# Implementation of Winnowing Algorithm for Document Plagiarism Detection

1st Nurissaidah Ulinnuha
*Department of Mathematics*
*Universitas Islam Negeri Sunan Ampel*
Surabaya, Indonesia
nuris.ulinnuha@uinsby.ac.id

2nd Muhammad Thohir
*Department of Arabic Language*
*Universitas Islam Negeri Sunan Ampel*
Surabaya, Indonesia
muhammadthohir@uinsby.ac.id

3rd Dian Candra Rini Novitasari
*Department of Mathematics*
*Universitas Islam Negeri Sunan Ampel*
Surabaya, Indonesia
diancrini@uinsby.ac.id

4th Ahmad Hanif Asyhar
*Department of Mathematics*
*Universitas Islam Negeri Sunan Ampel*
Surabaya, Indonesia
hanif@uinsby.ac.id

5th Ahmad Zaenal Arifin
*Department of Mathematics*
*Universitas PGRI Ronggolawe*
Tuban, Indonesia
az_arifin@unirow.ac.id

***Abstract*— The rapid development of the internet influences information availability. It makes easier for someone to do the plagiarism of a work. The rise of information available online makes the habit of copy-paste without mentioning the reference to become easy so that scientific work unwittingly becomes the result of plagiarism from other scientific works. Plagiarism prevention efforts are envolving in various sector. Designing and developing plagiarism checker applications is the purpose of this paper. Specifically by knowing the percentage of similarity between the original document and the test document. This research using Winnowing algorithm because it can detect plagiarism in documents up to sub-section of the document. This research using three validates consisting of computational mathematicians, software engineering experts, and users to test the application feasibility. The experiment uses several scenarios and the result of effectiveness evaluation yields 82% sensitivity, 100% specificity, and 91% accuracy. The implemented system works effectively so the system can be used to detect document plagiarism.**

***Keywords—Plagiarism Detection, Research and Development, Winnowing Algorithm.***

## I. INTRODUCTION

The rapid development of information technology has an impact on the rapid dissemination of information. One example is the rapid development of the internet. This causes more and more information available and facilitates a person in making plagiarism [1][2][3]. The number of plagiarism cases by academics became a tragedy in Indonesia's education. The scientific work that made unconsciously becomes the result of plagiarism from other scientific works.

There are several online plagiarism checkers that used to detect plagiarism but are less effective considering the limitations of pages offered, such as Viper. According to Sunu [4], Viper only able to detect a maximum of 8 pages with long checks up to 20 minutes with good computer specifications and super fast internet connection. There is also Turnitin with a payment every year which is not cheap for university campuses in developing countries like Indonesia [5][6][7].

Plagiarism detection is actually a part of pattern recognition[8][9]. In this paper, the plagiarism detection application is built using Winnowing algorithm which is one part of the pattern recognition as a search algorithm for the same document. The system accepts inputs in the form of a text document with a .pdf or .txt extension and afterwards search for resemblance to the document database. Documents with similarity levels exceeded the threshold will be displayed in the system. The input of the Winnowing algorithm is document string and output of the hash value used as the document fingerprint. According to Niwattanakul, et al [10]. Fingerprints of both documents are processed with Jaccard's coefficient similarity function to get a percentage of document similarity. Data used in this study is the big data in university digital library in one of Indonesian University.

Winnowing algorithm is used as an algorithm to calculate text similarity in a document because Winnowing algorithm can cut the processing time of large files by utilizing the rolling technique of the hashing process [3][11]. In addition, the value of Winnowing algorithm's similarity accuracy is not only influenced by the input value of k-gram, but it is also influenced by the input window value that serves to separate the hash results on each gram.

This research aims to design and develop plagiarism detection application and to know the percentage of similarity between documents tested with Winnowing algorithm. In the end, plagiarism applications are expected to be used by many people.

## II. PRELIMINARIES

### *A. Plagiarism*

Plagiarism is an act of misappropriation, theft/robbery, publication, statement or declaring a person's own thoughts, writings or creations that the other person has intentionally or unintentionally without the reference [12].

Classification based on the proportion or percentage of words, sentences, hijacked paragraphs [13] divided into:

1. Minor plagiarism : < 30%
2. Medium plagiarism : 30-70%
3. Major plagiarism : > 70%

### *B.* Winnowing Algorithm

Winnowing Algorithm [11][14] is an algorithm used in plagiarism detection including similar small parts in many of documents. The input of this algorithm is a text document processed to produce an output of hash values collection called fingerprint. This fingerprint is used as the basis of comparison between text files that have been entered and used in plagiarism detection.

In general, the working principle of document resembling algorithm is described in Figure 1 :

a. Remove whitespace insensitivity, such as spaces or punctuation.
b. Forming a gram chain of size k.
c. Calculating hash values of each gram.
d. Divide into a particular window.
e. Selecting some hash values into document fingerprinting.
f. Determine the percentage of similarity between two documents with the Jaccard Coefficient equation

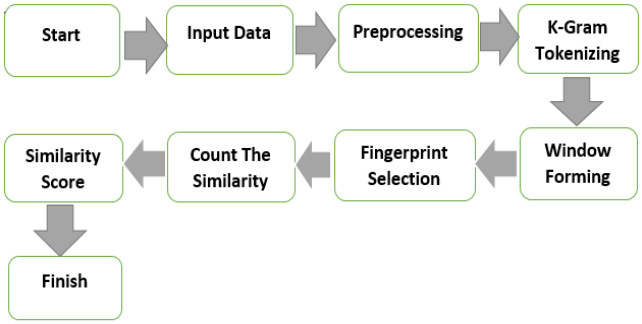


Figure 1. The Steps of Winnowing algorithm

The detailed steps of applying Winnowing algorithm are as follows [11] :

1. Preprocessing

The preprocessing is done in two steps: first, eliminating irrelevant characters in text documents, such as punctuation marks, spaces and second, changing the capitalization. Example, a sentence is given:

"Penelitian ini menggunakan algoritma Winnowing.".

After preprocessing, i.e. deleting spaces and punctuation marks, and converting all the letters into small and normal letters (not bold, not tilted and not underlined), resulting is in the following text:

Penelitianinimenggunakanalgoritmawinnowing

2. K-gram Method

The K-gram method [15][16] is a method used in the process of tokenization or separation of text, by forming substring along the character k of a string. Example: Cut a string along k. The value of k is 7. From the above sentence example, the result obtained as Figure 2.

| | | | | | | |
|---|---|---|---|---|---|---|
| penelit | eneliti | nelitia | Elitian | Litiani | itianin | tianini |
| aninime | ninimen | inimeng | nimengg | imenggu | menggun | engguna |
| Ggunaka | gunakan | unakana | nakanal | Akanalg | kanalgo | analgor |
| Algorit | lgoritm | goritma | oritmaw | Ritmawi | itmawin | tmawinn |
| Awinnow | winnowi | innowin | nnowing | | | |

Figure 2. The result k-gram with k=7

3. Rolling Hash

The hash function [17] is a function that receives a string input of arbitrary length and converts it into a fixed length output string. The output of the hash function is called hash-value or message digest. Hash value size generally smaller than the original string size.

The hash method equation is given by:

$$H_{(c_1 \ldots c_k)} = c_1 * b^{k-1} + c_2 * b^{k-2} + \cdots + c_{k-1} * b^{k} + c_k \quad (1)$$

where:
c = value of ASCII characters (decimal)
b = basis (prime)
k = sum of character (character index)

The advantage of rolling hashes is for the next hash value. To get the hash value of the k-grams method, the following hash rolling equation is used:

$$H_{(c_2 \ldots c_{k+1})} = \left(H_{(c_1 \ldots c_k)} - c_1 * b^{k-1}\right) * b + c_{k+1} \quad (2)$$

Using Equation (2) can save computational time when calculating the hash value of a gram. The result of calculating the hash value in gram is shown in Figure 3. Each number indicates the hash value of a gram.

119231 112854 117772 112856 117786 117272 122286 113275
110291 118844 116065 118663 115536 117083 112963 118109
113854 116411 124069 116436 108841 114495 109590 116736
109754 117232 115597 121649 122295 117677 123506
116937 112547 125607 116678 120502

Figure 3. The result of hash calculation each gram

4. Window Forming

The window is a grouping of several hash values with the specified size. From the window that has been formed, the smallest hash value on each window is selected to be the fingerprint of each document. The number 112854 is the smallest hash value of the window [119231, 112854, 117772, 112856, 117786].

The distance of fingerprint arrangement will be measured using the Jaccard coefficient with other fingerprint documents tested at a similarity level. The smaller the distance, the greater the level of similarity of the document.

5. Fingerprint Document

The fingerprint is a technique that aims to prevent unauthorized copying of a digital content. Fingerprints are not easily detected because they are designed in ways that make digital content difficult to fabricate [18].

Document fingerprinting is a method that can be used to detect document resemblance. Winnowing algorithm uses a fingerprint as a keyword used as a reference to look for similarities with the document being tested. The hash value of the document is divided using window w before determining the fingerprint of both the original document and the test document.

6. Jaccard Coefficient

Jaccard Coefficient is an equation used to find the degree of similarity between two text documents on Winnowing algorithm. This step is done by calculating the hash value and selecting the smallest fingerprint of two text documents[11]. The equation of Jaccard coefficient is given by:

$$Similaritas_{(d_i,d_j)} = \frac{|W(d_i) \cap W(d_j)|}{W(d_i) \cup W(d_j)} \quad (3)$$

where:

$W(d_i)$ = smallest fingerprint text document 1
$W(d_j)$ = smallest fingerprint text document 2

## III. RESEARCH METHOD

### A. Development Model

This research is a type of Research and Development (R&D). The ways undertaken in this development study include several phases[19], i.e:

1. Formulating potentials and problem.
2. Collecting data from university digital library
3. Product design
4. Design validation
5. Revised product design
6. Usage trial of small group product
7. Product revision
8. Usage trial of large group product
9. Product revision
10. Mass production

### B. Trial and Evaluation Product

The trial and evaluation product is intended to collect the data used as a basis for determining the effectiveness and attractiveness of the developed plagiarism detection application. Data obtained from trials are used to refine and enhance plagiarism detection applications. The trial will test the quality of the applications empirically.

a. Trial and Evaluation Design

The trial and evaluation design intended to get feedback directly from the user about the product quality being developed. Prior to testing, first create a design or draft application design that will be developed. First, the design of the application discussed in a Focus Group Discussion (FGD) with people who are considered experts and have competency in the field of computational mathematics and in the field of software engineering. After the application design as FGD results are completed, the next step is the implementation of the program which then through the process of consulting with computer programming experts. The results of consultation with computer programming experts is a product revision. The next phase is usage trials of a large group of university lecturer as a user application. Trials aim to improve the product so that when developed or used, the product has been completely valid and has a certain quality.

b. Subject Test

A subject test is a group of academic lecturers. The first phase is small group trials with two lecturers research subjects with areas of computational mathematics expertise and in the field of software engineering. The second phase is large group trials with 19 lecturers.

c. Type of Data

The data collected in this study is:

(1) Data on the process of developing plagiarism detection applications in accordance with predetermined development procedures, including data containing input from computational mathematicians and software engineering experts.

(2) Application feasibility data based on the assessment results. The data includes:
   (a) Qualitative data: the value of each assessment criteria.
   (b) Quantitative data: assessment score.

d. Data Collection Instruments

(1) Assessment of Small Group Test

In computational mathematics instruments, the components used are effectiveness, correctness, termination, efficiency, and complexity as shown in Table 1. In the instruments of software engineering experts, good application criteria can be reviewed from components of application compatibility, feature completeness, and application display as shown in Table 2.

TABLE 1. GUIDANCE INSTRUMENT EXPERT OF COMPUTATION MATH

| No | Aspect of Assessment | Point of Instrument | Number of Point |
|---|---|---|---|
| 1 | *Effectiveness* | 1, 2, 3 | 3 |
| 2 | *Correctness* | 4 | 1 |
| 3 | *Termination* | 5 | 1 |
| 4 | *Efficiency* | 6 | 1 |
| 5 | *Complexity* | 7, 8 | 2 |

Data obtained from computational mathematicians and software engineers assessment in the form of numbers. The number converted into qualitative data based on the scoring of results. The media score was analyzed by searching for average ratings. The questionnaire instruments were arranged using a Likert scale with a rating scale of 1 to 5.

TABLE 2. GUIDANCE INSTRUMENT SOFTWARE ENGINEERING EXPERT

| No | Aspect of assessment | Point of Instrument | Number of Point |
|---|---|---|---|
| 1 | Corresponding | 1, 2, 3, 4, 5 | 5 |
| 2 | Feature Completeness | 6,7,8,9,10,11,12 | 7 |
| 3 | Display | 13,14,15,16,17, 18,19,20,21 | 9 |

(2) Assessment of Large Group Test

Another name of the instrument for application feasibility is called usability evaluation. Usability evaluation aims to find out how well the application can be operated by the user. The first step in the evaluation of usability is to give tasks to the user while interacting with the system being tested. After all tasks have been completed by the user, the next step is to give a questionnaire containing questions that represent the five aspects of usability.

Questionnaires containing questions that represent the five aspects of usability, namely ease of learning (learnability), memorability, efficiency, errors, and

satisfaction[20] as shown in Table 3. The learnability aspect is an aspect that measures the ease of the user performing simple tasks when first using the application. Memorability aspect is done to measure the speed of the user in remembering the design and function of the application. The aspect of efficiency is used to measure the speed of the user in the work of a task. Errors aspect is used to see the possibility of user error. Satisfaction is an aspect that measures the level of user satisfaction in using the application.

TABLE 3. GUIDANCE INSTRUMENT OF ASSESSMENT FOR ASPECT USABILITY

| No | Aspect of assessment | Point of Instrument | Number of Point |
|---|---|---|---|
| 1 | Learnability | 1, 2, 3 | 3 |
| 2 | Efficiency | 4, 5, 6, 7 | 4 |
| 3 | Memorability | 8,9,10,11,12 | 5 |
| 4 | Errors | 13, 14, 15, 16, 17, 18, 19, 20 | 8 |
| 5 | Satisfaction | 21,22,23 | 3 |

TABLE 4. GUIDELINES

| Qualitative Data | Score |
|---|---|
| SS (Strongly Agree) | 5 |
| S (Agree) | 4 |
| CS (Quite Agree) | 3 |
| TS ( Not Agree) | 2 |
| STS (Strongly Disagree) | 1 |

The questionnaire has 23 questions that have represented the five aspects of usability given to the respondents.

e. Data Analysis Technique

This research uses descriptive analysis according to the development procedure performed. The first phase of development is done by collecting the reference material plagiarism system. The next step is designing and manufacturing the application followed by small group test for input suggestion improvement for the application. The last phase is a user's feasibility rating.

Analytical steps to determine the application feasibility is done as follows:

1. Change the assessment in qualitative form to quantitative with the provisions in Table 4
2. Calculating the score per item question using the formula:

$$score = \frac{5*n.SS+4*n.S+3*n.CS+2*n.TS+1*n.STS}{5*n} \quad (4)$$

Where:

$nSS$ = the number of respondents strongly agree
$nS$ = the number of respondents answered agree
$nCS$ = the number of respondents answered quite agree
$nTS$ = the number of respondents answered disagree
$nSTS$ = the number of respondents answered strongly disagree
$n$ = the number of respondents.

3. Calculate the average score using the average score formula = $\frac{\sum x}{N}$, where $\sum x$= the total score of all questions and N is the number of respondents
4. Change the average score to a qualitative score with the assessment criteria in Table 5.

TABLE 5. CLASSIFICATION OF WITNESSES CONCLUSIONS OF USABILITY EVALUATION RESULT

| Value | Conclusion |
|---|---|
| 0% - 20% | Strongly disagree that the application is very easy to understand and understand |
| 21% - 40% | Do not agree that the application is very easy to understand and understand |
| 41% - 60% | Simply agree that the application is very easy to understand and understand |
| 61% - 80% | Agreed that the application is very easy to understand and understand |
| 81% - 100% | Strongly agree that the application is very easy to understand and understand |

## IV. RESULT OF DEVELOPMENT

### A. User Interface

In the main application view, there are two main menus namely Data Input menu and Process menu. There are two options for input user document data. The first option copies the text from the input document. The next step, click the Plagiarism Detection button. When any document in the database document has similarity level above the threshold, i.e. 30%, the system will switch from the Input Data menu to the Process menu. However, if the similarity of the input document is below the threshold, a dialog message will appear stating that no database document is similar to the input document.

On the Process menu page, list of the similar document will be shown. On this page, documents have been sorted in descending order based on similarity level. The smaller the value of the similarity level, the more different the two documents. The Process menu page is shown in Figure 4.

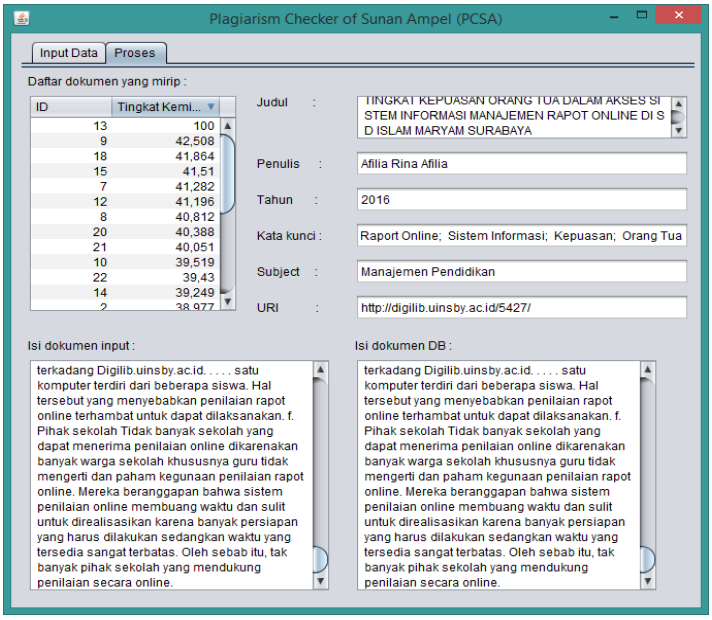


Figure 4. Display of Process menu page

### B. Presentation of Product Test Result Data

#### a) Validation of computational mathematics experts

The instrument for validating algorithm consists of 8 questions. Comments and suggestions obtained from computational mathematicians validation serve as a basis for improving the

efficiency of the algorithm before the application is tested to the user. A scoring diagram per aspect by a computational mathematician is shown in Figure 5.

The maximum score of the overall ideal answer is 40, whereas the computational mathematician assigns 34. The results obtained from the computation mathematician's validation questionnaire are 85% with the description that the application's algorithm eligible for use with slight revisions. Based on the eligibility criteria, Winnowing algorithm that applied to the plagiarism detection application is valid and feasible to use. However, there are several revisions needed to improve the efficiency of the algorithm.

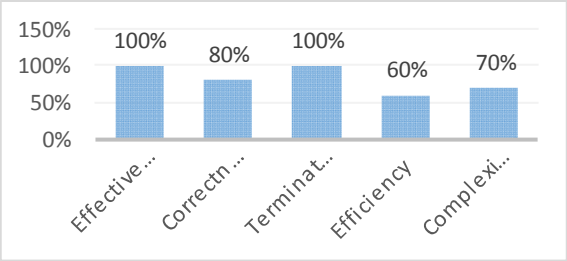


Figure 5. Application appraisal diagram of a computational mathematician

*b) Validation of Software Engineering Expert*

Data validation test results of software engineering experts obtained from two experts in the field of software engineering. Comments and suggestions that obtained from software engineering experts will be the base to improve application performance before application are tested to users.

The result of calculation for the whole item question is 86,67%. Based on the eligibility criteria, the application is reviewed from application compliance, feature completeness, and display. The result shows that application included invalid qualification and eligible to use. However, there are several revisions needed to improve application performance.

*c) User Validation*

User trials are conducted after obtaining valid results against trials that have been done by computational mathematicians and software engineering experts. Diagrams showing user ratings by aspect. From the assessment of large group trial data, it can be seen that the average aspect assessment is 92.27%. Based on the eligibility criteria, the plagiarism detection application included in the qualification is valid and feasible to use.

*C. Presentation of Data Testing*

The purpose of testing to ensure whether the application is built in accordance with the analysis and design done so that the desired goal is achieved. Based on the results of system testing, it can be deduced that functionally, the system can produce the expected output. To summarize the process, the input document involved is an abstract document.

Table 6 describes the application successfully through the testing phase. Applications can detect the resemblance of test documents derived from database documents with full 100% resemblance scores. Applications can detect the resemblance of test documents whose contents come from some of the contents of the database document. The application can detect the resemblance of test documents whose contents come from database documents even though there are grammatical changes. Applications can accurately detect the resemblance of test documents that are composite content from two database documents.

TABLE 6. SUMMARY OF OUTPUT TESTING DOCUMENTS

| No | Type of Test | Level of Change | Result |
|---|---|---|---|
| A | **System Test** | - | 100% |
| B. | **Result Test** | | |
| 1 | Full abstract | 100% | 100% |
| 2 | Partially abstract | 40% | 41% |
| | | 60% | 59% |
| | | 80% | 78% |
| 3 | Abstract grammar changes | 40% modified grammar and 60% content | 61% |
| | | 60% modified grammar and 40% content | 47% |
| | | 80% modified grammar and 20% content | 36% |
| 4 | Combined abstract | 60% abstract 1 40% abstract 2 | 49% abstract 1 40% abstract 2 |
| | | 40% abstract 1 60% abstract 2 | 40% abstract 1 48% abstract 2 |

Detection of similarity testing by manipulating the document through several scenarios ranging from 100% to 20% similarity level, generating a similarity level of documents of 90.12%. The application can detect the resemblance of test documents whose contents come from the database document despite any grammatical changes. Applications can accurately detect the resemblance of test documents that are composite content from two database documents.

*D. Measuring Effectiveness*

Winnowing method was tested on 100 abstract documents. 50 of the 100 abstract documents are plagiarized in different ways such as simple copy paste, altering some terms with synonyms and altering sentence structure (paraphrase) from a repository document. To mention the plagiarized document, given a limit of 30%. Therefore, if the similarity level between the two documents is more than 30%, then the system considers the input document as a plagiarized document.

To evaluate the effectiveness of the implemented system, three common parameters were used for testing: sensitivity, specificity, and accuracy.

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{FP+TN}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Where :

True Positive (TP) : the documents which are copied and are recognized as copies

False Positive (FP) : the documents which are not copied but are recognized as copies

False Negative (FN) : the documents which are copied but are recognized as the originals

True Negative (TN) : the documents which are not copied and are recognized as the originals

TABLE 7. PERFORMANCE EVALUATION OF WINNOWING ALGORITHM

| | Sensitivity | Specificity | Accuracy |
|---|---|---|---|
| Winnowing algorithm | 82% | 100% | 91% |

Results of performance evaluation of Winnowing algorithm shown in Table 7. Sensitivity score is 82%, which means the system ability to detect plagiarism to give a positive result for plagiarism document of 82%. The specificity score is 100%, which means the system ability to perform plagiarism detection to give negative results on the document is not plagiarism of 100%. The accuracy score is 91%, which means the system's ability to correctly detect all documents tested by 91%. It means that the implemented system detects copy paste, synonym replacement and active to passive active conversion with good performance.

## V. CONCLUSION

This application can be used and developed to detect document plagiarism, especially scientific papers. Application development conducted several tests, that is: (1) Tests conducted by computational mathematicians obtain a feasibility level of 85%. (2) Tests conducted by software engineering experts obtain a feasibility level of 86.67%. (3) Tests conducted by the user obtain a feasibility level of 92.27%. (4) The test results are similarities with several engineering scenarios tested with Winnowing algorithm of 90.12%. (5) The document plagiarism detection system using Winnowing algorithm yields 82% sensitivity, 100% specificity, and 91% accuracy. The system works effectively so it can be used to detect document plagiarism.

## ACKNOWLEDGMENTS

The authors are grateful to the Ministry of Religious Affairs of the Republic of Indonesia and Sunan Ampel State Islamic University for their support and cooperation.

## REFERENCES

[1] R. V Smith, L. D. Densmore, and E. F. Lener, “Ethics and the Scientist,” pp. 79–91, 2016.

[2] S. Solarino, *Ethical Behavior in Relation to the Scholarly Community : A Discussion on Plagiarism*. Elsevier Inc., 2015.

[3] R. Sutoyo, I. Ramadhani, and A. D. Ardiatma, “Detecting Documents Plagiarism using Winnowing Algorithm and K-Gram Method,” in *Cybernetics and Computational Intelligence (CyberneticsCom), 2017 IEEE International Conference on*, 2017, pp. 67–72.

[4] S. Wibirama, “Viper: cara mudah mendeteksi plagiarisme,” 2013. [Online]. Available: http://wibirama.staff.ugm.ac.id/2013/01/29/sunu-wibirama-viper-cara-mudah-mendeteksi-plagiarisme/. [Accessed: 29-May-2018].

[5] B. Marsh, “Turnitin . com and the scriptural enterprise of plagiarism detection,” vol. 21, pp. 427–438, 2004.

[6] G. P. V and J. D. Velásquez, “Engineering Applications of Artificial Intelligence Docode 5 : Building a real-world plagiarism detection system,” *Eng. Appl. Artif. Intell.*, vol. 64, no. July 2016, pp. 261–270, 2017.

[7] S. Vie, “A Pedagogy of Resistance Toward Plagiarism Detection Technologies,” *Comput. Compos.*, vol. 30, no. 1, pp. 3–15, 2013.

[8] S. Xie, M. Imani, E. R. Dougherty, and U. M. Braga-neto, “Nonstationary Linear Discriminant Analysis,” pp. 161–165, 2017.

[9] Q. Li, S. Authentication, and C. Technology, “Chapter 4 Non-Stationary Pattern Recognition,” pp. 61–73, 2012.

[10] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, “Using of Jaccard Coefficient for Keywords Similarity,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2013, vol. I.

[11] S. Schleimer, D. S. Wilkerson, A. Aiken, and U. C. Berkeley, “Winnowing : Local Algorithms for Document Fingerprinting,” 2003.

[12] N. Kock and A. Chandra, “Dealing with Plagiarism in the Information Systems Plagiarism and Ways to Address Them,” vol. 27, no. 4, 2003.

[13] S. Sastroasmoro and A. Einstein, “Beberapa Catatan tentang Plagiarisme *,” pp. 239–244, 2007.

[14] N. Elbegbayan, “Winnowing, a Document Fingerprinting Algorithm.” Linkoping University, pp. 1–7, 2005.

[15] A. Putera *et al.*, “K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm,” vol. 6, no. 07, 2017.

[16] G. Myles, “k-gram Based Software Birthmarks,” in *ACM Symposium on Applied Computing*, 2005, pp. 314–318.

[17] Z. Fuyao, “A String Matching Algorithm Based on Efficient Hash Function.” .

[18] S. J. (Eds. H.C.A. van Tilborg, “Encyclopedia of Cryptography and Security,” no. D, p. 838, 2011.

[19] B. Metode, P. Kuantitatif, and K. Dan, *Sugiyono Metode Penelitian Kuantitatif Kualitatif Dan R D DOWNLOAD*. 2017.

[20] J. NIELSEN, “Usability 101: Introduction to Usability,” 2012.