http://infor.seaninstitute.org/index.php/infokum/index
JURNAL INFOKUM, Volume 10, No.5, Desember 2022 ISSN : 2302-9706

IMPLEMENTATION OF HUFFMAN AND LZ78 ALGORITHM FOR CHARACTER COMPRESSION

Chairul Imam[1], Muhammad Furqon Siregar[2]

Department Of Information Technology, Department Of Informatics Universitas Battuta
Email : chairulimam313@gmail.com

Article Info
Received : 01/11/2022
Revised : 03/12/2022
Accepted : 09/12/2022

The Huffman algorithm is a time complexity of 0(n log n), because the Huffman algorithm performs one iteration process when merging two trees that have the smallest frequency at a root takes 0(log n), and the process is done many times until only there is only one tree left with a combination of the LZ78 algorithm using a dictionary technique in compressing one of the characters in the form of a string replaced by a table code that is made for reference in extended ASCII in inputting characters or strings.

Keywords: Algorithm huffman, LZ78.

1. INTRODUCTION

Data is something that has no meaning for the recipient and still requires processing. Data can be in the form of a situation, pictures, sounds, letters, numbers, mathematics, language or other symbols that we can use as material to see the environment, object, event or a concept, making everyone try to find it. a method that can be used to compress data. Huffman's algorithm, created by an MIT student named David Huffman, is one of the oldest and most famous methods of compression text. Huffman's algorithm uses a coding principle similar to code Morse, that is, each character (symbol) is encoded with only a series of bits, where frequently occurring characters are encoded with short bit sequences and characters that occur infrequently are encoded with a longer string of bits.

2. METHOD

The method used is as follows :

a. The Huffman Coding Algorithm is one of the algorithms that can be used to compress data so that the resulting data size is lower than the actual size. The example discussed this time is about compression and data from a sentence. Huffman Coding is an optimized prefix code that is usually used in lossless data compression. This algorithm uses a special method to select the representative of each symbol, which finally produces a prefix code. This algorithm is a very well-known method for making prefix code "Huffman Code" is widely used as a synonym of "prefix code" even for scripts that are not generated from Huffman Coding calculations.

b. The LZ78 algorithm was developed by Terry A. Welch from the previous compression method discovered by Abraham Lempel and Jacob Ziv in 1977. This algorithm uses a dictionary technique in its compression. Where the character string is replaced by a table code that is created every time a string is entered. The table is created to reference the next string input. The size of the dictionary table in the original LZW algorithm is 4096 samples or 12 bits, where the first 256 samples are used for the single character table (Extended ASCII), and the rest are used for character pairs or strings in the input data. The LZW algorithm performs compression by using table codes 256 to 4095 to encode byte pairs or strings. With this method multiple strings can be encoded with reference to strings that have appeared earlier in the text.

The complete LZW compression algorithm:

1. DICTIONARY initialized with all existing base characters : {'A'..'Z','a'..'z','0'..'9'}.
2. W the first character in the character stream.
3. K the next character in the character stream.
4. Check whether (W+K) is in DICTIONARY

If yes, then W W + K (combine W and K into a new string).
If not, then:
Outputs a code to replace the string W. Add the string (W+K) into the dictionary and provide the next number/code that has not been used in the dictionary for that string.

3. RESULT AND DISCCUSSION

3.1 Implementation and System Testing

Implementation is an action or implementation of a plan that has been prepared carefully and in detail. Implementation is usually carried out after the planning is considered complete. This implementation The decision support system consists of the following requirements :

a. Hardware Requirements
In order for a system to run as desired, it cannot be separated between hardware and software aswell as the specifications needed in designing the system
1. Processor core 13
2. Minimum Hard Disk 1000 GB
3. 3GB RAM
4. Monitor
5. Keyboard
6. Mouse

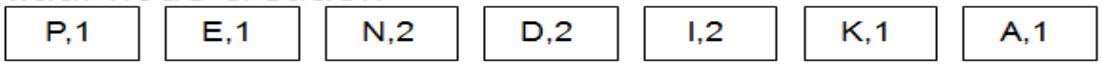
b. Software Requirements In order for this system to run smoothly, it is inseparable from the presence of software that clearly supports the above hardware, along with the software needed to run the system
1. Windows 7,8 and 10,11 Operating Systems.
2. Microsoft Visio

how to complete a character or sentence using the huffman algorithm :
that each ASCII character is usually represented by 8 bits. So for example a file contains a series of characters "ABACAD" then the file size is 6 x 8 bits = 48 bits or 6 bytes. If each character is given another code, for example A=1, B=00, C=010, and D=011, it means that we only need a file with a size of 11 bits (10010101011), which needs to be noted is that these codes must be unique. or in other words a code cannot be formed from other codes Frequency of occurrence of characters in words PENDIDIKAN can be seen in the table below.

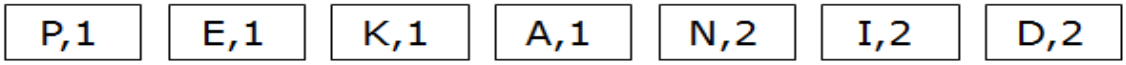
Table 1. Frequency Of Appearance Of Character

Character	Frequency	Big Memory
P	1	8 bit
E	1	8 bit
N	2	16 bit
D	2	16 bit
I	2	16 bit
K	1	8 bit
A	1	8 bit
Amount	10 character	80 bit

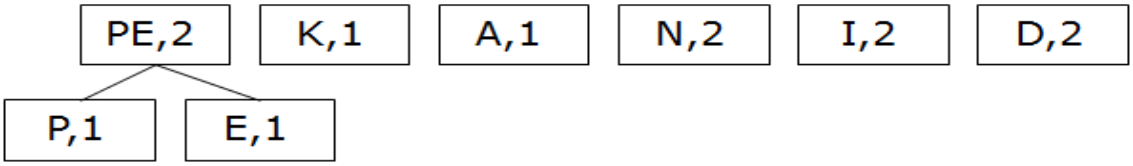
1. Initial node creation



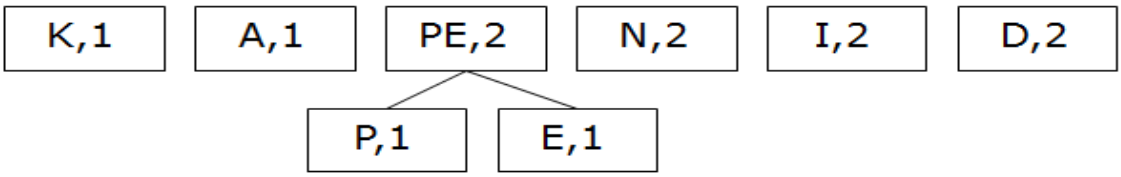
2. First sorting of nodes by frequency



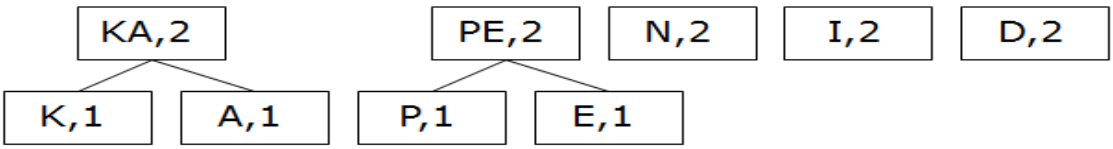
3. First merge



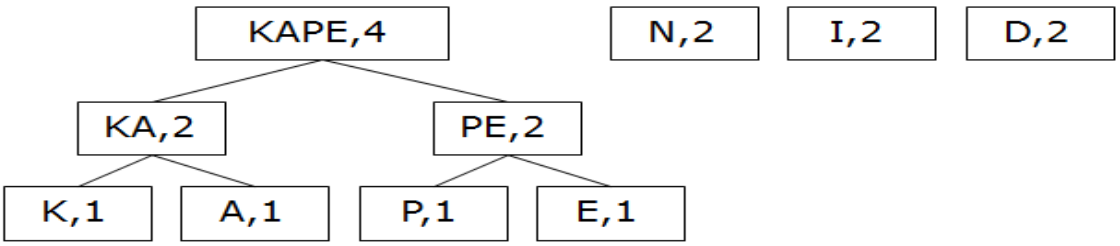
4. Second order



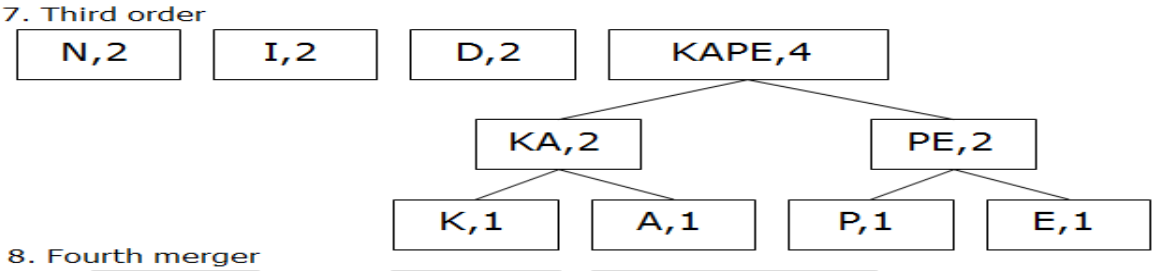
5. Second merger



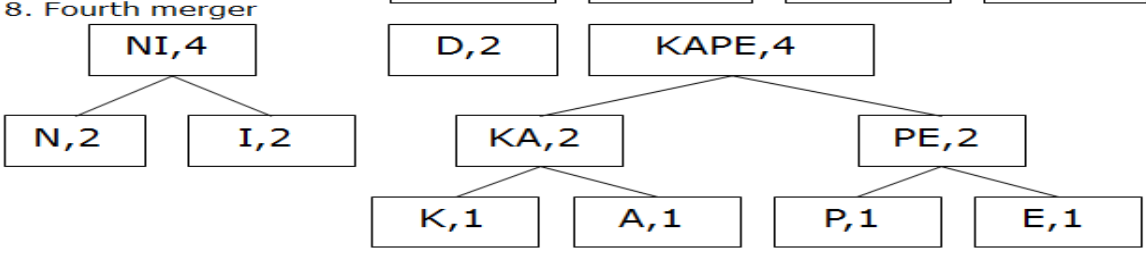
6. The third merge (because the nodes are already sorted)



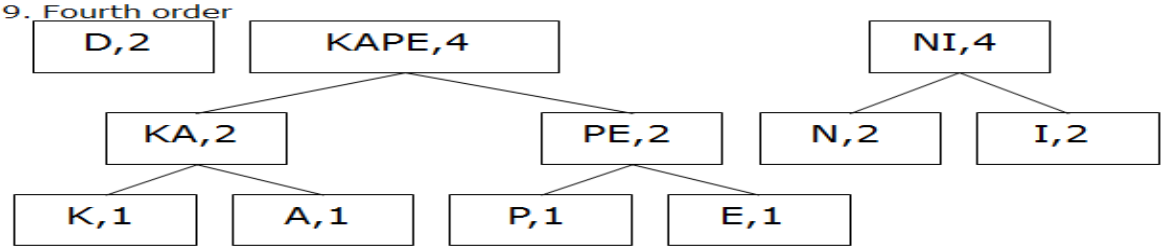
7. Third order



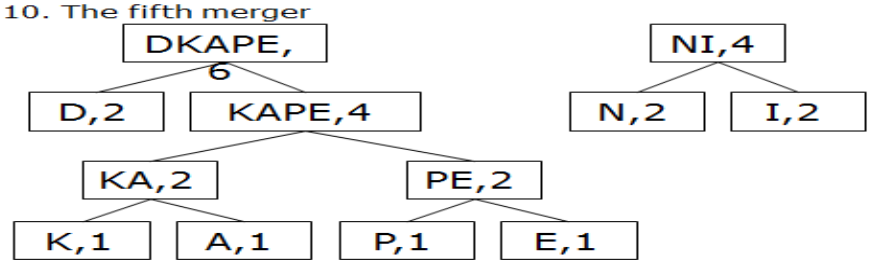
8. Fourth merger



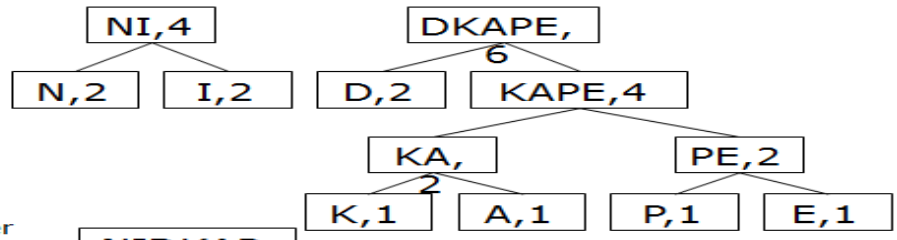
9. Fourth order



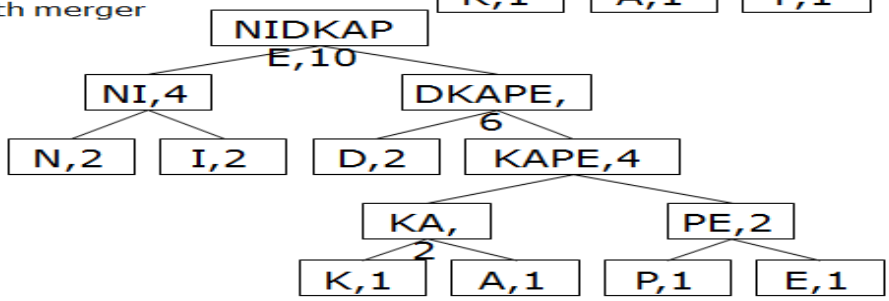
10. The fifth merger



11. Fifth order



12. The sixth merger



13. Giving 0 and 1 bit values for each leaf

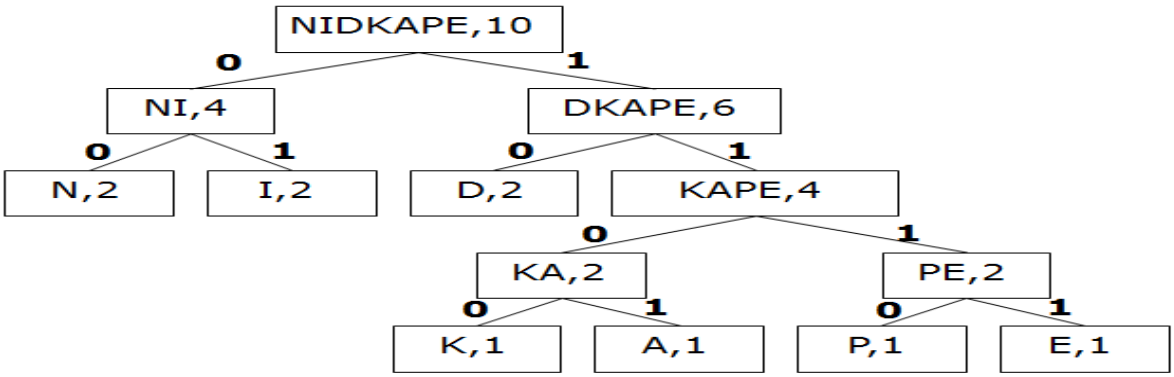


Table of compressed binary form

Character	Frequency	Bit Shape	Jml bit / Character	Big Memory
P	1	1110	4	4 bit
E	1	1111	4	4 bit
N	2	00	2	4 bit
D	2	10	2	4 bit
I	2	01	2	4 bit
K	1	1100	4	4 bit
A	1	1110	4	4 bit
Amount	10character			28 bit

4. CONCLUSIONS

Based on the previous discussion that has been carried out,some conclusions can be drawn as following Faster decompression process performed compared to compression process due to decompression process is not carried out in the process of forming a tree Huffman from the data but just directly read from the code table Huffman tree saved on files during the compression process.characters or sentences that have been compressed when the compression process is done once again then the size or sentence will be getting bigger a bit because Huffman and LZ78 algorithms are optimal compression is a character that two compressions times then the last process will not reduce the character size again. Occur increment of bytes in process compression second time because program save tree structure Huffman from compression first.

Reference

[1] Cormen, T.H, Charles E.L., Ronald L.R., and Clifford S. 2001. Introduction to Algoritms. Second Edition. London: McGraw- Hill Book Company.

[2] Blelloch, G.E., 2001, Introduction to Data Compression. Computer Science Department, Carnegie Mellon University.

[3] Haryanto, R.I., 2009. Kompresi Data dengan Algoritma Huffman dan Perbandingannya dengan Algoritma LZW dan DMC, Makalah IF2091 Strategi Algoritmik.

[4] Schindler, M., Practical Huffman Coding. http://www.compressconsult.com/huffman/ Diakses 25 Januari 2011

[5] Silalahi, B.P., Julio A., Danny D.S. Perbandingan Algoritma Huffman Statik dengan Algoritma Huffman Adaptif pada Kompresi Teks. Jurnal Penelitian.

[6] Imam, C., Muhammad Furqon Siregar and Auliana Nasution (2021) "Implementation of OSI Layer Based on Interactive Education Media", *Jurnal Mantik*, 4(4), pp. 2545-2551. doi: 10.35335/mantik.Vol4.2021.1206.pp2545-2551.

[7] Imam, C., & Siregar, M. F. (2021). Utilization Of Key Logger In Security Aspect Based On Spyware For Monitoring Laptop Using Refog Key Logger. *INFOKUM*, *10*(1), 105-111.

[8] M. F. Siregar, P. Sihombing and Suherman, "Analysis of Fuzzy Logic Method for Load Lifting Robot," *2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, 2019, pp. 125-129, doi: 10.1109/ELTICOM47379.2019.8943852.

[9] Muhammad Furqon Siregar, Chairul Imam, Auliana Nasution.(2020). Pemanfaatan Solenoid Valve Dan Sensor Hc-Sr04 Sebagai Pencuci Tangan Otomatis Berbasis Arduino Uno.

[10] Siregar, M. F., & Chairul Imam. (2022). Design Of Automatic Hand Washing System Using Solenoid Valve Based On Microcontroller. *INFOKUM*, *10*(03), 20-25.