

File Compression Application Design Using Half Byte Algorithm

Paska Marto Hasugian¹, Rizki Manullang²

Rekayasa Perangkat Lunak, STMIK Pelita Nusantara

Email: paskamarto86@gmail.com¹, riskymanulang87@gmail.com²

Keywords	Abstract. The need for large data storage capacity seems increasingly important. This need is caused by the data that must be stored more and more. The storage is not allocated in one place only. But they will also store data or files elsewhere as data backups. How much storage capacity must be provided to hold all the data. Half Byte algorithm is one of the data compression algorithms. The Half Byte algorithm utilizes four left bits that are often the same sequentially especially text files. When the first four bits of the character are received in a row seven or more times, the algorithm compresses the data with marker bits, then the first character of the same four-bit series is followed by the last four-bit pair of the next series.
Compression Half Byte Algorithm	

1. INTRODUCTION

Companies generally need a very large capacity, to store all the important data and files. The storage is not allocated in one place only [1]. But they will also store the data or files elsewhere. Although the saved yangg is the same, it is useful for data backup. Backups need to be done because there is no guarantee that a data storage area inside the computer will not be damaged [2]. How much storage capacity must be provided to hold all the data. However, the data can be compressed first so that the place that memory needs is less and the time it takes to communicate is shorter, resulting in fewer failures in data manipulation. The utilization method used today is by compressing finished files (original files) and then utilized and newly communicated [3]. After arriving the data of the utilization results on sipenerima, then done monitoring (Un-compress) to return to the original form, then the file can be used[4].

Compression is a technique for reducing the number of bits needed to store image file data so that the compressed file size is smaller than the original file, but does not experience a significant decrease in file quality. Thus, the required storage space capacity will be smaller and the time and cost of access becomes smaller as well [5]. Data compressing methods have evolved and each method has advantages and disadvantages, the capabilities of each method of data compressing are generally measured by parameters of size or capacity and compression speed, by compressing or shrinking the file size first, then the required storage capacity becomes smaller[7].

2. METHOD

2.1 Implementation Stages

1. The stage of the engineering system is the stage of finding the needs of the entire system that will be used in building a file expression system. This file expression system uses the Half-Byte algorithm method by utilizing the left bits that are often the same in sequence.
2. Analysis, The stage of analysis is the stage of processing the data that has been obtained and grouping the data according to the needs of the design.
3. Design stages are the interface stages of this file compression and decompression application. In this study, using BPM Diagrams to design which is done to include the design of pages in the application.
4. Coding, The coding stage is the stage after the design and application is implemented directly. in the implementation of the application.
5. Testing stage is the stage where there is no longer data from the user who wants to be implemented so that the needs of the system are considered to have been met.
6. In this stage, the comparison of compression ratios is done using the Half-Byte algorithm to find out how much the Ratio of Half-Byte algorithm to file compression compared to Zip and Winrar .
7. Testing is the most important thing in assessing the success of an application program, and aims to find out the errors and shortcomings in the program created.
8. After testing the application, at this stage will see the results achieved by the Half-Byte algorithm

ratio how much is achieved in performing file compression compared to Zip and Winrar

2.2. Half-Byte Algorithm

The Half-Byte algorithm utilizes four left bits that are often the same sequentially especially in text files. For example in a text file containing the words "fetch", in hexadecimal and binary characters are translated as [7]:

Tabel 1 Goritm Half-Byte

character	Heksadesimal	Biner
m	6D	01101101
e	65	01100101
n	6E	01101110
g	67	01100111
a	61	01100001
m	6D	01101101
b	62	01100010
i	69	01101001
l	6C	01101100

Compression with a half byte algorithm. The compression algorithm is: 1. Find a row of characters whose first 4 bits are the same in sequence, do compression. 2. Write a marker byte on a compression file, in the form of 8 rows of bits (1 byte) that can be selected arbitrarily as long as it is used consistently on all bytes of the compression marker. This marker byte serves to mark that the next character is a compression character, so it is not confusing when returning a compressed file to the original file. 3. Write the first character of 4 consecutive left bits from the original file. 4. Combine the 4 right bits of the second and third characters then write them into the compression file. Do this until the end of the series of characters with the same first 4 bits. 5. Close by writing a byte marker on the compression file. 6. Repeat steps 1-5 until the last character [5], [6].

3. RESULTS AND DISCUSSION

3.1 ConvertIng Characters to Binary Values

System Design In the manufacture of an application is very necessary, because with the design of the system, applications can be built in accordance with the design that has been made, in addition to the design of the system also makes it easier to facilitate the flow of applications to be built. The Half-Byte algorithm utilizes the same four left bits sequentially seven times or more especially in text files. For example in a text file containing the inscription "rahmadani", in hexadecimal and binary characters are translated as:

Table 2 Algorithms Half Byte

character	Heksadesimal	Biner
r	72	01110010
a	61	01100001
h	68	01101000
m	6D	01101101
a	61	01100001
d	64	01100100
a	61	01100001
n	6E	01101110
i	69	01101001

The algorithm in the file compression program is divided into two, namely compression algorithm with half bytes and compression algorithm with half bytes.

3.2 Compression Algorithm With Half Bytes

Compression Algorithm is an algorithm designed to compress text files then the need to compress files, as for the form of the algorithm is seen in pseudocode as follows:

Input:

Document File to be compressed

Process : Specify marker bits

If bit marker is specified Then

Compress text else files

Show Text file or End if initial character

Output : Compressed File Document.

3.3 Algorithms With Half Bytes

Compression Algorithm is an algorithm designed to perform the monitoring of text files then the need to do dkompress files, as for the form of the algorithm is seen in pseudocode as follows:

Input : Document file that has been compressed.

Process : Specify marker bits

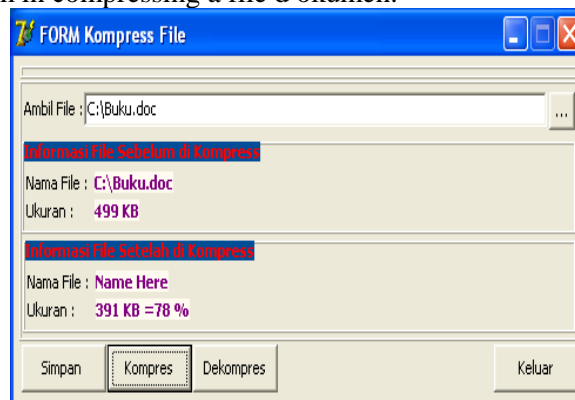
if bit marker is specified Then Decompress text else file

Show Text file or End if initial character

Output : Document original file.

3.4 HalfByte Algorithm Compress Program

Form kompress halfbyte algorithm is a form used to compress a document file using halfbyte algorithm, it appears in the image this algorithm compresses a file with a large 499 KB to 391 KB, from the large capacity obtained that this algorithm compresses a file almost 22% of the actual result, from both algorithms that appear in the image can be concluded that the huffman algorithm is better than the halfbyte algorithm in compressing a file d okumen.



Gambar 1 Form Kompress HalfByte

Form uncompressed halfbyte algorithm is a form used to compress a document file with which it has been compressed using halfbyte algorithm, it appears in the image that filesize the result of the compress is 391 KB by using Halfbyte Algorithm then after uncompressed the file back to the actual size of 499 KB.

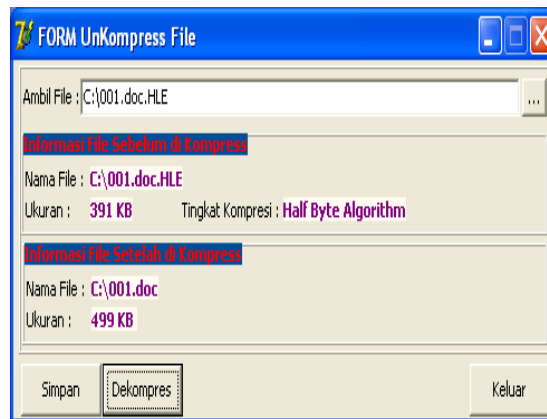


Figure 2 Form UnKompres HalfByte

4. Conclusion

The conclusions are as follows: Adding the first character of the original file, By combining the four right bits with the second and third characters, then added to the compressing file. Application designed using Borland application, the first thing to do is to design the konfress file form first by selecting the desired compression method, then designed the form uncompress file.

REFERENCE

- [1] H. T. Sihotang, "Perancangan Dan Implementasi Algoritma Arithmetic Coding Untuk Aplikasi Kompresi Data Video Dan Audio," *J. Mantik Penusa*, vol. 2, no. 1, 2018.
- [2] T. I. Saputra, F. Fauziah, and N. Hayati, "Implementasi Discrete Wavelet Transform Pada Aplikasi Kompresi Citra Medis," *J. InfomediaTeknik Inform. Multimed. Jar.*, vol. 4, no. 2, 2020.
- [3] S. K. Fatima, S. G. Fatima, S. A. Sattar, and A. Sheela, "An advanced data security method in WSN," *Int. J. Adv. Res. Eng. Technol.*, vol. 10, no. 2, 2019, doi: 10.34218/IJARET.10.2.2019.026.
- [4] S. Pirbhulal, O. W. Samuel, W. Wu, A. K. Sangaiah, and G. Li, "A joint resource-aware and medical data security framework for wearable healthcare systems," *Futur. Gener. Comput. Syst.*, vol. 95, 2019, doi: 10.1016/j.future.2019.01.008.
- [5] Y. Bulo, P. Sharma, and C. T. Bhunia, "Improving Error Correction Capability of Aggressive Packet Combining Scheme by using Half-Byte Packet Reverse Technique and Even - Odd Selection Method," *Int. J. Comput. Appl.*, vol. 137, no. 2, 2016, doi: 10.5120/ijca2016908701.
- [6] P. S. Anderson, "The PASC0 Half-Byte Hard Drive," *Phys. Teach.*, vol. 43, no. 3, 2005, doi: 10.1119/1.1869431.
- [7] X. Hu and Y. Zhao, "Research on Plaintext Restoration of AES Based on Neural Network," *Secur. Commun. Networks*, vol. 2018, 2018, doi: 10.1155/2018/6868506.
- [8] N. Nazar, R. S. Kumar, M. Rajeswari, and G. R. Gnanaking, "A secure model for hiding multimedia files within two cover images," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, 2019.
- [9] A. Sinambela and I. Ramli, "Kompresi Medulla Spinalis akibat Metastasis," *Radioter. Onkol. Indones.*, vol. 9, no. 1, 2018, doi: 10.32532/jori.v9i1.65.
- [10] M. A. Latif, S. D. Nasution, and P. Pristiwanto, "Analisa Perbandingan Algoritma Rice Codes Dengan Algoritma Goldbach Codes Pada Kompresi File Text Menggunakan Metode Exponential," *Maj. Ilm. INTI (Informasi dan Teknol. Ilmiah)*, vol. 13, no. 1, 2018.