# CHAPTER 1: INTRODUCTION

## 1.1 Rationale

Data is very important especially for large companies. Some companies require a software or a good tool to compress the data, because it can reduce the size of the data itself. There are some studies concerning data compression. Among others are those conducted by *by Yusof. Mohd Kamir, Mat Deris. Mohd Sufian, and Abidin. Ahmad Faisal Amri* which says "*Research about the previous compression algorithm is aimed to improve compression ratio". One of them is Lempel, Ziv, and Welch (LZW) compression algorithm, which was developed from that previously made by Ziv and Lempel* " [3]. The most popular technique for data compression is Lempel Ziv Welch (LZW) [9]. The first step of this technique is read data file. After that, it will read the characters in file one by one. Then, each character will be assigned a newcode. If the same characters are found, they are not assigned a newcode but use the existing code in data dictionary. LZW algorithm will loop until characters in data file are null.

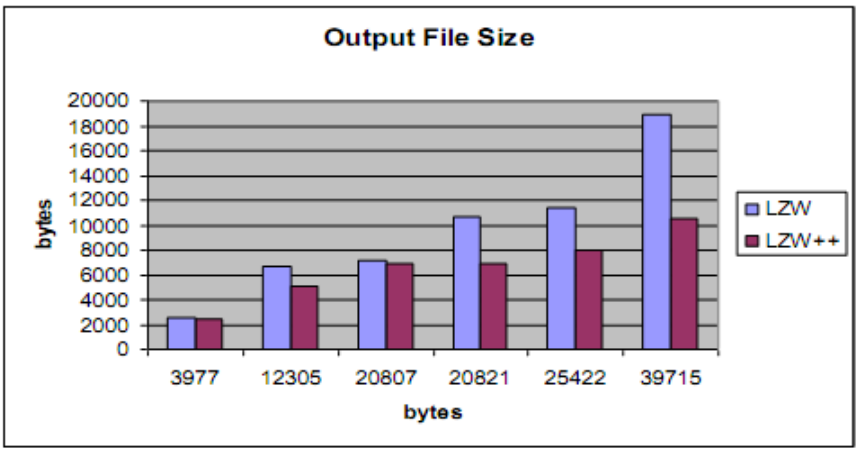


Figure 1.1: LZW vs. LZW++ (Txt Format) [3]

The LZW has a potential to be improved. In previous research *"Study of Efficiency and Capability LZW++ Technique in Data Compression" (Yusuf. Mohd.Kamir, Mat Deris. Mohd.Sufian, and Abidin, Ahmad Faisal Amri, 2009),* the experiment used text as data type (txt file). This technique (which is called LZW++ by the authors) proposed to read 3 (three characters) at a time instead of reading one by one character. Based on the results of the experiment, LZW ++ technique is better than Standard LZW technique in term of compressed file size. Figure 1.1 shows the comparison between the Standard LZW technique and LZW++ technique in terms of file size. The purple bar represents the result of the Standard LZW and the red bar represents the result of the enhanced LZW which means

LZW++ technique. The results also show that LZW++ technique can reduce original size compared to the original size.

## 1.2 Theoretical Framework

This research proposes new approaches to modify the Standard LZW algorithm to increase compression ratio. It can increase the LZW compression ratio by adopting the Standard LZW compression technique to read characters, but while creating a newcode for new characters, we also made it possible to create the second newcode to represent the characters which sequence is reversed from the new one. This idea is based on the study using Indonesian language dictionary that has been published by PUSAT BAHASA DEPARTEMEN PENDIDIKAN NASIONAL and using English file in the Canterbury Corpus (bible.txt). *Canterbury Corpus is* a collection of files intended to use as a benchmark for testing lossless data compression algorithms. This experiment was conducted using a sequence of 2 or 3 characters and their reversed sequence of characters, it searched a number of its occurrences. If the probability of its occurrences is high, then it is highly possible that by adding a second newcode, it can produce better compression ratio.

Appendix J and appendix K shows that when two characters are used, the number of occurrences of finding the reversed sequence is quite high. For example, the experiments showed that for pairs of *consonant-vowel* and *vowel-vowel*, the probability to find their reversed pairs is very high. Meanwhile, evaluation on *consonant-consonant* pairs shows that the probability to find their reversed sequences is almost zero. Based on this finding, it can be concluded that, in case of the pairs of *consonant-vowel* and *vowel-vowel*, the idea to implement reversed sequence of characters as a new code can be adopted.

## 1.3 Conceptual Framework/Paradigm

In this research, measurement is important. Variables that must be measured are file size and compression ratio. *"File size can be measured in Bytes (B) kilobytes (KB), and etc"* [4], while the compression ratio can be measured in percentage, using compression ratio formula explained in chapter 2.2.2.

A compression method is illustrated in the following block diagram:

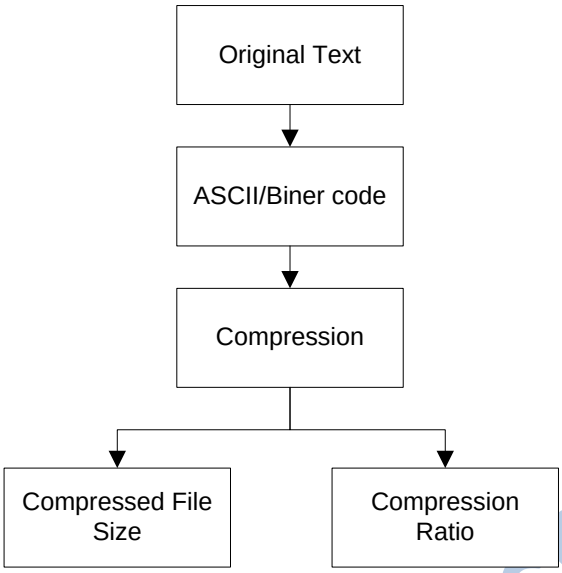


Figure 1.2: Block diagram compression

## 1.4 The Problem

The problem in this study is that the output file size of the Standard LZW method is still considered high.

## 1.5 Hypothesis

For overcoming the problem the following method was proposed. By adding a second newcode on a dictionary to represent a reverse sequence of characters, a compression ratio can be improved, because there is a high probability that a sequence of 2 or 3 characters and their reversed sequence in Indonesian as well as English language are found. In Standard LZW method, this characteristic was not exploited.

## 1.6 Assumption

The assumption proposed in this study is that there is no error on a transmission or on stored medium, so that the output stream read by the decoder is considered intact.

## 1.7 Scope and Delimitation

This thesis discusses the implementation of the modified LZW algorithm and the scope of this study are:

1. All of 8-bit standard characters have been stored in the dictionary.
2. Only text file will be processed
3. Time performance will not be discussed
4. Length of the codes will not be more than 16 bit
5. Only Roman characters are used
6. A tool to convert byte data into bit sequence is not developed
7. Compressed file is not produced. The size of the encoder's output is calculated only by the formula.

## 1.8 Importance of the Study

The benefits of this study are as follows

1. To achieve better compression ratio and to make the output file size smaller than Standard LZW.
2. To achieve the target, reverse sequence of characters are introduced on the Standard LZW Algorithm.