. 20, no. 1, p. 22, 2018, doi: 10.14710/transmisi.20.1.22-28.

[13] D. Kurniadi, S. F. C. Haviana, and A. Novianto, "Implementasi Algoritma Cosine Similarity pada sistem arsip dokumen di Universitas Islam Sultan Agung," *J. Transform.*, vol. 17, no. 2, p. 124, 2020, doi: 10.26623/transformatika.v17i2.1613.

[14] A. Z. Z. Abidin and A. Sukmadinata, "Sistem Deteksi Kerusakan pada Sistem Operasi Menggunakan Metode TF - IDF dan Cosine Similarity," *J. Ilm. Inform.*, vol. 8, no. 2, pp. 6–11, 2020.

[15] A. D. Fikri, "Perbandingan Metode Dice Similarity Dengan Cosine Similarity Menggunakan Query Expansion Pada Pencarian Ayatul Ahkam Dalam Terjemah Alquran Berbahasa Indonesia Skripsi" pp. 1–73, 2019.

[16] U. Hasanah and D. A. Muatiara, "Perbandingan metode cosine similarity dan jaccard similarity untuk penilaian otomatis jawaban pendek," *Semin. Nas. Sist. Inf. dan Tek. Inform.*, no. 2019: SENSITIF 2019, pp. 1255–1263, 2019.

[17] O. Nurdiana, J. Jumadi, and D. Nursantika, "Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia," *J. Online Inform.*, vol. 1, no. 1, p. 59, 2016, doi: 10.15575/join.v1i1.12.

[18] M. M. Sya'bani and R. Umilasari, "Penerapan Metode Cosine Similarity dan Pembobotan TF / IDF pada Sistem Klasifikasi Sinopsis Buku di Perpustakaan Kejaksaan Negeri Jember," *Justindo (J. Sist. Teknol. Indones.*, vol. 3, no. 1, pp. 31–42, 2018.

[19] Z. Mujahidin, "Implementasi Metode Rabin Karp Untuk Mendeteksi Tingkat Kesamaan Dua Dokumen," *J. Tugas Akhir*, 2013.

[20] D. Soyusiawaty and Y. Zakaria, "Book data content similarity detector with cosine similarity (case study on digilib.uad.ac.id)," 2018, doi: 10.1109/TSSA.2018.8708758.

[21] R. A. Sasmita and A. Z. Falani, "Pemanfaatan Algoritma TF/IDF Pada Sistem Informasi Ecomplaint Handling," *J. Link*, vol. 27, no. 1, pp. 27–33, 2018.

[22] D. Asmarajati, "Analisis Perbandingan Algoritma Tf-Idf Dengan Sql Query Untuk Kasus Pencarian Pada Sistem Informasi Dokumentasi Arsip (Sidokar)," *Device*, vol. 10, no. 1, pp. 1–8, 2020, doi: 10.32699/device.v10i1.1478.

[23] S. W. Kim and J. M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Comput. Inf. Sci.*, vol. 9, no. 1, 2019, doi: 10.1186/s13673-019-0192-7.

[24] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, 2018, doi: 10.5120/ijca2018917395.

[25] S. Mujilahwati, "Pre-Processing Text Mining Pada Data Twitter," *Semin. Nas. Teknol. Inf. dan Komun.*, vol. 2016, no. Sentika, pp. 2089–9815, 2016.

[26] F. Alzami, E. D. Udayanti, D. P. Prabowo, and R. A. Megantara, "Document Preprocessing with TF-IDF to Improve the Polarity Classification Performance of Unstructured Sentiment Analysis," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, 2020, doi: 10.22219/kinetik.v5i3.1066.

[27] G. Mediamer, adiwijaya@telkomuniversity ac id Adiwijaya, and S. Al Faraby, "Development of rule-based feature extraction in multi-label text classification," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 4, 2019, doi: 10.18517/ijaseit.9.4.8894.

[28] S. M. H. Dadgar, M. S. Araghi, and M. M. Farahani, "A novel text mining approach based on TF-IDF and support vector machine for news classification," 2016, doi: 10.1109/ICETECH.2016.7569223.

[29] P. Sun, L. Wang, and Q. Xia, "The Keyword Extraction of Chinese Medical Web Page Based on WF-TF-IDF Algorithm," in Proceedings - 2017 *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, *CyberC 2017*, 2017, vol. 2018-January, doi: 10.1109/CyberC.2017.40.

[30] Reni Nursyanti, R.Yadi Rakhman Alamsyah, and S. Perdana, "Perancangan Aplikasi Berbasis Web Untuk Membantu Pengujian Kualitas Kain Tekstil Otomotif," *J. Sist. Inf. dan Telemat.*, vol. 10, no. 1, pp. 5–13, 2019.