International Journal of Technology (2015) 2: 139-150
ISSN 2086-9614 © IJTech 2015

IMPLEMENTATION OF PORTER'S MODIFIED STEMMING ALGORITHM IN AN INDONESIAN WORD ERROR DETECTION PLUGIN APPLICATION

Marsel Widjaja[1*], Seng Hansun[1]

[1]*Computer Science Department, Faculty of Information and Communication Technology, Universitas Multimedia Nusantara, Jl.Scientia Boulevard, Gading Serpong, Tangerang, Banten-15811 Indonesia*

(Received: August 2014 / Revised: April 2015 / Accepted: April 2015)

ABSTRACT

Stemming is a process of finding the root of a word with some omission stages of prefixes and suffixes. Stemming for each language varies depending on the morphology of the language. Stemming has widely been used as a complementary stage in many activities related to a word or phrase. With so many stemming utilizations, numerous algorithms have been developed to conduct the stemming process. In this study, the authors would like to develop, improve, and implement Porter's stemming algorithm in a word error detector plugin application. The test results have shown that the modified Porter's stemming algorithm produces more accurate results in the analysis than Porter's original stemming algorithm with an average difference in precision of about 3%.

Keywords: Indonesian; Morphology; Porter; Stemming

1. INTRODUCTION

Indonesian is the main language Indonesian people use to communicate both orally and in writing. The selection of raw words in pronunciation and in writing a sentence are important elements. In terms of Standard Indonesian Grammar (Dept. of Cultural and Education, Republic of Indonesia, 1988), the *Dictionary of Indonesian* is the main reference in the use of raw words in the Indonesian language. In addition to communication, the use of proper Indonesian is required when writing formal documents, journals, reports, and so on.

According to Buckland (1998), a good document written in Indonesian must have a high level of formality, employ appropriate vocabulary, and avoid grammatical and spelling errors. A document must be rechecked and revised repeatedly to make sure there are no typographical errors. This can be a very time-consuming task if more than one document needs to be inspected.

Pustakers (www.pustakasekolah.com) defined stemming as a process used to maximize word processing by changing basically every word being said. It is necessary to obtain a word that is not in accordance with the Indonesian dictionary. Porter's stemming algorithm is one of the algorithms that can be used to find the root (raw) word in the Indonesian language.

Through previous studies, Porter's stemming algorithm has been applied to search applications and has also been compared with other stemming algorithms, such as Nazief and Adriani's algorithm (Agusta, 2009). In the present study, Porter's stemming algorithm is applied to a plugin application for Microsoft Word that will quickly and thoroughly check the validity

[*] Corresponding author's email: marselwidjaja@yahoo.co.id, Tel. +62-21- 5422.0808, Fax. +62-21- 5422.0800

Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v6i2.456

the validity of each word in a document written in the Indonesian language in order to eliminate typographical errors. Thus, the document obtained with a high degree of formality to minimize the level of Indonesian language word errors.

The purpose of this study is to implement Porter's stemming algorithm as a Microsoft Word plugin application and improve it with the proposed method.

2. LITERATURE

2.1. Indonesian Morphological Structure

Morphology is the study of the form of the words in a particular language. Root words in Indonesian can be developed into other words due to the affix-related rules that exist in Standard Indonesian Grammar. There are various types of affixes in morphology, examples of which include the following:

1. Prefix: per-, me-, ter-, di-, ber-, and so on.
2. Infix: -el-, -em-, and -er-.
3. Suffix: -an, -kan, and -i.
4. A confix has a variety of functions, among which include the following:
   a) Remuneration functional form of the verb, including the following: me-, ber-, per-, -kan, -i, and ber-an.
   b) Remuneration functional of the noun, including the following: pe-, ke-, -an, ke-an, per-an, -man, -wan, and -wati.
   c) Remuneration functional form of the adjective, including the following: ter-, -i, -wi, -iah.
   d) Remuneration functional form of the words, including the following: ke- and se-.
   e) Remuneration functional form of function words, including the following: se- and se-nya.

2.2. Docx

A Docx file is generated from the Microsoft Word typing software, which was first released in 1983. Microsoft Word has been used by many people throughout the world and is thus one of the world's most popular typing software programs. The Word file format is now considered the standard digital document format (Roy, 2001).

2.3. Algorithm

Based on Utami and Sukrisno's (2005) definition, an algorithm is a method or logic of the sequence of work to solve problems systematically. An algorithm will generate the appropriate output from the desired input.

2.4. Stemming

Pustakers (www.pustakasekolah.com) defined stemming as a process of changing the words in the document to obtain a root word with certain rules. Stemming is used to maximize the information retrieval in a document. Implementation of the Indonesian stemming algorithm is different from the English stemming process because both languages have a different morphology. For example, if the English had to eliminate a prefix and a suffix, the Indonesian must also eliminate the confix. There are many other variations of augmentation that must be reckoned with when implementing the Indonesia stemming algorithm. The first stemmer for English is Lovin's stemmer (Lovins, 1968).

2.5. Purpose of a Stemming Algorithm

According to Moral et al. (2014), a stemming algorithm has three main objectives. The first is a grouping of words according to their topics. Many words from the same root derivation and derivation generated through additional affix (prefix, infix, and / or suffix). The second goal of a stemming algorithm is related to the process of finding information that has the same root, and thus the grouping term by the root word makes it easy to index the documents. The third goal is the incorporation of a variety of the same root to reduce the words that are taken into account in the process of collecting data, thereby reducing the space required to store the structures used by the information retrieval system.

2.6. Porter's Stemming Algorithm

According to Milutinovich (2006), Porter's stemming algorithm was first discovered in 1979 by Martin Porter in a computer lab. Porter's stemming algorithm is a process of removing the suffix morphology and inflection of a word in English as part of the normalization that occurs when creating the information retrieval system. Porter's algorithm, which was originally developed for English, was developed for Indonesian by WB Frakes in 1992. The stemming process using Porter's algorithm is illustrated in Figure 1.

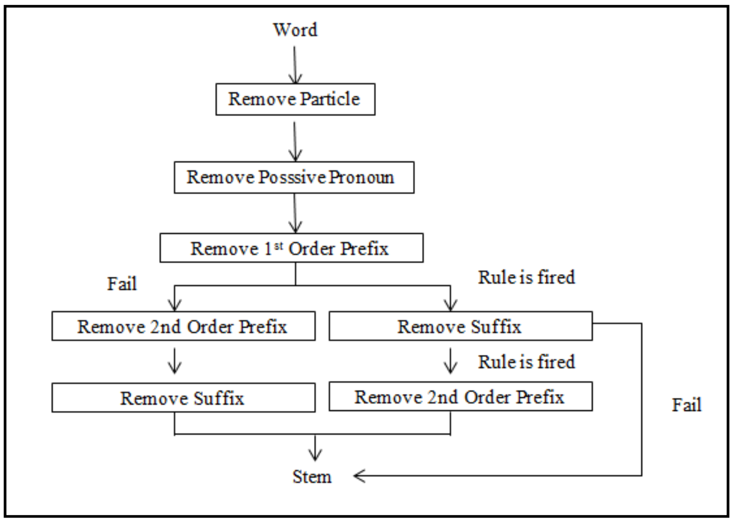


Figure 1 Porter's Algorithm (Agusta, 2009)

Agusta (2009) outlined the following measures undertaken in Porter's algorithm:
1. Remove the particle.
2. Remove the possessive pronoun.
3. Remove the first prefix. If there is no first prefix, then go to step 4a; otherwise go to step 4b.
4. a. Remove the second prefix and then proceed to step 5a.
   b. Remove the suffix; if there is no suffix then the word is assumed to be a root word. If the suffix is found, then go to step 5b.
5. a. Remove the suffix. The final remaining word is assumed to be the root word.
   b. Remove the second prefix. The final remaining word is assumed to be the root word.

According to Agusta (2009), there are five phases in the algorithm rules for Porter's algorithm for Indonesian, which are presented in Table 1 below.

Table 1 Rules for inflectional particle

Suffix	Replacement	Measure Condition	Additional Condition	Example
-kah	NULL	2	NULL	bukukah
-lah	NULL	2	NULL	pergilah
-pun	NULL	2	NULL	bukupun

In the second rule, there are examples of suffixes being changed to base words with possessive pronouns.

Table 2 Rules for inflectional possessive pronoun

Suffix	Replacement	Measure Condition	Additional Condition	Example
-ku	NULL	2	NULL	bukuku
-mu	NULL	2	NULL	bukumu
-nya	NULL	2	NULL	bukunya

In the third rule, there is an example of a prefix being changed to the word base with a second order derivational prefix.

Table 3 Rules for second order derivational prefix

Prefix	Replacement	Measure Condition	Additional Condition	Example
ber-	NULL	2	NULL	berlari -> lari
bel-	NULL	2	Ajar	belajar -> ajar
be-	NULL	2	k*er	bekerja -> kerja
per-	NULL	2	NULL	perjelas -> jelas
pel-	NULL	2	Ajar	pelajar -> ajar
pe-	NULL	2	NULL	pekerja -> kerja

In the fourth rule, there are examples of first order prefixes for basic words.

Table 4 Rules for first order derivational prefix

Prefix	Replacement	Measure Condition	Additional Condition	Example
meng-	NULL	2	NULL	mengukur -> ukur
meny-	S	2	V…*	menyapu -> sapu
men-	NULL	2	NULL	menduga -> duga
mem-	P	2	V…	memaksa-> paksa
mem-	NULL	2	NULL	membaca-> baca
me-	NULL	2	NULL	merusak -> rusak
peng-	NULL	2	NULL	pengukur-> ukur
peny-	S	2	V…	penyapu -> sapu
pen-	NULL	2	NULL	penduga -> duga
pem-	P	2	V…	pemaksa -> paksa
pem-	NULL	2	NULL	pembaca -> baca
di-	NULL	2	NULL	diukur -> ukur
ter-	NULL	2	NULL	tersapu -> sapu
ke-	NULL	2	NULL	kekasih -> kasih

Finally, there are example of adding suffixes to base words.

Table 5 Rules for derivational suffix

Suffix	Replacement	Measure Condition	Additional Condition	Example
-kan	NULL	2	Prefix €{ke, peng}	tarikkan -> tarik
-an	NULL	2	Prefix €{di, meng, ter}	makanan -> makan
-i	NULL	2	Prefix €{ber, ke, peng}	tandai – tanda

2.7. Porter Stemmer Errors

According to Karaa (2013), Porter's stemming algorithm also has its drawbacks, the most significant of which are overstemming and understemming errors. Overstemming occurs when words that are cut produce a basic word with a different meaning. In the case of understemming, the words are derived from the same root word; however, if it is stemmed, it does not produce the same stem word. It certainly reduces the efficiency and performance of Porter's stemming algorithm.

2.8. Affixes and Dilution

Widya (2013) described an affixed word as a word that underwent the basic process of adding affixes with the aim of ensuring a clearer meaning in the use of the word. Dilution is the removal of the first letter of said base when the process of adding affixes is completed. Some words have to undergo a process of rule-based augmentation dilution. The following are the types of additives used in Indonesian.

1. Prefix meng- and peng-.
   The prefix meng- can be added to the base word when the initial letters form the word vowels "k," "h," "g," and "kh."
   Consider the following examples:
   • Meng-ambil and peng-ambil.
   • Meng-uap and peng-uap.
   • Meng-harap and peng-harap.
   • Meng-gunting and peng-gunting.
   • Meng-khotbah and peng-khotbah.
   The initial letter "k" undergoes a dilution process. Consider the following examples:
   • Meng-kaji becomes mengaji.
   • Peng-kaji becomes pengaji.
2. Prefix me- and pe-.
   The prefix me- can be added on the basis of the initial letters of words when the form includes the letters "l," "m," "n," "ny," "ng," "r," "y," and "w."
   Consider the following examples:
   • Me-latih and pe-latih.
   • Me-makan and pe-makan.
   • Me-namai and pe-nama.
   • Me-nyatakan and pe-nyanyi.
   • Me-nganga and pe-ngidap.
   • Me-rapikan and pe-robek.
   • Me-yakinkan and pe-yakin.
   • Me-warnai and pe-warna.
3. Prefix men- and pen-.
   The prefix men- can be added on the basis of the initial letters of words when the form

includes the letters “d,” “j,” “sy,” and “t.”
Consider the following examples:

- Men-datangi and pen-datang.
- Men-jegal and pen-jegal.
- Men-syukuri and pen-syukur.
- Men-tanam becomes menanam and pen-tanam becomes penanam (having a dilution for the initial letter “t”).

4. Prefix mem- and pem-.
The prefix mem- can be added on the basis of the initial letters of words when the form includes the letters “b,” “p,” and “f.”
Example:
   - Mem-bantai and pem-bantai.
   - Mem-pukul becomes memukul and pem-pukul becomes pemukul (having a dilution for the initial letter “p”).
   - Mem-fokuskan and pem-fokus.

5. Prefix meny- and peny-.
The prefix meny- can be added on the basis of the initial letters of words when the form includes the letter is “s.”
Example:
   - Meny-sapu becomes menyapu (having a dilution for the initial letter “s”).

3. APPLICATION DESIGN

The design process in this study uses a flowchart design. The main flowchart used in the system is included below in Figure 2.

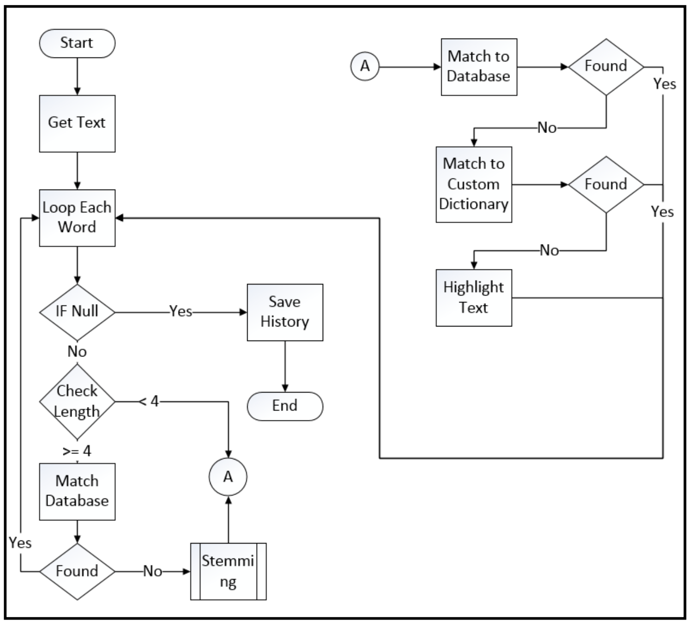


Figure 2 Process analysis with Porter's algorithm flowchart

Figure 2 illustrates the process flow of the text input obtained from the user, and then the input text is parsed per word in order to perform the stemming process using Porter's stemming algorithm. Then the word is compared to a database to confirm legitimacy; if it does not comply, then the word is highlighted. The history of the process results was stored in the database.

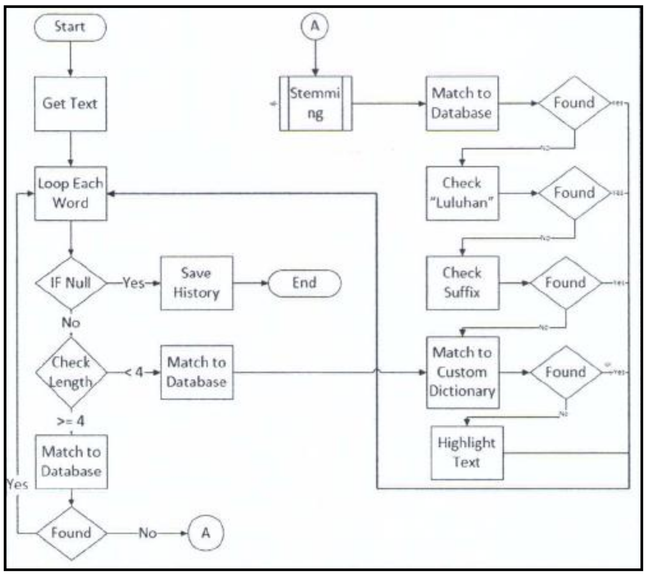


Figure 3 Process Analysis with Porter's modified algorithm flowchart

Figure 3 shows the process flow, which is similar to the previous process flow, but with some modifications. For example, several processes were added before the highlighted words to obtain more accurate results than the previous process flow.

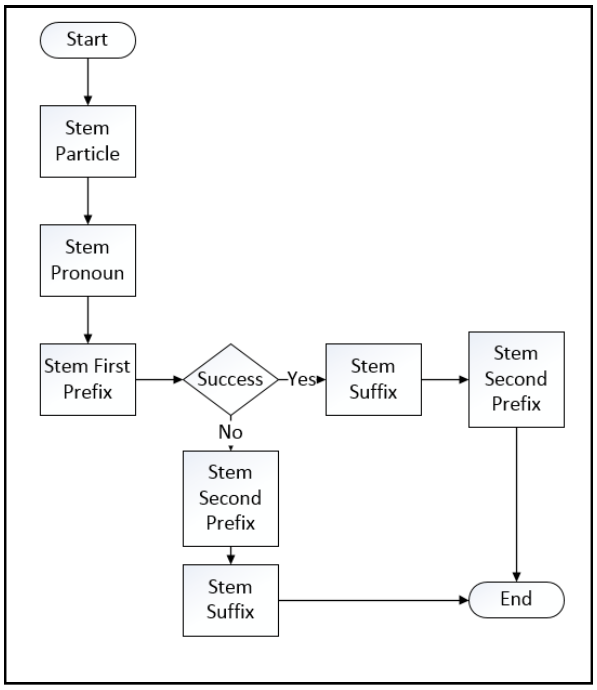


Figure 4 Stemming process with Porter's algorithm flowchart

Figure 4 illustrates that the stemming process begins with stemming the particle phase, followed by a pronoun stemming and first prefix stemming. If the first prefix in the word is found, then the process continues to the suffix stemming process and the last second prefix stemming; however, if it is not found in the first prefix, the prefix is the second stemming passed first, followed by suffix stemming.

According to the flowchart in Figure 4, the stemming process applies both to Porter’s original algorithm and Porter’s modified algorithm. The difference lies in the process of stemming the first prefix that has some modifications from its original process. The following modifications were made in the first table prefix.

Table 6 Rules for first order derivational prefix with modified prefix

Prefix	Replacement	Measure Condition	Additional Condition	Example
meng-	NULL	2	NULL	mengukur -> ukur
meng-	K	2	V…*	mengaji -> kaji
meny-	S	2	V…*	menyapu -> sapu
men-	NULL	2	NULL	menduga -> duga
men-	T	2	NULL	menari -> tari
mem-	P	2	V…	memaksa-> paksa
mem-	NULL	2	NULL	membaca-> baca
me-	NULL	2	NULL	merusak -> rusak
peng-	NULL	2	NULL	pengukur-> ukur
peny-	K	2	V…*	pengali -> kali
pen-	S	2	V…	penyapu -> sapu
pen-	NULL	2	NULL	penduga -> duga
pem-	T	2	NULL	penari -> tari
pem-	P	2	V…	pemaksa -> paksa
pe-	NULL	2	NULL	pembaca -> baca
di-	NULL	2	NULL	perusak -> rusak
ter-	NULL	2	NULL	diukur -> ukur
ke-	NULL	2	NULL	tersapu -> sapu
se-	NULL	2	NULL	kekasih -> kasih
ku-	NULL	2	NULL	sewaktu -> waktu

As indicated in Table 6, there are some additional rules for the first prefix, including modifications and additions to the existing theories based on affixes and dilution that exist on a theoretical basis.

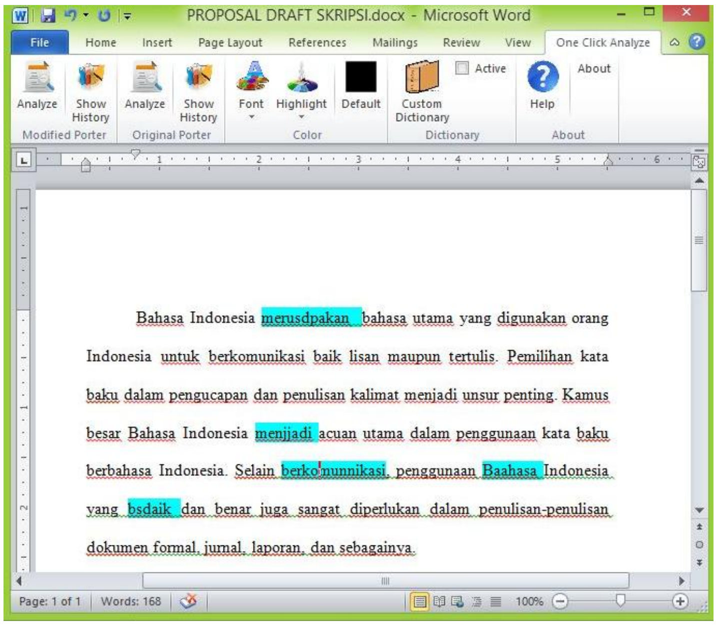
Furthermore, the measures for Porter’s stemming algorithm have been modified as follows:

1. Remove the particle.
2. Remove the possessive pronoun.
3. Remove the first prefix. If there is no first prefix, go to step 4a; otherwise go to step 4b.
4. a. Remove the second prefix and then proceed to step 5a.
   b. Remove the suffix; if the suffix is not found then the word is assumed to be a root word. If it is found, then go to step 5b.
5. a. Remove the suffix. If the root word is found in the dictionary, then the algorithm stops. If it is not found, skip to step 6.
   b. Remove the second prefix. If the root word is found in the dictionary, then the algorithm stops. If it is not found, skip to step 6.

6. First, check the prefixes that are diluted. If the root word is found in the dictionary, then the algorithm stops.
7. Return the suffix. If the root word is found in the dictionary, then the algorithm stops.
8. Check the custom dictionary; if it is found, then the algorithm stops.

4. IMPLEMENTATION AND TESTING RESULT

Figure 5 below presents the plugin interface in the Microsoft Word application. The new ribbon can be found in the Microsoft Word menu tab with the name "One Click Analyzer." There are 10 buttons and one checklist on this ribbon.



Bahasa Indonesia merusdpakan bahasa utama yang digunakan orang Indonesia untuk berkomunikasi baik lisan maupun tertulis. Pemilihan kata baku dalam pengucapan dan penulisan kalimat menjadi unsur penting. Kamus besar Bahasa Indonesia menjjadi acuan utama dalam penggunaan kata baku berbahasa Indonesia. Selain berkomunnikasi, penggunaan Baahasa Indonesia yang bsdaik dan benar juga sangat diperlukan dalam penulisan-penulisan dokumen formal, jurnal, laporan, dan sebagainya.

Figure 5 Whole interface of Microsoft Word

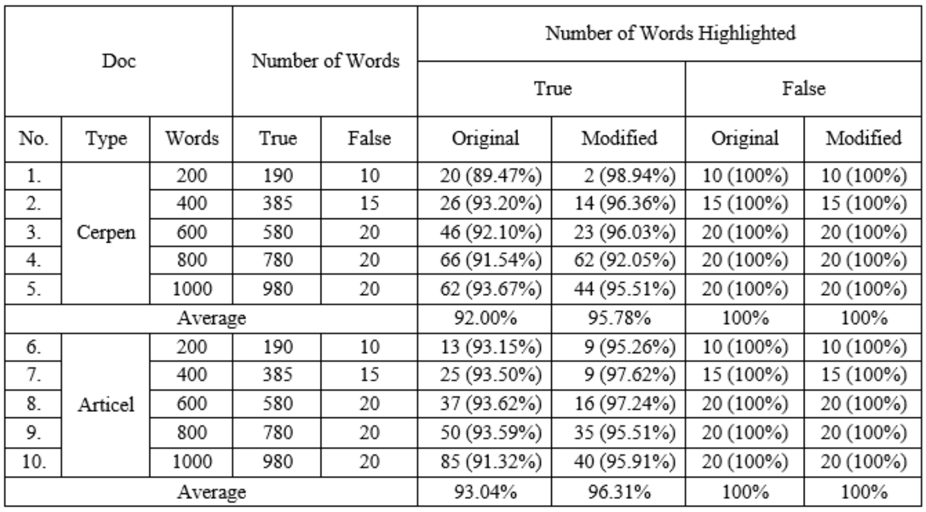
According to the results of the analysis illustrated in Figure 5, there are some incorrect words there are some incorrect words which the plugin managed to detect (as illustrated with the blue highlights).

In the "One Click Analyzer" menu tab there are two analyze buttons. The first analyze button can be used to analyze the text with Porter's modified algorithm, while the second can be used to analyze the text with Porter's original algorithm.

The test was conducted to see the results of the analysis of Porter's stemming algorithm and Porter's modified stemming algorithm. The test was done with a few words that were intentionally typed incorrectly to test the accuracy of the analysis of both versions of Porter's algorithm. In addition, the test was also performed to see the accuracy of the analysis of the correct words. Table 7 presents the results of the analysis with some sample documents in tabular form.

Table 7 shows some of the experimental results from the samples of the short stories ("cerpen") and articles. The first experiment was conducted on short stories, which included five different sample documents that each has a varying number of words. First, there is a short story with 200 words in which there are 10 incorrect words. The experiments were performed using Porter's original algorithm; 10 incorrect words were successfully analyzed, while 20 correct words are regarded as incorrect words. The second experiment was performed using Porter's modified algorithm.

Table 7 Analysis of the test results



Doc			Number of Words		Number of Words Highlighted			
					True		False	
No.	Type	Words	True	False	Original	Modified	Original	Modified
1.	Cerpen	200	190	10	20 (89.47%)	2 (98.94%)	10 (100%)	10 (100%)
2.		400	385	15	26 (93.20%)	14 (96.36%)	15 (100%)	15 (100%)
3.		600	580	20	46 (92.10%)	23 (96.03%)	20 (100%)	20 (100%)
4.		800	780	20	66 (91.54%)	62 (92.05%)	20 (100%)	20 (100%)
5.		1000	980	20	62 (93.67%)	44 (95.51%)	20 (100%)	20 (100%)
Average					92.00%	95.78%	100%	100%
6.	Articel	200	190	10	13 (93.15%)	9 (95.26%)	10 (100%)	10 (100%)
7.		400	385	15	25 (93.50%)	9 (97.62%)	15 (100%)	15 (100%)
8.		600	580	20	37 (93.62%)	16 (97.24%)	20 (100%)	20 (100%)
9.		800	780	20	50 (93.59%)	35 (95.51%)	20 (100%)	20 (100%)
10.		1000	980	20	85 (91.32%)	40 (95.91%)	20 (100%)	20 (100%)
Average					93.04%	96.31%	100%	100%

As a result, 10 incorrect words were successfully analyzed, whereas only two correct words were considered incorrect. The second experiment was conducted on a 400-word short story with 15 incorrect words. Both algorithms successfully analyzed 15 incorrect words; however, Porter's original algorithm regarded 26 correct words as incorrect words, while Porter's modified algorithm regarded 14 correct words as incorrect words. The third experiment was conducted on a 600-word short story with 20 incorrect words. Both algorithms successfully analyzed all 20 incorrect words; however, Porter's original algorithm regarded 46 correct words as incorrect words, while Porter's modified algorithm regarded 23 correct words as incorrect words. The fourth experiment used an 800-word short story with 20 incorrect words. Both algorithms successfully analyzed all 20 incorrect words; however, Porter's original algorithm regarded 66 correct words as incorrect words, while Porter's modified algorithm regarded 62 correct words as incorrect words. The fifth experiment was conducted on a 1,000-word document with 20 incorrect words. Both algorithms successfully analyzed all 20 incorrect words; however, Porter's original algorithm regarded 62 correct words as incorrect words, while Porter's modified algorithm regarded 44 correct words as incorrect words.

The next stage of the experiment involved using different kinds of articles, including five different types of documents each with a different number of words. The first experiment used a 200-word article with 10 incorrect words. Both algorithms successfully analyzed 10 incorrect words; however, Porter's original algorithm regarded 13 correct words as incorrect words, while Porter's modified algorithm regarded 9 correct words as incorrect words. The second experiment used a 400-word article with 15 correct words. Both algorithms successfully analyzed 15 incorrect words; however, Porter's original algorithm regarded 25 correct words as incorrect words, while Porter's modified algorithm regarded 9 correct words as incorrect words. The third experiment used 600-word article with 20 incorrect words. Both algorithms successfully analyzed 20 incorrect words; however, Porter's original algorithm regarded 37 correct words as incorrect words, while Porter's modified algorithm regarded 16 correct words as incorrect words. The fourth experiment used an 800-word article with 20 incorrect words. Both algorithms successfully analyzed 20 incorrect words; however, Porter's original algorithm regarded 50 correct words as incorrect words, while Porter's modified algorithm regarded 35 correct words as incorrect words. The fifth experiment used a 1,000-word article with 20 incorrect words. Both algorithms successfully analyzed 20 incorrect words; however, Porter's original algorithm regarded 85 correct words as incorrect words, while Porter's modified algorithm regarded 40 correct words as incorrect words.

Based on all of the experimental results it can be concluded that Porter's original algorithm and Porter's modified algorithm are capable of perfectly analyzing all of the incorrect words (100%); however, there are still shortcomings in analyzing the correct words. As the results in Table 7 suggest, it can be concluded that the use of Porter's modified algorithm produced better results in analyzing the correct words. The accuracy of the modified algorithm in analyzing the correct words was 96.31%, while the accuracy of Porter's original algorithm was 93.04%. Porter's modified algorithm has more complex algorithms and more complete prefix, postfix, and suffix rules tables. Porter's modified algorithm is designed to minimize the errors and the shortcomings of Porter's original algorithm in the process of analyzing the correct words. The results of the analysis of Table 7 are presented below in the form of graphs.

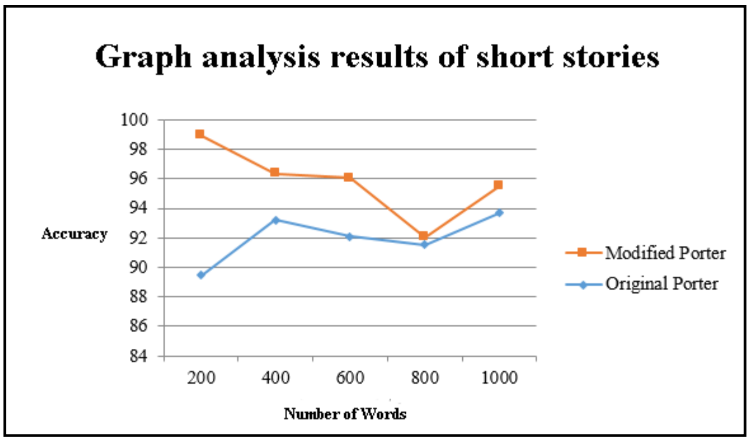


Figure 6. Analysis result of short story type for correct words

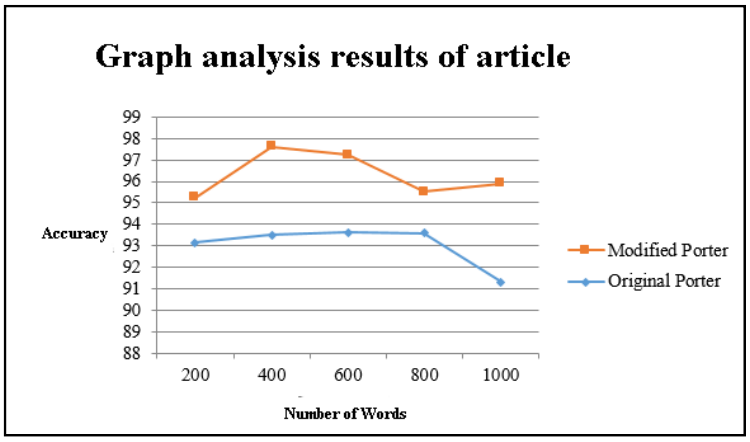


Figure 7 Analysis result of article type for correct words

5. CONCLUSION

This word error detection plugin application is good enough to analyze the existing posts in Microsoft Word. In addition, Porter's original algorithm and Porter's modified algorithm have been successfully implemented and produced good results. From the experimental results it can be concluded that the accuracy of Porter's modified algorithm is higher than Porter's original algorithm. Through the research conducted, Porter's algorithm can be developed further so that even more accurate results can be obtained. Furthermore, Porter's modified algorithm can be used together with other stemming algorithms in order to fill the gap, such as Nazief and Adriani's algorithm. While Nazief and Adriani's algorithm can provide better results for the stemming process, this approach will consume more time than Porter's original stemming algorithm. In addition, the development of this application can be improve by adding some features, such as the detection of words in English, auto correct, correct word suggestion, and auto italic for foreign language words, among others.

6. REFERENCES

Agusta, L., 2009. Comparison of Porter Stemming Algorithm and Nazief & Adriani's Algorithm for Stemming Indonesian Text Documents. National *Conference on Systems and Informatics*. KNS&I09–036

Buckland, M., 1998. *What is a Digital Document*? Available online at http://people.ischool.berkeley.edu/~buckland/digdoc.html, Accessed on 12 March 2014.

Dept. of Cultural and Education, Republic of Indonesia, 1988. Tata Bahasa Baku Bahasa Indonesia. Balai Pustaka [in Bahasa]

Karaa, Wahiba Ben Abdessalem, 2013. *A New Stemmer to Improve Information Retrieval*. *International Journal of Network Security & Its Applications* (IJNSA). Volume 5(4). Pp. 143-154 Available online at http://airccse.org/journal/nsa/5413nsa11.pdf,

Lovins, J.B., 1968. Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, Volume 11(1/2), pp. 22–31

Milutinovich, J., 2006. *The Porter Stemming Algorithm*. Available onlineat http://tartarus.org/~martin/PorterStemmer/index.html, Accessed on 27 February 2014

Moral, C., de Antonio, A., Imbert, R., Ramírez, J., 2014. A Survey of Stemming Algorithms in Information Retrieval, *Information Research*, Volume 19(1), p. 605. Available online at http://InformationR.net/ir/19-1/paper605.html,

Pustaker, *Algorithm stemming*. Available online at http://www.pustakasekolah.com/algoritma-stemming.html. Accessed on 27 February 2014

Roy, A.A., 2001. *History of the Personal Computer: The People and the Technology*. Allan Publishing.

Utami, E., Sukrisno, 2005. *10 Langkah Belajar Logika dan Algoritma*. Menggunakan Bahasa C dan C++ di gnu/Linux. Yogyakarta: C.V. Andi OFFSET [in Bahasa]

Widya, E., 2013. *Kata Berimbuhan*. Available online at http://basindosaka.blogspot.com/2011/12/kebahasaan.html, Accessed on 18 June 2014 [in Bahasa]