Stemming Algorithm for Indonesian Digital News Text Processing

PM Prihatini[1*], IKG Darma Putra[2], IAD Giriantari[3], and M Sudarma[4]
[1]Study Program of Doctoral Engineering Science, Faculty of Engineering, Udayana University, and Department of Electrical Engineering, Politeknik Negeri Bali
[2]Department of Information Technology, Faculty of Engineering, Udayana University
[3,4]Department of Electrical Engineering, Faculty of Engineering, Udayana University
*Email: manikprihatini@pnb.ac.id

***Abstract*— Stemming is the process of finding the basic word of a word in the text. The stemming algorithm built by Nazief-Adriani is the best stemming algorithm for Indonesian, and has been refined by Asian Jelita. However, references related to the Nazief-Adriani stemming algorithm are still difficult to find given that the algorithm is an internal publication. Therefore, in this study, will be built stemming algorithm for Indonesian news digital text based on the stemming algorithm Nazief-Adriani and Jelita Asian. The evaluation in this study was done before and after the addition of rules and more complete basic word dictionary. Both evaluations were performed by calculating Precision, Recall and F-Measure values between automatic and manual stemming results. Preliminary tests of the stemming algorithm Nazief-Adriani and Jelita Asian found some new basic words, abbreviations, entities and foreign terms that appear common in the news text but have not stored in the basic word dictionary. Furthermore, there are some unrecognized affixed words in defined rules. The addition of basic words, abbreviations, entities and foreign terms to the basic word dictionary, along with the addition of rules can improve the performance of the stemming algorithm built on this study. Thus, the completeness basic word dictionary and the accuracy of rules play a very important role in the success of an automatic stemming algorithm.**

Keywords—stemming; algorithm; Indonesian; dictionary; rules

I. INTRODUCTION

Stemming is a process in text processing that aims to find the basic word of the original word that appears in the text. Stemming has an important role because the stemming results are used to extract the existing features in the text. The word that appears in a text has various forms. There is a basic word, there is also a word containing affixes. Various forms of this word would get different treatment. The basic word and the word containing affixes are considered different entities in a text, so at the extraction of features, these two forms will also be considered as distinct features. Different features have different values. This is certainly very influential on the results of feature extraction. Therefore, the word containing affixes needs to be stemmed first to determine the basic word, so they will be have the same form. These words will be the same feature and reinforce its value in a text.

Stemming is strongly influenced by the language type of text. Stemming that used for one type of language may not be used for other languages. Therefore, in this study, stemming is directed to Indonesian texts. In Indonesian text, the word is formed from morphological rules involving inflection and derivational structures [1]. Inflection is the simplest structure and is expressed in the suffix of the particles (*-lah*, *-kah*, *-pun*, *-tah*), for example, the word "*pergilah* (go)" and possessive pronouns (*-ku*, *-mu*, *-nya*), for example, the word "*bukumu* (your book)". These two forms can appear together in a word, for example, the word "*anakmulah* (your child)". The derivational structure appears in the form of a combination of prefix, basic word, and suffix. In this combination, the basic word can begin with the prefix (*di-*, *ke-*, *se-*, *be-*, *te-*, *me-*, *pe-*), for example, the word "*pembeli* (buyer)", or end with the suffix (*-i*, *-an*, *-kan*), for example, the word "*tangisan* (crying)", or combine by prefix and suffix, for example, the word "*pembelian* (purchases)". In this combination, there are several disallowed prefix-suffix such as *be-i*, *di-an*, *ke-i*, *ke-kan*, *me-an*, *se-i*, *se-kan*, and *te-an* [2]. In a derivational structure, a word may change when given a prefix, for example, the word "*tari* (dance)" added by the prefix of "*me-*", then the word becomes "*menari* (dancing)". In addition, a prefix or combination of prefix-suffix can also be added to already prefix or combination of prefix-suffix, for example, the prefix of "*me-*" added to the combination of prefix-suffix "*perjuangkan* (struggle)", and then the word becomes "*memperjuangkan* (struggling)". Even, the basic word can also be inserted with infix, for example, the word "*peran* (role)" that is inserted with infix "*-em-*", and then the word becomes "*pemeran* (cast)". Therefore, automation of stemming process for Indonesian text with computerized system requires a proper algorithm so that the result of automatic stemming is able to approach the result of stemming done manually.

There are many studies of stemming for Indonesian texts have been conducted. Nazief and Adriani has built a stemming algorithm based on morphological rules involving allowed and disallowed affixes such as prefix, suffix, infix, and combination of prefix-suffix [2]. The stemming process of this algorithm begins with the removal of the inflexion suffix, followed by the removal of the suffix derivation, then the removal of the derivation prefix and the disallowed prefix-suffix checking.

(p-issn: 2579-5988, e-issn: 2579-597X)

The removal of the prefix is done a maximum of two times. This algorithm is also equipped with a recoding process to restore the initials of letters removed from the word due to the prefix attached to it. Vega has built stemming algorithms without using basic word dictionary, but uses only morphological rules by segmenting words into smaller components according to defined rules [3]. The accuracy generated by this algorithm is not as good as other stemming algorithms because it requires correct and complete morphological rules. Arifin and Setiono has built a simpler stemming algorithm but still use the basic word dictionary, removal prefix-suffix of the word and recoding process [4]. The stemming process of this algorithm begins with the deletion of the entire prefix and is followed by the removal of all the suffixes. Deletion is done a maximum of two times for the prefix and three times for the suffix. If the basic word is not found after the removal process, the prefix-suffix is returned to the original word to be matched with the entire combination of prefix-suffix thereby minimizing the error rate. Tala has built a stemming algorithm for Indonesian text based on the stemming algorithm Porter-like stemmer [1]. The stemming process begins with the deletion of particles, followed by the removal of the possessive pronouns, then the removal of the first order prefix. If it success, then do the suffix removal and the second order prefix. Instead, the second order prefix and suffix are removed. Asian Jelita has built a stemming algorithm based on the algorithm built by Nazief-Adriani and adding some features that did not exist in the previous algorithm [5]. Additional features include adding rules for plurals, changing the rules for some type of prefix, and adding some rules for combination of prefix-suffix. This algorithm produces a correct value of 95.3% compared to the original algorithm with a correct value of 92.1%. Prihatini has built the stemming algorithm for Indonesian news digital based on the Tala's stemming algorithm and adding word checking process in the basic word dictionary at the beginning of the stemming process and checking disallowed prefix-suffix at the end of the stemming process [6]. The stemming results of the study showed a precision value of 97% and recall of 64%. The difference in value between precision and recall indicates that the stemming algorithm built on Porter-like stemmer algorithm for Indonesian news digital still cannot give the best result.

From the study, it can be seen that the stemming algorithm built by Nazief-Adriani is the best stemming algorithm to date, and has been refined by Asian Jelita. However, references related to the Nazief-Adriani stemming algorithm are still difficult to find given that the algorithm is an internal publication. Whereas a good stemming algorithm is needed in several areas of study such as text pre-processing [7], information retrieval [8], text summarization [9], text clustering [10], text detection [11], and text classification [12]. Therefore, in this study, will be built stemming algorithm for Indonesian news digital text based on the stemming algorithm Nazief-Adriani and Jelita Asian. The discussion in this study is divided as follows. Section II discusses about research method including the dataset, text tokenization, case folding, filtering, stemming and evaluation. Section III discusses about the results of the evaluation and its analysis using Precision, Recall and F-Measure metrics. Section IV discusses about the conclusions obtained in the study.

II. Research Method

A. Corpus

This study is built for digital news text in Indonesia by taking data from digital media detikcom. The dataset used in this study is 125 news files from channel detikFinance, detikNet, detikSport, detikNews and detikHot [6]. The results of this study were compared with previous study results to show the best level of performance between the two algorithms for the same corpus.

B. Text Processing

The digital news text collected on the corpus cannot be directly stem because it still consists of a collection of paragraphs. Before the stemming process is beginning, this text must be processed first. In this study, text processing consists of text tokenization, case folding and filtering. The flowchart is shown in Fig.1.

The tokenization of the text in this study consists of three stages: splitting text into paragraphs, splitting paragraphs into sentences and splitting sentences into words. An example of the result of splitting the text into a paragraph is shown in Table 1. Examples of the result of splitting the paragraph into sentences are shown in Table 2. Examples of the result of splitting the sentences into words are shown in Table 3.

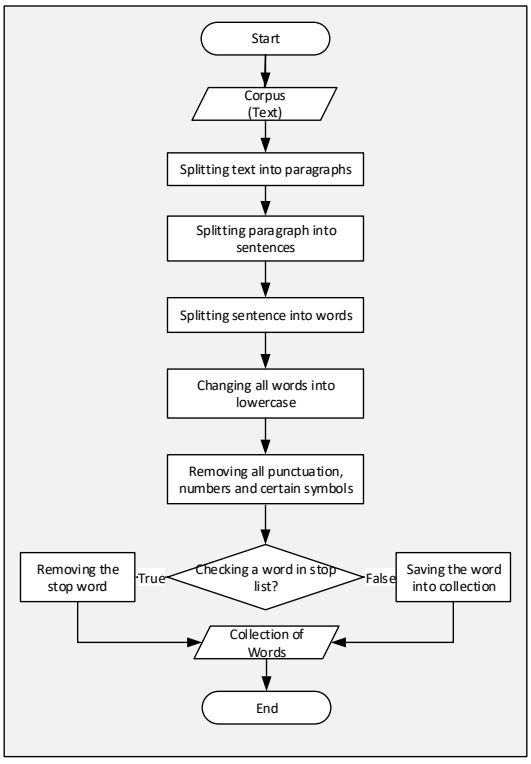

Fig. 1. Flowchart of Text Processing

In this study, case folding is used to convert all words into lowercase, followed by the removal of punctuation, numbers and certain symbols. Examples of case-folding outcomes are shown in Table 4.

Filtering in this study aims to eliminate words that do not affect the contents of a text. This process requires a list of words called a stop list. This study using stop list consisting of 902 words [1]. Examples of filtering outcomes are shown in Table 5.

TABLE I. SPLITTING THE TEXT INTO A PARAGRAPH

No.	Paragraph
1	Kementerian Perindustrian (Kemenperin) mencatat kebutuhan baja nasional mencapai 900.000 ton per tahun. Sekitar 40% dari total kebutuhan dipasok dari baja impor.
2	Melihat tingginya ketergantungan impor, Kemenpun pun mengeluarkan beberapa langkah, termasuk mendorong produksi baja dalam negeri.

TABLE II. SPLITTING PARAGRAPHS INTO SENTENCES

No.	Sentences
1	Kementerian Perindustrian (Kemenperin) mencatat kebutuhan baja nasional mencapai 900.000 ton per tahun.
2	Sekitar 40% dari total kebutuhan dipasok dari baja impor.
3	Melihat tingginya ketergantungan impor, Kemenpun pun mengeluarkan beberapa langkah, termasuk mendorong produksi baja dalam negeri.

TABLE III. SPLITTING THE SENTENCES INTO WORDS

No.	Words	No.	Words	No.	Words
1	Kementerian	5	kebutuhan	9	900
2	Perindustrian	6	baja	10	ton
3	(Kemenperin)	7	nasional	11	per
4	mencatat	8	mencapai	12	tahun.

TABLE IV. CASE-FOLDING OUTCOMES

No.	Words	No.	Words	No.	Words
1	kementerian	5	kebutuhan	10	ton
2	perindustrian	6	baja	11	per
3	kemenperin	7	nasional	12	tahun.
4	mencatat	8	mencapai		

TABLE V. FILTERING OUTCOMES

No.	Words	No.	Words	No.	Words
1	kementerian	5	kebutuhan	10	ton
2	perindustrian	6	baja		
3	kemenperin	7	nasional		
4	mencatat	8	mencapai		

C. Stemming

The stemming algorithm in this study is based on the Nazief-Adriani stemming algorithm refined by Asian Jelita. The flow chart of the Nazief-Adriani stemming algorithm is shown in Fig.2. The flowchart of the stemming algorithm Porter-like stemmer for Indonesian news digital text is shown in Fig.3.

The Nazief-Adriani stemming algorithm can be explained as follows.

1. The algorithm checks an original word into the basic word dictionary. If successful, then algorithm stops and the word expressed as a basic word. If it fails, the algorithm goes to the next step.
2. The algorithm removes the inflection suffix ("-*lah*"," –*kah*", "-*ku*", "-*mu*"," –*nya*"). If it success and the inflection suffix is a particle ("-*lah*" or "-*kah*"), the algorithm removes the inflection possessive pronouns ("-*ku*", "-*mu*", "-*nya*").
3. The algorithm removes the derivation suffix ("-*i*", "-*an*", "-*kan*"). If it success, the algorithm goes to the step 4. If step 4 fails:
    a. If derivation suffix is "-*an*" and the last character of the word is "-*k*", then the algorithm removes the character of "-*k*". Then, goes to step 4. If it fails, goes to step 3b.
    b. The algorithm restores the removed suffix (("-*i*", "-*an*", "-*kan*") to the original word.
4. The algorithm removes the derivation prefix, consist of several steps:
    a. If the removal of suffix is success in step 3, algorithm checks the disallowed prefix-suffix. If algorithm found it, the algorithm returns.
    b. If the current prefix similar with the previous prefix, the algorithm returns.
    c. If the removal of derivation prefix has done for three times, the algorithm returns.
    d. The algorithm checks the type of derivation prefix and removes the prefix.
    e. If the basic word is found, the algorithm returns. Instead, step 4 repeats again to remove the second prefix.
    f. The algorithm performs the recoding process, depends on the prefix type.
5. If all step is fail, the algorithm stops and returns to the original word.

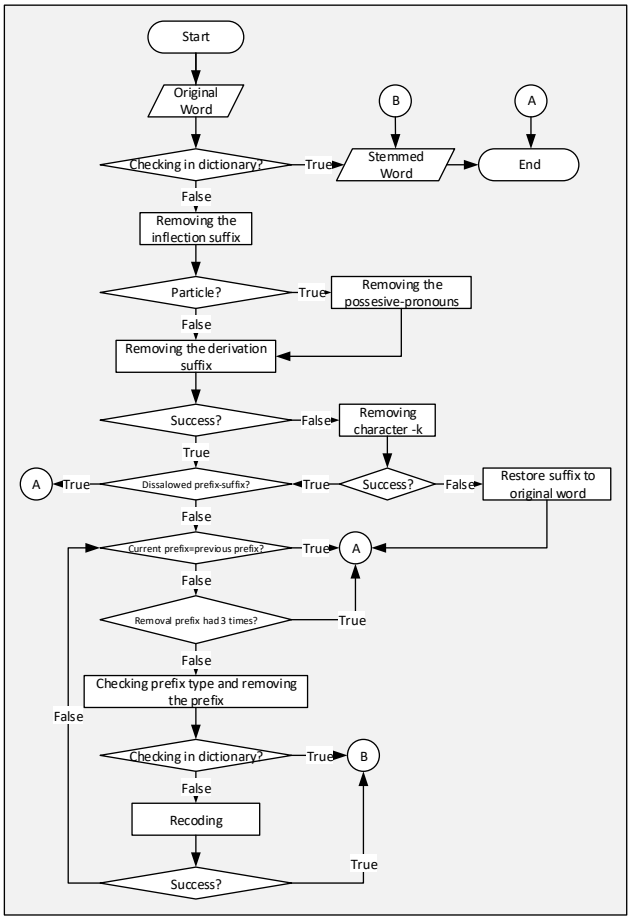

Fig. 2. Flowchart of Nazief-Adriani Stemming Algorithm

The improvements made by Asian Jelita, then referred as stemming model I, can be explained as follows.

1. Uses more complete dictionary.
2. Adding a rule for handling the plurals word, for example, the word "*buku-buku* (books)".
3. Adding the inflection particle suffix "*pun*", for example, the word "*siapapun* (who)".
4. Modifying the prefix type of "*ter-*", "*pe-*", "*mem-*", and "*meng-*".
5. Adding a new rules for combination prefix-suffix "*ber-lah*" for the word "*bersekolah* (be at school)", "*ber-an*" for the word "*berbadan* (having the body of)", "*men-i*" for the word "*menilai* (to mark)", "*di-i*" for the word "*dimulai* (to be started)", "*pe-i*" for the word "*petani* (farmer)", and "*ter-i*" for the word "*terkendali* (to be controlled)".

Stemming Algorithm Porter-like stemmer, and then referred as stemming model II, can be explained as follows.

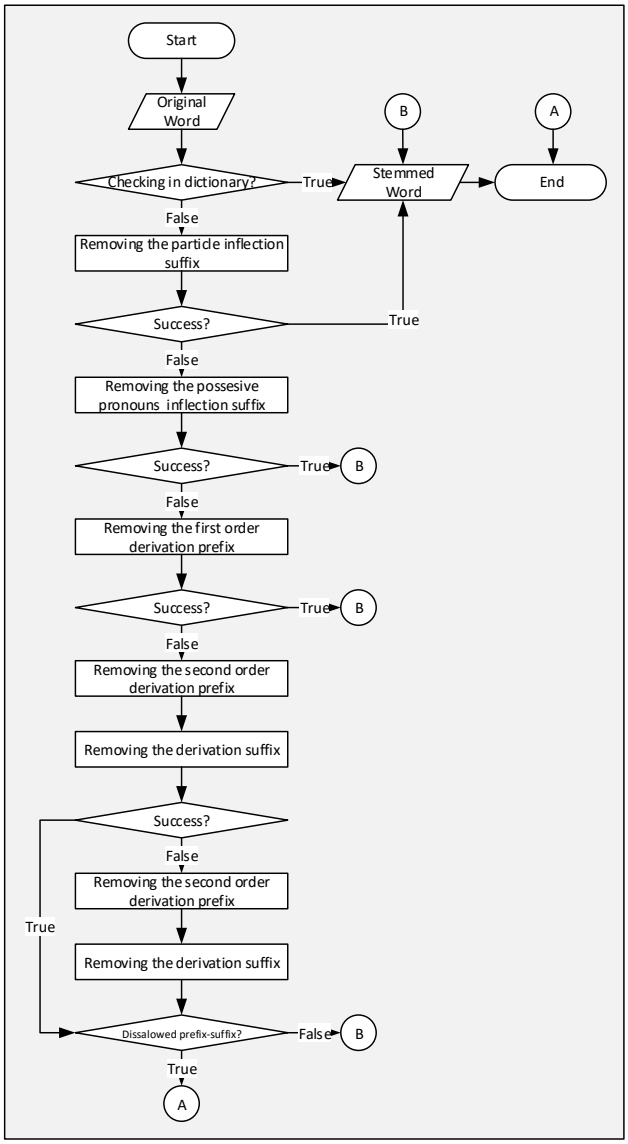

Fig. 3. Flowchart of Stemming Algorithm Model II

1. The algorithm checks an original word into the basic word dictionary. If successful, then algorithm stops and the word expressed as a basic word. If it fails, the algorithm goes to the next step.
2. The algorithm removes the particle inflection suffix ("*-lah*"," *–kah*", "*-tah*", "*-pun*"). If it fails, goes to the next step.

3. The algorithm removes the inflection possessive pronouns ("*-ku*", "*-mu*", "*-nya*"). If it fails, goes to the next step.
4. The algorithm removes the first order derivation prefix. If it success, the algorithm removes the second order derivation prefix for the first order and removes the derivation suffix. If it fails, the algorithm removes the second order derivation prefix for the possessive pronouns and removes the derivation suffix.
5. The algorithm checks the disallowed prefix-suffix. If the algorithm found it, the algorithm returns.
6. If all step is fail, the algorithm stops and returns to the original word.

In this study, the addition of some rules on the stemming model I, and then referred as stemming model III, can be explained as follows.

1. Adding some new basic words, abbreviations, entities, and foreign terms that found in the result of stemming model I into the basic word dictionary.
2. Adding a new rules for:
    a. If the particle inflection suffix is *–kah,* and the derivation prefix is *me-*, then removing the derivation prefix without removing the particle inflection suffix, for the word "*menikah* (married)".
    b. If the particle inflection suffix is *–kah,* and the derivation prefix is *se-*, then removing the derivation prefix without removing the particle inflection suffix, for the word with two prefix (*ber-* followed by *se-*), example "*bersedekah* (giving charity)".
    c. If the derivation suffix is *–an,* and the derivation prefix is *ke-* or *ter-* or *me-* or *pe-*, then removing the derivation prefix without removing the derivation suffix, for the word "*kedelapan* (eight)", "*terdepan* (front)", "*menekan* (push)", "*menelan* (swallow)"," *pemeran* (cast)".
    d. If the derivation suffix is *–an,* and the derivation prefix is *se-*, then removing the derivation prefix without removing the derivation suffix, for the word with two prefix (*ke-* followed by *se-*), example "*kesekian* (so and so)".
    e. If the derivation suffix is *–i,* and the derivation prefix is *peng-* or *pem-* or *mem-* or *men-*, then removing the derivation prefix without removing the derivation suffix, for the word "*pengendali* (controller)", "*pembeli* (buyer)", "*membeli* (buy)", "*mencuri* (steal)".
    f. If the derivation suffix is *–i,* and the derivation prefix is *ber-*, then removing the derivation prefix without removing the derivation suffix, for the word with two prefix (*pem-* followed by *ber-*), example "*pemberi* (giver)".

D. Evaluation

The evaluation on the results of stemming built in this study was done twice: before and after the addition of rules and some new words into basic word dictionary. Both evaluations were performed by calculating Precision, Recall and F-Measure values between automatic stemming results and manual stemming results [13].

Precision is calculated based on the similarity between the results of manual stemming with the stemming result given by the system, as in (1). Recall is calculated based on the success of the system in finding the basic word in the stemming process, as in (2). F-Measure is calculated based on the value of Precision and Recall, as in (3).

$$P=TP/(TP+FP) \quad (1)$$

$$R=TP/(TP+FN) \quad (2)$$

$$F=(2*P*R)/(P+R) \quad (3)$$

TP refers to the number of words that are successfully stemmed. *FP* refers to the number of false stem words of all words that are successfully stemmed. *FN* refers to the number of words that are not successfully stemmed.

III. Results and Discussion

A. Text Processing

The corpus which consists of 125 files is processing to produce a list of words that ready for stemming. The results of text processing are shown in Table 6.

B. Stemming Model I

At this step, the evaluation is performed using the Nazief-Adriani stemming algorithm that has been refined by Asian Jelita. The basic word dictionary used consists of 28,527 words. The evaluation results from this model are shown in Table 7. Of 19,129 words that must be stemmed, this algorithm is able to find 13,113 successful stem words and 6,122 unsuccessful stem words. From 13,113 successful stem words, there are 12,917 true results (*tp*) and 196 false results (*fp*). From 6,122 unsuccessful stem words, there are 496 true results (*tn*) and 5,626 false results (*fn*).

TABLE VI. TEXT PROCESSING OUTCOMES

No.	Step	Number of Words
1.	Tokenization	36,097
2.	Case Folding	34,981
3.	Filtering	19,129

TABLE VII. STEMMING OUTCOMES

No.	Stemming Model	TP	FP	FN
1.	I	12,917	196	5,626
2.	II	13,069	427	7,372
3.	III	16,756	41	1,932

TABLE VIII. COMPARATIVE OF STEMMING MODEL

No.	Stemming Model	Precision	Recall	F-Measure
1.	I	0.9851	0.6966	0.8161
2.	II	0.9678	0.6427	0.7724
3.	III	0.9976	0.8966	0.9444

From the results of this evaluation, stemming model I has a performance value that is 0.9851 (99%) for Precision, 0.6966 (70%) for Recall, and 0.8161 (82%) for F-Measure.

The error that occurs in the stemming algorithm model I caused by several factors:

1. There are basic words that have not been stored in the basic word dictionary, such as word "*pegawai* (employee)", "*derita* (suffer)", "*doa* (pray)", etc.
2. There are abbreviations that often appear in the text and are considered already common in the writing of news texts, such as "*apbn*", "*bumn*", "*cagub*", "*jpg*", etc.
3. There are many entities such as place names, day names, names of months that have not been stored in the dictionary of the word base, such as "Indonesia", "*senin* (Monday)", "*juni* (June)", etc.
4. There are unrecognized English terms, such as the word "background", "banking", "building", "capture", etc.
5. There are words that contain affixes that do not match with existing rules, such as word "*menikah* (married)", "*bersedekah* (giving charity)", "*kedelapan* (eight)", "*terdepan* (front)", "*menekan* (push)", "*menelan* (swallow)"," *pemeran* (cast)", "*kesekian* (so and so)", "*pengendali* (controller)", "*pembeli* (buyer)", "*membeli* (buy)", "*mencuri* (steal)", "*pemberi* (giver)".

C. Stemming Model II

The evaluation results for stemming model II is taken from the evaluation results which has been done in the previous study using the same corpus [6]. The results show stemming model II has a performance value that is 0.9678 (97%) for Precision and 0.6427 (64%) for Recall. F-Measure value can be calculated from the Precision and Recall value.

$$\text{F-Measure} = (2*0.9678*0.6427)/(0.9678+0.6427)$$
$$= 0.7724$$
$$= 77\%$$

D. Stemming Model III

At this step, the evaluation is performed using a stemming model III algorithm to overcome the weakness in stemming model I. The solution is done by adding 147 new basic words found on the stemming result of model I into the basic word dictionary. This is done to increase the amount of the basic words dictionary collection so as to increase the number of words that are recognized during stemming process. There is also the addition of 188 abbreviations found in the results of stemming model I into the basic word dictionary. This is done based on observations of the contents of the news text in each channel indicating that the abbreviations appear in large quantities for news in the same channel. That is, the abbreviation plays an important role in the text. The same applies to the 123 entities and 671 foreign terms in English. This all addition keeps the number of words stored in the words dictionary number becomes 29,656 words. The next step is adding a rule for unrecognized affixed words to stemming.

The evaluation results from stemming model III are shown in Table 7. Of 19,129 words that must be stemmed, this algorithm is able to find 16,797 successful stem words and 2,428 unsuccessful stem words. From 16,797 successful stem words, there are 16,756 true results (*tp*) and 41 false results (*fp*). These false results appear because there is a case that an original word matches with two rules. When the word matches with the first rule and the result is appear in the dictionary, so the result is assumed as basic word and the algorithm returns. Even though, the original word more suitable for the second rule. For example, the word "mengalahkan (beating)" is stemmed to be "*alah*" than "*kalah* (lose)". This cause the result does not match with the result from manual stemming. From 2,428 unsuccessful stem words, there are 496 true results (*tn*) and 1,932 false results (*fn*). These missing results appear because there are still many entities like name of people and organization that most appear in the text but have not been recognized in stemming process because the entities have not stored in dictionary. From the results of this evaluation, stemming model III has a performance value that is 0.9976 (~100%) for Precision, 0.8966 (90%) for Recall, and 0.9444 (94%) for F-Measure.

The comparison of evaluation results for stemming models I, II and III can be seen in Table 8. The performance of the stemming model III increases 1% for Precision, 29% for Recall, and 16% for F-Measure than stemming model I. The performance of the stemming model III increases 3% for Precision, 40% for Recall, and 22% for F-Measure than stemming model II. The comparison results show that stemming model III has the highest Precision, Recall and F-

Measure values. This is because the stemming model III uses a more complex algorithm than the stemming model I and II. Stemming model III has added more basic words, abbreviations, entities and foreign terms to the basic word dictionary so it can improve the compatibility of a word with the basic words in the dictionary. Furthermore, the addition of rules in stemming also plays an important role because it is able to increase the number of words that has successfully stemmed. These two steps show that the number of words in the basic words dictionary and the accuracy and completeness of rules play a very important role in the success of a stemming algorithm.

IV. CONCLUSION

The development of stemming algorithm in this study is based on the algorithm Nazief-Adriani that has been refined by Asian Jelita. The evaluation of 125 digital news texts shows the use of the Nazemma-Adriani stemming algorithm and Jelita Asian provides better performance than the Porter-like stemmer algorithm. Preliminary tests of the stemming algorithm Nazief-Adriani and Jelita Asian found some new basic words that have not been stored in the basic dictionary. In addition, there are many abbreviations, entities and foreign terms that appear common in the news text but have not stored in the basic word dictionary. Furthermore, there are some unrecognized affixed words in defined rules. The addition of basic words, abbreviations, entities and foreign terms to the basic word dictionary, along with the addition of rules into the rule list can improve the performance of the stemming algorithm built on this study. The performance values of the stemming algorithm are 0.9976 (~100%) for Precision, 0.8966 (90%) for Recall, and 0.9444 (94%) for F-Measure. Thus, the completeness of words in the basic words dictionary and the accuracy and completeness of rules play a very important role in the success of a stemming algorithm. In the future, it is desirable that more studies is done to build a complete words basic dictionary and the appropriate and complete morphological rules for the Indonesian language, so that it can be used as a reference in processing the Indonesian language text automatically.

REFERENCES

[1] F.Z. Tala, A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. The Netherlands, 2003.

[2] B.A. Nazief and M. Adriani, Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia. Jakarta: Faculty of Computer Science, University of Indonesia, 1996.

[3] V.B. Vega and S. Bressan. "Indexing the Indonesian Web: Language Identification and Miscellaneous Issues." in *Tenth International World Wide Web Conference*. 2001. Hongkong.

[4] A.Z. Arifin and A.N. Setiono. "Classification of Event News Documents in Indonesian Language Using Single Pass Clustering Algorithm." in *Proceedings of the Seminar on Intelligent Technology and its Applications (SITIA)*. 2002. Surabaya, Indonesia.

[5] J. Asian, H.E. Williams, and S.M.M. Tahaghoghi. "Stemming Indonesian." in *Proceedings of the Twenty-eighth Australasian conference on Computer Science*. 2004. Australian Computer Society, Inc. .

[6] P.M. Prihatini and I.K. Suryawan. "Text Processing Application Development for Indonesian Documents Clustering." in *The 1st International Joint Conference on Science and Technology (IJCST)*. 2016. Bali, Indonesia.

[7] A.F. Hidayatullah and M.R. Ma'arif, "Pre-Processing Tasks in Indonesian Twitter Messages," IOP Conf. Series: Journal of Physics: Conf. Series, vol. 801, 2016.

[8] A. Pirkola, "Morphological Typology of Languages for Ir," Journal of Documentation, vol. 57, 2001.

[9] A. Najibullah. "Indonesian Text Summarization Based on Naïve Bayes Method ". in *Proceeding of the International Seminar and Conference 2015: The Golden Triangle (Indonesia-India-Tiongkok) Interrelations in Religion, Science, Culture, and Economic*. 2015. Semarang, Indonesia.

[10] P.M. Prihatini, et al., "Fuzzy-Gibbs Latent Dirichlet Allocation Model for Feature Extraction on Indonesian Documents," Contemporary Engineering Sciences, vol. 10, pp. 403-421, 2017.

[11] M. Widjaja and S. Hansun, "Implementation of Modified Porter Stemming Algorithm to Indonesian Word Error Detection Plugin Application," International Journal of Technology, vol. 6, pp. 139, 2015.

[12] A. Hidayatullah, C. Ratnasari, and S. Wisnugroho, "Analysis of Stemming Influence on Indonesian Tweet Classification," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 14, pp. 665-673, 2016.

[13] C. Goutte and E. Gaussier. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation." in *Proceedings of the European Colloquium on IR Resarch (ECIR'05), LLNCS 3408* 2005. Springer.