Jurnal Elektronik Ilmu Komputer Udayana
Volume 8, No 3. February 2020 p-ISSN: 2301-5373
e-ISSN: 2654-5101

Lemmatization in Balinese Language

I Gede Angga Purnajiwa Arimbawa[a1], Ngurah Agus Sanjaya ER[a2]

[a]Informatics Department, Udayana University
Bali, Indonesia
[1]arimbawaangga@gmail.com
[2]agus_sanjaya@unud.ac.id

Abstract

Lemmatization is a process to extracting root word from an affixed word with the aim of reducing variations of the word into the root word. Previous researches on extraction of root word in Balinese Language has been done with rule- based methods to remove affixes from words. The weakness of the rule-based method is that it must comply with the set of rules provided. However, writings in Balinese often contain typographical errors because speakers tend to write words according to how the word is spoken instead of following the correct rules. In this research, we apply the Levenshtein distance method to overcome the aforementioned shortcoming. After all the rules applied to a given word fail, the Leven-shtein distance method is used to list all words that are "close". Next, we select the closest word as the root word of the given input. Based on the experiments, our proposed method achieved an accuracy of 96.01 %.

Keywords: *Lemmatization, BalineseLanguage, LevenshteinDistance, Rule-Based, LexiconBased*

1. Introduction

Indonesia is a country that has abundant wealth, not only wealth of natural resources, Indonesia also has rich cultural heritage. One form of Indonesian cultural is the Regional Language. Until 2017, the Language Agency under the Ministry of Education and Culture Indonesia has mapped at least 652 regional languages in Indonesia [1]. One of the regional languages in Indonesia with speakers that is quite a lot is Balinese.

Balinese is one of the Austronesian languages which is the mother tongue for Balinese tribes who live on the Bali island, Indonesia. Balinese language is one of top ten regional languages with the most speakers in Indonesia makes this language important. Balinese language is used in government affairs, education, and other matters in addition to everyday conversation.

Balinese has unique characteristics, because to express a meaning can use a variety of words, this is due to the level of language (sor singgih basa), this makes the vocabulary in the Balinese language very rich. Balinese also has another uniqueness, namely the difference in words when written and pronounced as the example of the word "malajah" (English: Learning), written "malajah" but pro- nounced "melajah", because of its uniqueness, many Balinese speakers write words like how to pronounce and not writing according to the correct rules.

Not much different from Indonesian, Balinese also has rules in its use. In Balinese every word can have affix. The words affixed in the Balinese language can be distinguished according to the place attached to the basic form or origin, namely prefix, suffix, infix, confix, simulfix, and a combination of affixes. Prefix is an affix that is affixed at the beginning of a basic word. Suffix is an affix that is affixed at the end of a basic word. Infix is an affix affixed in the middle of a basic word. Confix are single affixes that occur from a combination of prefixes and endings that form a unit [2].

Simulfix (simultaneous affix) is an affix that lies only in the prefix of the base word, simulfix is the formation of words that are considered not standard and are generally used in the language of conversation or local language [3].The combination of affixes is a supplementary combination of prefixes, suffixes and infixes in a basic word. The process of extracting basic words in the field of text mining is very important, because it can reduce variations in words in the form of basic words.

To extract basic words, two methods can be used, namely by stemming and lemmatization. Stemming and lematization have the same goal, to find the basic words or general words of a word. It’s just that the stemming approach is simpler because it only stem the affixes, while lemmatization use a better approach by using the vocabulary / data dictionary and the morphology analysis of the word.

In some studies that have been reviewed and related to extraction of basic words in Balinese Languange, it is assumed that the word to be processed is in accordance with the applicable rules (a standard word) and does not consider the state of error. Even though in Balinese, writing a word is often written not in accordance with the rules, because speakers are more likely to write in the way they pronounce the word. So in this study, we contribute to making a basic word extraction system in Balinese Language that is able to handle writing errors that are not in accordance with the standard rules.

2. Literature

The process of finding basic words in text is very useful in the field of Natural Language Processing and Text Mining, because this process can reduce the dimensions of the data to be processed. Some research on Balinese Language has ever been done. [4] Used the Rule-based method with The Porter Stemmer algorithm for Indonesian Language which has been supported by the Balinese language morphology, then refined by [5] by adding rules to infix, confix, simulfix and affix combinations. The advantages possessed by the based method are that it is easily approved and validated and easy to apply to a simple domain, but it will be more difficult to apply to domains with a higher level of complexity [5].

Stemming or lematization with rule-based approaches has been done in various languages such as the Indonesian language [6, 7, 8, 9, 10] and English [11]. The research conducted by [6, 7, 8, 9, 10, 11] managed the stemming and lemmatization with fairly good accuracy, but cannot be done on words with typographic errors. As we know typographical errors often occur in making documents. In Balinese Language, typography in writing is very possible because the pronunciation and writing of a word in Balinese language are often different, and speakers tend to write how the word pronounced rather than writing the correct rules. For example the word “malajah” (English: Learn- ing), is pronounced “melajah” but is written “malajah”, the letter “a” in Balinese language is often pronounced with “e”. The word “gerepe” (English: fingering) is pronounced “grepe” but is written “gerepe”.

There are several methods that can be used to overcome stemming or lemmatization failure due to typographic errors. One method is Levenshtein Distance. Calculation of distance values is obtained from the matrix used to calculate the number of differences in strings between two strings. Calculation of the distance between these two strings is determined from the minimum number of change operations to make string A become string B. Levenshtein Distance is used to compare input words with all words listed in vocabulary and return words that have the shortest difference distance as a result.

Levenshtein Distance has been widely used, in [12] it is used to check the spelling of answers, in [13] is used to test the similarity of the exam participant’s name on the LJK (Computer Answer Sheet) answer sheet and is used to de- tect the similarity of text document [14]. In [12, 13] test is a word so that the distance calculation process used is each letter, while at [14] test is per document so that the distance calculation process used is each word.

In this research, the method of lematization for Balinese Language was developed used the rule-based and lexicon-based approaches, and using Levenstein Distance which was tested in letters in each word to overcome the failure of lemmatization due to typographic errors.

3. Reseach Methods

This research consists of several stages, including the preparation of a list of basic words (lexicon- based), preparation of rule-based rules, system design, and results and discussion.

3.1. Ruled-Based

Rule based is built based on the Balinese affix rules, namely l, namely prefixes, suffixes, infixes, confixes, simulations, and combinations of affixes. In Balinese there are prefixes {n-, ma-, pa-, ka-, sa-, a-, pre-, pari-, pati-, then-, saka-, kuma-}, suffix {-a , -ang, -an, -in, -e, -ne, -n, -ing}, infixes {-in -, - um-, - el -, - er-}, confixes {pa-an, time, food, bra}, simulfix {ma-n, pa-n}, combination of affixes {ma-food, ma-in, ma-n-ang}. The structure of the affix rule is built by utilizing Regular Expression. The table of the rules of affixing is shown in tables 1,2,3,4,5 and 6.

Information:
ˆ ... = Begins with ... a$ = Ends with a [aiueo] = Vocal Letter
[ˆaiueo] = Consonant Letter [a-z] = All Alphabets
⇒ = change
None = Empty String

Table 1. Rule For Deleting Prefiks

No	Aturan	Action	Contoh	
			Dari	Ke
1	ˆng[aiueo]	ˆNg ⇒ None	Ngidih	Idih
2	ˆng[w]	ˆNg ⇒ None	Ngwangun	Wangun
3	ˆn[aiueo]	ˆN ⇒ t,d	Negul	Tegul
			Nunduk	Dundun
4	ˆny[aiueo]	ˆNy ⇒ c,j,s	Nyacad	Cacad
			Nyaring	Jaring
			Nyampat	Sampat
5	ˆng[aiueo]	ˆNg ⇒ k,g	Ngutang	Kutang
			Ngambar	Gambar
6	ˆm[aiueo]	ˆM ⇒ b,p	Mapag	Bapag
			Matek	Patek
7	ˆnga[ˆaiueo]	ˆNga ⇒ None	Ngamaling	Maling

Table 2. Rule For Deleting Suffiks

No	Aturan	Action	Contoh	
			Dari	Ke
1	[a-z]*a$	a$ ⇒ None	Daara	Daar
2	[a-z]*[aiueo]na$	na$ ⇒ None	Anggona	Anggo
3	[a-z]*[ˆaiueo]ang$	ang$ ⇒ None	Jemakang	Jemak
4	[a-z]*[aiueo][ny]ang$	[ny]ang$ ⇒ None	Gedenang	Gede
5	[a-z]*[ˆaiueo]an$	an$ ⇒ None	Cenikan	Cenik
6	[a-z]*[aiueo]nan$	nan$ ⇒ None	Dawanan	Dawa
7	[a-z]*[ˆaiueo]in$	in$ ⇒ None	Jagurin	Jagur
8	[a-z]*[aiueo]nin$	nin$ ⇒ None	Jumunin	Jumu
9	[a-z]*[ˆaiueo]e$	e$ ⇒ None	Payuke	Payuk
10	[a-z]*[aiueo]ne$	ne$ ⇒ None	Bajune	Baju
11	[a-z]*[ˆaiueo]ne$	ne$ ⇒ None	Baasne	Baas
12	[a-z]*[aiueo]nne$	nne$ ⇒ None	Giginne	Gigi
13	[a-z]*[aiueo]n$	n$ ⇒ None	Bukun	Buku
14	[a-z]*[aiueo]ning$	ning$ ⇒ None	Rikalaning	Rikala

Table 3. Rule For Deleting Infiks

No	Aturan	Action	Contoh	
			Dari	Ke
1	[ˆaiueo]*in[aiueo][a-z]*	in ⇒ None	Sinurat	Surat
2	ˆin[aiueo][a-z]*	in ⇒ None	Inucap	Ucap
3	[ˆaiueo]*um[aiueo][a-z]*	um ⇒ None	Rumaksa	Raksa
4	ˆum[aiueo][a-z]*	um ⇒ None	Umawak	Awak
5	[ˆaiueo]*el[aiueo][a-z]*	el ⇒ None	Telapak	Tapak
6	[ˆaiueo]*er[aiueo][a-z]*	er ⇒ None	Gerudug	Gudug

Table 4. Rule For Deleting Konfiks

No	Aturan	Action	Contoh	
			Dari	Ke
1	ˆpa[a-z]*an$	ˆpa ⇒ None an$ ⇒ None	Pasirepan	Sirep
2	ˆka[a-z]*an$	ˆka ⇒ None an$ ⇒ None	Kasengsaraan	Sengsara
3	ˆma[a-z]*an$	ˆma ⇒ None an$ ⇒ None	Majemakan	Jemak
4	ˆbra[a-z]*an$	ˆbra ⇒ None an$ ⇒ None	Majemakan	Jemak

Table 5. Rule For Deleting Simulfiks

No	Aturan	Action	Contoh	
			Dari	Ke
1	ˆmam[a-z]*	ˆmam ⇒ b,p	Mamuduh	Buduh
2	ˆpang[a-z]*	ˆpang ⇒ k,g	Pangalung	Kalung

Table 6. Rule For Deleting Affiks Combination

No	Aturan	Action	Contoh	
			Dari	Ke
1	ˆma[a-z]*an$	ˆma ⇒ None an$ ⇒ None	Makurenan	Kuren
2	ˆman[a-z]*in$	ˆma ⇒ t,d in$ ⇒ None	Manuturin	Tutur
3	ˆmang[a-z]*ang$	ˆmang ⇒ None ang$ ⇒ None	Mangorahang	Orah

3.2. Lexicon-Based

Some of Balinese basic words are used to ensure that the results of melting the affixed word are in accordance with their basic form and each basic word is also used to compare with the input word. The list of basic words used is 10,000 words collected by scrapping from the website *https://dictionary.basabali.org*. The list of basic words in vocabulary is arranged alphabetically.

3.3. Levenshtein Distance

Levenshtein Distance algorithm (also known as Edit-Distance) is an algorithm of calculating similarity between 2 Strings. This algorithm calculates the number of edit operations needed to convert string A to String B. The most common way to do the calculation is using a dynamic programming approach and utilizing the Matrix. The matrix is initialized in size (m, n) m is the size of string A and n is the size of String B. The matrix is filled from the upper left corner to the lower right corner. Every horizontal or vertical step has to do with inserting or deleting. The cost to carry out operations (insert or delete) is 1 for each operation. Mathematically, can be described as follows.

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j)+1 \\ \operatorname{lev}_{a,b}(i,j-1)+1 \\ \operatorname{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{1}$$

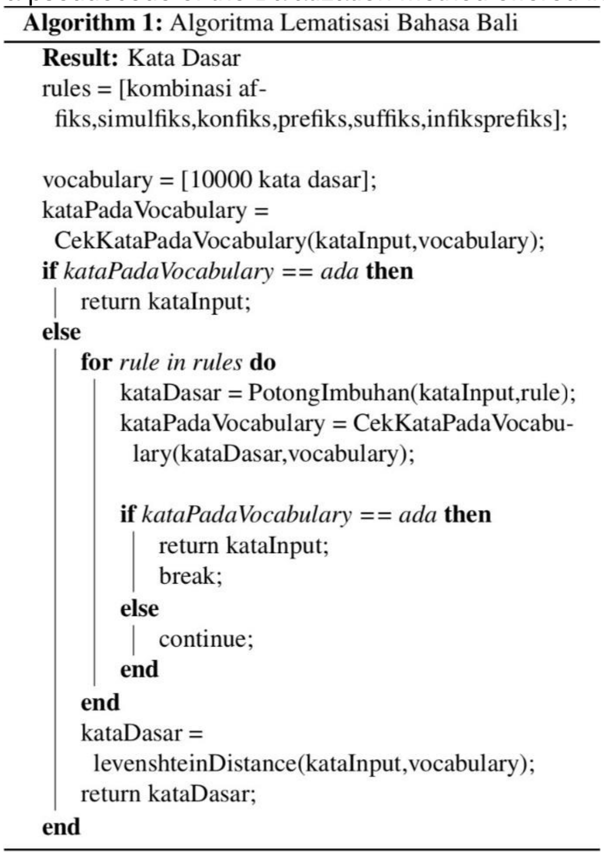
Contains an explanation of the research stages that illustrate the logical sequence to get the research output in line with the expectations and system overview. If there are images and tables, it should be presented with table and Figure names which are accompanied by sequence numbers. In the manuscript, number citations consecutively in square brackets [11], also number tables and figures consecutively as showed in Table 1 and Figure 1.

3.4. Design System

In this study, using a combination of rule-based and lexicon-based for the process of word tightening in the Balinese language. If there are conditions where the base word is not found, it will be handled using the Levenshtein Distance algorithm. The following is a system process flow.

a. Enter the word.
b. Check whether there is a vocabulary, if there is then input is the basic word, if not, do steps 3-4.
c. Check whether the input word qualifies for leaching using affix combination rules. If it meets the requirements then do a preaching affix, then check the vocabulary, if found the right result, then return it as a result. If it doesn't meet the requirements or isn't found in the vocabulary then go to the next step.
d. Check whether the input word meets the requirements to be dissolved using the simulations fixation. If it meets the requirements then do a preaching affix, then check the vocabulary, if it is found that is suitable then return it as a result. If it does not meet the requirements or is not found in the vocabulary then go to the next stage.
e. Check whether the input word meets the requirements to be dissolved using the confix combination rule. If it meets the requirements then do a preaching affix, then check the vocabulary, if found the right result, then return it as a result. If it doesn't meet the requirements or isn't found in the vocabulary then go to the next step.
f. Check whether the input word qualifies for leaching using the prefix combination rule. If it meets the requirements then do a preaching affix, then check the vocabulary, if found the right result, then return it as a result. If it doesn't meet the requirements or isn't found in the vocabulary then go to the next step.
g. Check whether the input word meets the requirements to be dissolved using suffix combination rules. If it meets the requirements then do a preaching affix, then check the vocabulary, if found the right result, then return it as a result. If it doesn't meet the requirements or isn't found in the vocabulary then go to the next step.
h. Check whether the input word qualifies for leaching using the infix combination rule. If it meets the requirements then do a preaching affix, then check the vocabulary, if found the right result, then return it as a result. If it doesn't meet the requirements or isn't found in the vocabulary then go to the next step.
i. Match input with all words found in the vocabulary using the Leven-shtein Distance algorithm. Search for the word with the smallest calculation distance from the input. Return results as the basis for recommendations.

The following image is a pseudocode of the Balatization method offered in this study.



```
Algorithm 1: Algoritma Lematisasi Bahasa Bali
Result: Kata Dasar
rules = [kombinasi af-
  fiks,simulfiks,konfiks,prefiks,suffiks,infiksprefiks];

vocabulary = [10000 kata dasar];
kataPadaVocabulary =
  CekKataPadaVocabulary(kataInput,vocabulary);
if kataPadaVocabulary == ada then
    return kataInput;
else
    for rule in rules do
        kataDasar = PotongImbuhan(kataInput,rule);
        kataPadaVocabulary = CekKataPadaVocabu-
          lary(kataDasar,vocabulary);

        if kataPadaVocabulary == ada then
            return kataInput;
            break;
        else
            continue;
        end
    end
    kataDasar =
      levenshteinDistance(kataInput,vocabulary);
    return kataDasar;
end
```

Figure 1. Pseudocode Lematization

3.5. Calculate Accuracy

To calculate the results of the given lipidisation accuracy, the authors use equation 1.

$$s = {}^{\%}\& \times 100 \qquad (2)$$

In equation 1, s is the accuracy of lipatization. t is the number of words that have been successful in being properly diluted, n is the total number of words.

To prove the proposed method can provide optimal results, the authors do the testing by comparing the results given by the system against the baseline provided. The author uses 250 sentences as a baseline where each sentence contains words from 4 to 23 words per sentence (an average of 9 words per sentence).

4. Result and Discussion

To prove the proposed method works optimally, the authors test by comparing the results given by the system against the baseline provided. The author uses 250 sentences as test data, each of which contains words ranging from 4 to 23 words per sentence (an average of 9 words per sentence). Table 7 provides some examples of test data used and the results provided. Bold words indicate that the word is experiencing addictions. Test data is collected from story documents and Balinese news. From tests conducted on 250 test sentences, an accuracy of 96.01% is obtained when using Levenshtein Distance and 93.65% when do not use Levenshtein Distance. Figure 1 shows a comparison graph of the accuracy of the lipatization method with Levenshtein Distance and without Levenshtein Distance.

The graph shows the lematization method with Levenshtein Distance obtaining a higher accuracy value compared to the lematization method without using Levenshtein Distance. The failure experienced by this method is to find the basic words of a word because the order of rules used is incorrect. A word can be successfully written by a rule because it meets the conditions of the rule and provides a return value for a word, but the word given is not the correct answer.

For example the word “ajaka” (Indonesian: invited) has the basic word “invite”, but because there is a rule to melt prefix a- so that it produces the word “jaka” (Indonesian: palm tree leaves) where the word exists in vocabulary, so the results of the organization are wrong. This shows that the order of affix leaching rules is very influential on the results given. This problem can be solved by labeling words, so that the word “ajaka” which acts as a predicate is not possible to produce an output in the form of “object”. Behind the addition of Levenshtein Distance which causes increased accuracy, Levenshtein Distance also increases the computation time for longer because input words that do not get the right rules will check all words in the vocabulary.

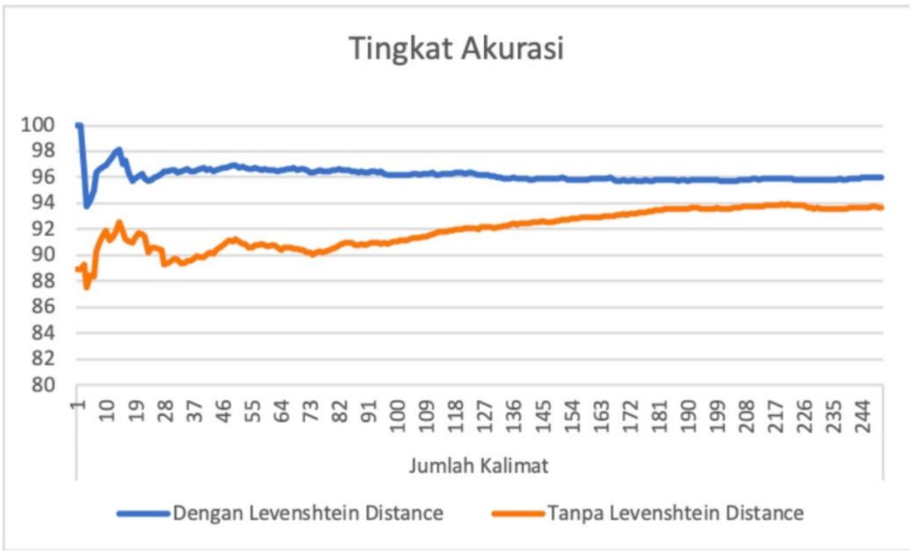


Figure 2. Comparison of testing accuracy of the lemmatization method

Table 7. Example od Test Data

No	Kalimat Input	Kalimat Hasil
1	ada ***katuturan*** satur i siap selem ***ngelah pianak pepitu***	ada ***tutur*** satua i siap selem ***gelah anak pitu***
2	jero meong tiang mriki jagi ***maembon*** mawinan tiang ***madue pianak pianak*** kari alit	jero meong tiang mriki jagi ***embon*** mawinan tiang due ***anak anak*** kari alit
3	tiang baang ***kampidne*** me ***pianakne*** len milu ***mesaut***	tiang baang ***kampid*** me ***panak*** len milu ***saut***
4	***ningeh*** tutur meng kuuke teken ***pianak pianakne*** buka keto lantas i siap selem ***nundunin pianak pi-anakne***	***dingeh*** tutur meng kuuk teken ***anak panak*** buka keto lantas i siap selem ***dundun anak panak***
5	ento meng kuuk nagih ***ngamah*** iraga	ento meng kuuk nagih ***amah*** iraga
6	tunden ia ***ngubuhin*** cai ***nganti*** tumbuh bulu	tunden ia ***ubuh*** cai ***kanti*** tumbuh bulu
7	joh ***pakeberne*** lantas ***ngenceg*** duur batune	joh ***keber*** lantas ***enceg*** duur batun
8	i sugih ***jumahne nyeksek baasne*** maan latah dadua	i sugih ***jumah seksek baas*** maan latah dadua
9	mara ***kedenga*** bek ***limane misi*** mas teken selaka	mara ***kedeng*** bek ***lima isi*** mas teken selaka
10	***anake*** bengong ***mabalih*** baan ***kuatne*** i ubuh	***anak*** bengong ***balih*** baan ***kuat*** i ubuh

5. Conclusion

From the research conducted it can be concluded that the method of lematization by utilizing the Leven- shtein Distance algorithm has better performance compared to the method of lipatization without using the Levenshtein Distance algorithm with an accuracy of 96.01%. Order rule used very influential on the results of lipatization.

The future suggestion for this research is to add the word labeling process so that incorrect results caused by incorrect rules can be avoided.

References

[1] Kementerian Pendidikan dan Kebudayaan. *Badan Bahasa Petakan 652 Bahasa Daerah di Indonesia*. 2018. URL: https://www.kemdikbud.go.id/main/blog/2018/07/badan- bahasa - petakan - 652 - bahasa - daerah - di-indonesia (visited on 07/24/2018).

[2] Abdul Chaer. *Linguistik umum*. Rineka Cipta Jakarta, Indonesia, 1994.

[3] Gorys Keraf. "Diksi dan Gaya Bahasa. 1984". In: *Jakarta: PT Gramedia* (1984).

[4] Gusti Ngurah Mega Nata and Putu Pande Yudias- tra. "Stemming teks sor-singgih Bahasa Bali". In: *E- Proceedings KNS&I STIKOM Bali* (2017), pp. 608– 612.

[5] Made Agus Putra Subali and Chastine Fatichah. "Kombinasi Metode Rule-Based dan N-Gram Stem- ming untuk Mengenali Stemmer Bahasa Bali". In: *Jurnal Teknologi Informasi dan Ilmu Komputer* 6.2 (2019), pp. 219–228.

[6] Derwin Suhartono. "Lemmatization technique in bahasa: Indonesian". In: *Journal of Software* 9.5 (2014), p. 1203.

[7] Yusup Miftahuddin, Jasman Pardede, and Renita Dewi. "Penerapan Algoritma Lemmatization pada Dokumen Bahasa Indonesia". In: *MIND Journal* 3.2 (2018), pp. 47–56.

[8] Reina Setiawan et al. "Flexible affix classification for stemming Indonesian Language". In: *2016 13th International Conference on Electrical Engineer- ing/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE. 2016, pp. 1–6.

[9] Agus Zainal Arifin, IPAK Mahendra, and Hen- ning Titi Ciptaningtyas. "Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language". In: *Proceeding of International Conference on Information & Com- munication Technology and Systems (ICTS)*. 2009, pp. 149–158.

[10] Wahyu Hidayat. "Ekstraksi Kata Dasar Secara Berjenjang (Incremental Stemming) Berbasis Atu- ran Morfologi Untuk Teks Berbahasa Indonesia". In: *Jurnal Infotel* 9.2 (2017), pp. 166–171.

[11] Joe ̈l Plisson, Nada Lavrac, Dunja Mladenic, et al. "A rule based approach to word lemmatization". In: *Proceedings of IS-2004* (2004), pp. 83–86.

[12] Uli Fitrianti and Mutammimul Ula. "Implemen- tasi Algoritma Levenshtein Distance dan Algoritma Knuth Morris Pratt pada Aplikasi Asmaul Husna Berbasis Android". In: *Jurnal Sistem Informasi* 1.2 (2017).

[13] Guntur Syahputra and Jijon Raphita Sagala. "Deteksi Tingkat Kemiripan Data Siswa Peserta Ujian Berbasis Komputer Pada Bimbingan Belajar Expert Dengan Menggunakan Metode Levenshtein Distance". In: *Algoritma: Jurnal Ilmu Komputer Dan Informatika* 1.01 (2017).

[14] Guntur Syahputra and Jijon Raphita Sagala. "Anal- isis kinerja algoritma levenshtein distance dalam mendeteksi kemiripan dokumen teks". In: *LOGIK@* 6.2 (2016), pp. 131–143.