Indonesian Words Error Detection System using Nazief Adriani Stemmer Algorithm

Anton Yudhana[1], Abdul Fadlil[2]
Electrical Engineering Department
Universitas Ahmad Dahlan Yogyakarta, Indonesia

Muhamad Rosidin[3]
Informatics Engineering Department
Universitas Ahmad Dahlan, Yogyakarta, Indonesia

***Abstract*—Stemming in each language has a different process and is determined according to the structure of the language. Stemming is mostly used as a complete step in the processing of words and phrases. There are many stemming algorithms available, and some used as a process for word processing. One function of stemming is to detect word errors in Indonesian. In this study, researchers created the Indonesian words error detection system using Nazief and Adriani algorithm. In the trials conducted, the system will accept text input obtained from the user. Then the system will preprocess the text. In this study, there are three stages of preprocessing, namely tokenization, case folding, and filtering. After the stages in preprocessing are finished, the system will call each word for the process of stemming. The results of the stemming will be compared with the base words available in the database. If it does not match, then the word is highlighted and is considered an error word. The first finding is the Nazief Adriani's algorithm can be able to detect words error until 100%. The second finding is the Nazief Adriani's algorithm also detect non-words error, the accuracy of detecting is 97.464%.**

Keywords*—Indonesian; word error; stemming; Nazief and Adriani stemmer algorithm; detection system

I. Introduction

Affixes can be easily found in Indonesian because it uses a lot of affixes. Affixes can be used in all Indonesian words and it can be combining each other [1]. There are three types of affixes in Indonesian, namely prefixes, insertions, and suffixes. It is not simple to separate words that contain affixes into base words. There are base words whose letters initially change when given an affix. This rule makes it difficult to use the right words. The word that containing affixes can be changed into base words using the stemming algorithm. The implementation of stemming is very large because stemming is the most important part of text mining. An example of stemming development can be found in research about plagiarizing. Indeed, the phenomenon of plagiarizing in the scope of Indonesian education has long occurred so that educational institutions are tainted by plagiarism act [2]. Based on the description above, the researchers will develop the Indonesian words error detection system using Nazief Adriani's algorithm.

Typing has two ways, namely typing by looking at the keyboard and typing without looking at the keyboard [3]. Nowadays, to detect error words in a document is very difficult because it has been checked manually.

In the previous study, stemming algorithm implementation has been carried out in detecting word errors, for example, in the research of Marsel Widjaja and Seng Hansun (2015) [4]. The stemming can be done with Nazief and Adriani, Porter, Confix Stripping, Enhanced Confix Stripping, Porter Stemmer, and Modified Porter Stemmer algorithm.

II. Research Method

A. Proposed Method

The ability of the algorithm used in this method will be tested. Then the test results showed error words and processing time. The steps used in this study can be seen in Fig. 1.

In Fig. 1, it can be seen that the initial stage in carrying out this research is the design of the Indonesian words error detection system. Then implement a design that has been created using the PHP programming language. Then enter the Nazief Adriani algorithm into the Indonesian words error detection system. At the last stage is the testing of system functions.

B. Nazief Adriani's Stemmer Algorithm

The stemming algorithm in this study is based on Nazief and Adriani's algorithm [5]. The flow chart of the Nazief Adriani algorithm can be seen in Fig. 2.

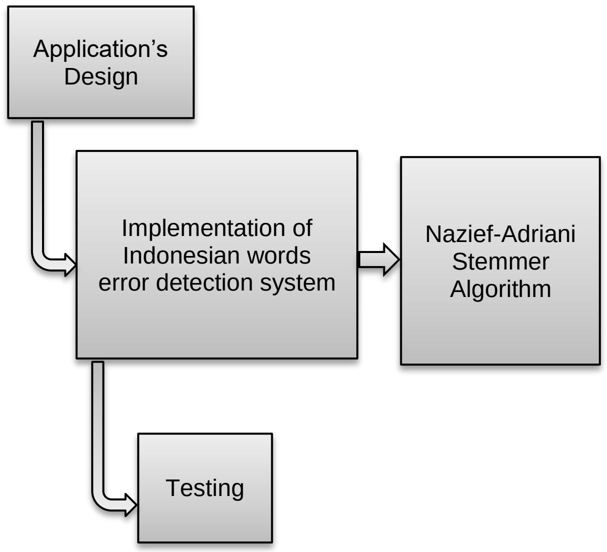

Application's Design
Implementation of Indonesian words error detection system
Nazief-Adriani Stemmer Algorithm
Testing

Fig. 1. Proposed Method.

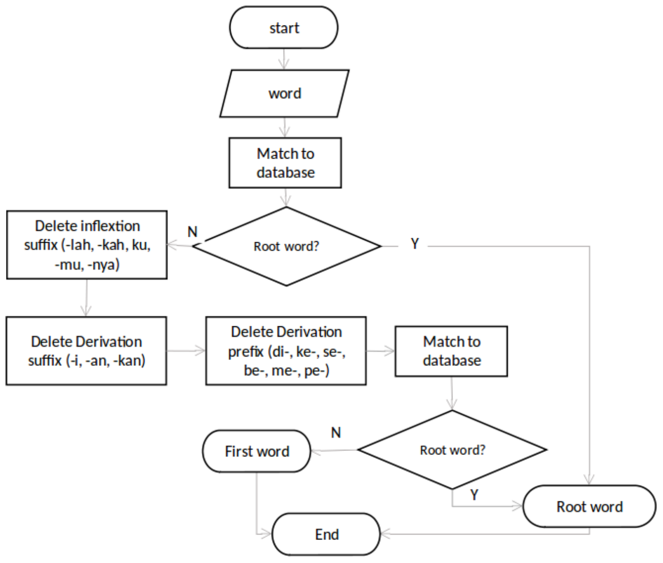

start
word
Match to database
Delete inflextion suffix (-lah, -kah, ku, -mu, -nya)
N
Root word?
Y
Delete Derivation suffix (-i, -an, -kan)
Delete Derivation prefix (di-, ke-, se-, be-, me-, pe-)
Match to database
N
First word
Root word?
Y
Root word
End

Fig. 2. Flow Chart of the Nazief Adriani Stemmer Algorithm.

Nazief Adriani's algorithm has been used in research [6], [7], [8], dan [9]. The algorithm created by Bobby Nazief and Mirna Adriani has the following stages:

1) Check the original word: The algorithm checks the original word towards a base word dictionary. If it works, thus algorithm stops, and the word is declared as the base word. If it fails, the algorithm goes to the next step.

2) Remove the inflection suffix: The algorithm removes the inflection suffix ("- lah", "–kah", "-ku", "-mu", "-nya"). If it works and the inflection suffix is a particle ("-lah" atau "-kah"), the algorithm eliminates possessive pronoun inflection ("-ku", "-mu", "-nya").

3) Remove the derivation suffix: The algorithm removes the derivation suffix ("-i", "-an", "-kan"). If it works, thus the algorithm continues to step 4. If step 4 fails, the algorithm continues to step a, as follow:

a) Delete the character"-k": If the derivation suffix is "-an" and the last character of the word is "-k", the algorithm removes the "-k". Then, proceed to step 4. If it fails, go to step b.

b) Restore the original word: The algorithm returns the deleted suffix ("-i", "-an", "-kan") to the original word.

4) Remove the derivation prefix: The algorithm removes the derivation prefix, consists of several steps:

a) Unauthorized prefixes and suffixes: If the removal of inflection suffixes in step 3 is performed, the algorithm checks for unauthorized prefixes and suffixes. If the algorithm finds it, the algorithm will returns.

b) Similar prefixes: Check if the current prefix is similar to the previous prefix, then the algorithm is returned.

c) Limitation of derivation prefix deletion: If removal of derivation prefix has been performed three times, the algorithm is returned.

d) Check and delete the derivation prefix: The algorithm checks the type of derivation prefix and removes the prefix.

e) Find the root word: If the root word is found, the algorithm is returned. Instead, step 4 repeats again to removes the second prefix.

f) Recoding: The algorithm does the recoding process, depending on the type of prefix.

5) Recording: The algorithm is recording the process.

If the algorithm is failed in doing all steps above, then the first word is assumed to be the base word. So the process is complete [10].

III. Proposed System

A. Flowchart System

The design's process of the study uses the main flowchart that can be observed in Fig. 3.

Fig. 3 shows the main flowchart in this study. Initially, the system will accept text input obtained from the user. Then the system will be preprocessing the text. There are three stages of preprocessing. The first stage is case folding, where the contents of the text will be changed to the default form, usually lowercase. The second stage is tokenizing, where the system will parse the input text into units of words. The third stage is filtering, where the system will eliminate characters that are not needed in the next process. After the preprocessing stages are completed, the system will call each word using an array to perform the stemming process using the Nazief Adriani algorithm. The results of the stemming will be matched with the base words available in the database to confirm their validity. If it does not match, then the word is highlighted and is considered an error word.

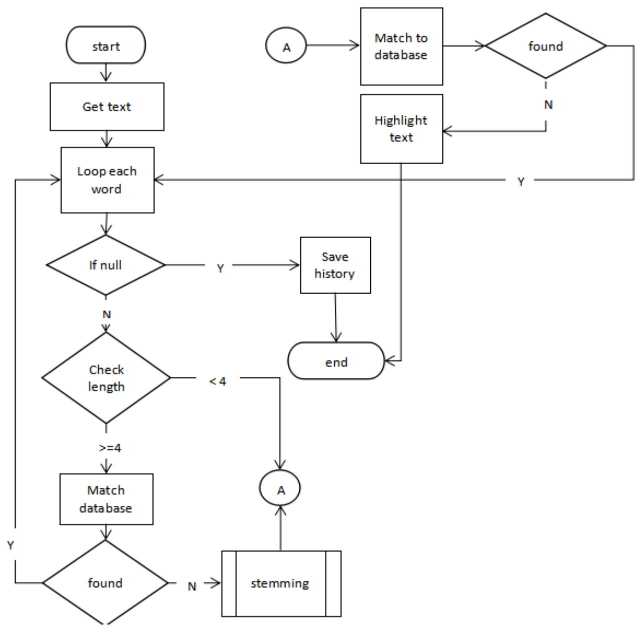


Fig. 3. Main Flowchart System.

B. *Information Retrieval*

Information Retrieval is the stage in identifying or retrieving documents from directories (files) as feedback in requests for information [11]. Information retrieval of researchers explains that queries are the basis for providing better search engine performance [12]. Specific techniques are needed to retrieve documents relevant to user requests, one of the techniques that can be used is Information Retrieval (IR) [13].

C. *Text Processing*

Text preprocessing is part of building the text corpus. Building a text corpus has two main steps, namely collecting and preprocessing [14]. Text preprocessing is an early stage of semantic analysis (meaning accuracy) and syntactic analysis (arrangement accuracy) [15]. The steps in Indonesian text processing consist of; case folding, tokenizing, stopword removal, and stemming. Before the process of stemming begins, the document must be preprocessed. In this study, text processing consists of tokenization, case folding, filtering and stemming [16].

D. *Preprocessing*

Preprocessing is to eliminate characters and words that are not relevant to the document [17]. Omitting the information will facilitate and improve word processing [18]. Preprocessing in text mining is expected to reduce the processing time by eliminating unnecessary words or text from texts or documents. [19]. At this stage, a combination of four preprocessing methods that are commonly used includes: tokenization, case folding, stop word removal, and stemming.

1) Case folding: The process of changing a capital letter into lowercase letters in a document (a-z). In this study, the case folding process is done by calling a function directly in the PHP programming language.

2) Tokenizing: Tokenizing is the stage used to separate or eliminate input strings based on each word from its constituents or separate each word arranged in the document. The omitted part can be numbers, characters or symbols, and punctuation in addition to the letters of the alphabet [20].

3) Filtering: Remove words that have been listed in the stopword or stoplist. Stopwords are words that often appear in large amounts of text and are considered to have no significance [21]. In this study, there is no words are deleted because each word will be verified in the database.

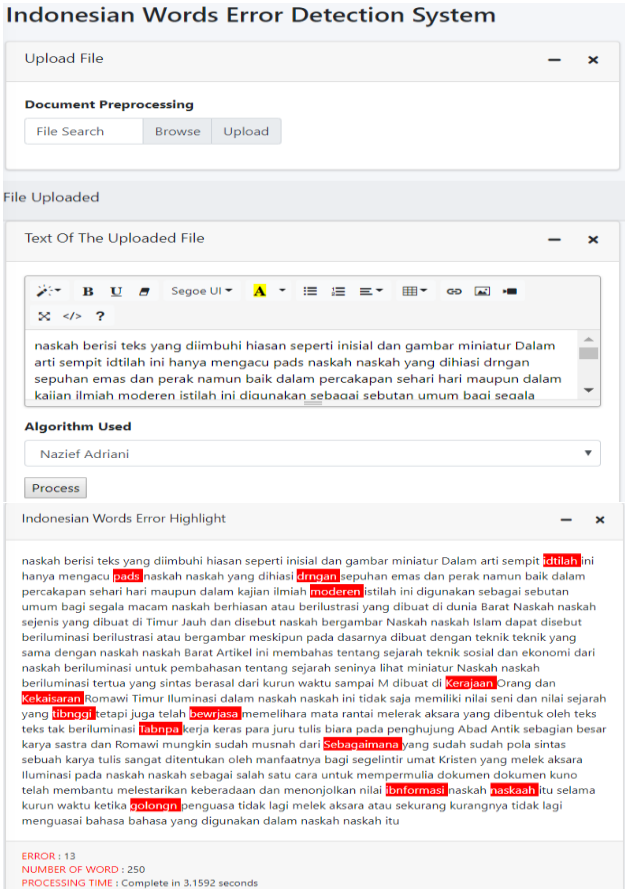
4) Stemming: The stemming process in Indonesian is more complicated than English because there are variations of affixes that must be removed to get the base word [22]. The structure of Indonesian morphology has a higher level of complexity than English [23]. Besides Indonesian and English, stemming can be used in Arabic, as in research [24]. Stemming is more efficient for Arabic retrieval than for English [25]. Stemming is the process of determining the base words of words that contain affixes. Nazief Adriani is one of the most commonly used stemming algorithms. [26]. There is also a pretty good porter algorithm in the process of stemming [27]. Stemming is implemented in the appropriate affix. In Indonesian, the same intonation can give different meanings depending on the topic domain of the word or term. For example, the Indonesian greeting "kemeja" with the same intonation can be written as "ke meja" (go to the table) or "kemeja" (a dress) [28].

IV. Result and Analysis

A. *System Interface*

Display system interface created using the PHP programming language. Processed documents will be stored in the system. Display on the head of the page for the title of the application. In the middle, there is a document input form and an upload button. If the upload button is clicked, it will display the contents of the document. At the bottom, there are algorithms and process buttons. If the process button is clicked, it will display the word that has been highlighted and interpreted as a word error. In addition, there is also a table view of the results of stemming. The system interface display can be seen in Fig. 4 and highlights the words in Fig. 5.

In addition to the page containing the word error highlight in Fig. 5, there is also a table displaying the results of the stemming process using Nazief and Adriani's algorithm. The words displayed are only words that have an affix and the results of the stemming. But with the wrong word, the results of stemming cannot be seen. The stemming result table can be seen in Fig. 6.



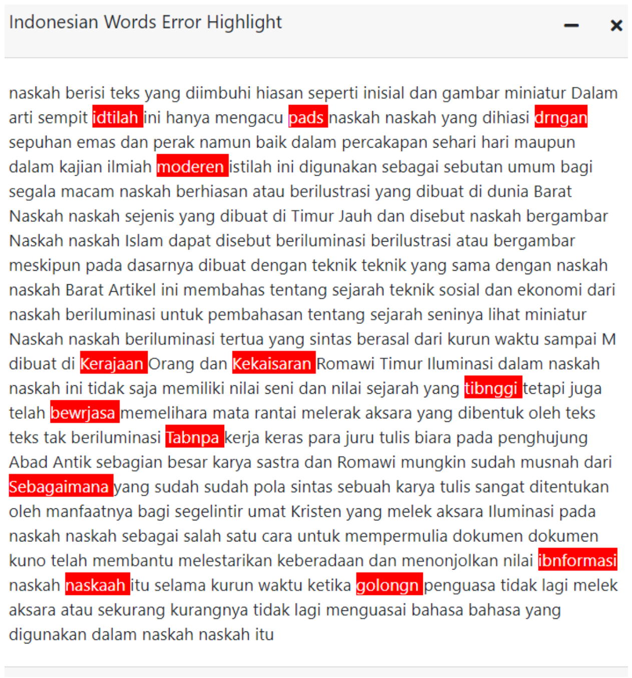
naskah berisi teks yang diimbuhi hiasan seperti inisial dan gambar miniatur Dalam arti sempit idtilah ini hanya mengacu pads naskah naskah yang dihiasi drngan sepuhan emas dan perak namun baik dalam percakapan sehari hari maupun dalam kajian ilmiah moderen istilah ini digunakan sebagai sebutan umum bagi segala

Nazief Adriani

naskah berisi teks yang diimbuhi hiasan seperti inisial dan gambar miniatur Dalam arti sempit idtilah ini hanya mengacu pads naskah naskah yang dihiasi drngan sepuhan emas dan perak namun baik dalam percakapan sehari hari maupun dalam kajian ilmiah moderen istilah ini digunakan sebagai sebutan umum bagi segala macam naskah berhiasan atau berilustrasi yang dibuat di dunia Barat Naskah naskah sejenis yang dibuat di Timur Jauh dan disebut naskah bergambar Naskah naskah Islam dapat disebut beriluminasi berilustrasi atau bergambar meskipun pada dasarnya dibuat dengan teknik teknik yang sama dengan naskah naskah Barat Artikel ini membahas tentang sejarah teknik sosial dan ekonomi dari naskah beriluminasi untuk pembahasan tentang sejarah seninya lihat miniatur Naskah naskah beriluminasi tertua yang sintas berasal dari kurun waktu sampai M dibuat di Kerajaan Orang dan Kekaisaran Romawi Timur Iluminasi dalam naskah naskah ini tidak saja memiliki nilai seni dan nilai sejarah yang tibnggi tetapi juga telah bewrjasa memelihara mata rantai melerak aksara yang dibentuk oleh teks teks tak beriluminasi Tabnpa kerja keras para juru tulis biara pada penghujung Abad Antik sebagian besar karya sastra dan Romawi mungkin sudah musnah dari Sebagaimana yang sudah sudah pola sintas sebuah karya tulis sangat ditentukan oleh manfaatnya bagi segelintir umat Kristen yang melek aksara Iluminasi pada naskah naskah sebagai salah satu cara untuk mempermulia dokumen dokumen kuno telah membantu melestarikan keberadaan dan menonjolkan nilai ibnformasi naskah naskaah itu selama kurun waktu ketika golongn penguasa tidak lagi melek aksara atau sekurang kurangnya tidak lagi menguasai bahasa bahasa yang digunakan dalam naskah naskah itu

ERROR : 13
NUMBER OF WORD : 250
PROCESSING TIME : Complete in 3.1592 seconds

Fig. 4. Interface of Indonesian Words Error Detection System.


Indonesian Words Error Highlight

naskah berisi teks yang diimbuhi hiasan seperti inisial dan gambar miniatur Dalam arti sempit idtilah ini hanya mengacu pads naskah naskah yang dihiasi drngan sepuhan emas dan perak namun baik dalam percakapan sehari hari maupun dalam kajian ilmiah moderen istilah ini digunakan sebagai sebutan umum bagi segala macam naskah berhiasan atau berilustrasi yang dibuat di dunia Barat Naskah naskah sejenis yang dibuat di Timur Jauh dan disebut naskah bergambar Naskah naskah Islam dapat disebut beriluminasi berilustrasi atau bergambar meskipun pada dasarnya dibuat dengan teknik teknik yang sama dengan naskah naskah Barat Artikel ini membahas tentang sejarah teknik sosial dan ekonomi dari naskah beriluminasi untuk pembahasan tentang sejarah seninya lihat miniatur Naskah naskah beriluminasi tertua yang sintas berasal dari kurun waktu sampai M dibuat di Kerajaan Orang dan Kekaisaran Romawi Timur Iluminasi dalam naskah naskah ini tidak saja memiliki nilai seni dan nilai sejarah yang tibnggi tetapi juga telah bewrjasa memelihara mata rantai melerak aksara yang dibentuk oleh teks teks tak beriluminasi Tabnpa kerja keras para juru tulis biara pada penghujung Abad Antik sebagian besar karya sastra dan Romawi mungkin sudah musnah dari Sebagaimana yang sudah sudah pola sintas sebuah karya tulis sangat ditentukan oleh manfaatnya bagi segelintir umat Kristen yang melek aksara Iluminasi pada naskah naskah sebagai salah satu cara untuk mempermulia dokumen dokumen kuno telah membantu melestarikan keberadaan dan menonjolkan nilai ibnformasi naskah naskaah itu selama kurun waktu ketika golongn penguasa tidak lagi melek aksara atau sekurang kurangnya tidak lagi menguasai bahasa bahasa yang digunakan dalam naskah naskah itu

Fig. 5. Indonesian Words Error Highlight.

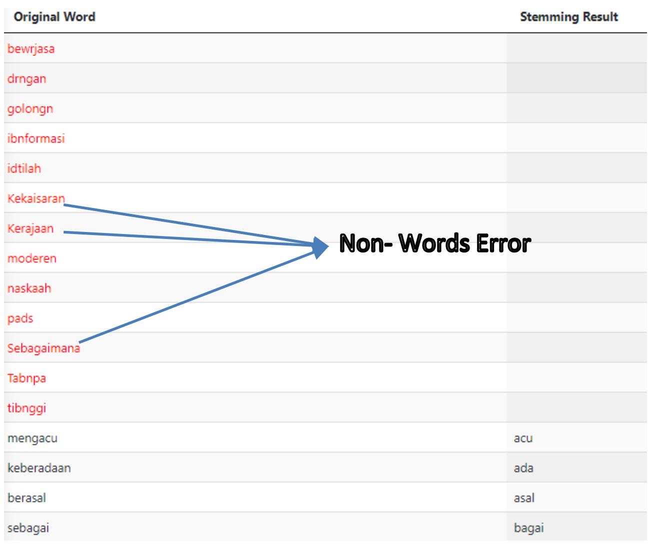

Original Word	Stemming Result
bewrjasa	
drngan	
golongn	
ibnformasi	
idtilah	
Kekaisaran	
Kerajaan	
moderen	
naskaah	
pads	
Sebagaimana	
Tabnpa	
tibnggi	
mengacu	acu
keberadaan	ada
berasal	asal
sebagai	bagai

Fig. 6. The Interface of Stemming Result Table.

B. Document Testing

The documents to be tested in the word error detection system contain sentences in Indonesian. The amount of documents tested is six documents that can be seen in the following Table I.

C. Test Result

The results of the word error detection system in Indonesian using the Nazief Adriani algorithm are quite good. In the six documents tested, 100% succeeded in detecting word errors in the document, but some words that were considered correct were also detected as errors. As seen in document text1, the detection ability is 98.75%, complete data on the test results are presented in Table II, and the graph can be seen in Fig. 7.

Based on all the experimental results in Table II, it can be concluded that the Nazief Adriani algorithm can analyze all the wrong words up to (100%). However, there are still deficiencies in analyzing the correct words. Based on the results shown in Table II, it can be concluded that the average accuracy of Nazief Adriani's algorithm in analyzing the correct words is 97.464. The results of the analysis of Table II are presented below in the graphical form.

TABLE. I. TEST DOCUMENT INFORMATION

Document Name	Number of Words	Description
text1.pdf	250 words	Randomly copy articles on the internet.
text2.pdf	500 words	Randomly copy articles on the internet. Including text1.pdf
text3.pdf	1000 words	Randomly copy articles on the internet. Including text1 and text2.pdf
cerpen1.pdf	1384 words	Copied from compass stories titled “Seragam” written by clippers
cerpen2.pdf	1592 words	Copied from compass stories, titled “Dua Wajah Ibu” written by clippers
cerpen3.pdf	1630 words	Copied from compass stories, titled “Tangan-Tangan Buntung" written by clippers

TABLE. II. WORD ERROR DETECTION SYSTEM

Document			**Number of Words**		**Number of Word Highlight**	
No	***Type***	***Words***	***True***	***False***	***True***	***False***
1	text1.pdf	250	240	10	3 (98,75%)	10 (100%)
2	text2.pdf	500	480	20	12 (97,5%)	20 (100%)
3	text3.pdf	1000	960	40	23 (97,6%)	40 (100%)
4	cerpen1.pdf	1400	1384	16	33 (97,6%)	16 (100%)
5	cerpen2.pdf	1620	1592	28	84 (94, 72%)	28 (100%)
6	cerpen3.pdf	1650	1630	20	68 (95,87%)	20 (100%)
Average					97,464 %	100 %

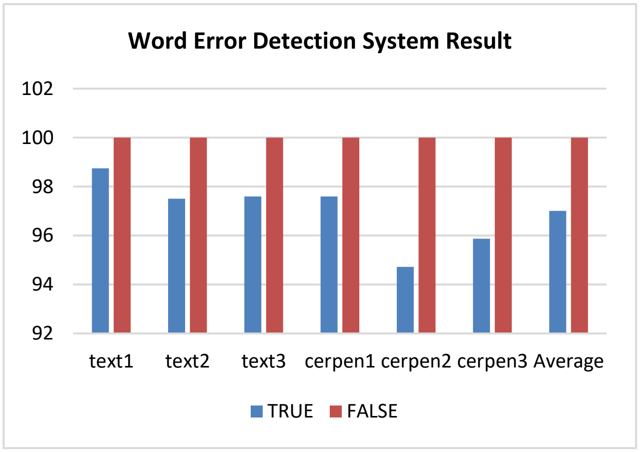


Fig. 7. Chart of Word Error Detection System Result.

Nazief Adriani's algorithm has more complex steps and is designed to minimize errors and lack of other stemming algorithms in the process of analyzing correct words. Based on the results of the study, it was found there were differences between the studies conducted by Marsel Widjaja and Seng Hansun (2015). That study implemented the porter stemmer modified algorithm in the Indonesian word error detection plugin application and was able to analyze all the wrong words up to (100%). However, there are still deficiencies in analyzing correct words with an average of 96.31%. Whereas in this study, the Indonesian word error detection system using Nazief Adriani algorithm can analyze all the wrong words up to (100%). However, there are still deficiencies in analyzing correct words with an average of 97,464%. Thus, it can be concluded that the Nazief Adriri algorithm has an average accuracy that is better than the modified porter stemmer algorithm.

The results of the processing speed on the word error detection system in Indonesian documents with the stemming process using the Nazief Adriani stemmer algorithm is quite good. In the six documents that were tested with three attempts, the average time required to process one word is smaller or equal to 0.030 seconds/word, the data on the complete test results are presented in Tables III, IV, and V.

Table III explains the details of the processing speed in the first experiment. In the text1.pdf document containing 250 words, it has a processing time of 6.8554 seconds with a processing speed of 0.0274 seconds/word. In a text2.pdf document containing 500 words, it has a processing time of 13.9129 seconds with a processing speed of 0.0278 seconds/word. In the text3.pdf document containing 1000 words has a processing time of 27.6677 seconds with a processing speed of 0.0277 seconds/word. In the cerpen1.pdf document containing 1400 words has a processing time of 40.4160 seconds with a processing speed of 0.0289 seconds/word. In the cerpen2.pdf document containing 1620 words has a processing time of 41.2138 seconds with a processing speed of 0.0254 seconds/word. In the cerpen3.pdf document containing 1650 words has a processing time of 47.3358 seconds with a processing speed of 0.0287 seconds/word.

Table IV explains the details of the processing speed in the second experiment. In the text1.pdf document containing 250 words, it has a processing time of 6.8958 seconds with a processing speed of 0.0276 seconds/word. In a text2.pdf document containing 500 words, it has a processing time of 13.7238 seconds with a processing speed of 0.0274 seconds/word. In the text3.pdf document containing 1000 words has a processing time of 27.9247 seconds with a processing speed of 0.0279 seconds/word. In the cerpen1.pdf document containing 1400 words has a processing time of 41.3115 seconds with a processing speed of 0.0295 seconds/word. In the cerpen2.pdf document containing 1620 words has a processing time of 41.4399 seconds with a processing speed of 0.0256 seconds/word. In the cerpen3.pdf document containing 1650 words has a processing time of 46.9563 seconds with a processing speed of 0.0285 seconds/word.

Table V explains the details of the processing speed in the third experiment. In the text1.pdf document containing 250 words, it has a processing time of 6.8409 seconds with a processing speed of 0.0274 seconds/word. In a text2.pdf document containing 500 words, it has a processing time of 13.8641 seconds with a processing speed of 0.0277 seconds/word. In the text3.pdf document containing 1000 words has a processing time of 27.7267 seconds with a processing speed of 0.0277 seconds/word. In the cerpen1.pdf document containing 1400 words has a processing time of 42.2806 seconds with a processing speed of 0.0302 seconds/word. In the cerpen2.pdf document containing 1620 words has a processing time of 41.2279 seconds with a processing speed of 0.0254 seconds/word. In the cerpen3.pdf document containing 1650 words has a processing time of 47.1392 seconds with a processing speed of 0.0286 seconds/word.

The average processing speed of all experiments can be seen in Table VI and graphs in Fig. 8.

TABLE. III. PROCESSING SPEED OF TRIAL 1

No	**Document**		**Trial 1**	
	Type	***Words***	***Time(s)***	***s/word***
1	text1.pdf	250	6.8554	0.0274
2	text2.pdf	500	13.9129	0.0278
3	text3.pdf	1000	27.6677	0.0277
4	cerpen1.pdf	1400	40.4160	0.0289
5	cerpen2.pdf	1620	41.2138	0.0254
6	cerpen3.pdf	1650	47.3581	0.0287

TABLE. IV. PROCESSING SPEED OF TRIAL 2

No	**Document**		**Trial 2**	
	Type	***Words***	***Time(s)***	***s/word***
1	text1.pdf	250	6.8958	0.0276
2	text2.pdf	500	13.7238	0.0274
3	text3.pdf	1000	27.9247	0.0279
4	cerpen1.pdf	1400	41.3115	0.0295
5	cerpen2.pdf	1620	41.4399	0.0256
6	cerpen3.pdf	1650	46.9563	0.0285

TABLE. V. PROCESSING SPEED OF TRIAL 3

No	**Document**		**Trial 3**	
	Type	***Words***	***Time(s)***	***s/word***
1	text1.pdf	250	6.8409	0.0274
2	text2.pdf	500	13.8641	0.0277
3	text3.pdf	1000	27.7267	0.0277
4	cerpen1.pdf	1400	42.2806	0.0302
5	cerpen2.pdf	1620	41.2279	0.0254
6	cerpen3.pdf	1650	47.1392	0.0286

TABLE. VI. THE AVERAGE OF THREE TRIALS

No	Doc		Avg	
	Type	***Words***	***Time(s)***	***s/word***
1	text1.pdf	250	6.8640	0.0275
2	text2.pdf	500	13.8336	0.0277
3	text3.pdf	1,000	27.7730	0.0278
4	cerpen1.pdf	1,400	41.3360	0.0295
5	cerpen2.pdf	1,620	41.2939	0.0255
6	cerpen3.pdf	1,650	47.1512	0.0286

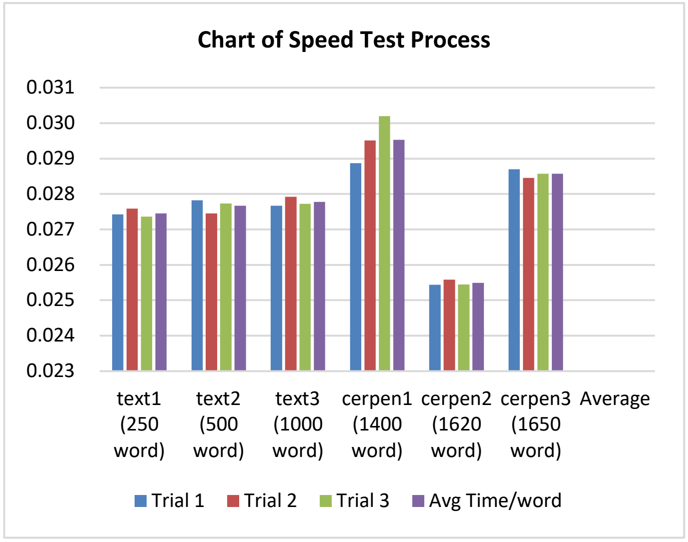


Fig. 8. Chart of Speed Test Process.

V. Conclusion

Indonesian words error detection system is good enough to detect word errors. From the trials conducted, the Nazief Adriani algorithm is 100% successful in detecting incorrect words in the six documents prepared. But some words that should be correct or non-word errors are detected as word errors, so they are highlighted by the system. Thus, Nazief Adriani's algorithm has been successfully implemented in the system and gives good results. In improving the accuracy of Nazief Adriani's algorithm, algorithm modifications can be made. Besides being modified, other algorithms can also be added, such as porter stemmer, confix stripping (CS), or enhanced confix stripping (ECS). The development of an Indonesian word error detection system can be improved by adding other features, such as automatic correction and correct word suggestions.

References

[1] S. Vinsensius B. Vega S., Bressan, "Indexing the Indonesian Web: Language Identification and Miscellaneous Issues." Poster Proceedings of the tenth International World Wide Web Conference, pp. 46–47, 2001.

[2] S. Sunardi, A. Yudhana, and I. A. Mukaromah, "Plagiarism Detection Implementation Using N-Gram and Jaccard Similarity Methods On Winnowing Algorithm (English)," Transmisi, vol. 20, no. 3, p. 105, 2018.

[3] M. I. K. Islam, M. T. Habib, M. S. Rahman, M. R. Rahman, and F. Ahmed, "A context-sensitive approach to find optimum language model for automatic Bangla spelling correction," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 11, pp. 184–191, 2018.

[4] M. Widjaja and S. Hansun, "Implementation of porter's modified stemming algorithm in an Indonesian word error detection plugin application," Int. J. Technol., vol. 6, no. 2, pp. 139–150, 2015.

[5] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, "Stemming Indonesian," Conf. Res. Pract. Inf. Technol. Ser., vol. 38, no. January, pp. 307–314, 2005.

[6] A. S. Rizki, A. Tjahyanto, and R. Trialih, "Comparison of stemming algorithms on Indonesian text processing," Telkomnika (Telecommunication Comput. Electron. Control., vol. 17, no. 1, pp. 95–102, 2019.

[7] T. Mardiana, T. B. Adji, and I. Hidayah, "Stemming influence on similarity detection of abstract written in Indonesia," Telkomnika (Telecommunication Comput. Electron. Control., vol. 14, no. 1, pp. 219–227, 2016.

[8] A. F. Hidayatullah, C. I. Ratnasari, and S. Wisnugroho, "Analysis of Stemming Influence on Indonesian Tweet Classification," Telkomnika (Telecommunication Comput. Electron. Control., vol. 14, no. 2, pp. 665–673, 2016.

[9] T. Winarti, D. Kerami, E. T. P. Lussiana, and S. A. Sudiro, "Improving stemming algorithm using morphological rules," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 7, no. 5, pp. 1758–1764, 2017.

[10] 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.

[11] H. L. Agnew, "Our Natural Language," Orig. Czech Natl. Renasc., no. 1982, pp. 51–92, 2017.

[12] N. Yusuf, M. A. B. M. Yunus, and N. B. Wahid, "A comparative analysis of web search query: Informational vs. navigational queries," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 9, no. 1, pp. 136–141, 2019.

[13] G. Mediamer, - Adiwijaya, and S. Al Faraby, "Development of Rule-Based Feature Extraction in Multi-label Text Classification," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 9, no. 4, pp. 1460–1465, 2019.

[14] S. M. Isa, R. Suwandi, and Y. P. Andrean, "Optimizing the hyperparameter of feature extraction and machine learning classification algorithms," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 3, pp. 69–76, 2019.

[15] W. C. F. Mariel, S. Mariyah, and S. Pramana, "Sentiment analysis: A comparison of deep learning neural network algorithm with SVM and naïve Bayes for Indonesian text," J. Phys. Conf. Ser., vol. 971, no. 1, pp. 0–8, 2018.

[16] P. M. Prihatini, I. K. G. D. Putra, I. A. D. Giriantari, and M. Sudarma, "Stemming Algorithm for Indonesian Digital News Text Processing," Int. J. Eng. Emerg. Technol., vol. 2, no. 2, pp. 1–7, 2017.

[17] A. Yudhana, Sunardi, and I. A. Mukaromah, "Implementation of winnowing algorithm with dictionary English-Indonesia technique to detect plagiarism," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 5, pp. 183–189, 2018.

[18] H. Alani and S. Saad, "Schema matching for large-scale data based on ontology clustering method," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 7, no. 5, pp. 1790–1797, 2017.

[19] M. Javed and S. Kamal, "Normalization of unstructured and informal text in sentiment analysis," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 10, pp. 78–85, 2018.

[20] A. Fadlil, "Application of Student Questionnaire Retrieval System Using Application of Information Retrieval for Student Opinion (English)," J. Teknol. Inf. dan Ilmu Komput., vol. 6, no. 1, pp. 33–40, 2018.

[21] A. Yudhana, Sunardi, and A. Djalil, "Implementation of Pattern Matching Algorithm for Portable Document Format," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 11, pp. 509–512, 2017.

[22] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," M.Sc. Thesis, Append. D, vol. pp, pp. 39–46, 2003.

[23] E. Da Costa, H. Tjandrasa, and S. Djanali, "Text mining for pest and disease identification on rice farming with interactive text messaging," Int. J. Electr. Comput. Eng., vol. 8, no. 3, pp. 1671–1683, 2018.

[24] H. Omar, M. Dahab, and M. Kamal, “Stemmer Impact on Quranic Mobile Information Retrieval Performance,” Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 12, pp. 135–139, 2016.

[25] A. Nwesri, “Effective Retrieval Techniques for Arabic Text,” 2008.

[26] A. Yudhana, A. D. Djayali, and Sunardi, “Plagiarism Detection System for Scientific Papers Using Pattern Matching Algorithms (English),” Jurti, pp. 178–187, 2017.

[27] S. Alnofaie, M. Dahab, and M. Kamal, “A Novel Information Retrieval Approach using Query Expansion and Spectral-based,” Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 9, pp. 364–373, 2016.

[28] S. N. Hidayatullah and Suyanto, “Developing an adaptive language model for Bahasa Indonesia,” Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 488–492, 2019.