EPI International Journal of Engineering
Volume 2, Number 1, February 2019, pp. 46-55
DOI: 10.25042/epi-ije.022019.09

*p*ISSN 2615-5109
*e*ISSN 2621-0541

Autonomy Stemmer Algorithm for Legal and Illegal Affix Detection Use Finite-State Automata Method

Ana Tsalitsatun Ni'mah[a]*, Dwi Ari Suryaningrum[b], Agus Zainal Arifin[c]

[a]Informatics Department, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember. Email: anatsalits@gmail.com
[b]Informatics Department, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember. Email: dwiari.suryaningrum@gmail.com
[c]Informatics Department, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember. Email: agusza@cs.its.ac.id

Abstract

Stemming is the process of separating words from their affixes to get a basic word. Stemming is generally used when preprocessing in text-based applications. Indonesian Stemming has developed research which is divided into two types, namely, stemming without dictionaries and stemming using dictionaries. Stemming without dictionaries has a disadvantage in the results of removal of affixes which are sometimes inappropriate so that it results in over stemming or under stemming, while stemming using dictionaries has a disadvantage during the stemming process which is relatively long and cannot eliminate affixes to compound words. This study proposes a new stemming algorithm without a dictionary that is able to detect legal and illegal affixes in Indonesian using the Finite-State Automata method. The technique used is rule-based Stemmer based on Indonesian language morphology with Regular Expression. Test results were carried out using 118 news documents with 15792 words. The first test results on the autonomy stemmer algorithm obtain the correct word which amounts to 10449 of the total number of words processed, which means getting an average accuracy of 66%. The second test results on the autonomy stemmer algorithm get the results of the average speed of 0.0051 seconds. The third test result is being able to do the elimination of affixes to compound words.

Keywords: Autonomy stemmer; confix stripping stemmer; finite state method; porter Indonesian language; regular expression; stemming

1. Introduction

Stemming is a basic word separation process from its affixes based on the morphological mapping of various variants of affixed words [1]. Stemming in informatics is used in text processing which is generally when searching for information, translations, etc. [2-4]. Morphology is a very important thing in stemming algorithms. Morphology is a process of forming words [2, 5, 6]. Words that experience morphology in Indonesian is affixed words, rephrase words and compound words. English only has one type of affix word, suffix, whereas in the morphology of the Indonesian language there are several types of affixes, namely: prefix, insertion, suffix, combined prefix ending, and foreign affixes.

The Indonesian Stemming Algorithm was first developed by Nazief and Adriani [7]. The Stemming algorithm is called Confix Stripping (CS). The Confix Stripping (CS) algorithm performs the affix decapitation process by referring to the Indonesian dictionary at each step. The algorithm was developed again by Arifin and Setiono [8]. The development of this algorithm simplifies the affixing rule. Tala [1] conducted research on stemming Indonesian without using a dictionary [1]. The stemming algorithm refers to the Porter algorithm, that algorithm is a stemming algorithm used in English, Tala applies the algorithm to Indonesian. Furthermore, Putra et al. [7] doing research on various types of stemming in Indonesian. The study presents several Indonesian languages stemming algorithms, including Confix Stripping (Nazief and Adriani), Modified Confix Stripping (Arifin and Setiono), Vega etc. The research was conducted again on Confix Stripping by Adriani et al. [9]. The results of the study concluded that the most stable algorithm for stemming Indonesian at that time was Confix Stripping. Arifin, et al. [10] developed the Confix Stripping Algorithm. The algorithm is named Enhanced Confix Stripping Stemmer (ECS). ECS modifies the rules in Confix Stripping. Apart from some of there are still many more studies on Indonesian stemming [2, 3, 5, 6].

There are some researches on stemming, including carrying out the affixing process with the Brute Force technique/table lookup or stemming based dictionary, there are also some that use affix removal techniques. The basic research of Indonesian stemming with affix removal technique is the Indonesian Porter stemming algorithm [1], while the research basis for a dictionary-based stemming technique is Confix Stripping [7]. The development of the Porter stemming algorithm for Indonesian has been compared with the Confix Stripping stemming algorithm [12]. The study, developed by Agusta [12], has mentioned several comparisons. Among other

*Corresponding author. Tel.: +62 857 3612 4000
Jl. Raya Telang - Kamal
Bangkalan, Indonesia Postcode 69162

things, the Porter Stemming Algorithm process takes a shorter time than the Confix Stripping algorithm, the Porter stemming algorithm has a smaller accuracy compared to Confix Stripping Algorithm with an average difference of 20%. The process of Confix Stripping dictionary stemming algorithm is very influential on stemming results, the more complete the dictionary is used, the more accurate stemming results will be. According to Tahitoe and Purwitasari's research [11] the Enhanced Confix Stripping algorithm that they developed still lacked that is unable to stem compound words. According to Widjaja and Hansun research [6], the Indonesian Porter stemming algorithm also has its drawbacks, namely over stemming and under stemming. This, of course, will reduce the efficiency and performance of the stemming algorithm [6].

This study proposes a new stemming algorithm without a dictionary that is able to detect legal and illegal affixes in Indonesian using the Finite-State Automata method. The purpose of this study is to get stemming results that have high accuracy and speed by not relying on dictionaries during the removal process so that they can do the elimination of affixes to compound words.

2. Method

2.1. *Finite-state automata and regular expression*

Automata is a process sequence that automatically receives input and produces discrete output. The input circuit received is a string or language that is recognized by automata. If the input circuit is received and recognized, the engine produces output [5].

Finite-State Machine is an abstract machine in the form of mathematical theory by getting discrete outputs and inputs during the process that can recognize the simplest language (regular language) and can be implemented significantly where the system in an internal configuration called a state [5]. FSM works by means of the machine reading the input memory in the form of a tape, which is 1 character at a time (from left to right) using a read head which is controlled by a finite state control box where there are a number of finite states on the machine. The FSM is always in a condition called the initial state when starting to read a tape. State changes occur on the machine when the next character is read. When the head arrives at the end of the tape and the condition encountered is the final state, then the string contained on the tape is said to be received by FSM (Strings are the property of the language if the FSM language is accepted). FSM is stated simply by the regular expression language.

Regular expressions or often referred to as Regex are formulas for searching patterns of sentences or strings. Regex is very helpful in finding sentence patterns. So experiments with all possible sentence patterns need not be done. Regular expressions are generally used by many word processors or text editors and other tools to search for and manipulate sentences based on a certain pattern. At low levels, the regex can search for a word fragment. At a high level, the regex is able to control the data. Both searching, deleting and changing [5].

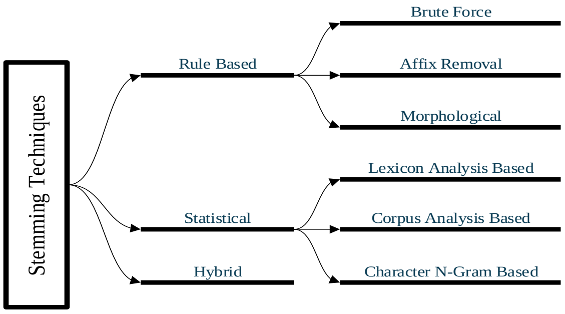

Figure 1. Stemming techniques

2.2. *Stemming*

The Stemming method can be classified into 3 techniques, namely [13]: rule-based, statistical, and hybrid can be seen in Fig. 1.

2.4.1. *Rule-based stemmer*

This stemmer is a more accurate stemmer compared to other stemmer techniques because this technique pays attention to the language rules in the stemming process. Stemmer is categorized into 3, namely: Brute Force method, Affix Removal method, and morphology method.

The Brute Force method is also known as the Table lookup techniques which is a stemming process carried out on the basis of a search table that contains a collection of basic words or basic word dictionaries.

Affix removal method is to delete the ending or prefix of words so that they turn them into basic words. Most stemmers currently used use this type of approach. The Affix removal method is based on two principles namely iteration and the other is the longest match [14]. This method starts at the end of the word and works towards the beginning. No more than one process is permitted in one class the deletion process. Some stemming algorithms that use this approach are Lovins and MF Porter [14]. In Basaha Indonesia, the stemming algorithm that uses this technique is the Indonesian Porter. The recharge process only occurs once every time the process.

Morphological methods are stemming techniques that use the language morphology rules in the process of eliminating affixes. This method allows the simultaneous removal of affixes in one deletion process, in contrast to the affix removal method.

2.4.2. *Statistical stemmer*

This Lexicon technique is a technique that groups words according to similarity. The process of stemming is done by finding the closest distance from the meaning of the word that has been collected. Corpus techniques are similar to the Lexicon Technique, the difference is that if Lexicon collects words based on meaning, the corpus collects morphologically or similarly written words.

The N-gram method was coined in 1974 by Adamson and Boreham. N-grams come from grams that are more than 2 or digram. A digram is a pair of consecutive letters [14]. This approach, linking the pair's words on the basis of the unique digram both have. To calculate this measurement using the Dice coefficient. For example, the term information and informative can enter into grams as follows [14]:

information => in nf fo or rm ma at ti io on
unique digrams = in nf fo or rm ma at ti io on
informative => in nf fo or rm ma at ti iv ve
unique digrams = in nf fo or rm ma at ti iv ve

Thus, "information" has ten digrams, all of which are unique, and "informative" also have ten digrams, all of which are unique. Two eight digram sharing words are unique: in, nf, fo, or, rm, ma, at, and ti. After the digram is unique for the pairs of words that have been identified and counted, the size of the similarity based on them is calculated. The similarity measure used is the Dice coefficient, which is expressed in Eq. 1.

$$S = \frac{2C}{A+B} \qquad (1)$$

where A is the number of digrams unique in the first word, B the number of digram is unique in seconds, and C digram number is unique which is shared by A and B. For the example above, the dice coefficient will be the same (2 x 8) / (10 + 10) = 0.80. The size of the similarity is determined for all terms in the database. Once the similarities are calculated for all the words their partners are grouped as groups. The Dice Coefficient value gives us a clue that the basic word for this pair is in the first 8 digrams [14].

2.4.3. Hybrid stemmer

This Hybrid technique is a technique that combines several techniques. For example, the lookup table technique is combined with affix removal or something else. The stemming algorithm that uses this technique is Confix Stripping. Confix Stripping removes affixes based on Indonesian morphology and matches them into an Indonesian language dictionary table with the deletion process adjusted to an affix removal rule, one by one.

2.3. Indonesian morphology

The technique used in the Algorithm of this study is based on the word grammar contained in the Indonesian Grammar guidebook from the Ministry of Education and Culture [15]. The basic prefix is the most basic prefix and has not experienced developer. Consists of 6 affixes namely meng, peng-, ber-, di-, ter- and se-. There are some basic prefixes that have developed if strung together with a few basic words with several rules.

Meng- and peng- becomes me- and pe- if coupled with a basic word that starts with the letter /r, l, m, n, w, y, ng, ny/. Example :

- Meng-/peng- + rawat : care (merawat), nurse (perawat)
- Meng-/peng- + lamar : apply (melamar), applicant (pelamar)
- Meng-/peng- + minum : drink (meminum), drinker (peminum)
- Meng-…-i + nama : name (menamai)
- Peng-…-an + nama : naming (penamaan)
- Meng-…-i + waris : inherit (mewarisi)
- Peng- + waris : heir (pewaris)
- Meng-…-kan + yakin : convincing (meyakinkan)
- Peng-…-an + yakin : confidence (peyakinan)
- Meng- + nganga : gaping (menganga)

Meng- and peng- change into mem- and pem- if coupled with basic words that begin with the letters /b, f, v, pr/. Exceptions to the basic words beginning with /pr/ if they meet peng- then letter p melts, e.g. peng- + proses : pemroses. Example :

- Meng-/peng - + bawa : carry (membawa), carrier (pembawa)
- Meng-/peng - + fitnah : slander (memfitnah), slander (pemfitnah)
- Meng-/peng - + vonis : sentencing (memvonis), verdict (pemvonis)
- Meng- + produk + i : produce (memproduksi)

Meng- and peng- change into men- and pen- if coupled with basic word that begin with letter /d, c, j, z, s(consonant), t(consonant)/. Example :

- Meng-/peng - + dakwah : preaching (mendakwah), preacher (pendakwah)
- Meng-/peng - + curi : steal (mencuri), thief (pencuri)
- meN-/peN- + jual : sell (menjual), seller (penjual)
- Meng-…-i + ziarah : visit (menziarahi)
- Peng- + ziarah : pilgrims (penziarah)
- Meng-…-i + syukur : grateful (mensyukuri)
- Peng-…-an + syukur : thanksful (pensyukuran)

Meng- and peng- remain as meng- and peng- if coupled with basic word that begin with letters /k(konsonan), g, h, kh, and vokal/. Example :

- meng-/peng- + ganggu : disturb (mengganggu), disturber (penggangu)
- meng-/peng- + hasut : provoke (menghasut), provoker (penghasut)
- meng-/peng- + khitan : circumcise (mengkhitan), circumcision (pengkhitan)
- meng-/peng- + atur : arrange (mengatur), arranger (pengatur)
- meng-/peng- + ekor : follow (mengekor), follower (pengekor)
- meng-/peng- + inap : lodge (menginap), lodger (penginap)
- meng-…-i + obat : treat (mengobati)
- peng-…-an + obat : treatment (pengobatan)
- meng-/peng - + ukur : measure (mengukur), measure (pengukur)

Meng- and peng- change into meny- and peny- if coupled with basic word that begin with letters /s(vokal)/ and that letter s melts. Example :

- meng-/peng- + sayang : love (menyayang), loving (penyayang)
- meng-/peng- + sapa : greet (menyapa), greeter (penyapa)
- meng-/peng- + sulap : juggle (menyulap), juggler (penyulap)
- meng-/peng- + sikat : brushing (menyikat), brush (penyikat)

Meng- and peng- change into menge- and penge- if coupled with basic word which consists of only one syllable. Example :

- meng-/peng- + cat : paint (mengecat), painter (pengecat)
- meng-/peng- + bom : bomb (mengebom), bomber (pengebom)
- meng-/peng- + las : weld (mengelas), welder (pengelas)
- meng-/peng- + pel : swab (mengepel), swabber (pengepel)
- meng-/peng - + cek : check (mengecek), checker (pengecek)
- meng-/peng- + tes : test (mengetes), tester(pengetes)

Meng- and peng- change into mem- and pem- if coupled with basic word that begin with letters /p(vokal)/ and that letter p melts. Example, meng-/peng- + pukul : hit (memukul), hitter (pemukul).

Meng- and peng- remain as meng- and peng- if coupled with basic word that begins with letters /k(vocal)/ but the letter k melts. Example, meng-/peng- + kikis : scrape (mengikis), scraper (pengikis).

Meng- and peng- change into men- and pen- if coupled with basic word that begins with letters /t(vocal)/ and the letter t melts. Example, meng-/peng- + tukar : swap (menukar), swapper (penukar).

Ber- change into be- if coupled with basic word that begins with letters /r(vocal)/. Example, ber- + regu : team (beregu). Ber- change into bel- if coupled with the basic word /ajar/. Example, ber- + ajar : study (belajar). Ber- remain as ber- if coupled with basic word that begins with letters /all consonants except r/. Example, ber- + canda : joking (bercanda).

Per- remain as per- if coupled with basic word that begins with letter /consonant/. Example, per- + tanda : sign (pertanda). Per- change into pel- if coupled with the basic word /ajar/. Example, per- + ajar : student (pelajar). Per- change into pe- if coupled with basic word that begins with letter /r(vocal), tani, tinju/. Exceptions if per- is added to the –an suffix, then per- remain as per-. Example, per- + tani : farmer (petani).

Ter- change into te- if coupled with basic word /r(vokal)/. Example, ter- + rasa : feel (terasa). Ter- remain as ter- if coupled with all consonant or vocal letter /(vocal), (consonant)/. Example, ter- + indah : most beautiful (terindah).

The basic suffix is the most basic suffix. There is no development or change as in the prefix. The suffix is only three, namely -an, -kan, and -i.

The rules for the combined prefix and suffix are explained in a legal and illegal table Affix, can be seen in Table 1 [15]. The table describes the prefix rules and their derivatives combined with suffix suffixes. There are some that are the rules of Confix Stripping, other rules are obtained from Indonesian grammar from the 2015 graduation [15].

In line with the rules, a combination of words or commonly called compound words, including special terms, the elements are written separately. However, if the combination of words gets a prefix and suffix at the same time, the combined elements of the word are written in a series. The basic form of responsibility also must be written a series if you get the prefix and suffix at once. Therefore, writing the correct form of the word is accountability, not responsibility, accountability, or accountability. Combined payoffs on phrases have the same rules as affixes to compound words.

Table 1. Illegal and legal affix

Prefix		Suffix	
		Illegal	Legal
Ber	Ber	-i	-kan, -an
	Berke	-kan, -i	-an
Me	Me	-an	-kan, -i
	Mem	-an	-kan, -i
	Men	-an	-kan, -i
	Meng	-an	-kan, -i
	Menge	-an, -i	-kan
	Meny	-an	-kan, -i
Pe	Pe	-kan, -i	-an
	Pem	-kan, -i	-an
	Pen	-kan, -i	-an
	Penge	-kan, -i	-an
	Peng	-kan, -i	-an
	Peny	-kan, -i	-an
Per	Per	-	-kan, -an, -i
	Perse	-kan, -i	-an
	Pember	-kan, -i	-an
	Memper	-an	-kan, -i
	Diper	-an	-kan, -i
	Di	-an	-kan, -i
Ke	Ke	-kan, -i	-an
	Keter	-kan, -i	-an
	Kese	-kan, -i	-an
	Se	-kan, -i	-an
	Ter	-an	-kan, -i

2.4. *Indonesian stemming*

Indonesian stemming was first developed by Nazief and Adriani in 1996. The developed Stemming used a checking technique on the basic word dictionary in each process of removing the affix. Furthermore, there are also those who develop Indonesian stemming without using a dictionary in the process of eliminating the affix, the research was carried out by Tala in 2005. Stemming without the dictionary only uses affix removal techniques as Porter did.

2.4.1. *Nazief and Adriani (confix stripping)*

Nazief and Adriani [9] stemming algorithms were developed based on dictionary lookup table techniques of basic words and Indonesian language morphological rules which group affixes into prefixes (prefixes), insertions (infix), suffixes (suffixes) and combined prefixes (confixes). This algorithm uses a dictionary of basic words and supports recoding, namely the rearrangement of words that experience an excessive stemming process [11].

The Indonesian morphology rules used in the Confix Stripping algorithm are grouped into the following categories [11]:

a) Inflection suffixes are groups of endings that do not change the basic word form.
   1) Particle (P), which includes “-lah”, “-kah”, “-tah”, and “-pun”.
   2) Possessive Pronoun (PP), including “-ku” , “- mu”, and “-nya”.
b) Derivation Suffixes (DS) is a collection of original Indonesian endings which are directly added to the basic words, namely the ending “-i”, “-kan”, dan “-an”.

Table 2. Illegal affix of confix stripping

Prefix	Suffix
Be-	-i
Di-	-an
Ke-	-i, -kan
Me-	-an
Se-	-i, -kan

c) Derivation Prefixes (DP) is a collection of prefixes that can be directly given to pure base words, or to basic words that have received additions up to 2 prefixes.
   1) Prefixes that can be morphological ("me-", "be-", "pe-", and "te-")
   2) Prefixes that are not morphological ("di-", "ke-" and "se-")

These rules are used in the process of stemming algorithms by Nazief and Adriani. But not all composite prefixes are allowed by Confix Stripping [9]. Some affix combinations that are not allowed can be seen in Table 2.

The Confix Stripping algorithm has the following processes [16]:

a) Search for words that will be in the dictionary system. If it's found, it is assumed that the word is root word. Then the algorithm stops.
b) Inflection Suffixes ("-lah", "-kah", "-ku", "-mu", or "-nya") are discarded. If it is in the form of particles ("-lah", "-kah", "-tah" or "-pun") then this step is repeated again to delete obsessive pronouns ("-ku", "-mu", or "-nya"), if there is.
c) Remove Derivation Suffixes ("-i", "-an" or "-kan"). If the word is found in the dictionary, the algorithm stops. If not then go to step c1. Inflective affixes always in sequence. This algorithm first removes the inflection particle (P) suffix {"-kah", "-lah", "-tah", atau "-pun"}, and then each suffix change the ownership pronoun {"-ku", "-mu", or "-nya"}.
   1) If "-an" has been deleted and the last letter of the words Is "–k", the "-k" is also deleted. If the word is found in the dictionary, the algorithm stops. If not found then do step c2.
   2) The deleted suffix ("-i", "-an" or "-kan") is returned, proceed to step d.
d) Derivation Prefix is removed. If in step 3 there is a suffix that is deleted then go to step d2.
   1) Check prefix-suffix combination tables that are not permitted. If it is found, the algorithm stops, if it does not go to step 4b.
   2) For i = 1 to 3, specify the type of prefix then delete the prefix. If the root word has not been found, do step 5, if the algorithm has stopped. Note: if the second prefix equals the first prefix of the stop algorithm.
e) Recoding.
f) If all steps have been completed but it does not work, the initial word is assumed to be root word. Process complete.

After a number of experiments and analyses, several words that could not be stemmed using Confiz Stripping Stemmer were conducted. Analysis by the Enhanced Confix Stripping Stemmer algorithm for words that failed to be stemmed as follows:

a) Lack of decapitation of the word prefix rules in the format "mem+p...", "men+s...", and "peng+k...". This happened to word "mempromosikan", "memproteksi", "mensyaratkan", "mensyukuri", dan "pengkajian".
b) The lack of relevance of the rules for the decapitation of the word prefix in the format "menge+basic word" and "penge+basic word", as in the words "mengerem" and "pengeboman".
c) There are elements in some basic words that resemble an affix. Words like "pelanggan", "perpolitikan", and "pelaku" fail to be stemmed because the end of "-an", "-kan" and "-ku" should not be eliminated.

To correct the errors above, the ECS Stemmer algorithm performs several improvements as follow:

a) Make modifications and additions to the rules.
b) Add an additional algorithm to overcome end-chopping errors that should not be done. This algorithm is called Returns Suffix loop, and is done if the recoding process fails.
c) Return all prefixes that have been removed before, resulting in the word model as follows: [DP+[DP+[DP]]] + basic word. Decapitation of the prefix is followed by a search process in the dictionary then performed on the word that has been returned to that model.
d) Return the suffix according to the sequence of models in Indonesian. This means that the return starts from DS ("-i", "-kan", "-an"), then PP("-ku", "-mu", "-nya"), and finally P ("-lah", "-kah", "-tah", "-pun"). For each return, do steps 3) to 5) below. Especially for the "-kan" suffix, the first return starts with "k", then it continues with "an".
e) Check in the basic word dictionary. If found, the process is stopped. If it fails, then do the prefix process based on the rules.
f) Perform recoding if needed.
g) If checking in the base word dictionary still fails after recoding, then the omitted prefixes are returned again.

This algorithm still has several disadvantages that must be corrected, i.e.:

- Elimination of affixes to compound words that have combined additions.
- Over stemming and under stemming
- The speed of the stemming process

2.4.2. Ledy Agusta (porter)

According to Milutinovich, Porter's stemmer algorithm was first discovered in 1979 by Martin Porter in the computer lab. Porter stemming algorithm is a process of removing English morphology suffixes and inflections of words. The Porter algorithm, which was originally developed for English, was developed for Indonesian by Frakes [6]. Porter's stemmer works well in English [17]. Porter stemmer has become the standard stemmer for English and the same stemming approach has been adopted for other languages i.e. Romance (French, Italian, Portuguese and Spanish), Germanic (Dutch and German), Scandinavian languages (Danish, Norwegian and Swedish), Finnish and Russia [18]. Porter stemmer is a linear stemmer step, applying morphological rules sequentially allows the elimination of affixes gradually [19].

Table 3. Regular expression of autonomy stemmer

<table>
<tr><th>No</th><th>Prefix</th><th>Suffix</th><th>Deleted</th><th>Replacement</th><th>Example word</th></tr>
<tr><td>1</td><td>Belajar</td><td>[klt]ah|pun</td><td>Bel</td><td>-</td><td>Learn (Belajarkah)</td></tr>
<tr><td>2</td><td>Ber[eo][a-z]{3,}</td><td>[klt]ah|pun</td><td>Be</td><td>-</td><td>Swim (Berenang)</td></tr>
<tr><td>3</td><td>Ber[^eo][a-z]{3,}</td><td>Kan[km]u|nya|kah|pun</td><td>Ber</td><td>-</td><td>Together (Bersama)</td></tr>
<tr><td>4</td><td>Be[^aiueor][e][r][a-z]{2,}</td><td>[klt]ah|pun</td><td>Be</td><td>-</td><td>Work (Bekerja)</td></tr>
<tr><td>5</td><td>Keber[^e][a-z]{3,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Keber</td><td>-</td><td>Success (Keberhasilan)</td></tr>
<tr><td>6</td><td>Mempelajar</td><td>I|[klt]ah|pun</td><td>Mempel</td><td>-</td><td>Learn (Mempelajari)</td></tr>
<tr><td>7</td><td>Memper[a-z]{3,}</td><td>Kan|an|[klt]ah|pun</td><td>Memper</td><td>-</td><td>Enslave (Memperbudakkan)</td></tr>
<tr><td>8</td><td>Mempe[^aiueor][e][r][a-z]{2,}</td><td>Kan|an|[km]u|nya|[klt]ah|pun</td><td>Mempe</td><td>-</td><td>Employ (Mempekerjakan)</td></tr>
<tr><td>9</td><td>Menyanyi</td><td>Kan|nya|[klt]ah|pun</td><td>Me</td><td>-</td><td>Sing (Menyanyi)</td></tr>
<tr><td>10</td><td>Meny[auieo][a-z]{2,}</td><td>Kan|i|[km]u|nya|[klt]ah|pun</td><td>Meny</td><td>S</td><td>Sweep away (Menyapu)</td></tr>
<tr><td>11</td><td>Menge[bcks][a-z]{3,}</td><td>Kan|nya|[klt]ah|pun</td><td>Menge</td><td>-</td><td>Validate (Mengesahkan)</td></tr>
<tr><td>12</td><td>Meng[ioeu][a-z]{3,}</td><td>Kan|i|[km]u|nya|[klt]ah|pun</td><td>Meng</td><td>K</td><td>Reduce (Mengurangi)</td></tr>
<tr><td>13</td><td>Men[aiueo][a-z]{3,}</td><td>Kan|i|[km]u|nya|[klt]ah|pun</td><td>Men</td><td>T</td><td>Dance (Menari)</td></tr>
<tr><td>14</td><td>Mem[aiueo][a-z]{3,}</td><td>Kan|i|[km]u|nya|[klt]ah|pun</td><td>Mem</td><td>P</td><td>Cut it off (Memotongkan)</td></tr>
<tr><td>15</td><td>Memper[a-z]{4,}|diper[a-z]{4,}|meng[akgh][a-z]{3,}|men[cdjt][a-z]{3,}|mem[bfvp][a-z]{3,}|me[lrwy][a-z]{3,}|di[a-z]{3,}</td><td>Kan|i|[km]u|nya|[klt]ah|pun</td><td>Memper|diper|meng|men|mem|me|di</td><td>-</td><td>Use (Mempergunakan), Used (Dipergunakan), Expect (Mengharapkan), Get (Mendapatkan), Justify (Membenarkan), Revealed (Mewahyukan), Delivered (Diantarkan)</td></tr>
<tr><td>16</td><td>Penge[bcks][a-z]{3,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Pe|penge</td><td>-</td><td>Singer (Penyanyi), Verifier (Pengesah), Clicker (Pengeklik), Bomber (Pengebom), Checker (Pengecek)</td></tr>
<tr><td>17</td><td>Pe[m][a-z]{4,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Pe</td><td>-</td><td>Entry (Pemasukan)</td></tr>
<tr><td>18</td><td>Peny[auieo][a-z]{2,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Peny</td><td>S</td><td>Poet (Penyair)</td></tr>
<tr><td>19</td><td>Peng[ioeu][a-z]{3,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Peng</td><td>K</td><td>Corrector (Pengoreksi)</td></tr>
<tr><td>20</td><td>Pen[aiueo][a-z]{3,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Pen</td><td>T</td><td>Dancer (Penari)</td></tr>
<tr><td>21</td><td>Pem[aiueo][a-z]{3,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Pem</td><td>P</td><td>Cutting (Pemotongan)</td></tr>
<tr><td>22</td><td>(ke*se[a-z]{10,}|keter[a-z]{3,}|ke[a-z]{3,}|peng[akgh][a-z]{3,}|pen[cdjt][a-z]{3,}|pem[bfvp][a-z]{3,}|pe[lrwyp][a-z]{5,}</td><td>An|[km]u|nya|[klt]ah|pun</td><td>Kese|se|keter|ke|peng|pen|pem|pe</td><td>-</td><td>Limitations (Keterbatasan), Negligence (Kelalaian), Barrier (Penghalang), Creation (Penciptaan), Generation (Pembangkitan), Throwing (Pelemparan)</td></tr>
</table>

The steps of this algorithm are as follows [12]:

a) Remove particle.
b) Remove obsessive pronoun.
c) Remove the first prefix. If it doesn't exist, then proceed to step d. Whereas if there is, then proceed to step e.
d) Delete the second prefix, then proceed to step f.
e) Delete suffix, if it is not found, the word is assumed to be root word. Whereas if found, then proceed to step g.
f) Remove suffix. Then the final word is assumed to be a basic word.
g) Remove the second prefix. Then the final word is assumed to be a basic word.

This algorithm still has several disadvantages, i.e. Over stemming and Under stemming.

2.4.3. *Autonomy stemmer*

The ECS Stemmer algorithm uses the basic word lookup table technique and the removal process using affix removal techniques. The Indonesian Porter algorithm only uses affix removal techniques. Both of them use Indonesian language morphology as the basis for deletion. The modification that we propose is to use the Indonesian Language morphology rules as a reference for eliminating affixes by applying them to the Regular Language Expression. The steps of the proposed stemming algorithm can be seen in Fig. 2.

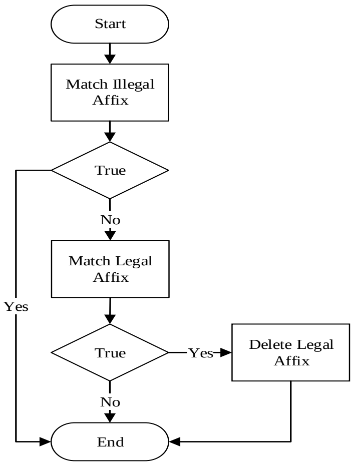

Figure 2. Flowchart autonomy stemmer

With the following information:

a) Analyzing the words that will be stemmed, if the word character is ≤ 3, then the process is complete.
b) Analyzing the words that will be stemmed, if it has an illegal compounding according to Table 1, the process is complete.
c) Analyzing the words that will be stemmed, if the word structure matches the regular expression formula in Table 3, then delete the affix.

The following is a description of Table 3:

a) If the word contains learning and or accompanied by particles, then the stemming process is carried out and has a teaching output.
b) If the word contains the prefix character "ber[eo][a-z]{3,}" and the suffix "[klt]ah|pun" which means the prefix meets letters e and o then meets the letters a to z with the number characters of at least 3 and or ending in particles. Then the stemming process is carried out, the output is the suffix prefix and the particle suffix is removed.
c) If the word contains the prefix character "ber[^eo][a-z]{3,}" and the ending "kan[km]u|nya|kah|pun" which means the prefix meets except the letters e and o then meets with letters a to z with a minimum number of characters of 3 and or ending in, substitutes for property and particles. Then the stemming process is carried out, the output is the prefix ber and the ending is deleted.
d) If the word contains the prefix character "be[^aiueor][e][r][a-z]{2,}" and the suffix "[klt]ah|pun" which means the prefix meets except letters a, i, u, e, o, r then meet with the letter e, then meet with the letter r, then meet with the letters a to z with a minimum number of characters 2 and or end of particles. Then the stemming process is carried out the output is the suffix prefix and the particle suffix is removed.
e) if the word contains the prefix character "keber[^e][a-z]{3,}" and the suffix "an|[km]u|nya|[klt]ah|pun" which means the prefix is met except letter e then meets with letters a to z with a minimum number of characters 3 and or ending in or ending –an, substitute for property and particles. Then the stemming process is carried out, the output is the suffix prefix of the succession and the ending is deleted.
f) And for the next number, how to read it is almost the same as the way above.

3. Result and Discussion

This chapter explains the results and discussion in our study. Our experiment is done in several processes, i.e. the first to input 118 news document, then the next step is to do preprocessing on punctuation and conjunctions. After preprocessing is done, the program separates paragraph into the table in every word to do the word stemming process. The amount from this process is 15792 words that will be in the system. The next step of the experiment in this research is to classify the results of the stemming. The following is a complete explanation of the trial process in this study:

3.1. News Document Dataset Input

The dataset used in this experiment is a crawl news document of some 118 online news sites. After preprocessing, the dataset obtained 15792 words stem.

3.2. Classification of Result of Stem Errors and Fixes

The process of classifying the results of errors and stemming improvements in our study uses a manual method by categorizing the types of words in the stem. Our truth is prediction by matching stemming words with the Indonesian language dictionary dataset. If the stemming yield word is not found in the dictionary database, the word trust counts 0. Here are some of the classifications:

Table 4. Example of compound word

Id	Word	Autonomy	ECS	Porter	Correct result
1	Notified (Diberitahukan)	Beritahu	Beritahu	Beritahu	Beritahu
2	Disseminate (Menyebarluaskan)	Sebarluas	Menyebarluaskan	Sebarluas	Sebarluas
3	Pulverization (Penghancurleburan)	Hancurlebur	Penghancurleburan	Nghancurlebur	Hancurlebur
4	Tell (Memberitahukan)	Beritahu	Beritahu	Beritahu	Beritahu
5	Responsibility (Pertanggungjawaban)	Tanggungjawab	Pertanggungjawaban	Rtanggungjawab	Tanggungjawab
6	Sign (Menandatangani)	Tandatangan	Menandatangani	Andatangan	Tandatangan
7	Underline (Menggarisbawahi)	Garisbawah	Menggarisbawahi	Garisbawah	Garisbawah
8	Accountable (Mempertanggungjawabkan)	Tanggungjawab	Mempertanggungjawabkan	Tanggungjawab	Tanggungjawab
9	Multiplied (Dilipatgandakan)	Lipatganda	Dilipatgandakan	Lipatganda	Lipatganda

Table 5. Example over stemming and under stemming

Id	Word	Autonomy	ECS	Porter	Correct Result
1	Policy (Kebijakan)	Bijak	Bija	Bija	Bijak
2	Agreed (Disetujuinya)	Setuju	Disetujui	Setujui	Setuju
3	Perpetrator (Pelaku)	Pelaku	Pela	Laku	Pelaku
4	Legislation (Perundang-undangan)	Undang-undang	Perundang-undangan	Rundang-undang	Undang-undang
5	Country (Negeri)	Negeri	Neger	Negeri	Negeri
6	Economy (Ekonominya)	Ekonomi	Ekonom	Ekonomi	Ekonomi

a) Improvements in stemming compound words

One of our aims to conduct research on modification of stemmer is to correct errors in compound word stem. Some of them are summarized in Table IV. In the table, there are 9 examples of compound words that are given combined additions. Can be seen in the results that the algorithm that we propose gets the best results among other algorithms.

b) Improvements to over stemming and under stemming

There are some over stemming and under stemming that can be corrected by the algorithm that is proposed. This is showed in Table V. In the table we only list 6 words that are successful in the algorithm that we propose, while the other words we include in Appendix A.

3.3. Calculation of Average Results

The calculation of the average stemming yield that described in this point. There are 3 calculations that are done, i.e. the number of word truth, the average speed of the process and the percentage of the word truth. Here is an explanation of how to calculate it:

a) Average word truth

The average truth of the word we counted from 118 datasets processed which produced 15,792 words. The results obtained from the word truth in the modification algorithm get 10,449 correct words, in the ECS algorithm get 11,530 correct words, and in the Porter, the algorithm gets as many as 9,043 words correctly. Word errors in the modification algorithm get 5,343 incorrect words, in the ECS algorithm get 4,262 incorrect words, and in the Porter, the algorithm gets 6,749 incorrect words. Can be seen in Table 6 and Fig. 3.

b) Average process speed

The average speed obtained to process 118 news documents with 15,792 words can be seen in Table 6. In the Modification algorithm the speed reaches 0.0051 seconds, at ECS it reaches 1.9195 seconds and the Porter algorithm reaches 0.0039 seconds.

c) Percentage of the word truth

We summarize the overall average in Table 6. The truth in the Modification algorithm that we propose is 66% word truth, the ECS algorithm gets 73% and the Porter algorithm gets 57%. We get a percentage of no more than 70% because in the dataset we use is not the whole standard language document, so there are still many non-standard words that are the results of the stem that we cannot match in the dictionary when the word truth calculation process.

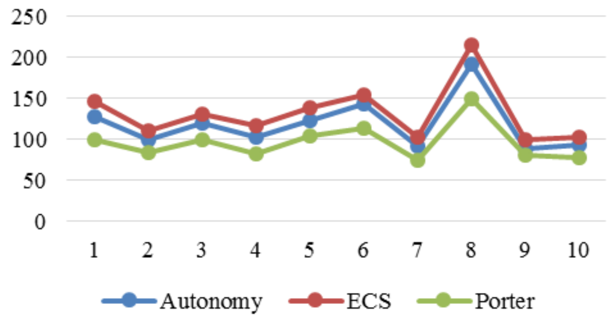

Figure 3. Comparison of true word

Table 6. Percentage of true word

Word	Autonomy	Porter	ECS
True Word	10.449	9.043	11.530
False Word	5343	6749	4262
Time (second)	0,0051	0,0039	1,9195
The precision of True Word (%)	66%	57%	73%
The precision of False Word (%)	34%	43%	27%

4. Conclusion

This study got the results of the initial goal of getting stemming results that have high accuracy and speed by not relying on the dictionary during the process of removing the additive so that it can do the elimination of affixes to compound words. From the results of the trial obtained accuracy of 10,449 true words with an accuracy of 66%, while Porter gets 9,043 correct words with an accuracy of 57%. The second objective of this study was to be able to get a faster stemming time from ECS which was equal to 0.0051, while ECS obtained a stem processing time for 15,792 words of 1.9195 seconds. This study has several shortcomings, so there is a need for further development, namely improvements to over stemming and under stemming in words that have foreign affixes, Sanskrit additions, and inserts.

Reference

[1] Tala, F. Z., A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia, Master of Logic Project, Institute for Logic, Language and Computation, Universiteit van Amsterdam, Netherland. 2003.

[2] Purwarianti, A. A Non-Deterministic Indonesian Stemmer. International Conference on Electrical Engineering and Informatics 17-19 July, Bandung, Indonesia. 2011.

[3] Setiawan, R., Kurniawan, A., and Budiharto, W., Flexible Affix Classification for Stemming Indonesian Language. 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2016.

[4] Agbele K.K., Adesina A.O., Azeez N.A., and Abidoye A.P.Context-Aware Stemming Algorithm for Semantically Related Root Words. Afr J Comp & ICT Vol 5. No. 4, ISSN 2006-1781, June. 2012.

[5] Latt, T. M. and Thida, A., An Analysis of Myanmar Inflectional Morphology Using Finite-State Method. 17th International Conference on Computer and Information Science (ICIS). June 6-8, Singapore. 2018.

[6] Widjaja, M., and Hansun, S., Implementation Of Porter's Modified Stemming Algorithm In An Indonesian Word Error Detection Plugin Application. International Journal of Technology (IJTech) 2: 139-150 ISSN 2086-9614. 2015.

[7] Putra R. B. S. and Utami E., Non-formal Affixed Word Stemming in Indonesian Language. International Conference on Information and Communications Technology (ICOIACT). 2018.

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

[9] Adriani, M., Nazief, B., Asian, J., Tahaghoghi, S. M. M., and Williams, H. E. 2007. Stemming Indonesian: A confix-stripping approach. ACM J. Educ. Resour. Comput. 6, 4, Article 13. 2007.

[10] Arifin, A.Z., dkk. Enhanced Confix Stripping Stemmer And Ants Algorithm For Classifying News Document In Indonesian Language. International Conference on Information & Communication Technology and Systems ISSN 2085-1944 : 149-156. 2009.

[11] Dwiyoga, A., dan Purwitasari, D. Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming. Skripsi Jurusan Teknik Informatika, Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember (ITS). 2010.

[12] Agusta, L., Comparison of Porter Stemming Algorithm and Nazief & Adriani's Algorithm for Stemming Indonesian Text Documents. National Conference on Systems and Informatics. 2009.

[13] Kaur P. and Buttar P. K., Review On Stemming Techniques. International Journal of Advanced Research in Computer Science Volume 9, No. 5, September-October 2018 ISSN No. 0976-5697 DOI: http://dx.doi.org/10.26483/ijarcs.v9i5.6308. 2018.

[14] Sharma D., Stemming Algorithms: *A Comparative Study and their Analysis*. International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 4– No.3, September. 2012.

[15] http://badanbahasa.kemdikbud.go.id/lamanbahasa/sites/default/files/Buku%20Penyuluhan%20BPK.pdf. 2015.

[16] Widayanto H. dan Huda A. F., *Comparison Nazief Adriani And CS Stemmer Algorithm For Stemm Real Data*. e-Proceeding of Engineering : Vol.4, No.3 ISSN : 2355-9365 page 5215 - 5222 Desember. 2017.

[17] Haroon M., *Comparative Analysis of Stemming Algorithms for Web Text Mining*. I.J. Modern Education and Computer Science, 2018, 9, 20-25 Published Online September 2018 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijmecs. 2018.

[18] Kassim M. N., Maarof M. A., Zainal A. and Wahab A. A., *Enhanced Affixation Word Stemmer with Stemming Error Reducer to Solve Affixation Stemming Errors.* Journal of Telecommunication, Electronic and Computer Engineering ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 8 No. 3. 2016.

[19] Karaa W. B. A., *A New Stemmer To Improve Information Retrieval*. International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4, July 2013 DOI : 10.5121/ijnsa.2013.5411. 2013.

Appendix A

No	Word	Autonomy	ECS	Porter	Current Result
1	Kebijakan	Bijak	Bija	bija	bijak
2	Disetujuinya	Setuju	Disetujui	setujui	setuju
3	Pemangkasan	Pangkas	Mangkas	mangkas	pangkas
4	Perbankan	Perbankan	Ban	rban	bank
5	Pertanggungjawaban	Tanggungjawab	Pertanggungjawaban	rtanggungjawab	tanggungjawab
6	Pergerakan	Pergerakan	Gera	rgera	gerak
7	Ketidakpastian	Tidakpasti	Ketidakpastian	tidakpasti	tidakpasti
8	Menahan	Menahan	Nah	ahan	tahan
9	Negeri	Negeri	Neger	negeri	negeri
10	Ekonominya	Ekonomi	Ekonom	ekonomi	ekonomi
11	Dibayangi-bayangi	Bayangi-bayang	Dibayangi-bayangi	bayangi-bayang	bayang-bayang
12	Kehandalan	Handal	Kehandalan	handal	handal
13	Perlahan	Perlahan	Perlahan	rlah	perlahan
14	Pelaku	Pelaku	Pela	laku	pelaku
15	Pemilu	Pemilu	Milu	milu	pemilu
16	Direspon	Respon	Direspon	respon	respon
17	Depannya	Dep	Dep	depan	depan
18	Menggerakan	Gera	Gera	gera	gerak
19	Terhadap	Hadap	Terhadap	hadap	hadap
20	Menjabarkan	Jabar	Menjabarkan	jabar	jabar
21	Ketidakstabilan	Tidakstabil	Ketidakstabilan	tidakstabil	tidakstabil
22	Kehati-hatian	Hati-hati	Kehati-hatian	hati-hati	hati-hati
23	Merespon	Respon	Merespon	respon	respon
24	Walaupun	Walau	Walaupun	walaupun	walau
25	Diekspektasikan	Ekspektasi	Diekspektasikan	ekspektasi	ekspektasi
26	Pemungutan	Pungut	Mungut	mungut	pungut
27	Dilipatgandakan	Lipatganda	Dilipatgandakan	lipatganda	lipatganda
28	Berjumlah	Jumlah	Berjum	jumlah	jumlah
29	Keikutsertaan	Ikutserta	Keikutsertaan	ikutserta	ikutserta
30	Mempertanggungjawabkan	Tanggungjawab	Mempertanggungjawabkan	tanggungjawab	tanggungjawab
31	Kepengurusan	Pengurus	Kepengurusan	pengurus	pengurus
32	Bekerjasama	Kerjasama	Bekerjasama	bekerjasama	kerjasama
33	Menutup-nutupi	Tutup-nutup	Menutup-nutupi	utup-nutup	tutup-tutup
34	Penghancurleburan	Hancurlebur	Penghancurleburan	nghancurlebur	hancurlebur
35	Sekian	Sekian	Kian	kian	sekian
36	Ditandatangani	Tandatangan	ditandatangani	tandatangan	tandatangan
37	Menghawatirkan	Hawatir	menghawatirkan	hawatir	hawatir
38	Buka-bukaan	Buka-buka	buka-bukaan	buka-buka	buka-buka
39	Walaupun	Walau	walaupun	walaupun	walau
40	Perundang-undangan	Undang-undang	perundang-undangan	rundang-undang	undang-undang
41	Ketidakstabilan	Tidakstabil	ketidakstabilan	tidakstabil	tidakstabil
42	Ketidakpastian	Tidakpasti	ketidakpastian	tidakpasti	tidakpasti
43	Menggarisbawahi	Garisbawah	menggarisbawahi	garisbawah	garisbawah
44	Menyebarluaskan	Sebarluas	menyebarluaskan	sebarluas	sebarluas
45	Beragam	Ragam	agam	agam	ragam