ISSN: 1978-0524

Stemming javanese affix words using Nazief and Adriani modifications

Aji Prasetya Wibawa [a,1,*], Felix Andika Dwiyanto [a,2], Ilham Ari Elbaith Zaeni [a,3], Rizal Kholif Nurrohman [a,4], AN Afandi [a,5]

[a] *Department of Electrical Engineering, Universitas Negeri Malang, Indonesia*
[1] *aji.prasetya.ft@um.ac.id,* [2] *ayikgugun@gmail.com,* [3] *ilham.ari.ft@um.ac.id,* [4] *publicationcenter@um.ac.id,* [5] *an.afandi@um.ac.id*
** corresponding author*

ABSTRACT

Stemming is the process of finding a basic word with several stages of affix removal. The main reason for stemming is to check spelling and machine translation and to support the effectiveness of the retrieval process. This study uses the Nazief and Adriani algorithm for stemming Javanese-influenced words. The first step taken is data collection and making a basic word dictionary. Then do the stemming process. Before stemming, modifications are made to the rules. The rules of the Nazief and Adriani algorithm, which are based on the morphology rules of the Indonesian language, are modified to suit the morphological rules of the Javanese language. Of the 366 words that were tested, it produced 351 correct basic words and 15 basic words that experienced errors. The results show that this algorithm can be used for stemming Javanese with an accuracy value of 95.9%.

Keywords:
Stemming
Javanese language
Nazief and Adriani

I. Introduction

The Javanese language has high morphological complexity because of the many variations of affixes in the word [1]. Complexity is also due to a prefix that changes depending on the first character of the root word. Similar to Indonesian, Javanese has a prefix consisting of a prefix, an infix, a suffix, and a combination of the three.

The use of Javanese has begun to decline. Indonesian teenagers no longer considered fashionable and relevant regional languages in the era of globalization [2]–[4]. It takes effort to reinvigorate the local language. There needs to be a serious effort to prevent fading level Java language capabilities, such as making an interpreter or learn the vocabulary of the dictionary [5] [6].

But there are things that must be understood that in the dictionary, there are no word affixes. The word affix can not be translated directly by the dictionary, so an alternative is needed by using the stemming algorithm[5].

Stemming algorithms in Information Retrieval system serves to reduce the number of indexes of a document. Besides that, it is used to classify words that have similar basic words and meanings but have different affixes [7]. In other words, stemming is the process of extracting the basic words from the original words and separating affixes [8]. This process has an important role because it is one of the first steps in the Information Retrieval process [9]. To produce the process of taking the prefix, infix, end or combination correctly, it is necessary to learn the morphology of a language correctly.

One of the stemming algorithms is the Nazief and Adriani method [10], [11]. This method has the advantage of a high degree of accuracy, and little overstemming and understemming [12]. Various studies on Javanese stemming have been done before [1], [5]. However, the use of Nazief and Adriani for Javanese has not proven successful.

This article will discuss the application of Nazief and Adriani stemming methods to Javanese. Due to the difference between the affix between Javanese and Indonesian, this method must be modified so that the stemming process runs well. Therefore this article is written systematically to discuss research methods, the results of the discussion as well as conclusions and suggestions for future development.

DOI: 10.26555/jifo.v14i1.a17106
W : http://journal.uad.ac.id/index.php/JIFO
E : jifo@uad.ac.id

II. Method

A. Javanese Morphology

In Javanese morphology, affixes are grouped into several categories as follows [13], [14]:

1) *Ater - ater (Prefix)*

This group can be divided into three:

a. *Ater - ater Anuswara*, which is a prefix that uses nasal noises because it produces buzzing sounds, including: "n-", "ny -", "m-", and "ng-".

b. *Ater - ater Tripurusa*, as opposed to *anuswara*, this prefix separates passive sentences, *including*: "no-", "kok-", "di-".

c. *Ater - ater Liya*, prefixes other than *Anuswara* and *Tripurusa*, include: "a-", "pa-", "pan-", "pam-", "pang-", "pi-", "pre-", "no - "" tar- "," ka- "," ke "," sa- "," kuma- "," we- "," kapi- ".

2) *Seselan* (Inserts)

Is an affix that is located in the middle of the word. In Javanese there are four insertions, namely: "-um-", "-in-", "-el-", "-er-".

3) *Penambang* (Suffixes)

Affixes located at the end of the word, among others: "-a", "-i", "-e", "-an", "-en", "-ana", "-ake", "-na", "-Ne", "-my", "-mu".

B. Data Set

This study uses data in the form of various words with Javanese influences. The development of Nazief and Adriani algorithm is based on the morphology of the Indonesian language, so before the stemming process is carried out it is necessary to modify the rules in accordance with the morphology of the Javanese language. Modified rules follow the rules defined in the Javanese Paramasastra book [15] and Javanese Language Structure [13]. The rules for decapitation for Java are shown in Table 1.

Table 1. Rules for the decapitation of Nazief and Adriani prefix for the Javanese language

No	Format words	Prefix	Example
1	kumaKD	kuma-KD	kumawani : kuma-wani
2	kamiKD	kami-KD	kamituwa : kami-tuwa
3	kapiKD	kapi-KD	kapilare : kapi-lare
4	pangKD	pang-KD	pangurip : pang-urip
5	kokKD	kok-KD	kokjupuk : kok-jupuk
6	panKD	pan-KD	pantelung : pan-telung
7	praKD	pra-KD	pralambang : pra-lambang
8	tarKD	tar-KD	tarbuka : tar-buka
9	takKD	tak-KD	takjupuk : tak-jupuk
10	dakKD	dak-KD	dakpangan : dak-pangan
11	diKD	di-KD	dibuntel : di-buntel
12	paKD	pa-KD	pawarta : pa-warta
13	piKD	pi-KD	pikukuh : pi-kukuh
14	saKD	sa-KD	sakarep : sa-karep
15	aKD	a- KD	agawe : a-gawe

In Table 1, the KD symbol states the basic word. For example, the word "kumawani" will be decapitated into the prefix "kuma" and the basic word "wani". Some suffixes in Javanese and examples of their use are shown in Table 2.

Table 2. Rules for decapitation of Nazief and Adriani suffixes for Javanese

No	Format words	Prefix	Example
1	ana	KD-ana	silihana : kuma-wani
2	ane	KD-ane	tandurane : kami-tuwa
3	ake	KD-ake	garapake : kapi-lare
4	an	KD-an	tulisan : tulis-an
5	na	KD-na	gambarna : gambar-na
6	ne	KD-ne	segane : sega-ne
7	ku	KD-ku	montorku : montor-ku
8	mu	KD-mu	klambimu : klambi-mu
9	a	KD-a	turua : turu-a
10	e	KD-e	omahe : omah-e
11	i	KD- i	tanduri : tandur-i

From the rules of decapitation in Table 2 an example can be drawn on the word "silihana" to be decapitated into the basic word "silih" and the suffix "ana".

Table 3. Rules for decapitation of Nazief and Adiani complex prefix for the Javanese language

Rules	Format words	Beheading	Example
J1	ng{r|l}V...	ng-{r|l}V...	ngrakit : ng-rakit
J2	m{l}V...	m-{l}V...	mlembang : m-lembang
J3	m{b|u}A...	m-{b|u}A...	mbacok : m-bacok
J4	n{j|d}...	n-{j|d}...	njaba : n-jaba; ndableg : n-dableg
J5	ng{g}...	ng-{g}...	nggembol : ng-gembol
J6	kV...	k-V...	kelingan : k-eling-an
J7	kaA...	ka-A	kapendhem : ka-pendhem
J8	keC...	ke-C...	kegodhok : godhok
J9	ngA...	ng-A...	ngababi : ng-abab-i
J10	pV...	p-V...	padusan : p-adus-an
J11	nyA...	ny-cA... | ny-sA...	nyangkem : ny-cangkem
J12	mV...	m-pV...	metani : m-petan-i
J13	mV...	m-wV...	minihi : m-winih-i
J14	nV... dimana A!=”j,d”	n-tV...	napok : n-tapok
J15	ngA...	ng-kA...	nginthil : ng-kinthil
J16	ngV...	ng-wV...	ngetan : ng-wetan

Table 3 shows that the symbol V represents the vowel, C for consonants and A for vowels / consonants. For example, the word "ngrakit" will be decapitated into the prefix "ng" and the basic word "rakit".

The related words used to test the stemming algorithm were taken from the Complete Javanese Dictionary, a Javanese-language newspaper called "PanjebarSemangat" and several Javanese-language sites.

The data collected from various sources is 449 words. The words used in the test consist of 235 words with prefix, 40 words with suffix, and 91 words with prefix and suffix, while 83 words are not used because there are no rules for decapitation the infix. The number of each word in each rule is shown in Figure 1.

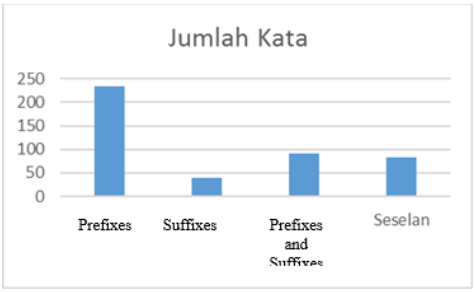

Fig. 1. Graph of total words

The test results will be manually checked to identify the correct base words or those experiencing errors. The results are evaluated using a Javanese dictionary. Then do the evaluation by calculating the value of accuracy. The calculation of the accuracy value is shown in Formula (1).

$$Accuracy = \frac{correct\ member\ of\ word}{total\ member\ of\ words}\ X\ 100\% \quad (1)$$

C. Nazief dan Adriani Stemming Algorithm

The development of Nazief and Adriani algorithms is based on the Indonesian morphology rules. This algorithm is grouped into prefixes, infixes, suffixes and combined suffix prefixes. In addition, it is also supported by the existence of a basic word dictionary and rearrangement of words that experience excessive stemming.

Indonesian morphology is grouped into several categories as follows [16]:

Inflection suffixes are groups of suffixes that do not change the basic word form. This group can be divided into two:

Particle (P), which includes "-lah", "-lah", "-tah", and "-pun".

Possessive Pronoun (PP) or its pronouns, including "ku", "- mu", and "nya".

Derivation Suffixes (DS) is a collection of original Indonesian suffixes which are directly added to the basic words namely suffixes "-i", "-kan", and "-an".

Derivation Prefixes (DP) are collections of prefixes that can be directly given to pure basic words, or to basic words that have been added up to 2 prefixes. These include:

Prefixes that have morphology ("me", "be-", "pe-" and "te-")

The prefix that has no morphology ("di-", "ke" and "se").

Based on the classification of the affixes above, the form of affixed words in Indonesian can be modeled as follows:

[DP+ [DP+ [DP+]]] Root Word [[+DS] [+PP] [+P]]

So that the Nazief and Adriani algorithms can be used for the Javanese language, the beheading rules are modified according to the morphology of the Javanese language. In addition, adjustments to the rules of beheading for the complex initials of the Nazief and Adriani algorithm in Indonesian, amounting to 33 rules [11], became 16 rules in Javanese as shown in Table 3.

The similarity in word formation structure between Javanese and Indonesian [5] allows algorithm modification. Modifications to this algorithm are packaged in the form of flowcharts.

Pada permulaan pemrosesan, dan pada setiap langkah selanjutnya, periksa kata yang saat ini diinputkan terhadap kamus kata dasar. Jika kata ditemukan, kata tersebut dianggap sebagai kata dasar dan proses berhenti. At the beginning of processing, and at each subsequent step, check the word that is currently entered into the base word dictionary. If a word is found, the word is considered as the basic word and the process stops.

First delete the suffix that does not affect the spelling of the words "" me "," - your "," e "}. For example "klambimu" (your clothes) becomes "klambi" (clothes). If the word is found in the dictionary, the process is stopped.

Remove the suffixes {"-a", "-i", "-en"}. Then check the dictionary. If the word is found, the process stops. If the word is not found, then proceed to removing the next suffix.

Remove the suffixes {"-an", "-ak", "-n"}. For example in the word "njagongi", the word will be stemmed as "njagong". Because it is not a valid root word, it continues to delete the prefix.

Deleting prefixes ("n-", "ny -", "m-", "ng-", "no-", "kok-", "di-", "a-", "pa-", "pan - "," pam- "," pang- "," pi- "," pre- "," no- "" tar- "," k "," p "," ka- "," th ", "Sa-", "kuma-", "we-", "kapi-"):

Stop the process if:

The prefix that is considered identical to the prefix that was deleted before

Three prefixes have been removed

Identify the type of prefix. This prefix consists of two types:

Standard ("dak-", "kok-", "di-", "a-", "pan-", "pam-", "pang-", "pi-", "pre-", "tak- "" Tar- "," tok - "," ke "," sa- "," kuma- "," Kami- "," kapi- ") which can be removed immediately.

Complexes ("n-", "ny -", "m-", "ng -", "k -", "p -", "ka -", "pa-") which have different variants, can morphology as per the basic word attached. Therefore it is necessary to use the rules in Table 3.

In the previous step, partial removal of the word "njagongi" to "njagong" was done. This step will remove the "n-" prefix to get "jagong". This is a valid base word, so the process stops. If no prefix above is appropriate, the process stops and the algorithm identifies that the root word is not found.

If the word searched in the dictionary is not found, repeat step 4 (recursive process) again. If the word is found, the process stops.

If after removal the recursive prefix, the word has not been found. Then the recoding process is carried out with reference to Table 3. The columns in Table 3 show the prefix and character encoding variants to be used when the first syllable of the base word starts with a certain letter. Not all prefixes have a recoding character.

If all steps are still unsuccessful, the algorithm will return the original word before stemming.

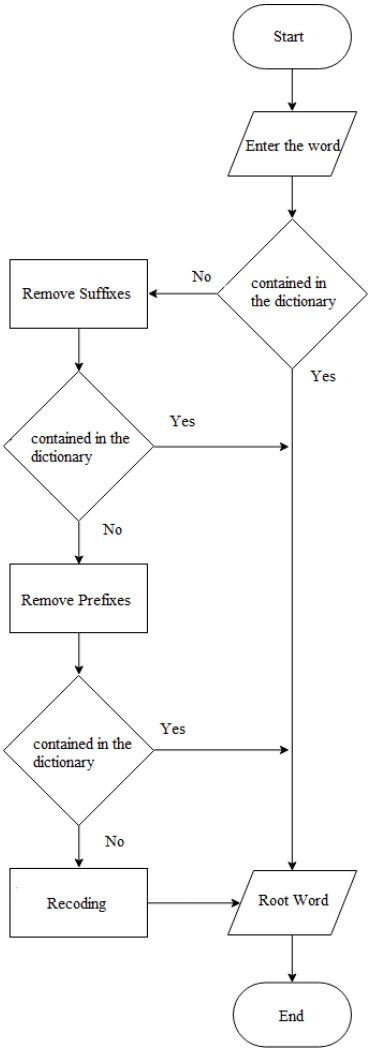

Fig. 2. Flowchart algoritme nazief and adriani for Javanese language

III. Results and Discussion

From the test results using Javanese words, the algorithm is made capable of stemming words that have prefixes, suffixes and a combination of prefixes and suffixes. However, there are a number of influential words that cannot be stemming.

The first mistake is the words *"kamigilan"* and "kamitegen". The words *"kamigilan"* and *"kamitegan"* produce incorrect output. The decapitation that was supposed to be *"kami-gila-n"* is actually the letter *"a"* at the end of the word, which is carried out because it reads as a suffix that causes overstemming. The word that should be *"gila"* becomes *"gil"*.

The second mistake is in the words *"pamulang"* and *"pamriksa"*. The word *"pamulang"* produces a fixed word that should be *"wulang"*. This is because if *"pam"* is removed, it will produce the word *"ulang"*, so there is a need for rules that eliminate the prefix *"pam"* and the addition of the letter *"w"* at the beginning of the word. While the word *"pamriksa"* cannot be done stemming because there are no rules that eliminate the prefix *"pam"* and the addition of the letter *"p"* at the beginning of the word.

The third mistake is the word *"koktekani"*. This error is caused by overstemming. The letter *"ani"* at the end of the word should be omitted only the letters *"n"* and *"i"*, in fact the letter *"a"* participated removed. That is because the existence of *"an"* at the end of the word is considered as a suffix that must be removed.

The fourth mistake in the word that has an effect on *"Pa"* which is *"pamomong"* and *"pamudha"* cannot be done stemming. Each of these words should be *"momong"* and *"mudha"*. The error lies in the prefix that should be removed only *"pa"* it actually eliminates the prefix *"pam"* contained in the rules as well.

The fifth mistake is the words *"sadinane", "relapse", "gawana", "segane" and "tekane"*. The word has the same error, which is overstemming at the end. The letter *"an"* at the end of a word is considered as a suffix that must be removed.

The sixth mistake is the word *"katawakake"*. The word was beheaded as *"tawa"*. Because there is no suffix *"kake"* beheading is done in stages by eliminating *"e"* and *"ak"*. The letter *"a"* in the word *"tawa"* is carried out with the omission due to the suffix *"ak"* so that overstemming occurs.

The seventh mistake in the word *"pangenan"*. The fault lies in the prefix.. The existence of the prefix *"Pang"* and *"Pan"* causes overstemming. The *"p"* prefix that should have been removed, it's the *"pang"* that was removed.

The eighth mistake of the word *"Nithik"*. The word should be distemming to *"thithik"*. Because there are no rules that change the prefix *"n"* to *"th"* so the word fails to stemming.

Of the errors in the words above, overstemming occurs at most words with the character *"an"* at the end of the word. As for the results of the experiments conducted in this study are shown in Table 4.

Table 4. Table Styles

Rules	Correct word	Wrong word	Accuracy (%)
Prefixes	228	3	98.7
Suffixes	37	3	92.5
Prefixes and Suffixes	86	9	90.5
Total	351	15	95.9

The accuracy value shows that the Nazief and Adriani algorithm can be used for Javanese and has good accuracy for Javanese. However, modifications made have not been able to stem words with infixes, the need for the establishment of insert rules in accordance with the morphology of the Javanese language.

IV. Conclusion

Based on the results of research that has been done that this algorithm can be used to perform Javanese stemming with an accuracy rate of 95.9%. Good accuracy is found in the prefix rule with minimal errors while in the suffix, there are still many errors. The system created is only capable of stemming words that contain prefixes and suffixes, so it is not possible for words that contain infixes.

In the future, it is expected that there will be developments that can correct errors, especially in the suffix rules, which are still largely wrong. In addition, it is also necessary to establish the appropriate infix rules of the morphology of the Javanese language so that words that have infixes can be done stemming. To produce an optimal root word also needs a complete root word.

References

[1] F. Amin, W. Hadikurniawati, S. Wibisono, H. Februariyanti, and J. S. Wibowo, "A Hybrid Method of Rule-Based and String Matching Stemmer for Javanese Language," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 19, pp. 4973–4982, 2017.

[2] N. J. Smith-hefner, "Language Shift, Gender, and Ideologies of Modernity in Central Java, Indonesia," vol. 19, no. 1, pp. 57–77, 2009.

[3] D. E. Subroto, M. Dwirahardjo, and B. Setiawan, "Endangered Krama and Krama Inggil Varieties of the Javanese Language," *Linguist. Indones.*, vol. 26, no. 1, pp. 89–96, 2008.

[4] S. Suwadji, "Javanese Language Today," in *Lokakarya Pengajaran Bahasa dan Sastra Jawa*, 1996, pp. 55–61.

[5] M. Madia, "Stemming Bahasa Jawa untuk Mencari Akar Kata dalam Bahasa Jawa dengan Aturan Analisis Kontrasif Afiksasi Verba," Universitas Islam Negeri Maulana Malik Ibrahim, 2016.

[6] A. Setiyowati, "Intereferensi Morfologi dan Sintaks Bahasa Jawa dalam Bahasa Inidonesia pada Kolom 'piye ya?' Harian Suara Merdeka." p. 76, 2008.

[7] [A. D. Tahitoe and D. Purwitasari, "Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming," in *Jurnal Ilmiah*, 2010, pp. 1–15.

[8] V. Gupta, N. Joshi, and B. Vidyapith, "Design & Development of a Rule Based Urdu Lemmatizer," *1st Int. Conf. Futur. Trend Comput. Anal. Knowl. Manag. IEEE*, no. July, 2015.

[9] C. Moral, A. de Antonio, R. Imbert, and J. Ramírez, "A Survey of Stemming Algorithms in Information Retrieval," *Inf. Res. An Int. Electron. J.*, p. 22, 2014.

[10] J. Asian, "Effective Techniques for Indonesian Text Retrieval," *Ph.D Thesis*, pp. 1–286, 2007.

[11] J. Asian, B. Nazief, and H. Williams, "Stemming Indonesian : A confix-Stripping Approach," no. January, 2007.

[12] M. S. H. Simarangkir, "Studi Perbandingan Algoritma - Algoritma Stemming untuk Dokumen Teks Bahasa Indonesia," *J. Inkofar*, vol. 1, no. 1, pp. 41–47, 2017.

[13] P. Purwadi, *Struktur Bahasa Jawa*. Yogyakarta: Media Abadi, 2005.

[14] A. B. Setiyanto, *Parama Sastra Bahasa Jawa*. Yogyakarta: Panji Pustaka, 2007.

[15] A. B. Setiyanto, *Parama Satra: Javanese Language*. Yogyakarta: Panji Pustaka, 2007.

[16] J. Wibowo, "Aplikasi Penentuan Kata Dasar dari Kata Berimbuhan pada Kalimat Bahasa Indonesia dengan Algoritma Stemming," *J. Ris. Komput.*, vol. 3, no. 5, pp. 346–350, 2016.