# The Similarity of Essay Examination Results using Preprocessing Text Mining with Cosine Similarity and Nazief-Adriani Algorithms

**Rika Rosnelly[1], Dedy Hartama[2], Muhammad Sadikin[3], Cindy Paramitha Lubis[4], Mutiara S. Simanjuntak[5],Sandy Kosasi[6]**

[1,2,3,4,5]Faculty of Engineering and Computer Science, Universitasof Potensi Utama, Medan, Indonesia
[6]STMIK Pontianak, Indonesia
[1]rikarosnelly@gmail.com,[2]dedyhartama@amiktunasbangsa.ac.id, [3]dicky.aries.3@gmail.com, [4]cindyparamitha96@gmail.com, [5]sarahwatymearl@gmail.com,[6]sandykosasi@yahoo.co.id

**Article History**: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27January 2021; Published online: 05April 2021

**Abstract**. Exams are one way to measure the level of students' ability to participate in learning. One type of exam given to students is the essay type. This study focuses on making automatic assessments for essay-type exams using cosine similarity. This method has several stages such as folding Case, tokenizing, filtering, stemming, analyzing, weighing of words in documents with cosine similarity. The stemming process uses the Nazief & Adriani algorithm. The results of this study are to conclude that the choice of words that are considered as keywords in the answer key greatly affects the results of the system's assessment. This is evidenced by testing applying the cosine law of 89.5%. However, there are several types of questions that are significantly different because there are unique characters in the database and answer keys that do not contain keywords that match the correct answer.

**Keywords:**automatic, stemming, assessment, database, answer key

## 1. Preliminary

At this time the world is feeling the impact of the Coronavirus Disease (Covid 19) pandemic. Indonesia is one of the countries that has been badly affected, especially in the field of education, causing schools and universities to be unable to carry out the face-to-face learning process. Learning is transferred by applying online methods or online learning using media such as google classroom, zoom, WhatsApp, and other methods[1]. The development of technology in the digital era as it is today certainly brings many benefits to society and one of them is education institutions[2].

The application of online learning methods is also applied to the implementation of the Mid-Semester Examination (UTS) and the Final Semester Examination (UAS). Exam questions given to students can be in the form of essays or multiple choices. Multiple choice questions are filled by selecting answers from those provided. It is different from essay questions which require students to provide the answers they have according to the student's understanding. The answers produced by students are not solely right or wrong, but there is also a possibility that it is close to correct. As an application, if the perfect answer is 100 then the wrong answer is given a value of 0 and the close answer is given a value of 40, and so on. Therefore,

Previous research was conducted by Rahimi Fitri and Arifin Noor Asyikin by applying the Cosine Similarity algorithm to the student essay exam assessment cases. The Prepossessing Text Mining stage, precisely at the stemming stage, does not apply an algorithm, so the process of determining the root word is ineffective and does not have criteria[3]. The stemming process influences the accuracy of information retrieval. Stemming is done by removing the affixes contained in words.

Another study was conducted by Saipech, Pongsakorn, and Pusadee in the case of detecting similarity in test results for Thai by applying the Cosine Similarity algorithm. In this study, the DCB approach was applied to Thai for the word segmentation process. Prepossessing in this study was limited to Word Segmentation and Stop word elimination[4].

The choice of a method or algorithm for a case must also be precise because it depends on the objectives and the results of its accuracy[5]. In this study, the authors used the Prepossessing Text mining stage by applying the Nazief and Adriani algorithms at the Stemming stage for Indonesian words. Many algorithms have been developed to carry out the Indonesian stemming process, including the Nazief and Andriani algorithms,

Porter's algorithms, and the Arifin and Setiono algorithms.[6]. The research scenario with stemming resulted in an average similarity value of 10% higher than without stemming[7].

The author applies the Cosine Similarity method to analyze student answers to produce a similarity in these answers. Then combined with the Nazief & Adriani algorithm for the stemming process of words.

### A. Information retrievals System

*Information retrievals System* or the information retrieval system is one of the clumps of computer science relating to information retrieval in document collections both in content and in the context that must be found to realize the desire of users for information[8].

Information that can be obtained from the Information Retrievals System can be in the form of text, pictures, audio, and video which are useful for searching for information and maintaining information.[9].

### B. Stemming

A process contained in the IR (Information retrieval) system is a stemming process. This stemming process is responsible for transforming the words contained in a document into the root word by applying certain rules.[9].

*Stemming* It is also one of the steps used for booster performance (improving performance). Information retrieval in Indonesian text is intended to remove suffixes, confixes, and prefixes, of course, different from English text where the stemming process is used to remove suffixes [10].

### C. Nazief -Adriani Algorithm

The Nazief-Adriani algorithm was first developed by Bobby Nazief and Mirna Adriani. The Nazief and Adriani stemming algorithm was developed based on Indonesian morphological rules which are grouped by prefixes, suffixes, and confixes called conjunctions.[11].

The basic word dictionary is used for the Nazief & Adriani Algorithm and is supported for recording, such as the compilation of words that are subjected to an excessive stemming process. The grouping of affixes into several categories according to the morphological rules of Indonesian is as follows[12]:

1. *Inflection suffixes* are a group of suffixes whose root word does not change. For example, the word "eat" which is given the "-lah" ending would become "eat". This group can be divided into two:
- *Particle* (P) such as, "pun", "tah", "- kah" and "-lah"
- *Possessive pronoun* (PP) or possessive pronouns such as "-ku", "-nya" and "-mu".

2. *Derivation suffixes* (DS) is a collection of original Indonesian suffixes added directly to the root words, namely the suffix "-kan", "-an" and "-i" ,.
3. *Derivation prefixes*(DP) is a pure root word that is immediately given a prefix or a root word that has been added up to two prefixes. It includes things like:
- "Be-", "te", "pe-" and "me" which are morphological prefixes
- "Ke-", "se-" and "di-" or prefixes have no morphology.

The form of affix words in Indonesian based on the affix classification above can be modeled as follows [13]:

$$\left[\mathrm{DP} + \left[\mathrm{DP} + [\mathrm{DP} +]\right]\right] \text{Kata Dasar } [[+\mathrm{DS}][+\mathrm{PP}]]$$

Information:
DP: Derivation prefixes
DS: Derivation suffixes
PP: Possessive pronoun

The rules used in the Nazief & Adriani algorithm are as follows[13]:
1. The combinations of prefixes that are not allowed are “se-kan”, “be-i”, “ke-kan”, “ke-i”, “me-an”, “te-an” and “se-i”.
2. It is not permissible to use affixes repeatedly.
3. If a word consists of only one or two letters, the process cannot be carried out.

4. Prefix added to change the original form of the root word or prefix that has been previously given for example the prefix "men" can change to "men-", "mem", meng- "and" meny- ". Therefore we need a rule in dealing with morphology.

The Nazief & Adriani algorithm made by Bobby Nazief and Mirna Adriani has processing stages as outlined in the following formula[13]:

$Prefiks\ 1 + Prefiks\ 2 + Katadasar + Sufiks\ 3 + Sufiks\ 2 + Sufiks\ 1$ ..... (1)

1. First look for the word you want to import into the basic word dictionary. If found, it is assumed that the word is the root word. Then the algorithm stops.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, or “-nya”) are discarded. If in the form of particles ("-lah", "-kah", "- tah" or "-pun") then this step is repeated again to remove the Possesive Pronouns ("-ku", "-mu", or "-nya") , If there is.
3. Removal of Derivation Suffixes (“-i”, “-an” or “-kan”). If a word is found in the dictionary base, the algorithm stops. If not then go to step c1

a. If the word "-an" has been deleted and the last letter of the word is "-k", then "-k" is also deleted. If the word is found in the dictionary, the algorithm stops. If not found then do step c2.
b. Deleted suffixes (“-i”, “-an” or “-kan”) are returned, go to step 4

4. Remove Derivation Prefix. If in step c any suffixes are removed then go to step d1, otherwise go to step d2.

a. Check the table of disallowed prefix-suffix combinations. If found, the algorithm stops, otherwise
b. go to step d2.
c. For i = 1 to 3, specify the prefix type then remove the prefix. If the root word has not been found, do step e, if so, the algorithm stops. Note: if the second prefix is the same as the first prefix the algorithm stops.

5. Recording. If all steps have been completed but are not successful, the initial word is assumed to be the root word. Process complete

### D. Cosine Similarity Method

*Cosine Similarity* is a measure of the similarity used in retrieval information and the size of the point of view between the document vector Da (point (ax, bx)) and Db (point (ay, by)). Each vector is represented in each word in the document (text) which is compared in the form of a triangle so that the law of cosine can be applied to state that[14]:

$$Similarity = \cos(\theta) = \frac{A.B}{||A||.||B||} \quad \text{...Pers. (2)}$$

Simply put, Cosine Similarity is used to compare the similarity level of a document with the cosine degree concept where the results are limited between 0 and 1. Documents are said to be different if the value is 0. Documents are said to be similar if the results of cosine similarity 1

$$\cos(\theta) = \frac{\sum_{i=0}^{n} A.B}{\sqrt{\sum_{i=1}^{n}(A_i)^2.}\sqrt{\sum_{i=1}^{n}(B_i)^2.}} \text{.....Pers. (3)}$$

Information:

A = Student Answers
B = Lecturer Answer
Ai = Weight of word i in block Ai
Bi = Weight of word i in block Bi
i = The number of words in the sentence
n = Number of vectors

## 2. Research Methods

The research methodology used in this study is a qualitative research method. The qualitative method is research that aims to understand the phenomena experienced by research subjects as a whole in the form of words and language in a natural context. The stages of research carried out in this study are as follows:

1. Data collection

Collecting data on questions and answer keys for the Data Mining course at AMIK Tunas Bangsa Medan, North Sumatra which will be tested as well as collecting supporting concepts or theories in this research.

2. Essay Exam Modeling

Determine the right keywords from the answer key as a reference for examination assessments, check the answers from students by making the keywords as a reference for the correct answers then calculate all the resulting values from the weight calculation of each question and add them to the maximum value of the questions which will become the final score college student.

3. Essay Exam Architecture

The method used is to match the answer key with the answer from the student and to fix the system functions when an error occurs.

4. Algorithm Implementation

The stages of matching answers between student answers and answer keys using the Cosine Similarity method are accompanied by the results of data processing. The creation of an automatic assessment system has several stages, namely, preprocessing and analyzing. Preprocessing includes several stages such as: tokenizing, filtering, stemming[15], while the analyzing stage is the calculation of weighting of words in documents with cosine similarity. This process is the process of changing words into basic words. Stemming process in this exam sample uses Nazief and Adriani algorithms.

## 3. Results and Discussion

The trial process used 60 student samples with three different questions then the results of the assessment of the answer keys were compared with the results of students' answers using the Nazief & Adriani algorithm calculation process and combining the cosine similarity method.
The framework of this study is shown in Figure 1.

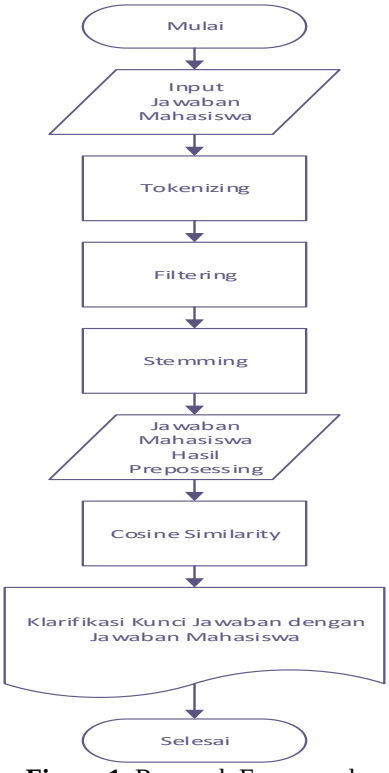


**Figure 1.** Research Framework

The following are the results of calculations using the Nazief & Adriani algorithm method and cosine similarity as follows:

1. Case Folding and Tokenizing

At the case folding stage, a change will be made to all small letters. Can be seen in table III.1:

**Table III.1.** Case Folding

| Sentence | Result of Case Folding |
|---|---|
| A (Student Answers) | If outlook is overcast then yes if outlook rain and wind is false then yes or wind is true then no if outlook is sunny and humidity is <77.50 then no or humidity <= (less or equal to) 77.500 then yes. |
| B (Answer Key) | if outlook = overcast then yes if outlook = rain and wind = false then yes or wind = true then no if outlook = sunny and humidity => 77,500 then no or humidity <= 77,500 then yes |

Then proceed with the tokenizing stage by breaking the sentence into several words. The tokenizing stages can be seen in table III.2:

**Table III.2.** Tokenizing

| Sentence | Tokenizing results |
|---|---|
| A (Student Answers) | if, outlook, overcast, then, yes, if, outlook, rain, and, wind, false, then, yes, or, wind, true, then, no, if, outlook, sunny, and, humidity, then, no, or, humidity, more, less, or, equal, then, yes. |
| B (Answer Key) | if, outlook, overcast, then, yes, if, outlook, rain, and, wind, false, then, yes or, wind, true, then, no, if, outlook, sunny, humidity, then, no, or, humidity, then, yes |

**2.** Filterin**g**

In the filtering stage, the process of deleting words that are considered to have no effect on the core of the sentence is carried out. Data decapitation is done by deleting the words "di-, ke-, se-". The stages of filtering results can be seen in table III.3:

**Table III.3.** Filtering

| Sentence | Filtering Results |
|---|---|
| A (Student Answers) | if, outlook, overcast, then, yes, if, outlook, rain, and, wind, false, then, yes, or, wind, true, then, no, if, outlook, sunny, and, humidity, then, no, or, humidity, more, less, or, equal, then, yes. |
| B (Answer Key) | if, outlook, overcast, then, yes, if, outlook, rain, and, wind, false, then, yes or, wind, true, then, no, if, outlook, sunny, humidity, then, no, or, humidity, then, yes |

3. Stemming

At the stemming stage, word variables are broken down into basic words. At this stemming stage using the Nazief & Adriani algorithm process can be seen in table III.4:

**Table III.4.** Nazief & Adriani Stemming Algorithm

| Sentence | Stemming Results |
|---|---|
| A (Student Answers) | if, outlook, overcast, then, yes, if, outlook, rain, and, wind, false, then, yes, or, wind, true, then, no, if, outlook, sunny, and, humidity, then, no, or, humidity, more, less, or, equal, then, yes. |
| B (Answer Key) | if, outlook, overcast, then, yes, if, outlook, rain, and, wind, false, then, yes or, wind, true, then, no, if, outlook, sunny, humidity, then, no, or, humidity , then, yes |

4. Analyzing

At the analyzing stage using the cosine similarity method to achieve the similarity value of sentences A and B. The analyzing stage is obtained from the number of each word of the Nazief & Adriani algorithm which can be seen from Table III.5:

**Table III.5.** The result of the stemming algorithm Nazief & Adriani

| No. | Word | A | B |
|---|---|---|---|
| 1 | if | 3 | 3 |
| 2 | outlook | 3 | 3 |
| 3 | overcast | 1 | 1 |
| 4 | then | 5 | 5 |
| 5 | yes | 3 | 3 |
| 6 | rain | 1 | 1 |
| 7 | and | 2 | 1 |
| 8 | wind | 2 | 2 |
| 9 | false | 1 | 1 |
| 10 | or | 3 | 2 |
| 11 | true | 1 | 1 |
| 12 | no | 2 | 2 |
| 13 | sunny | 1 | 1 |
| 14 | humidity | 2 | 2 |
| 15 | more | 1 | 0 |
| 16 | small | 1 | 0 |
| 17 | same | 1 | 0 |
| 18 | with | 1 | 0 |

After finding the results of the stemming algorithm Nazief & Adriani, the calculation is done using equation 3, namely:

$$Similarity = \frac{(3x3)+(3x3)+(1x1)+(5x5)+(3x3)+(1x1)+(2x1)+(2x2)+(1x1)}{\sqrt{(3+3+1+5+3+1+2+2+1+3+1+2+1+2+1+1+1+1)X}\ \frac{(3x2)+(1x1)+(2x2)+(1x1)+(2x2)+(1x0)+(1x0)+(1x0)+(1x0)}{\sqrt{(3+3+1+5+3+1+1+2+1+2+1+2+1+2+0+0+0+0}}}$$

$$Similarity = \frac{77}{9{,}274\ X\ 8{,}602}$$

$$Similarity = \frac{77}{79{,}755}$$

$$Similarity = 0{,}965$$

After calculating the cosine similarity process, the results obtained were 0.965 with a similarity percentage of answers of 96.5%.

5. Value calculation process

The results of the assessment were obtained from one of the students who obtained a similarity presentation result of 96.5%, 82%, 90% with a total of 3 questions. The following is the calculation of the score:

$$Nilai = \frac{96{,}5 + 82 + 90}{3} = 89{,}5$$

From the value above, it can be concluded that the mean level of the calculation of the value is above 80%, meaning that the answer keys and the results of the students' answers that have been compared have significant word similarities with a total value of 89.5%.

## 4. Conclusion

The conclusion results by applying the Cosine Similarity and Nazief & Adriani Algorithms to the Essay Exam Assessment concluded that the choice of words that are considered as keywords in the answer key greatly affects the results of the assessment of the system. The results of the tests carried out get a match accuracy value of 89.5%

From this research, several suggestions are given for further research, including paying attention to synonyms and anonymous words, then it is hoped that better weighting of the answer keys will be used which will be used as a reference in conducting assessments to improve the performance of the results of student answers.

## 5. Acknowledgement

We would like to extend our grateful for the support from STMIK Pontianak for funding this research fully.

## References

1. W. Darmalaksana, RYA Hambali, A. Masrur, and Muhlas, "Analysis of Online Learning during the WFH Pandemic Covid-19 as a Challenge for 21st Century Digital Leaders," *UIN Sunan Gunung Djati Bandung*, vol. 1, no. 1, pp. 1–12, 2020.
2. S. Adnan, TS Gunawan, H. Nasir, and M. Kartiwi, "Development of Social Network Analysis Platform," *IEEE 2019 Int. Conf. Smart Instrumentation, Meas. Appl. (ICSIMA 2019)*, no. August, pp. 27–29, 2019, doi: 10.1109 / ICSIMA47653.2019.9057327.
3. F. Rahimi and AN Asyikin, "Automatic Essay Assessment Application Using the Cosine Similarity Method," *Shaft Tech.*, vol. 7, no. 2, pp. 88–94, 2015, [Online]. Available: http://ejurnal.poliban.ac.id/index.php/porosteknik/article/view/218.
4. P. Saipech and P. Seresangtakul, "Automatic Thai Subjective Examination using Cosine Similarity," *ICAICTA 2018 - 5th Int. Conf. Adv. Informatics Concepts Theory Appl.*, pp. 214–218, 2018, doi: 10.1109 / ICAICTA.2018.8541276.
5. M. Sadikin, R. Rosnelly, and TS Gunawan, "Comparison of Classification Accuracy Levels of Permanent Lecturer Admission Using the Naive Bayes Classifier and C4 Methods. 5, "vol. 4, pp. 1100–1109, 2020, doi: 10.30865 / mib.v4i4.2434.
6. D. Novitasari, "Comparison of Porter's Stemming Algorithm with Arifin Setiono to Determine the Level of Accuracy of Basic Words," *STRING (Unit of Writing Ris. And Inov. Teknol.*, vol. 1, no. 2, p. 120, 2017, doi: 10.30998 / string.v1i2.1031.
7. MZ Naf'an, A. Burhanuddin, and A. Riyani, "Application of Cosine Similarity and TF-IDF Weighting to Detect Document Similarities," *J. Linguist. Computational*, vol. 2, no. 1, p. 23, 2019, doi: 10.26418 / jlk.v2i1.17.
8. D. Wahyudi, T. Susyanto, and D. Nugroho, "Implementation and Analysis of Stemming Algorithm Nazief & Adriani and Porter in Indonesian Language Documents," *J. Ilm. SINE*, vol. 15, no. 2, 2017, doi: 10.30646 / sinus.v15i2.305.
9. A. Prasidhatama and KM Suryaningrum, "Comparison of Nazief & Adriani Algorithm and Idris Algorithm for Basic Word Search," *J. Technol. and Manaj. Inform.*, vol. 4, no. 1, pp. 1–4, 2018, doi: 10.26905 / jtmi.v4i1.1773.
10. HT Nugroho, "The Influence of the Nazief-Adriani Stemming Algorithm on the Performance of the Winnowing Algorithm to Detect Indonesian Plagiarism," *J. Ultim. Comput.*, vol. 9, no. 1, pp. 36–40, 2017, doi: 10.31937 / sk.v9i1.572.
11. MW Sardjono, M. Cahyanti, M. Mujahidin, and R. Arianty, "Detection of Word Similarities for Indonesian Writing Titles Using the Nazief-Adriani Stemming Algorithm," *Sebatik 2621-069X*, pp. 138–146, 2018.
12. A. Bastian, H. Sujadi, and PA Sukmana, "Design and Design of Essay Exam Assessment Applications Using the Nazief & Andriani Algorithm and the Cosine Similarity Method," *infotech J.*, vol. 4, no. 2, pp.

62–68, 2018, [Online]. Available: http://jurnal.unma.ac.id/index.php/infotech/article/viewFile/1168/1068.

13. PN Banjarmasin, Y. Anistyasari, E. Hariadi, JT Informatics, and UN Surabaya, "NEW ALGORITHM FOR THE FORMATION OF BASIC WORD IN INDONESIAN STEMMING PROCESS," *Pros. SNRT (Seminar Nas. Ris. Ter.*, vol. 5662, no. November, pp. 70–76, 2019.
14. M. AGUS SALIM, "Development of Online-Based Essay Assessment Applications Using the Nazief and Adriani Algorithms with the Cosine Similarity Method," *It-Edu*, vol. 2, no. 01, 2017.
15. H. Hartono, OS Sitompul, T. Tulus, EB Nababan, and D. Napitupulu, "Hybrid Approach Redefinition (HAR) model for optimizing hybrid ensembles in handling class imbalance: A review and research framework," *MATEC Web Conf.*, vol. 197, 2018, doi: 10.1051 / matecconf / 201819703003.
16. Rosnelly, R., Gunawan, T., Paramitha, C., & Sadikin, M. (2020). Decision Support System Application Evaluation of Transformer Isolation Condition with Simple Additive Weighting (SAW) Method. Jurnal Abdimastek (Pengabdian Masyarakat Berbasis Teknologi), 1(1), 41-48