Available online at JDR Website: http://journal.unublitar.ac.id/jdr

Journal Of Development Research, 5 (2), November 2021, Pages 106-114

DOI: https://doi.org/10.28926/jdr.v5i2.150

Semantic Search for Scientific Articles by Language Using Cosine Similarity Algorithm and Weighted Tree Similarity

Muhamad Aldi Rifai[(1)], Indra Gita Anugrah[(2)]

Universitas Muhammadiyah Gresik, Indonesia

E-mail: [(1)]aldi_170602@umg.ac.id, [(2)]indragitaanugrah@umg.ac.id

Received: 12 May 2021; Revised: 21 October 2021; Accepted: 15 November 2021

Abstract

The activity of writing scientific articles by the academic community at universities is one of the activities that is often carried out, but when writing scientific articles problems arise regarding the difficulty of finding ideas, literature studies, and reference sources that you want to use as references when writing. Sometimes when we search in search engines, we have trouble finding the right document because usually, the keywords we are looking for are not in the title section but other parts of the structure. Since most search engines only match titles, other structures are usually excluded from matching. So that the search results that we do sometimes don't match what we want. In addition, each scientific article has many language differences in its structure as contained in the abstract section. To detect similarities through the structure of scientific articles, an algorithm is used namely weighted tree similarity, while the algorithm used to detect language uses N-grams and to check the level of similarity in keyword text with text in scientific articles using cosine similarity. The test results show that by combining the N-Gram, Cosine Similarity Algorithm, and Weighted Tree Similarity, the average value of accuracy, precision, and recall is quite high, which is above 0.9.

Keywords: Semantic Search; Scientific Articles; Cosine Similarity; Weighted Tree Similarity

Introduction

Writing scientific articles is an activity that is often carried out by academics in university, but in writing scientific articles we often have difficulty finding ideas or reference sources in writing scientific articles. Current technological developments make it easier to find referral sources, but often the search results we get are not what we expect, because sometimes the keywords we are looking for are actually in the structure of scientific article content, but sometimes search engines are only able to read the title so that the results the resulting search sometimes does not match or even excludes scientific articles because there are no keywords in the title.

In addition to the titles of scientific articles, there are other general structures such as abstracts, keywords, authors, and years that can be used to matching based on the keywords that are searched. For that, we need a process of retrieving data from the internet or what is called Web Scraping. Then for each structure in the scientific article, a weighting will be given based on the level of a structure against the search results.

Searches on search engines mostly use information retrieval, while searches using information retrieval must go through several processes, one of which is the stemming process. Stemming is the process of changing sentences into basic words by removing affixes. Many scientific articles are found in Indonesian but the abstracts are in a different language, namely English. Therefore, language detection is needed to determine the stemming algorithm to be used, because the stemming algorithm in Indonesian and English is different.

Based on the problem of finding reference sources in writing a scientific article, a similarity

detection system is needed based on the structure of the scientific article. One of the algorithms used to present the structure of the article is the weighted tree similarity. In a semantic search that uses the weighted tree similarity algorithm, metadata is arranged based on a tree that has labeled nodes, labeled branches, and is weighted (Sarno & Rahutomo, 2015).

Research on the application of the weighted tree similarity algorithm for semantic search resulted in several important conclusions, including the search accuracy using the weighted tree similarity algorithm which was higher than full-text search and ordinary metadata search (Sarno & Rahutomo, 2015). In this study, besides using the weighted tree similarity algorithm, also combines other algorithms, namely cosine similarity to measure the level of similarity and the results are considered effective.

From the research (Wahyuni et al., 2017) using cosine similarity and TF-IDF weighting obtained an accuracy rate of 98%. Furthermore, (Sugiyamto et al., 2014) tried to compare Jaccard and cosine similarity on the document similarity test, with the results showing that the similarity test using the cosine similarity had a higher level of accuracy, namely 0.949808 compared to Jaccard of 0.949077. For language detection in (Zaman et al., 2015) research on language detection systems using N-Gram, the performance is good enough to detect languages with an average F-measure of 0.93.

After the author studies the literature, the algorithms that will be used by the author are N-Gram, Cosine Similarity, and Weighted Tree Similarity. N-Gram is used to represent the language of the sentence before the stemming process is carried out, Cosine Similarity is used to check the similarity of the text, and Weighted Tree Similarity is used to represent the structure of the article and to detect similarities based on structure.

Material and Method

Figure 1 is the steps that the author took to research the creation of a scientific article search system.

Document Scrapping

Web Scraping or Document Scrapping is the process of taking an HTML document from the internet to retrieve certain data from the page for other purposes.

In this study, the authors used scientific article document sources from the internet on the website neliti.com, the website provides various kinds of scientific article documents.

Figure 2 is the flow of the Document Scrapping process, starting from the input of keywords and then the system will process it to obtain data by the desired structure, namely title, abstract, keywords, author, and years later the data is saved to the database.

Language Dectection

In this language detection process, the data in the database is taken and then the language detection process is carried out, for the detection of the author's language using the N-gram algorithm. Figure 3 shows the process of detecting language, in this process, begins by entering a text or sentence then the sentence will be cut using the N-gram concept.

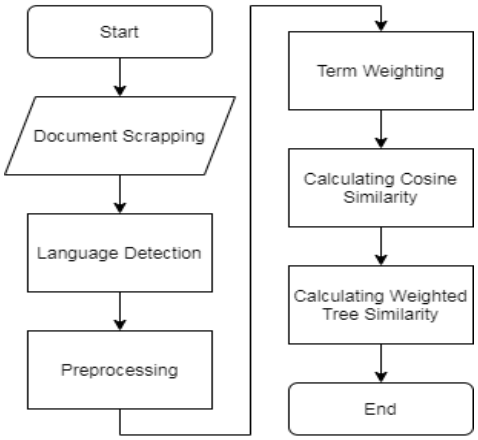

Figure 1. Flowchart Diagram

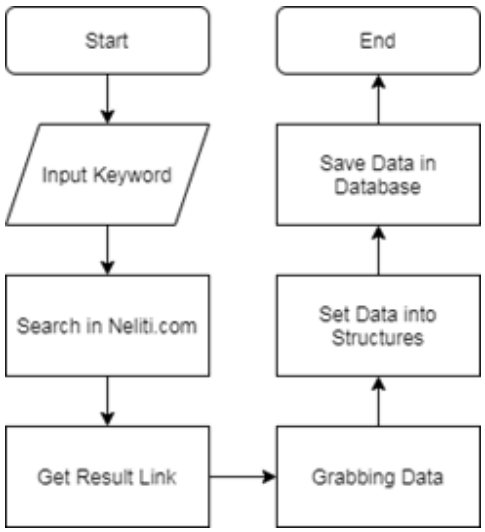

Figure 2. Scraping Process

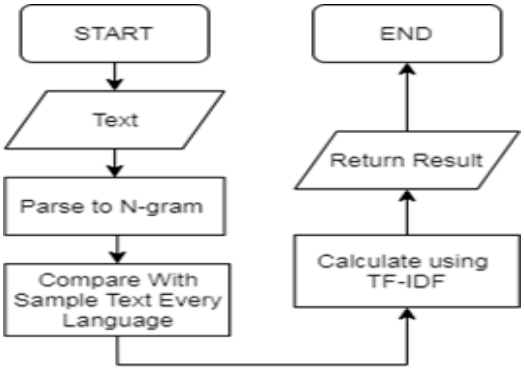

Figure 3. Language Dectection Process

The use of N-gram for language detection is based on the assumption that the N-gram distribution pattern of a language is unique because it is related to the frequency of use of letters, or letter pairs, either vowels or consonants from a language which are generally different from other languages (Zaman et al., 2015).

N-grams are distinguished by the number of character pieces of n. To assist in retrieving word pieces in the form of these letter characters, padding is done with blanks at the beginning and end of a word. For example, the word "TEXT" can be broken down into the following n-grams ("_" represents blank) :

uni-grams : T, E, X, T
bi-grams : _T, TE, EX, XT, T_
tri-grams : _TE, TEX, EXT, XT_
quad-grams : _TEX, TEXT, EXT_
quint-grams : _TEXT, TEXT_

After the text is converted to N-gram, then it is matched with the sample text data in each language. The matching results are calculated using the TF-IDF formula.

Preprocessing

The next stage after the language has been detected is preprocessing. The preprocessing process is carried out so that the data used is clean from noise, has smaller dimensions, and is more structured so that it can be processed further. The preprocessing stage has several processes, namely case folding, stop words removal, tokenizing, and stemming (Hermawan & Bellaniar Ismiati, 2020).

The following are the stages in prepocessing:

1. Case Folding. Case folding is the first process in a series of document preprocessing. In this process all the letters in the document are converted to lowercase. Only letters a through z are accepted (Riyani, 2019).
2. Stopword Removal. The stopword removal stage is the stage of removing unnecessary words from the text. Stopwords are non-descriptive words that can be discarded in a bag-of-words approach (Naf'an et al., 2019). At this stage the author uses the stopword removal function in the stemmer library literature.
3. Stemming. At this stage, the process of returning various word formations is carried out into the same representation. Stemming is where the process of word mapping on a sentence is rewarding to be the original word (without the prefix, suffix, insertion, combination) that is executed specific algorithm (Rofiqi et al., 2019). The stemming stage in this study for Indonesian uses the stemmer literary algorithm, while for English it uses the snowball stemmer algorithm. The results of this stemming process will be followed by the next stage to do word weighting using the tf-idf algorithm (Melita et al., 2018).
4. Tokenizing. Tokenization is the process of dividing text in the form of sentences or paragraphs in a document into certain tokens (Naf'an et al., 2019). At this stage the text will be compiled based on the terms of the stemming results.

Term Weighting

At this stage the search query and scientific article dataset are weighted words or terms to calculate the frequency of appearance of each search query word in each scientific article in the dataset. TFIDF method is a method for calculate the weight of each word the most commonly used in information retrieval. This method is also known to be efficient, easy and have accurate results (Maarif, 2015).

The weighting formula for term weighting for this study uses the TF-IDF formula.

$w_{i,j}$ = number of occurences of i in j

$tf_{i,}$ = number of documents containing i

N = total number of documents

Cosine Similarity

Cosine Similarity measures the similarity between two documents or text. In Cosine Similarity the document or text is considered as a vector. For text matching, the values of vectors A and B is the term-frequency vector of the document (Samuel et al., 2018). Cosine

$$w_{i,j} = tf_{i,j} \; x \; log\left(\frac{N}{df_i}\right) \quad (1)$$

Similarity is used to calculate the similarity of scientific articles, the formula for cosine similarity is as follows:

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

$A =$ Vector A, which will be compared as similar

$B =$ Vector B, which will be compared as similar

$A.B =$ Dot product between vector A and vector B

$|A| =$ Vector length A

$|B| =$ Vector length B

$|A||B| =$ Cross product between A and B

Weighted Tree Similarity

Documents to be calculated for the similarity are represented in a tree that has the characteristics of labeled nodes, labeled branches, and weighted branches (Sarno & Rahutomo, 2015). The article can be split to create a structure. Splits are determined to follow the path that might control of a process (Anugrah et al., 2016). An example of a tree representation in a scientific article in this study can be seen in Figure 3. In Figure 4, the tree structure in scientific articles is divided into 5, namely title, abstract, keywords, authors, and year. Each structure is given a preference weight according to the level of importance of the structure. The weights used in this study are title 0.25 Abstract 0.35 Keywords 0.2 Authors 0.15 and Year 0.05.

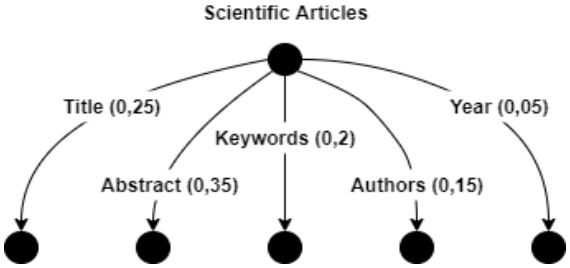

Figure 4. Tree Structure in the Article

Result and Discussion

This chapter discusses the implementation of a search system for scientific articles by language using the cosine similarity algorithm and weighted tree similarity.

Implementation Results

A scientific article search system has been successfully created by following the flow in Figure 1. Starting from the scrapping process with the source from the neliti.com website, the data taken include the title, abstract, keywords, author, and year. Then the language checking process was carried out using the library language detection from Patrick Schur. The results of the detection of the language are used as a reference to the next stage, namely the stemming stage, at the steamming stage if the results of language detection are detected in Indonesian, the algorithm used for the stemming process is the literary algorithm, but if the results are detected in English, the snowball library used is the snowball library porter's algorithm.

After the preprocessing is complete, the next stage is the term weighting stage, after getting the results of term weighting these results will be calculated using the Simillarty Cosain formula in equation 2, after obtaining the Simillarty Cosain Value for each structure, the next calculation will be carried out using the weighted tree similarty algorithm.

Table 1 is the results of experiments using the keyword "sistem" with a total of 140 scientific articles. From the results of the weighted tree similarity, the similarity level in the Extended Weighted Tree Similarity algorithm is determined in the value range 0 to 1 (Suharso et al., 2017). The value of 1 indicates that the scientific article has a high similarity to the keyword, otherwise if the result is 0, the smaller the level of similarity to the keywords.

Implementation in the System

In the search system, there are 3 main pages,

Table 1. Results Of Experiments

Article	Structure	Language Dectection		Term Weighting	Cosain Similarity	WTS
		Id	En			
D1	Title	0,402898185	0,340650202	0,867374435	1	0,8
	Abstract	0,378574402	0,275078044	12,05558032	1	
	Keywords	0,403598015	0,344913151	1,191885526	1	
	Authors	0,525674786	0,38676761	0	0	
	Year	0	0	0	0	
D2	Title	0,444623656	0,312405416	0,867374435	1	0,8
	Abstract	0,381581686	0,272882414	3,214821419	1	
	Keywords	0,413946869	0,317955408	1,191885526	1	
	Authors	0,435047951	0,398517873	0	0	
	Year	0	0	0	0	
D3	Title	0,352258065	0,312525028	1,734748869	1	0,6
	Abstract	0,380478668	0,271540062	6,429642839	1	
	Keywords	0	0	0	0	
	Authors	0,628076464	0,322819594	0	0	
	Year	0	0	0	0	
D4	Title	0,385529954	0,333419355	1,734748869	1	0,8
	Abstract	0,400957336	0,262278876	10,44816961	1	
	Keywords	0,396837861	0,382088285	1,191885526	1	
	Authors	0,650490884	0,563253857	0	0	
	Year	0	0	0	0	
D5	Title	0,461840176	0,334017595	0,867374435	1	0,6
	Abstract	0,35817898	0,267086368	6,429642839	1	
	Keywords	0	0	0	0	
	Authors	0,43916129	0,379516129	0	0	
	Year	0	0	0	0	
...	...	...	...	...	...	...
D140	Title	0,480346476	0,480346476	0	0	0
	Abstract	0,477377732	0,248740895	0	0	
	Keywords	0,48608871	0,324596774	0	0	
	Authors	0,406214856	0,336430897	0	0	
	Year	0	0	0	0	

namely the scrapping page, the preprocessing page, and the search page.

Figure 5 is a scrapping page on this page, users can scrapping data to a research website, and the results of the scrapping will be used as a structured dataset to be saved to the database. Figure 6 is a preprocessing page where the dataset was processed, namely by detecting the language and carrying out the preprocessing stages. Figure 7 is a search page on this page, keywords and scientific articles will be matched using the Simillarty Cosain algorithm and Similarity weighted tree then the calculation values are sorted from the largest, and the results are displayed to the user as shown in Figure 8.

System Testing

To determine the level of accuracy of the research results, in this study the authors used a Confusion matrix. There are 4 terms as a representation of the result of the classification process on confusion matrix. The four terms are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

The values of True Positive and True Negative provide information for the classifier in classifying the data correctly, while False Positive and False Negative provide information when the classifier is wrong in classifying data. (Fibrianda

Figure 5. Scrapping Page

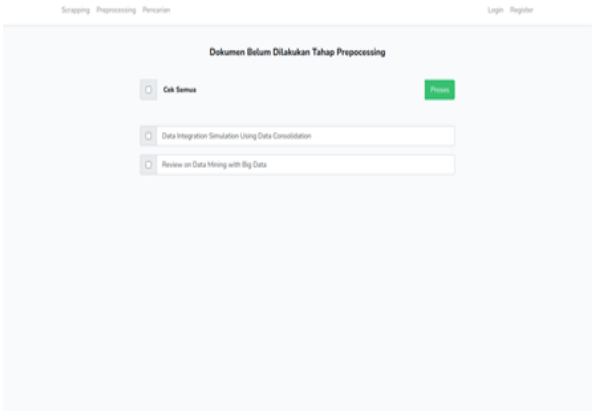

Figure 6. Preprocessing Page

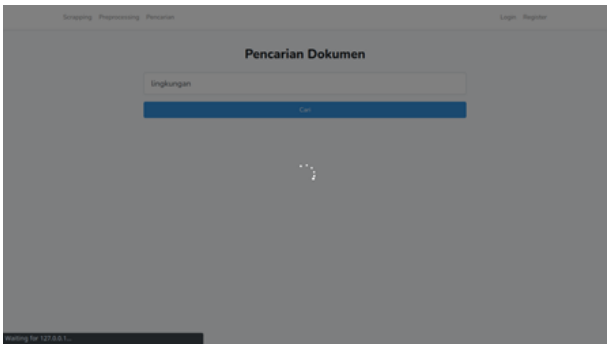

Figure 7. Searching Page

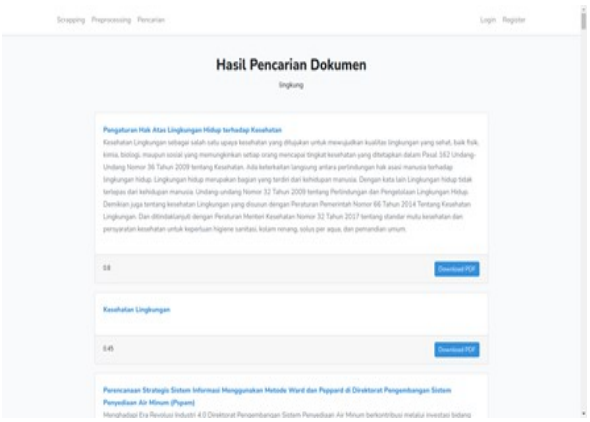

Figure 8. Result Page

& Bhawiyuga, 2018). For testing using the Confusion matrix, there are accuracy, precision and recall values. The accuracy value describes how accurate the model can classify correctly, in equation 3 is the formula for the accuracy value.

Precision describes the degree of accuracy between the requested data and the predictive results provided by the model. To calculate precision, you can use equation 4.

$$precision = \frac{TP}{TP + FP} \quad (4)$$

Recall describes the success of the model in recovering information. The recall formula is in equation 5.

$$recall = \frac{TP}{TP + FN} \quad (5)$$

In table 2, the test results when only using the cosine similarity method produce an average value of accuracy, precision, and recall that is almost the same in each language. In table 3, the test results when only using the weighted tree similarity

Table 2. Result Using Cosine Similarity Method

Query	TP	TN	FP	FN	Accuracy	Precision	Recall
sistem	19	111	4	6	0,928571429	0,826086957	0,76
Kesehatan	16	120	3	1	0,971428571	0,842105263	0,941176471
lingkungan	5	131	4	0	0,971428571	0,555555556	1
Study	40	89	10	1	0,921428571	0,8	0,975609756
Education	11	124	3	2	0,964285714	0,785714286	0,846153846
Learning	34	90	12	4	0,885714286	0,739130435	0,894736842
Average Indonesian					0,957142857	0,741249258	0,900392157
Average English					0,923809524	0,77494824	0,905500148

Table 3. Result Using Weighted Tree Similarity

Query	TP	TN	FP	FN	Accuracy	Precision	Recall
sistem	21	115	2	2	0,971428571	0,913043478	0,913043478
kesehatan	17	120	2	1	0,978571429	0,894736842	0,944444444
lingkungan	5	130	4	1	0,964285714	0,555555556	0,833333333
study	1	88	1	50	0,635714286	0,5	0,019607843
education	0	125	0	15	0,892857143	0	0
learning	7	93	2	38	0,714285714	0,777777778	0,155555556
Average Indonesian					0,971428571	0,787778625	0,896940419
Average English					0,747619048	0,425925926	0,0583878

Table 4. Result Using Cosine Similarity Method and Weighted Tree Similarity

Query	TP	TN	FP	FN	Accuracy	Precision	Recall
Sistem	21	116	2	1	0,978571429	0,913043478	0,954545455
Kesehatan	17	120	2	1	0,978571429	0,894736842	0,944444444
lingkungan	6	130	3	1	0,971428571	0,666666667	0,857142857
Study	1	86	1	52	0,621428571	0,5	0,018867925
Education	0	125	0	15	0,892857143	0	0
Learning	7	93	2	38	0,714285714	0,777777778	0,155555556
Average Indonesian					0,976190476	0,824815662	0,918710919
Average English					0,742857143	0,425925926	0,05814116

Table 5. Result Using Language Dectection, Cosine Similarity Method and Weighted Tree Similarity

Query	TP	TN	FP	FN	Accuracy	Precision	Recall
sistem	20	116	3	1	0,971428571	0,869565217	0,952380952
kesehatan	18	120	1	1	0,985714286	0,947368421	0,947368421
lingkungan	8	130	1	1	0,985714286	0,888888889	0,888888889
study	46	89	3	2	0,964285714	0,93877551	0,958333333
education	11	124	2	2	0,971223022	0,846153846	0,846153846
learning	43	92	3	2	0,964285714	0,934782609	0,955555556
Average Indonesian					0,980952381	0,901940842	0,929546087
Average English					0,96659815	0,906570655	0,920014245

Table 6. Average Accuracy, Precision, and Recall for Each Method

Method	Average Accuracy	Average Precision	Average Recall
Cosain Similarity (CS)	0,94047619	0,758098749	0,902946152
Weighted Tree Similarity (WTS)	0,85952381	0,606852276	0,477664109
CS + WTS	0,85952381	0,625370794	0,488426039
CS + WTS + Language Dectection	0,973775266	0,904255749	0,924780166

method resulted in a better average accuracy, precision, and recall values for Indonesian keywords. In table 4, the test results when using the cosine similarity method and weighted tree similarity resulted in better average accuracy, precision, and recall values for Indonesian keywords. In table 5 the results of the test Result using language detection, cosine similarity method, and weighted tree similarity produce an average value of accuracy, precision, and recall which is almost the same in every language.

Table 6 shows the comparison of the average values of accuracy, precision, and recall of all tested methods. The search results on scientific articles have the best accuracy, precision, and recall values when combining language detection, cosine similarity method, and weighted tree similarity. The accuracy value when all methods are combined is 0,973 while the precision is 0,904 and the recall value is 0,924.

Conclusion

Based on research, implementation, and testing, the following conclusions can be drawn: (1) The language detection process has a significant effect on the search results, it is proven when using only the cosine similarity and weighted tree similarity methods without the language detection process, the search results with the keyword education do not show results. In fact, there are 15 scientific articles that contain the keyword education. (2) The average test shows a number above 0.90 which means that the search for scientific articles based on language using the Cosain Simillarty algorithm and the Similarity weigted tree gives quite good similarity results and the search system also finds back information with fairly good results as evidenced by the recall value. an average of 0.924.

Suggestion

For further research, it might be possible to try with documents that have more sub-tree levels so that we can find out how stable this algorithm is and can also be combined with new methods or detection of languages other than Indonesian and English.

References

Anugrah, I. G., Sarno, R., & Anggraini, R. N. E. (2016). Decomposition using Refined Process Structure Tree (RPST) and control flow complexity metrics. *Proceedings of 2015 International Conference on Information and Communication Technology and Systems, ICTS 2015*, 203–208. https://doi.org/10.1109/ICTS.2015.7379899

Fibrianda, M. F., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2* (9), 3112–3123.

Hermawan, L., & Bellaniar Ismiati, M. (2020). Pembelajaran Text Preprocessing berbasis Simulator Untuk Mata Kuliah Information Retrieval. *Jurnal Transformatika*, *17*(2), 188. https://doi.org/10.26623/transformatika.v17i2.1705

Maarif, A. A. (2015). Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah. *Dokumen Karya Ilmiah | Tugas Akhir | Program Studi Teknik Informatika - S1 | Fakultas Ilmu Komputer | Universitas Dian Nuswantoro Semarang*, *5*, 4. mahasiswa.dinus.ac.id/

docs/skripsi/jurnal/15309.pdf

Melita, R., Amrizal, V., Suseno, H. B., Dirjam, T., Studi, P., Informatika, T., & Sains, F. (2018). *(TF-IDF) DAN COSINE SIMILARITY PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI SYARAH HADITS BERBASIS WEB (STUDI KASUS : SYARAH UMDATIL AHKAM)*. *11*(2).

Naf'an, M. Z., Burhanuddin, A., & Riyani, A. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. *Jurnal Linguistik Komputasional*, *2*(1), 23–27. https://doi.org/10.26418/jlk.v2i1.17

Rofiqi, M. A., Fauzan, A. C., Agustin, A. P., & Saputra, A. A. (2019). *Implementasi Term-Frequency Inverse Document Frequency (TF- IDF) Untuk Mencari Relevansi Dokumen Berdasarkan Query*. *1*(2), 58–64.

Samuel, R., Natan, R., & Syafiqoh, U. (2018). *Penerapan Cosine Similarity dan K-Nearest Neighbor (K-NN) pada Klasifikasi dan Pencarian Buku*. *1*(1), 1–6.

Sarno, R., & Rahutomo, F. (2015). Penerapan Algoritma Weighted Tree Similarity. *Jurnal Teknologi Informasi*, *7*(August), 39–46. http://download.portalgaruda.org/article.php?article=153043&val=5910&title=PENERAPAN ALGORITMA WEIGHTED TREE SIMILARITY UNTUK PENCARIAN SEMANTIK

Sugiyamto, S., Surarso, B., & Sugiharto, A. (2014). Analisa Performa Metode Cosine Dan Jacard Pada Pengujian Kesamaan Dokumen. *Jurnal Masyarakat Informatika*, *5*(10), 1–8. https://doi.org/10.14710/jmasif.5.10.1-8

Suharso, W., Qurrota, A., & Arifianto, D. (2017). *Pengembangan Sistem Deteksi Kesesuaian Dokumen Proposal Program Kreativitas Mahasiswa Dengan Metode Extended Weighted Tree Similarity*. 84–91.

Wahyuni, R. T., Prastiyanto, D., & Supraptono, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro*, *9*(1), 18–23. https://doi.org/10.15294/jte.v9i1.10955

Zaman, B., Hariyanti, E., & Purwanti, E. (2015). Sistem Deteksi Bahasa pada Dokumen menggunakan N-Gram. *Multinetics*, *1*(2), 21. https://doi.org/10.32722/vol1.no2.2015.pp21-26

