IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.16, No.1, January 2022, pp. 11~22
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.63686

Selection of the Best K-Gram Value on Modified Rabin-Karp Algorithm

Wahyu Hidayat*[1], Ema Utami[2], Andi Sunyoto[3]
[1,2,3]Magister Teknik Informatika, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
e-mail: ***[1]wahyu.1181@students.amikom.ac.id**, [2]ema.u@amikom.ac.id, [3]andi@amikom.ac.id

Abstrak

Algoritma Rabin-Karp digunakan untuk mendeteksi kemiripan dengan menggunakan teknik hashing, dari studi terkait telah dilakukan modifikasi pada proses hashing namun pada penelitian sebelumnya belum pernah dilakukan penelitian untuk pemilihan nilai k terbaik pada proses K-Gram. Pada tahap stemming menggunakan algoritma Nazief & Adriani untuk mentransformasikan kata-kata menjadi kata dasarnya. Peneliti menggunakan beberapa variasi nilai K-Gram untuk menentukan nilai K-Gram terbaik. Analisis dilakukan dengan menggunakan data publik Ukara Enhanced yang diperoleh dari Kaggle dengan total 12.215 data. Data jawaban essay siswa berjumlah 268 data pada group A dan 305 pada group B, setiap data jawaban siswa pada masing-masing group akan dibandingkan dengan jawaban sesama anggota group. Hasil penelitian nilai k = 3 memiliki kinerja terbaik yaitu memiliki interpretasi tertinggi 1-14% (derajat kemiripan kecil) dan 15-50% (derajat kemiripan sedang) dibandingkan nilai k = 5,7 , dan 9 yang memiliki jumlah hasil interpretasi tertinggi 0% -0,99% (Dokumen berbeda). Namun jika jawaban essay siswa yang dibandingkan memiliki interpretasi 100% (Persis sama), nilai k pada K-Gram tidak mempengaruhi hasil.

Kata kunci*—Similarity, Algoritma Stemming Nazief & Adriani , Algoritma Rabin-Karp, Dice's Similarity Coefficient*

Abstract

The Rabin-Karp algorithm is used to detect similarity using hashing techniques, from related studies modifications have been made in the hashing process but in previous studies have not been conducted research for the best k value in the K-Gram process. At the stage of stemming the Nazief & Adriani algorithm is used to transform the words into basic words. The researcher uses several variations of K-Gram values to determine the best K-Gram values. The analysis was performed using Ukara Enhanced public data obtained from the Kaggle with a total of 12215 data. The student essay answers data totaled to 258 data in the group A and 305 in the group B, every student essay answers data in each group will be compared with the answers of other fellow group member. Research results are the value of k = 3 has the best performance which has the highest some interpretations of 1-14% (Little degree of similarity) and 15-50% (Medium level of similarity) compared to values of k = 5, 7, and 9 which have the highest number of interpretation results 0%-0.99% (Document is different). However, if the students essay answers compared have 100% (Exactly the same) interpretations, the k value on K-Gram does not affect the results.

Keywords— *Similarity, Nazief & Adriani Stemming Algorithm, Rabin-Karp Algorithm, Dice Similarity Coefficient*

Received January 28th,2021; Revised August 25th, 2021; Accepted January 31th, 2022

1. INTRODUCTION

Research on determining the value of *k* has been widely studied by scientists, but for the selection of the best *k* value in the application of the Rabin-Karp algorithm has never been done. An experiment is needed to find out the best number of *k* to find out the effect of the *k* value on the similarity results that will be obtained in detecting document similarity. An essay is a test in the form of questions that expect answers to be written down, clear, and in writing. Each essay question given generally has an answer that must be explained and improvised by each student because the answer usually does not only cover the understanding of the theory but can be like each student's personal opinion and explanation that has the same meaning or purpose but with a different writing style. Text preprocessing can process the answer data from Ukara Enhanced because the dataset uses Indonesian language, so the Nazief & Adriani stemming algorithm is used at the stemming stage in text preprocessing to make the words in the students' text answers to the basic words. Similarity in the essay answers of each student can be identified similarity value with the Rabin-Karp algorithm by processing the basic word to the stages of parsing using the K-Gram method which will then be converted into a hash using a rolling hash and will match the hash results with other student hashes.

Preprocessing stage in text mining on the document is a case of folding, tokenization, filtering, and stemming [1,2]. Nazief & Adriani algorithm is used in the stage stemming in preprocessing text. This algorithm applies basic Indonesian morphological rules, checked collected allowed affixes and unallowed affixes, and uses a basic Indonesian word dictionary to compare basic words [3]. A. Jelita [4] there are several stemming algorithms in Indonesian, namely Nazief and Adriani's Algorithm, Arifin and Setiono's Algorithm, Vega's Algorithm, Ahmad, Yusoff, Sembok's Algorithm, and Idris. In testing Nazief & Adriani algorithm produces the best results, correctly stemming 93% of word occurrences in C_TR_MAJORITY, 92% of C_TR_UNIQUE, and 95.0% of C_TR_SUBJECTIVE.

Research by A. Rahmatulloh et al. [5] discuss the comparative performance of Porter and Nazief & Adriani stemming on the Winnowing algorithm for plagiarism detection. The results of the study concluded that testing the Winnowing algorithm without stemming had the results of 70.7% plagiarism similarity with processing speed of 0.711 s, Winnowing algorithm testing with stemming Porter had the results of plagiarism similarity of 65.7% with processing speed of 0.221 s, testing of the Winnowing algorithm with Nazief & Adriani stemming had results plagiarism similarity 70.5% with processing speed 0.476 s. Stemming Porters reduce the level of plagiarism similarity results very significantly but speed up processing while Nazief & Adriani stemming results are close to the same as without using stemming and also speed up processing.

K-Gram is used to make the order of grams by changing the results of the preprocessing text into a group of strings that are grouped into new strings where the new string collection is obtained from concatenation of the preprocessing text results with a length determined by the number of *k* values of K-Gram [6,7]. The stage of making a hash in the Rabin-Karp algorithm using rolling hashes [8]. rolling hash is a non-cryptographic hash function which allows the rapid computation of hash of each of the consecutive chunks. The fast computation of the rolling hash is due to the fact that the hash computation of a chunk utilizes the hash of the previous chunk [9].

Musthofa and Yaqin [10] applied the Rabin-Karp algorithm to correct automatic answers by matching essay answers with key answers. Because the manual system requires a lot of time, to speed up the correction of answers made an automatic answer correction system. In testing using Confusion Matrix in the application of Rabin-Karp algorithm using the value $k = 3$ and a dataset of 50 in the study resulted in 90% accuracy and 10% error rate. The automatic

essay grading system [11] conducts research with a dataset in the form of Japanese answer documents that will be romanized because the input is in the form of hiragana, katakana, or kanji into romaji. In this study using the Winnowing algorithm that uses hashing techniques and in fingerprint search using window techniques. By testing the dataset used with the parameters n = 2, w = 2, and p = 2. In experimental variations n unlike p and w, there are variations with an accuracy below 80% and therefore the parameter n is better with small numbers. the research resulted in an average accuracy of 86.86%.

The research of B. Leonardo and S. Hansun [12] discussing to detect the similarity of documents to other documents obtained from searches on Google Search using Rabin-Karp and Jaro-Winkler distance algorithms. Result of research are the similarity of text testing using the Rabin-Karp algorithm produces an average percentage of 51% and requires an average time of 0.594 minutes. Whereas Jaro-Winkler Distance produces an average of 35% and requires an average time of 0.992 minutes. The Rabin-Karp algorithm is effective than the JaroWinkler Distance algorithm. According to M. Bicer and X. Zhang [13] researching on the efficiency of the Boyer-Moore-Horspool algorithm, the Rabin-Karp algorithm, the Raita algorithm, and the Double-Hash algorithm on string similarity. Research reseults are the Double-Hash algorithm is more efficient in 5 different tests such as many patterns, timestamp patterns, very long patterns, very short patterns, and no patterns. The Double-Hash algorithm has a test duration of 5.63s, 5.74s, 5.67s, 6.43s, 6.20s. Subsequent research from D. D. Sinaga and S. Hansun [14] is Detecting the similarity of Indonesian documents using a combination of Confix-Stripping Algorithms in the stemming process so it can detect the prefix and suffix words. The result is the Rabin-Karp Algorithm has an average processing of 0.0123s and has an average accuracy rate of 89.1967% and the testing of the Rabin-Karp Algorithm without stemming processes has an average processing of 0.0103s.

Hashing process in Rabin-Karp algorithm using the modulo process, as defined the value of modulo can produce the same results so that it affects the results of accuracy because modulo can produce hashing that is not unique or in different cases can have the same value. Previous research on the hashing process eliminates modulo values, the results increase the syntax accuracy of word matching [15]. Rabin-Karp algorithm is used to match data from unique hashes formed from the hashing process of each data and the Rabin-Karp algorithm is used to identify the duplicate contents in the dataset [16,17]. After finding a unique hash value in the two data compared then the similarity value between the two is calculated using the Dice Similarity Coefficient. Dice Similarity Coefficient which is used to determine similarity between two documents, two queries, or a document and a query [18,19]. This research aims to determining the k value on K-Gram to decide selection of the best k value in the application of the modified Rabin-Karp algorithm in the removal of modulo in the hashing process to calculate the similarity between documents.

2. METHODS

This study uses the Ukara Enhanced student answer dataset from Kaggle. This data processed using text preprocessing, at the stemming stage using the Nazief & Adriani algorithm. Base words are cut and grouped into new strings according to the number of *k* on the K-Gram. Word cuts are changed to hashes using a rolling hash without modulo, and then compared with answers of other students with Rabin-Karp algorithm. Calculation of the similarity value using Dice's Similarity Coefficient and the similarity results are interpreted. These analysis process shown in Figure 1. The following explanation of the flow diagram in order to make the research aims will be divided into 3 processes: text preprocessing, Rabin-Karp algorithm, Dice's Similarity Coefficient. These three processes become the main methodology of this research. In

each process further explanations are carried out by providing sub-processes to clarify the steps to be taken.

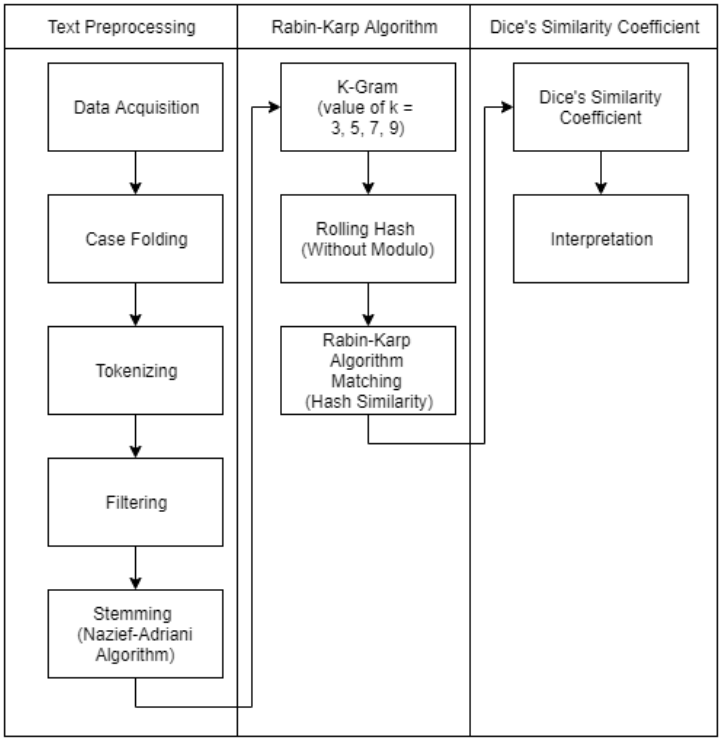

Figure 1 Sequence of analysis process

2.1 Text Preprocessing

Primary data is taken from the answer Ukara Enhanced from Kaggle, after that primary data collection then the next process is preprocessing text, which has a case of folding, tokenizing, filtering or stopword removal, and stemming [20]. In this research using Indonesian language data, therefore at the stemming stage using the Nazief & Adriani algorithm.

This research data comes from the raw text Ukara Enhanced answer dataset from students obtained from the Kaggle site, we use a total of 573 essay answers obtained in groups A and B. Student answers (true/false) have been labeled to the dataset. The language processed by word processor is only standard Indonesian according to *Kamus Besar Bahasa Indonesia* (KBBI). This study does not look at spelling or writing errors in documents, and is independent of synonyms or synonyms.

Case folding is used as a text converter to standard shapes or in lowercase letters and removes characters other than letters. Tokenizing or parsing is used for word separator text based on white space characters, tabulation, and spaces are considered as separators between words. The filtering stage is the stage of selecting important words and removing less important

The Nazief & Adriani algorithm was developed with basic word dictionary table search techniques and Indonesian morphological rules such as prefixes, insertions (infix), suffixes (suffixes) and combined prefixes (confixes). This algorithm uses a dictionary of basic words and supports re-coding by rearranging words that experience excessive stemming and have rules [21].

2.2 Rabin-Karp Algorithm

The Rabin-Karp Algorithm is the simplest string searching algorithm. This algorithm uses the hash function to discover the potential pattern in the input text. for the length of text n and pattern p of mutual length m, its average and best-case running time is O (n+m) in space O

(p), and also the worst-case time is O (nm) in space O (m) [22].

Determine the value of *k* with prime numbers, 3,5,7 and 9 so that the base word obtained is cut by the number of *k* in K-Gram which is then processed to be converted into a hash by rolling hash [23].

Change text that has been grouped with K-Gram into a hash using rolling hash. In previous studies have examined the comparison of hashing using modulo and without using modulo, the results if not using modulo syntactic accuracy of word matching increases. Some research on similarity algorithms such as Rabin-Karp and Winnowing using hashing technique is Rolling Hash [24].

$$H_{(c1\ldots ck+1)} = \left(H_{(c1\ldots ck+1)} - c1 * b^{(k-1)}\right) * b + c_{(k+1)} \quad (1)$$

Information:

$H_{(c1\ldots ck+1)}$: hash value
$c1$: the ASCII value of the l character in the string
l : string length
b : hash basis value

The hash value obtained will be sought using the Rabin-Karp algorithm by matching the same hash and supported by the answers of other students. After finding the unique hash value in both documents, then search for the hash value found in both of the matching processes (fingerprint). From the same number of hash findings and the total number of hashes in each document, the similarity values can be calculated [25].

2.3 Dice's Similarity Coefficient

The results of the hash comparison obtained will be calculated the similarity value using Formula Dice's Similarity Coefficient.

$$SC(X,Y) = 2\frac{|(X \cap Y)|}{|X|+|Y|} \quad (2)$$

Information:

X : The X represents the amount of fingerprints in document X
Y : Y itself represents the amount of fingerprints in document Y

After the process of finding the percentage value of Dice's Similarity Coefficient, then be interpreted according to the value of Dice's Similarity Coefficient. Grouping interpretations are shown in Table 1.

Table 1 Table Interpretation

Dice's Similarity Coefficient	Explanation
0%-0.99%	Document is different
1-14%	Little degree of similarity
15-50%	Medium level of similarity
51-99%	Approaching the resemblance
100%	Exactly the same

3. RESULTS AND DISCUSSION

Datasets grouped in 2 groups, each group has a different number of answers. Group A are 268 answer data, group B are 305 answer data. In testing using data in group A which has 35778 possibilities and B has 46360 possibilities for comparison of answers between students.

The next step is to remove punctuation marks (case folding), perform the tokenizing, filtering and stemming stages. At stemming stage it uses Nazief & Adriani algorithm for the process of determining standard word of a word with some predetermined rules. If all steps have been completed but are unsuccessful, then the first word is assumed to be the base word. The results of the stemming process are shown in Table 2.

Table 2 Results Stemming

Group	Answer ID	Stemming
1	1	["mahasiswa", "daftar", "batu", "batu", "buruk", "prosedur"]
1	2	["potong", "potong", "informasi", "mereplikasi", "coba", "tama", "butuh", "jenis", "sampel", "prosedur", "cuka", "wadah", "persis", "ukur", "massa", "sampel", "jenis", "wadah", "plastik", "pengaruh", "hasil", "coba", ""]
...	...	...
A	11643	["intetraksi", "adaptasi", "lingkung"]
...	...	...
B	12215	["sumbang", "euro", "upaya", "produksi"]

After text preprocessing step is the parsing step, which is the term that has gone through the preprocessing process cut into pieces per character. Cuts per character using the K-Grams method. After the intersection of characters is known, hashing is done at each intersection using rolling hash. Take the gram from "mah", with an ASCII value of 109, an ASCII value of 97, has an ASCII value of 104.

$$H(\text{mah}) = \left(109 * 10^{(3-1)}\right) + \left(97 * 10^{(3-2)}\right) + \left(104 * 10^{(3-3)}\right)$$

$$H(\text{mah}) = 10900 + 970 + 104$$

$$H(\text{mah}) = 11974$$

After knowing the hash of each K-Gram intersection in each document answer, then compare the hash results in each answer with the other answer hashes. The following hash results in *k* 3, 5,7 and 9 hashes are shown in Table 3.

Table 3 Result K-Gram 3, 5, 7 and 9 Hash Similarity

Group	Answer ID 1	Answer ID 2	K-Gram			
			k = 3	k = 5	k = 7	k = 9
A	11643	11644	10797	-	-	-
A	11643	11645	12837	-	-	-
A	11643	11646	10924 11337 12470 10797 11082 10936 12457 12685 10955	1093470 1134797 1248082 1080936 1109457 1094685 1246955	109348082 113480936 124809457 108094685 110946955	10934809457 11348094685 12480946955
A	11643	11647	11703	-	-	-
...	...	...	...	...	...	...
B	12214	12215	10903 10903	-	-	-

Then calculate the similarity, found the same 1 hash for document ID 11643 which has hash as many 27 and document ID 11644 which has hash as many 25 calculating the similarity in K-Gram 3 as follows :

$$S(11643,11644) = 2\frac{1}{27+25} = \frac{2}{52} = 0.03846$$
$$S(11643,11644) = 0.03846 * 100\%$$
$$S(11643,11644) = 3.85\ \%$$

The following results of Dice's Similarity Coefficient are shown in Table 4.

Table 4 Result Dice's Similarity Coefficient

Group	Answer ID 1	Answer ID 2	Similarity			
			K-Gram 3	K-Gram 5	K-Gram 7	K-Gram 9
A	11643	11644	3.85%	0.00%	0.00%	0.00%
A	11643	11645	5.00%	0.00%	0.00%	0.00%
A	11643	11646	43.90%	37.84%	30.30%	20.69%
A	11643	11647	4.55%	0.00%	0.00%	0.00%
...	...	...	...	...	...	...
B	12214	12215	8.70%	0.00%	0.00%	0.00%

Then change the value into interpretation which will be displayed on each variant of the K-Gram value so that different interpretations of each *k* value on the K-Gram are known. The results of interpretation are shown in Table 5.

Table 5 Result Interpretation

Group	Answer ID 1	Answer ID 2	Similarity			
			K-Gram 3	K-Gram 5	K-Gram 7	K-Gram 9
A	11643	11644	little degree of similarity	document is different	document is different	document is different
A	11643	11645	little degree of similarity	document is different	document is different	document is different
A	11643	11646	medium level of similarity	medium level of similarity	medium level of similarity	medium level of similarity
A	11643	11647	little degree of similarity	document is different	document is different	document is different
...	...	...	...	...	...	...
B	12214	12215	little degree of similarity	document is different	document is different	document is different

The result interpretation of Similarity Possibilities Answer Student Essay Group A are shown in Table 6.

Table 6 Result Interpretation Similarity Possibilities Answer Student Essay Group A

Dice's Similarity Coefficient	Similarity Hash K-Gram 3	Similarity Hash K-Gram 5	Similarity Hash K-Gram 7	Similarity Hash K-Gram 9
0%-0.99%	11546	24237	28175	31936
1-14%	13245	6995	5013	2341
15-50%	9930	4066	2275	1289
51-99%	1029	452	287	184
100%	28	28	28	28

The test results of the similarity possibilities in group A that have been interpreted. Diagram similarity possibilities data group A is shown in the Figure 2.

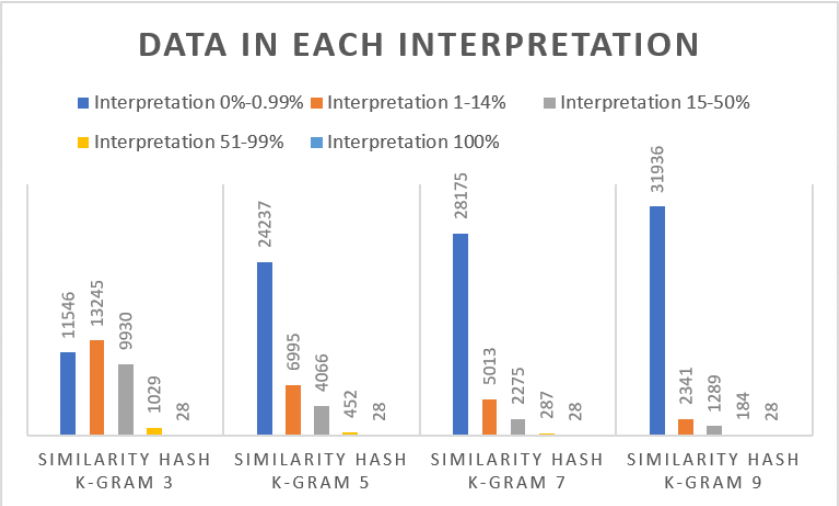

Figure 2 Diagram Similarity Possibilities Data Group A

Comparison of each answer data in group a has results of similarity at value of k = 3 show that the interpretation of 1-14% (Little degree of similarity) has the most members, as many as 13245. While at a value of k = 5, 7, and 9 shows that the interpretation 0%-0.99% (Document is different) has the most members, values are 25332, 34288, and 41183. The result interpretation of Similarity Possibilities Answer Student Essay Group B are shown in Table 7.

Table 7 Result Interpretation Similarity Possibilities Answer Student Essay Group B

Dice's Similarity Coefficient	Similarity Hash K-Gram 3	Similarity Hash K-Gram 5	Similarity Hash K-Gram 7	Similarity Hash K-Gram 9
0%	11777	25332	34288	41183
1-14%	15222	13059	8220	2564
15-50%	16884	7073	3324	2239
51-99%	2376	795	427	273
100%	101	101	101	101

According to these results, diagram similarity possibilities data group B is shown in the Figure 3. The results of similarity possibilities in group B at value of k = 3 show that the interpretation of 15-50% (Medium level of similarity) has the most members, as many as 16884. While at a value of k = 5, 7, and 9 shows that the interpretation 0%-0.99% (Document is different) the most values are 25332, 34288, and 41183.

In group A and B datasets testing, comparing each of student essay answers to their group resulting chances of similar answers that tested to every other student essay answer. Testing is done by varying the value of k at the K-Gram stage for the group A and B, which detects the possibility of the same answer between students who are different in each k value that is applied, but in both tests concluded that the value of k = 3 has good results because in that test produced the possibility of similar essay scores between students that spread evenly on each interpretation. However, different values of k = 5, 7, and 9 produce the possibility of similar values to essay answers among students that dominate the interpretation of 0%-0.99% (Document is different).

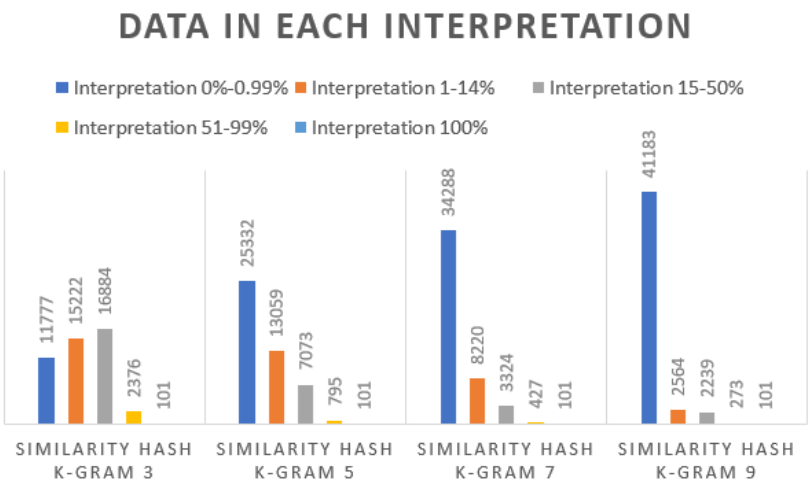

Figure 3 Diagram Similarity Possibilities Data Group B

The value of *k* = 3 results dominate in several interpretations which is able to detect the similarity of essay answers among students into interpretations of 1-14% and 15-50%. While using the values k = 5, 7, and 9 in every interpretation that is decreasing in number in each interpretation. But if the document does have 100% in common then in every test the various *k* values have the same results.

4. CONCLUSIONS

The results compare students answer tests in groups A and B with *k* values = 3, 5, 7 and 9 on K-Gram, thus can be concluded that the number of values on K-Gram affects Dice Similarity Coefficient results. Previous studies that applied the Rabin-Karp algorithm has the similar result, N-Gram value also affects the number of similarity values. The value of *k* = 3 has the best performance in detecting the similarity between students essay answers, which has the highest number of interpretations of 1-14% (Little degree of similarity) and 15-50% (Medium level of similarity) compared to values of *k* = 5, 7, and 9 which have the highest number of interpretation results 0%-0.99% (Document is different). But if the students essay answers compared have 100% (Exactly the same) interpretations, the *k* value on K-Gram does not affect results in each test.

REFERENCES

[1] A. N. Muhammad, S. Bukhori, and P. Pandunata, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier," *2019 Int. Conf. Comput. Sci. Inf. Technol. Electr. Eng.*, vol. 1, pp. 199–205, 2019. Available: https://doi.org/10.1109/ICOMITEE.2019.8920923 [Accessed: 28-Jan-2021]

[2] S. Wahyu Handani, D. Intan Surya Saputra, Hasirun, R. Mega Arino, and G. Fiza Asyrofi Ramadhan, "Sentiment analysis for go-jek on google play store," *J. Phys. Conf. Ser.*, vol. 1196, no. 1, 2019. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1196/1/012032/meta [Accessed: 28-Jan-2021]

[3] A. Amalia, D. Gunawan, Y. Fithri, and I. Aulia, "Automated Bahasa Indonesia essay

evaluation with latent semantic analysis," *J. Phys. Conf. Ser.*, vol. 1235, no. 1, 2019. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1235/1/012100/meta [Accessed: 28-Jan-2021]

[4] A. Jelita, "Effective Techniques for Indonesian Text Retrieval," *Ph.D Thesis*, pp. 1–286, 2007.

[5] A. Rahmatulloh, N. I. Kurniati, A. Z. Asyikin, I. Darmawan, and J. D. Witarsyah, "Comparison between the stemmer porter effect and nazief-adriani on the performance of winnowing algorithms for measuring plagiarism," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 4, pp. 1124–1128, 2019. Available: http://www.insightsociety.org/ojaseit/index.php/ijaseit/article/view/8844 [Accessed: 28-Jan-2021]

[6] S. Andysah Putera Utama, M. Mesran, R. Robbi, and S. Dodi, "K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm," *Int. J. Sci. Technol. Res.*, vol. 6, no. 07, pp. 350–353, 2017. Available: https://doi.org/10.31219/osf.io/yxjnp [Accessed: 28-Jan-2021]

[7] M. T. Pham and T. B. Nguyen, "The DOMJudge based online judge system with plagiarism detection," *RIVF 2019 - Proc. 2019 IEEE-RIVF Int. Conf. Comput. Commun. Technol.*, pp. 1–6, 2019. Available: https://ieeexplore.ieee.org/abstract/document/8713763/ [Accessed: 28-Jan-2021]

[8] R. M. Harpreet Kaur, "Granularity-Based Assessment of Similarity Between Short Text Strings," *Proc. ofthe Third Int. Conf. Microelectron. Comput. Commun. Syst.*, pp. 91–107, 2019. Available: https://link.springer.com/chapter/10.1007/978-981-13-7091-5_9 [Accessed: 28-Jan-2021]

[9] D. Chang, M. Ghosh, S. K. Sanadhya, M. Singh, and D. R. White, "FbHash: A New Similarity Hashing Scheme for Digital Forensics," *Digit. Investig.*, vol. 29, pp. S113–S123, 2019. Available: https://doi.org/10.1016/j.diin.2019.04.006 [Accessed: 28-Jan-2021]

[10] M. Misbah Musthofa and A. Yaqin, "Implementation of Rabin Karp algorithm for essay writing test system on organization XYZ," *2019 Int. Conf. Inf. Commun. Technol. ICOIACT 2019*, pp. 502–507, 2019. Available: https://doi.org/10.1109/ICOIACT46704.2019.8938562 [Accessed: 28-Jan-2021]

[11] A. Agung Putri Ratna, D. Lalita Luhurkinanti, I. Ibrahim, D. Husna, and P. Dewi Purnamasari, "Automatic Essay Grading System for Japanese Language Examination Using Winnowing Algorithm," *Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 565–569, 2018. Available: https://scholar.ui.ac.id/en/publications/automatic-essay-grading-system-for-japanese-language-examination- [Accessed: 28-Jan-2021]

[12] B. Leonardo and S. Hansun, "Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 2, pp. 462–471, 2017. Available: http://ijeecs.iaescore.com/index.php/IJEECS/article/view/6097 [Accessed: 28-Jan-2021]

[13] M. Bicer and X. Zhang, "An Efficient , Hybrid , Double-Hash String- Matching Algorithm," *2019 IEEE Long Isl. Syst. Appl. Technol. Conf.*, pp. 1–5, 2019. Available: https://doi.org/10.1109/LISAT.2019.8816827 [Accessed: 28-Jan-2021]

[14] D. D. Sinaga and S. Hansun, "Indonesian text document similarity detection system using rabin-karp and confix-stripping algorithms," *Int. J. Innov. Comput. Inf. Control*,

vol. 14, no. 5, pp. 1893–1903, 2018. Available: http://www.ijicic.org/ijicic-140519.pdf [Accessed: 28-Jan-2021]

[15] M. I. Errissya Rasywir, Yovi Pratama, Hendrawan, "Removal of Modulo as Hashing Modification Process in Essay Scoring System Using Rabin-Karp," *2018 Int. Conf. Electr. Eng. Comput. Sci.*, pp. 159–164, 2018. Available: https://doi.org/10.1109/ICECOS.2018.8605211 [Accessed: 28-Jan-2021]

[16] P. Sundari, S. Deepasamili, and C. Science, "PROGRESSIVE DUPLICATION DETECTION USING RABIN- KARP ALGORITHM," *Int. J. Res. Sci. Eng. Technol.*, vol. 3, no. 11, pp. 11–17, 2016. Available: https://ijrset.in/index.php/ijrset/article/view/191 [Accessed: 28-Jan-2021]

[17] K. E. Rajakumari, "Comparison of Token-Based Code Clone Method with Pattern Mining Technique and Traditional String Matching Algorithms In-terms of Software Reuse," *Proc. 2019 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2019*, pp. 1–6, 2019. Available: https://doi.org/10.1109/ICECCT.2019.8869324 [Accessed: 28-Jan-2021]

[18] M. Afzali and S. Kumar, "Text Document Clustering: Issues and Challenges," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 263–268, 2019. Available: https://doi.org/10.1109/COMITCon.2019.8862247 [Accessed: 28-Jan-2021]

[19] B. N. Prastowo *et al.*, "A Proposed Framework for Essay Answer Processing based on Computational," *Int. Conf. Educ. Assess. Policy*, vol. 2, p. 54, 2018. Available: https://doi.org/10.26499/iceap.v2i1.96 [Accessed: 28-Jan-2021]

[20] R. Annisa, I. Surjandari, R. Annisa, and I. Surjandari, "Opinion Mining on Mandalika Hotel Reviews Using Latent Opinion Mining on Mandalika Hotel Reviews Using Latent Dirichlet Allocation Dirichlet Allocation," *Procedia Comput. Sci.*, vol. 161, pp. 739–746, 2019. Available: https://doi.org/10.1016/j.procs.2019.11.178 [Accessed: 28-Jan-2021]

[21] A. T. Ni'mah, D. A. Suryaningrum, and A. Z. Arifin, "Autonomy Stemmer Algorithm for Legal and Illegal Affix Detection use Finite-State Automata Method," *EPI Int. J. Eng.*, vol. 2, no. 1, pp. 46–55, 2019. Available: http://cot.unhas.ac.id/journals/index.php/epiije/article/view/177 [Accessed: 28-Jan-2021]

[22] I. Obeidat and M. AlZubi, "Developing a faster pattern matching algorithms for intrusion detection system," *Int. J. Comput.*, vol. 18, no. 3, pp. 278–284, 2019. Available: https://zuscholars.zu.ac.ae/scopus-indexed-articles/811/ [Accessed: 28-Jan-2021]

[23] Riki, Edy, and Maryanto, "Plagiarism Detection Application Uses Winnowing Algorithm with Synonym Recognition for Indonesian Text Documents," *Selangor Sci. &Technology Rev.*, vol. 3, no. 1, pp. 1–14, 2019. Available: http://sester.journals.unisel.edu.my/ojs/index.php/sester/article/view/47 [Accessed: 28-Jan-2021]

[24] J. H. T. Purba, M. Zarlis, and Sawaluddin, "THE IMPLEMENTATION OF N-GRAM FOR ESSAY Faculty of Computer Science and Information Technology , Universitas Sumatera Utara , Medan , Indonesia Faculty of Computer Science and Information Technology , Universitas Sumatera Utara , Medan , Indonesia," vol. 7838, pp. 141–145, 2019. Available: https://eprajournals.com/hits_update.php?id=1688 [Accessed: 28-Jan-2021]

[25] A. Bahrul Khoir, H. Qodim, B. Busro, and A. Rialdy Atmadja, "Implementation of rabin-karp algorithm to determine the similarity of synoptic gospels," *J. Phys. Conf. Ser.*, vol. 1175, no. 1, 2019. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1175/1/012120 [Accessed: 28-Jan-2021]