CHAPTER I

INTRODUCTION

1.1 Background

Nowadays, so many people know and interact with digital text documents/files. Even elementary school students' assignments need to be saved as a digital text file. High school students and college students also use text files everyday for their assignments and project reports. Office employees also work with these on a daily basis, making text files an indispensable element to our daily life. To accommodate the need, many companies such as Microsoft, created a text processing program to create and modify text files known as Microsoft Word.

Sometimes, we encounter a very large and space-consuming text files, and to store that file in its regular size to our flash drive/hard disk, sometimes it could be a problem, and even worse if our storage media is almost full. So this project is created to compress such files in order to save space.

There are many types of compression algorithms but they are divided into two groups: lossy and lossless compression. *Lossy compression is a compression method that uses inexact approximations (or partial data discarding) for representing the content that has been encoded. Such compression techniques are used to reduce the amount of data that would otherwise be needed to store, handle, and/or transmit the represented content.* This means the compressed file cannot be uncompressed back to its original state, there are some loss of data. Hence the name "lossy compression".

On the other hand, *lossless compression allows the original data to be perfectly reconstructed from the existing compressed data.* Because we are compressing text file here, we don't want any single bit of missing information in the text. So, the compression being used here are all lossless types.

Lossless compression itself has two different main methods, statistical and dictionary based. *Statistical compression algorithms are a class of lossless compression algorithms based on the probability that certain characters will occur.* The algorithm used to represent

this method is Shannon – Fano. While dictionary-based compression is *a class of lossless data compression algorithms which operate by searching for matches between the text to be compressed and a set of strings contained in a data structure (called the 'dictionary') maintained by the encoder. When the encoder finds such a match, it substitutes a reference to the string's position in the data structure.* The algorithm used to represent this method is LZ77.

1.2 Scope

This project's source code is made using Java programming language. The main purpose is to compare the performance of Shannon – Fano and LZ77 algorithms. Does not focus on decompression. A GUI version of this project exists and runs well but with some restrictions and obstacles.

1.3 Objectives

This project is created using Java programming language. The project compares the performance of Shannon–Fano and LZ77 on a given text file. Detailed purpose:

1. Able to compile and run the program.
2. Can compress the text files properly
3. Can compare the file size from the generated output file from each algorithm.