CHAPTER 1 THE PROBLEM

1.1 Rationale / Motivation

Stemming is a processs to find root word from its complex form by removing all affixes are attached on it. Stemming have been applied in text clustering to simplify words without losing meaning [19], text classification to reduce number of words [18], summarization [20], information retrieval [17], and word-based text compression [6].

The first *Bahasa Indonesia* (Indonesian Language) stemmer algorithm was developed by Nazief-Adriani (1996), it was a stemmer algorithm with a confix stripping approach and a dictionary-based stemmer. The stemming processs was performed by stripping the shortest possible match of affixes. The dictionary look-up was performed before each stripping step and the stripping processs itself was implemented recursively [1]. We refer this algorithm as S_NA stemmer algorithm.

Jelita Asian improved S_NA algorithm by using more complete dictionary, adding rule to deal with plurals, adding rules of prefixes and suffixes, additional rules and adjusting rule precedence [4][7]. Stemmer algorithm was proposed by Jelita Asian called Confix Stripping Stemmer (CS Stemmer) so we refer this algorithm as CS stemmer algorithm. By improving S_NA stemmer algorithm, CS stemmer could achieve performance up to 97 %. From experiments, Jelita Asian showed CS stemmer had the best performance among others stemmer algorithms [4][5].

However, stemming words processs used CS stemmer algorithm still occurs failures. Ciptaningtyas [5] found failures of CS stemmer and succeeded to fix the failures by modifying some rules prefixes removal and adding the step called loopPengembalianAkhiran when the CS stemmer failed to stemm a word correctly, this method was called Enhanced Confix Stripping Stemmer. In that research, they showed that failures of CS Stemmer had been solved by Enhanced Confix Stripping Stemmer, but Ciptaningtyas did not report the number of performance statistically. However, adding step after CS stemmer failed would cause algorithm to run longer than CS stemmer.

A new method would be proposed by modifying the algorithm so failures occurance could be prevented. By preventing failures occurance would impact the accuracy improvement and reduce time comsumtion of the previous stemmer algorithms.

1.1 Theoretical Framework

A word in *Bahasa Indonesia* is formed from a root word attached with one or more combinations of affixes or just a root word itself. Affixes are attached to a root word consists of prefixes, infixes, and suffixes, combination of them called confix; prefixes are affixes that attached on the left side of root word; infixes are affixes inserted within a word; suffixes are affixes that attached on the right side of root word. Suffixes could be one or combination of a particle, a possessive pronoun, or a derivational suffix.

Stemming is a process removing all affixes attached on word to find a root word; Affix removals from word are performed in some steps. In dictionary – based stemmer, at the beginning and each step of affix removals process are followed by checking the word that its affix has been removed to the root word dictionary; if the word is found in root word dictionary then affix removals process will be stopped, otherwise, the process will be continued with the other affix removal until there is no affix are contained by the word or the word has less than three characters.

On process of removing affix from word, there are affixes that could be removed in a simple way by cutting affix syllable from the word, but there are affixes that generate morphological changes. As some prefixes generate morphological changes on process removal, so previous researchers have composed template formulas for derivation prefix rules as reference for affix removals process .

Some rules in template formulas for derivation prefix rules have more than one formulas, so these rules are ambiguous affixes especially prefix removal. The ambiguities in affix removals are :

1. Formulas in template formulas for derivation prefix that have more than one formula in one rule.
2. Root word that contains syllable the same with derivational prefix or suffix syllable with derivation prefix and suffix.
3. Derivational suffix *"-kan"* with *"-an"*.

Those ambiguities cause the stemmer algorithm failed to stemm word correctly, because the algorithm can not detemine which appropriate formula should be used to remove affix correctly.

However, affix removals steps sequence of stemmer algorithm can leads the algorithm to use appropriate formula for affix removal.

1.2 Conceptual Framework/Paradigm

The general model of words in *Bahasa Indonesia* is as follows :

[DP+[DP+[DP]]]+root-word+[[DS][PP][P]]

DP is derivational prefix, **DS** is derivational suffix, **PP** is possessive pronoun, and **P** is particle. As described on previouse section, all of them are affixes.

Stemmer algorithm basically is a sequence of steps to remove affixes from a word to get a root word. In the dictionary – based stemmer algorithm, at the beginning process and each step of affix removal, word is performed to check against root word dictionary; if the word is found in the root dictionary then process will be stopped and root word is found, otherwise, the process will be continued with affix removals. Stemming process wil be stopped if :

1. word is found in the root dictionary;
2. word does not contain any syllable the same woth affix;
3. word cosist of less than 3 characters

The sequence of Nazief – Adriani stemmer (S_NA) begins with suffixes removal (right side of root word) then followed by prefixes removal (left side of root word); As with this sequence S_NA has failures, CS stemmer improve the algorithm by adding a rule precedence so stemming process sequence begins not only from right to left side of root word but enable from left to right side of root word. However, by adding rule precedence CS stemmer still has failures even there were some words that could be stemmed by S_NA correctly but CS stemmer could not stemm correctly.

Ciptaningtyas had fixed some failures of CS stemmer by returning affixes that had been removed to the word if stemmer give a failure result then repeat the stemming process. By this method, the running time of stemmer will be increased.

1.3 Problem Statement

The sequence was proposed by S_NA and CS stemmer algorithm could not handle all affix removal ambiguities so both stemmer could not stemm some words correctly. Stemming sequence that runs from left to right side of root word that was proposed by S_NA stemmer aglorithm could

not handle the ambiguities. CS stemmer algorithm with adding rule precedence to enabling stemm word from left to right side root word and vice versa still could not handle the ambiguities, even some words that could be stemmed correctly by S_NA, could not be stemmed correctly by CS stemmer. Ciptaningtyas had fixed the failures by repairing the result if stemmer give failure result affect the increased of time consumption. Therefore, it is necessary to proposed new method that has capability to handle the ambiguities of affix removals to prevent the failure result occurance so will be composed a better stemmer algorithm that can give reulst correctly and less time consumption

1.4 Objectives

According to the problem statement and proposed method above, lead this study to achieve the objectives :

1. Handling the ambiguities of affix removals so the failures of S-NA and CS stemmer algorithm could be fixed.

2. By fixing the failures of S_NA and CS stemmer algorithm could improve stemming performance (accuracy and time consumption) of both stemmer algorithm.

1.5 Hypothesis

By rearrange the sequence of stemming process, algorithm could handle the ambiguities of affix removals and stemm words better than S_NA and CS stemmer so the performance of both previous stemmer could be imrpoved.

1.6 Assumption

Stemmer that is usually applied in text mining will return to the original word form if at the end of stemmming process the word was not found in root word dictionary, in other words, the stemmer failed to find root word form. However, for analisys purpose, in this study the output of stemmer

was form of word as is the result of stemming process, both the result word form is in the root dictionary or not, the word would not returned to the original form.

1.7 Scope and Delimitation

Delimitation in this study are :

1. According to algorithm, if result of stemming process was not found in root dictionary to be returned to original form but for analysis purpose in this study result still in form as result of stemming.
2. Stemmer would not stemm a word from foreign language or affixes that have been absorbed from foreign language into Bahasa Indonesia..
3. Result of stemming process would be considered as correct stemm if found in root directory although the meaning of original word and root word were not the same. In other word, stemming process just observe in morphological word but not semantically.
4. This study would not fix mistakes occur in conversion process from PDF file to text file.
5. This study would not examine infix and compsite words.

1.8 Importance of the Study

Stemmer was applied in various text mining application to improve application performance, such as :

- In Information Retrieval stemmer could improve performance by providing variant morphological searched terms and reduce size of index [9].
- In word based text compression, stemmer could simplify the dictionary as various word from could be represented by one word [6].
- Besides reduce size of document index, stemmer could increase text retrieval accuracy [10].
- In text classification stemmer reduce the number of features [18].

As described above, stemmer influences the performance of text mining applications especially for information retrieval. Therefore, it is important to improve stemmer algorithm performance so the applications performance could be improved.