Comparison of the BM25 and rabinkarp algorithm for plagiarism detection

To cite this article: I N S W Wijaya *et al* 2021 *J. Phys.: Conf. Ser.* **1810** 012032

Comparison of the BM25 and rabinkarp algorithm for plagiarism detection

I N S W Wijaya[1], K A Seputra[2], W G S Parwita[3]

[1,2]Department of Informatics Engineering, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha

[3]Teknik Informatika, STMIK STIKOM Indonesia, Denpasar, Bali, Indonesia

Email: wahyu.wijaya@undiksha.ac.id[1], agus.seputra@undiksha.ac.id[2], gede.suka@gmail.com[3]

Abstract. Plagiarism occurs because of the easy distribution of data. Plagiarism detection of documents such as student assignments and final projects requires a long process, often overlooked. However, to avoid plagiarism, a document must be checked for the level of plagiarism. Plagiarism detection can be done online / offline with the plagiarism checker. However, checking documents with plagiarism checkers such as Turnitin, Dupli Checker, Copyleaks, PaperRater, Grammarly and others requires additional fees. Several studies have been conducted to detect plagiarism. BM25 and Rabin Karp are examples of the Plagiarism Checker method. BM25 is tfidf based, while Rabin Karp is Hashing based. Each method needs to know its performance to detect plagiarism. Based on these problems, a study on the comparison of plagiarism detection with the BM25 algorithm with Rabin-Karp will be conducted. The case study is to use the article in Indonesian. The application of the BM25 and Rabin Karp algorithms goes through the Pre-Processing stage which consists of case folding, cleaning, tokenizing, filtering, and stemming. In this study, Sastrawi stemmer was used in this study . The test was conducted on twenty articles in Indonesian. The test results that are seen are the performance in the form of execution time.

1. Introduction

In Computer Science, there are several plagiarism detection methods. Several methods like Character-Based, Vector-Based, Syntax-Based, Semantic-Based, Structure-Based, Stylometric-Based Methods, etc. can be used for plagiarism detection[1]. Several studies on plagiarism detection have been conducted. The BM25 can be used to check for duplication of software bug reports. Duplication checking used the term weighting technique[2]. The measurement results of BM25 have an average result value that is closer to the threshold compared to the cosine similarity[3]. In this research, cosine similarity is an advanced stage after the text is being converted into a vector space model.

Before the BM25 method, the steps taken were preprocessing. Preprocessing has several stages. Tokenizing, Case folding, Cleaning,Stopword Removal, and Stemming. The stemming process can be done by several methods. The stemming algorithm with "arifinsetiono" method has a higher percentage of accuracy than the Porter method [4]. The “NaziefAndriani” stemming algorithm has higher accuracy than the “arifinsetiono” method. "NaziefAndriani" over stemming value is lower than that of arifinsetiono[5]. Several methods can be used to perform stemming such as Porter Stemmer[6], Nazief Andriani[7], and Sastrawi[8][9][10]. The development was carried out on the

"NaziefAndriani"algorithm. Sastrawi developed a more effective stemming algorithm. The literary algorithm uses the NaziefAndriani algorithm as a basis. For this reason, this study uses a literary algorithm.

There is similar research that discusses plagiarism detection. A case study is a student's final project document. Plagiarism detection can be done using the Rabin Karp algorithm. The Rabin Karp algorithm uses a string-matching approach [11]. Rabin Karp can be used to detect plagiarism in Indonesian-language documents[11].

BM25 is better than cosine similarity. However, it is not known whether the BM25 algorithm gives more accurate results when compared to the Rabin Karp algorithm. This is certainly worth researching. The BM25 and Rabin-Karp algorithms both use a simple string search approach, namely the Brute Force Algorithm.Based on these problems, a study on the comparison of plagiarism detection with the BM25 algorithm with Rabin Karp will be conducted. The case study was conducted by using "Bahasa" documents.

2. Methodology

Based on research conducted by the iThenticate organization in 2013, there wereten types of plagiarism, namely duplication, replication, paraphrasing, verbatim, misleading attributions, invalid sources, secondary sources, unethical collaborations, repeated research, and complete. Several factors caused plagiarism problems, namely high laziness, lack of knowledge, lack of training in mind and logic, and time constraints.[12].

2.1. Plagiarism Detection

The more plagiarism, the more application systems are used to detect plagiarism. Turnitin is an example of a plagiarism checker application. There werethree main stages in a plagiarism detection system. These stages wereheuristic retrieval, detailed analysis, and post-processing [13]. The plagiarism value wasobtained from the comparison between the document weight d to I indicated plagiarism and the original document weight that didnot change. The plagiarism value can be calculated using the formula in Equation 1[14].

$$Plagiarism_{di} = \frac{weight_{di}}{weight100\%_{di}} \times 100\% \qquad (1)$$

2.2. BM25

One method that can be used to determine the relationship between documents and queries wasthe BM25 method. Okapi BM25 provided a score and ranking on the document[15][16]. In the BM25 method, 3 main factors influenced the weight value, namely term frequency, inverse document frequency, and document length[17]. The formula for the BM25 method can be seen in Equation (2).

$$BM25_{(\mathrm{dj,q1:N})} = \sum_{i\in 1}^{N} IDF_{(qi)} \frac{TF_{(qi,dj)}.(k+1)}{TF_{(qi,dj)} + k.\left(1 - b + b.\frac{|d_j|}{L}\right)} \qquad (2)$$

2.3. Rabin Karp

Is a string matching algorithm. Rabin Karp useda hashing technique. This technique functions to compare the searched string (m) with the substring in the text (n). It will check if the hash value is the same. If the arrangement of strings and substrings is not the same, then an n-m shift to the right is performed. To calculate the plagiarism value, the last step wasto calculate the similarity. The method used was the Dice Similarity Coefficients. The formula can be seen in Equation 3.

$$S = \frac{2C}{A + B} \tag{3}$$

S is the value of similarity, while A, B, C are the number of k-grams of text 1, the number of k-grams of text 2, the same number of k-grams of text 1 and text 2, respectively[18].K gram with value = 5 produces the fastest time for the testing process[19].

Before calculating BM25 and Rabin Karp, pre-processing must be done. It was aimedat eliminating unnecessary terms in the text. The text pre-processing stage consistedof case folding, tokenizing, filtering, and stemming.

a. Case Folding is a process that aims to change uppercase letters to lowercase letters. This process is done to uniform all the words in the text and make searching easier [20].
b. Tokenizing is a process that aims to find words or tokens. In other words, it is the process of breaking a sentence into words[20].
c. Cleaning is a process that aims to remove punctuation marks, numbers, HTML tags, links, and others. This process can be interpreted as a process of filtering words without paying attention to existing punctuation marks.
d. Filtering is a process that aims to remove meaningless words. This deletion is based on a dictionary containing a list of non-essential words such as hyphens and pronouns.
e. Stemming is a process aimed at finding affixed words and returning them to their base form. Besides, stemming can reduce the form of inflection and usually refers to harsh heuristics, namely cutting off the ends of a word[20].

3. Flowchart Sistem

The system was developed to perform a plagiarism comparison test on a document. Comparisons were made to BM25 and the Rabin Karp algorithm. The two algorithms have different flows in checking a document. However, each algorithm also has similarities in pre-processing

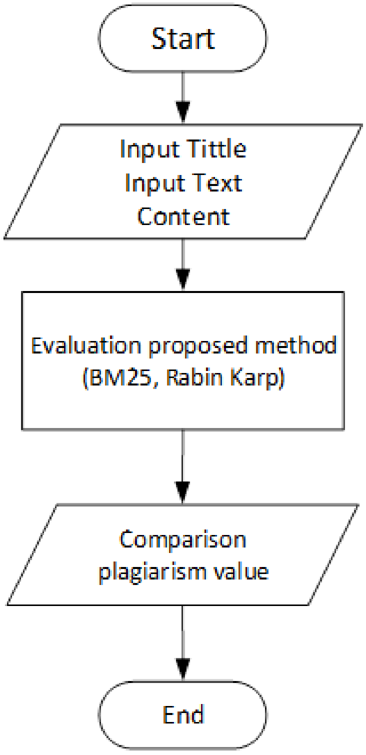


Figure 1. Proposed Flowchart System.

In-text mining, both in terms of information retrieval, sentiment analysis, plagiarism detection, etc., pre-processing is mandatory. It aimedto simplify a set of words that a document had. The effect wasto make it easier to select the words to be tested. The concern was the level of word connection in each of the test documents.

The system requireds input from the user. The allowed input wasIndonesian text. The text was then checked with the Rabin Karp algorithm and the BM25 algorithm. Test documents have been prepared in the database. In other words, the text entered by the user will be stored in the database as a test document. The document becamea test document. After the article wassuccessfully calculated, the results wereissued in tabular form by providing the doc id, title, value, or plagiarism level.

The development of a system for comparing plagiarism wasshown by a flowchart. The process startedwith input, pre-processing, BM25 algorithm, Rabin Karp algorithm, and output. The flowchart can be seen in Figure 2.

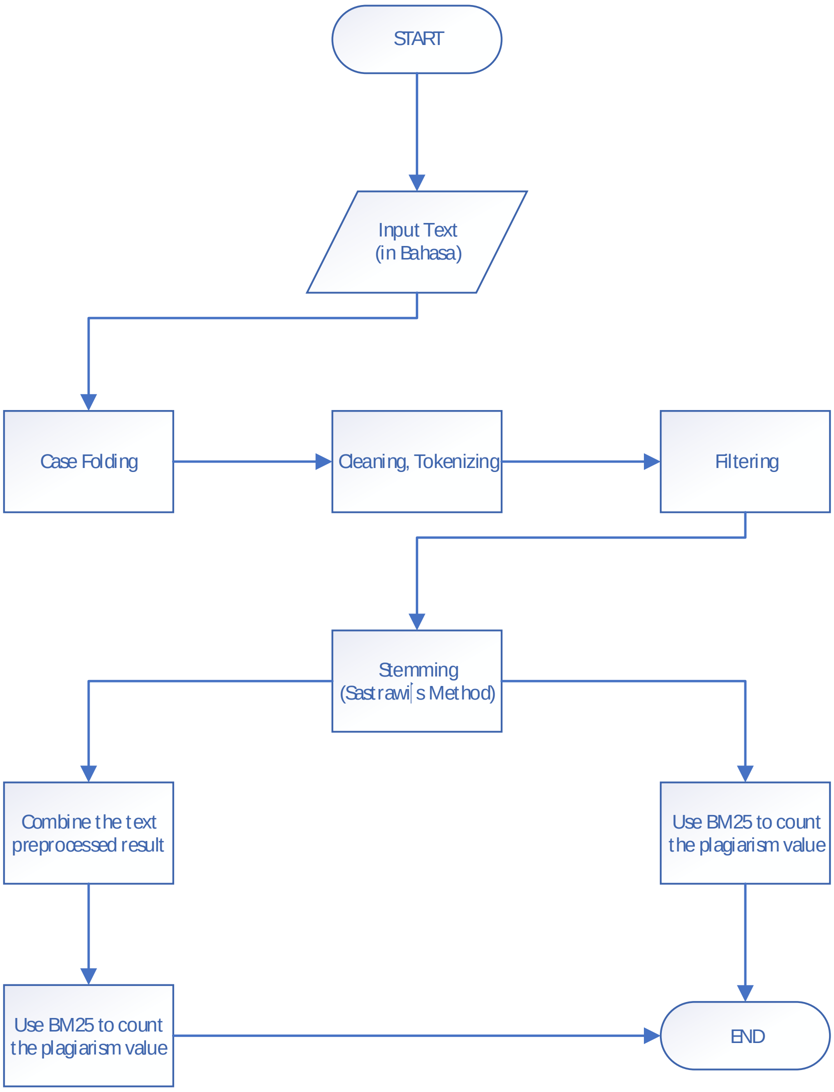


Figure 2. Flowchart System.

4. Result and Discussion

The system was developed with the PHP programming language. System development was using XAMPP version 3.2.4. Input to the system waslimited to text. The text wasthen stored in a database. The database design can be shown with the entity-relationship diagram in Figure 3.

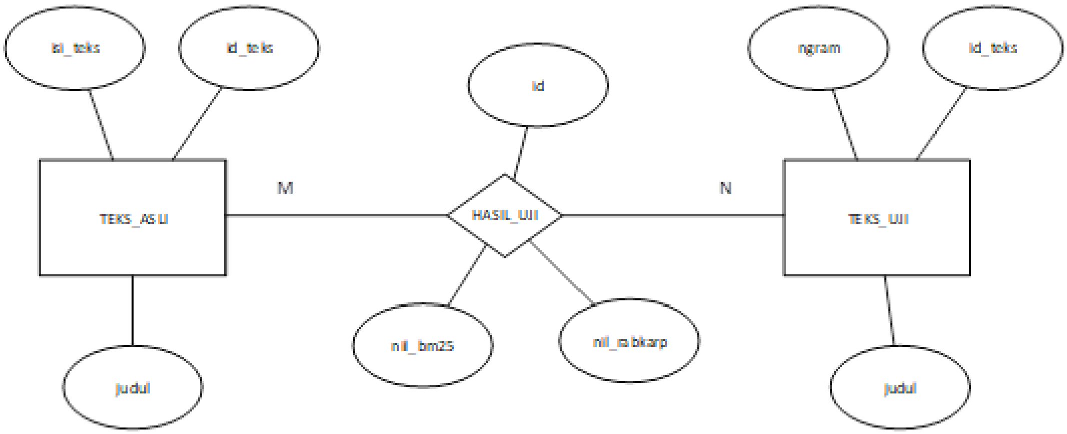
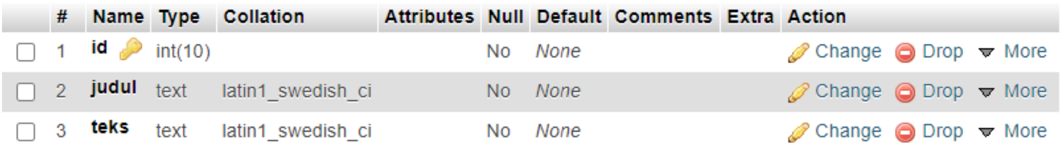


Figure 3. Entity Relationship Diagram.

The database design was aimed atproviding storage space for the documents that have been tested. The database design wasquite simple. This wasbecause of stop word lists written in the program code. So, the stop word list did notneed a table in the DBMS.

In Figure 3, it can be seen that there wasa relationship between the original text and the test text. This happenedbecause every time peopledo a test, of course, an original text will be tested on all test texts that have been stored in the database. The relationship between the two entities is calledHASIL_UJI. The relation of the test results has many to many cardinalities so that it requireedthe relation to have additional attributes. With the many to many cardinalities shown by this relation, in its application to the DBMS, it provided a new table, namely the table of test results relations. So that the DBMS will have three tables, namely the "teks_asli" table, the "teks_uji", and the "hasil_uji" table. The table structure built on the DBMS can be shown in figure 4,5,6.

1. Table Structure teks_asli

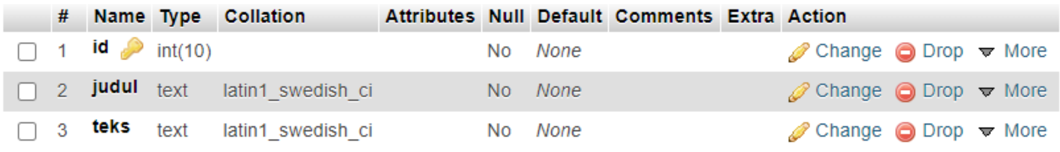


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id**	int(10)			No	*None*			Change Drop More
☐	2	**judul**	text	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**teks**	text	latin1_swedish_ci		No	*None*			Change Drop More

Figure 4. Table Structure teks_asli.

The teks_asli table has three attributes, namely id, "judul", and "teks". This attribute wasused to store the title data and the ngram value of the teks_asli. The id wasused to declare the primary key of the text to be tested.

2. Table Structure teks_uji



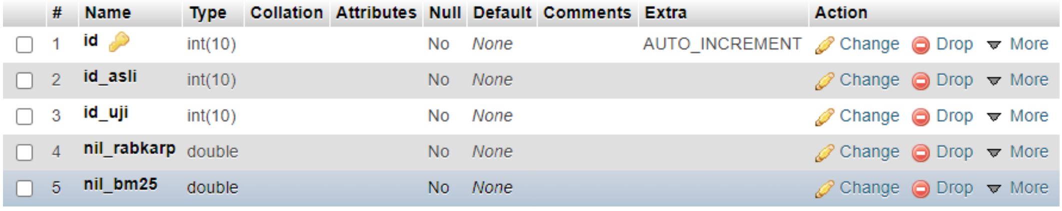
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id**	int(10)			No	*None*			Change Drop More
☐	2	**judul**	text	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**teks**	text	latin1_swedish_ci		No	*None*			Change Drop More

Figure 5. Table Structure teks_asli.

The “teks_uji” table also has the same three attributes as the teks_asli, but the text attribute holds the hash value of each preprocessed term.

3. Table Structure hasil_uji

The “hasil_uji” table stores the weighted values obtained through the Rabin Karp and BM25 algorithms.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id** 🔑	int(10)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**id_asli**	int(10)			No	*None*			Change Drop More
□	3	**id_uji**	int(10)			No	*None*			Change Drop More
□	4	**nil_rabkarp**	double			No	*None*			Change Drop More
□	5	**nil_bm25**	double			No	*None*			Change Drop More

Figure 6. Table Structure hasil_uji.

The researchers made a use case diagramsto describe the interaction that occuredbetween the system and the user. The use case diagram can be shown in Figure 7.Figure 7 provideedan overview of system users who can only do check_plagiarism. Other use cases such as algorithms and Pre-Processing have become atomic processes in the check_plagiarism.

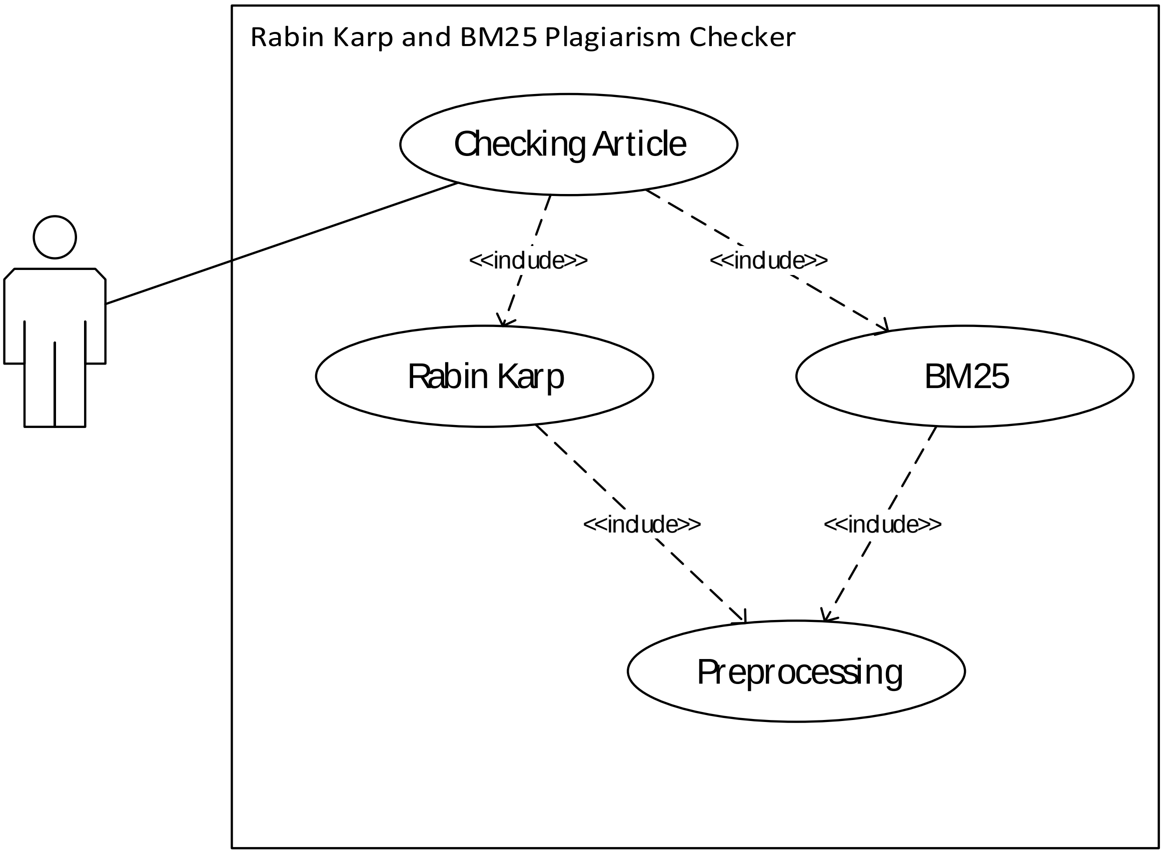


Figure 7. Use Case RabinBM System.

The researchersdeveloped a system using PHP language. Pre-processing wasdeveloped independently, except for the stemming process. The stemming process useda stemmer developed by a Sastrawi. The system was then given input as many as 20 articles. Then the 20 test Articles were tested.From the 20 articles, each was broken down into 10 articles. Then periodically reduced by 10% the numberof words [21]. The results can be seen in Table1.

Table 1. Similarity Test Result.

No	Article	Method	similar word ratio within Article										Execution Time
			100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	
1	A1	Rabin-Karp	100	95,38	89,31	83,73	75,31	68,32	58,40	48,20	35,31	17,61%	0,1139
		BM25	100	86,90	83,48	74,74	60,88	44,62	34,28	27,24	17,88	8,77%	0,5612
2	A2	Rabin-Karp	100	94,06	87,61	81,19	73,09	64,84	55,94	44,88	32,65	16,44%	0,0226
		BM25	100	88,31	80,84	73,91	62,39	53,79	38,82	28,74	17,24	7,39%	1,7700
3	A3	Rabin-Karp	100	95,16	89,53	82,15	75,45	66,53	56,55	46,97	34,39	18,23	0,0264
		BM25	100	91,01	76,66	68,98	60,46	49,62	38,79	28,83	18,53	11,21	1,6557
4	A4	Rabin-Karp	100	93,17	87,68	81,07	73,33	65,32	55,04	42,60	29,85	17,81	0,0186
		BM25	100	91,68	83,25	68,67	55,64	46,08	33,65	26,12	19,41	7,40	1,1735
5	A5	Rabin-Karp	100	94,74	89,35	81,96	74,70	66,33	56,95	46,15	32,28	18,49	0,0150
		BM25	100	93,33	82,73	62,80	55,07	48,93	38,42	22,65	14,44	7,76	0,5644
6		Rabin-Karp	100	94,86	88,66	82,70	75,25	66,07	55,29	45,39	31,60	15,57	0,0346
	A6	BM25	100	94,56	89,14	69,30	55,62	48,38	39,01	29,23	17,68	9,08	3,7400
7	A7	Rabin-Karp	100	95,22	89,11	83,06	76,92	69,77	59,02	50,89	33,11	17,06	0,0125
		BM25	100	82,82	74,02	67,62	51,47	39,04	28,29	17,60	11,49	5,22	0,5350
8	A8	Rabin-Karp	100	94,55	89,58	83,77	76,34	68,24	59,62	50,10	33,19	18,10	0,0260
		BM25	100	87,99	76,16	67,90	57,75	47,20	37,09	24,11	20,67	9,78	0,9687
9	A9	Rabin-Karp	100	94,35	88,82	81,42	76,51	68,88	59,54	48,77	34,23	20,78	0,0147
		BM25	100	86,47	77,07	61,41	43,65	37,34	32,92	24,19	16,90	8,51	0,4120
10	A10	Rabin-Karp	100	92,96	86,21	80,49	73,60	65,41	56,84	45,39	32,62	17,56	0,0153
		BM25	100	93,15	85,35	72,39	62,31	49,35	41,17	31,41	20,17	9,72	0,9794
11	A11	Rabin-Karp	100	95,16	89,53	82,15	75,45	66,53	56,55	46,97	34,39	18,23	0,0222
		BM25	100	91,01	76,66	68,98	60,46	49,62	38,79	28,83	18,53	11,21	1,5347
12	A12	Rabin-Karp	100	94,30	88,71	81,31	73,77	64,55	52,97	40,86	28,68	15,69	0,0143
		BM25	100	81,30	70,48	62,07	55,27	48,57	41,75	32,89	22,83	12,95	0,5788
13	A13	Rabin-Karp	100	95,38	89,31	83,73	75,31	68,32	58,40	48,20	35,31	17,61	0,0213
		BM25	100	86,90	83,48	74,74	60,88	44,62	34,28	27,24	17,88	8,77	0,5820
14	A14	Rabin-Karp	100	96,17	90,05	82,68	74,67	64,20	56,86	45,80	30,15	16,75	0,0143
		BM25	100	79,28	71,30	64,48	58,41	53,98	44,67	35,33	26,97	21,51	0,3910
15	A15	Rabin-Karp	100	94,00	89,97	84,99	76,98	68,32	59,29	47,85	32,63	16,71	0,0109
		BM25	100	79,52	72,87	66,20	56,70	49,45	41,23	31,68	19,24	9,72	0,1629
16	A16	Rabin-Karp	100	94,51	89,43	82,25	75,61	67,10	57,84	45,45	33,47	16,22	0,0180

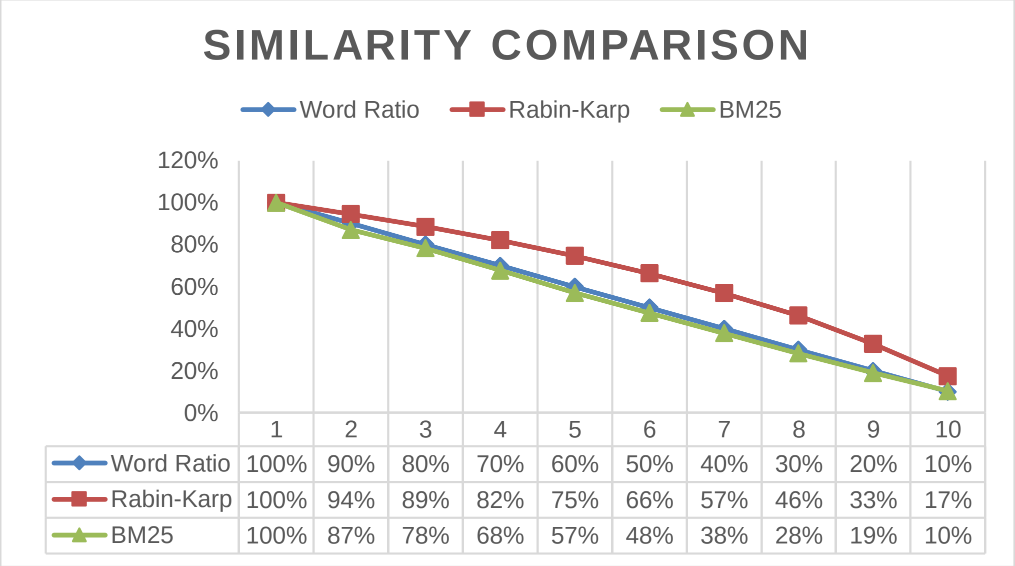
No	Article	Method	similar word ratio within Article										Execution Time
			100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	
		BM25	100	82,02	74,79	67,10	60,75	56,45	46,17	34,51	23,41	13,13	0,8688
17	A17	Rabin-Karp	100	91,77	85,06	77,30	70,63	62,45	54,01	42,52	30,22	14,81	0,0168
		BM25	100	85,86	80,19	70,51	62,58	54,21	46,46	39,80	23,28	11,05	0,5023
18	A18	Rabin-Karp	100	94,66	87,62	82,05	71,99	62,82	54,43	45,98	36,88	19,37	0,0093
		BM25	100	92,47	81,60	71,38	59,19	53,32	44,38	34,72	26,60	18,34	0,3313
19	A19	Rabin-Karp	100	93,82	87,85	81,18	73,91	64,78	55,69	44,07	31,74	16,26	0,0279
		BM25	100	92,86	84,52	70,56	59,53	45,73	33,03	26,53	18,75	12,48	2,6363
20	A20	Rabin-Karp	100	95,29	87,70	80,35	73,73	66,34	59,25	46,93	32,24	18,54	0,0106
		BM25	100	71,18	58,70	49,01	40,48	30,60	23,22	12,14	6,41	2,76	0,3292

These data werethe results of the comparison between the Rabin-Karp and BM25 methods. The comparison value wasthe result of calculating the similarity of the document. The similarity wasobtained by manipulating the input text. The input text has previously passed preprocessing and tokenizing. The next step wasto select words with a ratio of 100% to 10%. The selection results wereused to check the original text document. Testing wasalso seen based on the algorithm processing time in each test document. The test results wereaveraged to see the comparison of the level of similarity of each algorithm. The average value can be seen in table 2.

Table 2. Similarity and Time average Comparison.

Method	similar word ratio within Article										Execution Time
	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	
Rabin-Karp	100	94	89	82	75	66	57	46	33	17	0,0233
BM25	100	87	78	68	57	48	38	28	19	10	1,0139

The similarity of the input text waschecked based on n% of preprocessing words. N minus 10 per iteration until it reacheed10%. Then each algorithm gotthe average value shown in table 2. The graph in figure 8wasgiven to facilitate observation.



	1	2	3	4	5	6	7	8	9	10
Word Ratio	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
Rabin-Karp	100%	94%	89%	82%	75%	66%	57%	46%	33%	17%
BM25	100%	87%	78%	68%	57%	48%	38%	28%	19%	10%

Figure 8. Rabin-Karp and BM25 Similarity Comparison.

Based on Figures 8 the behavior of each algorithm can be observed. The ratio of plagiarism values for Rabin Karp tendedto be greater than that of BM25. However, if sorted in descending or ascending order, it will give the same results. These results wereinseparable from differences in how to calculate plagiarism. BM25 useda comparison of the tf and idf values of each test term with the articles in documents stored in the database. Meanwhile, Rabin Karp useda comparison of the hash value.

The test was carried out based on the occurrence of the same words as BM25.The result showed that the similarity value of BM25 was closer to the word ratio compared to BM25. This applieed to each test article. The hashing technique changed the similarity of a text by looking at the letter value that was changed in ASCII form. This can give the same value to several substrings that will be used for the similarity calculation process. When calculating the similarity, the numerator value was obtained based on the appearance of the same hash value in the test article with the input article. If the same substring was detected in the test article, the numerator value will be added. Then the appearance value will be doubled in the calculation of similarity. For this reason, it was necessary to improve the method of calculating similarity in the Rabin Karp algorithm. To find out the performance of each algorithm, an execution time test was conducted. The execution time was calculated from the text that has passed the preprocessing stage. This was done because Rabin Karp and BM5 both required a preprocessing stage. Based on the average testing of these 20 articles, the execution time of Rabin Karp was far superior to that of BM25.

5. Conclusion

Plagiarism is one of the problems faced by the world of education. Several methods have been proposed to solve this problem. The BM25 method, which is a weighting method for ranking, can be used to check plagiarism. Likewise the Rabin Karp method. Based on the research that has been done, it can be concluded that in terms of performance, especially the execution time, the Rabin Karp algorithm has a better performance than the BM25 algorithm. This happens because the hash value of the articles in the Rabin Karp algorithm can be stored in the database. Whereas in the BM25 algorithm the tf and idf values must be calculated when testing. For similarity, BM25 has a better average value compared to Rabin Karp if it is based on the proximity with the same word ratio.However, for the accuracy of the plagiarism value, it is not known which algorithm is better. So that in further research, precision, and recall testing can be carried out for the accuracy of the plagiarism value.

References

[1] Chowdhury H A and Bhattacharyya D K 2018 Plagiarism: Taxonomy, Tools and Detection Techniques

[2] Yang C Z, Du H H, Wu S S and Chen I X 2012 Duplication detection for software bug reports based on BM25 term weighting *Proc. - 2012 Conf. Technol. Appl. Artif. Intell. TAAI 2012* 33–8

[3] Dahniawati D, Indriati and Sutrisno 2019 Deteksi Plagiarisme pada Artikel Berita Berbahasa Indonesia menggunakan BM25 *J. Pengemb. Teknol. Inf. dan Ilmu Komput.***3** 4508–15

[4] Novitasari D 2017 Perbandingan Algoritma Stemming Porter dengan Arifin Setiono untuk Menentukan Tingkat Ketepatan Kata Dasar *STRING (Satuan Tulisan Ris. dan Inov. Teknol.***1** 120–9

[5] Simarangkir M S H 2017 Studi Perbandingan Algoritma - Algoritma Stemming Untuk Dokumen Teks Berbahasa Indonesia *J. Inkofar***1** 41–7

[6] Hajeer S I, Ismail R M, Badr N L and Tolba M F 2017 A new stemming algorithm for efficient information retrieval systems and web search engines *Intell. Syst. Ref. Libr.***115** 117–35

[7] Septian G, Susanto A and Shidik G F 2017 Indonesian news classification based on NaBaNA *Proc. - 2017 Int. Semin. Appl. Technol. Inf. Commun. Empower. Technol. a Better Hum. Life, iSemantic 2017***2018-Janua** 175–80

[8] Hidayatullah A F and Ma'arif M R 2016 Pre-processing Tasks in Indonesian Twitter Messages *Journal of Physics: Conference Series* pp 1–6

[9] Azmi S D and Kusumaningrum R 2019 Relevance Feedback using Genetic Algorithm on Information Retrieval for Indonesian Language Documents *J. Inf. Syst. Eng. Bus. Intell.***5** 171–82

[10] Yusliani N, Primartha R and Diana M 2019 Multiprocessing Stemming: A Case Study of Indonesian Stemming *Int. J. Comput. Appl.***182** 15–9

[11] Parwita W G S, Indradewi I G A A D and Wijaya I N S W 2019 String Matching based Plagiarism Detection for *2019 5th International Conference on New Media Studies* pp 54–8

[12] Wijaya H 2016 PLAGIARISME DALAM PENELITIAN pp 84–92

[13] k. Sanjalawe Y and Anbar M 2017 The Plagiarism Detection Systems for Higher Education - A Case Study in Saudi Universities *Int. J. Softw. Eng. Appl.***8** 33–49

[14] Gunawan D, Sembiring C A and Budiman M A 2018 The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents *J. Phys. Conf. Ser.***978** 1–6

[15] Sari S and Adriani M 2014 Learning to Rank for Determining Relevant Document in Indonesian-English Cross Language Information Retriecal using BM25 *International Conference on Advanced Computer Science and Information Systems (ICACSIS)* pp 309–14

[16] Kadhim A I 2019 Term Weighting for Feature Extraction on Twitter: A Comparison between BM25 and TF-IDF *2019 Int. Conf. Adv. Sci. Eng. ICOASE 2019* 124–8

[17] Russell S J and Norvig P 2003 *Artificial Intelligence A Modern Approach; PearsonEducation*

[18] Putra D A and Sujaini H 2015 Implementasi Algoritma Rabin-Karp untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah(CONTOH PLAGIAT) *J. Sist. dan Teknol. Inf.***4** 66–74

[19] Hartanto A D, Syaputra A and Pristyanto Y 2019 Best parameter selection of rabin-Karp algorithm in detecting document similarity *2019 Int. Conf. Inf. Commun. Technol. ICOIACT 2019* 457–61

[20] Manning C D, Raghavan P and Schütze H 2009 *Introduction to Modern Information Retrieval (2nd edition)* vol 53

[21] Yusuf B, Vivianie S, Marsya J M and Sofyan Z 2019 Analisis Perbandingan Algoritma Rabin-Karp dan Ratcliff / Obershelp untuk Menghitung Kesamaan Teks dalam Bahasa Indonesia *Semin. Nas. APTIKOM* 61–9