Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Application of Huffman algorithm and Unary Codes for Text File Compression

Sarwando Siboro[1)*], Bayu Angga Wijaya[2)], Mahendra Brutu[3)], Yelita Kristiani Lase[4)]
[1,2,3,4)] Universitas Prima Indonesia
Medan, Indonesia
[1)] sarwando2812@email.com, [2)]bayuanggawijaya@unprimdn.ac.id, [3)]mahendraberutu72@gmail.com,
[4)]laseyelitakristiani@gmail.com

Submitted : July 17, 2022 | **Accepted** : July 27, 2022 | **Published** : July 30, 2022

Abstract: Technique in carrying out data compression is an important point in technological developments. With compression in data in the form of text can include many uses, including for data transfer, copying and for backing up data. From its uses, this aspect is important for data security. There are many compression techniques on the data, including using huffman algorithms and unary code. One of its applications will be implemented on a text data that is widely used by digital actors in storing important data. The data must not be known by unauthorized parties in accessing the data. Therefore, huffman algorithms and unary code can solve this problem. By compressing the selected data also encrypts it as an extra security. The Huffman algorithm is a lossless compression algorithm or a technique that does not change the original data, by converting the unit of data content into bits. So this algorithm is widely used in the compression process. The Unary Codes algorithm is also a lossless compression technique that is generally used by combining several modification techniques. In this unary codes algorithm, each symbol in the string will be searched for its frequency. Then sorted from the last order (descending). The use of these two text data compression techniques results in a file size that is smaller than the original but can be returned to the original data

Keywords: Compression; Huffman; Unary Codes; Java; Algorithm;

INTRODUCTION

In addition to the improvement of current technology, data has a very important role. Data is not only introduced in the form of text, but information and data can also be images, sound, and video commonly called multimedia. The expansion in the use of data has created some problems in data storage and has indirectly expanded the interest in data storage. The larger the file size, the more extra space is needed in storage (Lamsah et al., 2020). Today there are several types of formats in files, one of which is the text format.

A text is a collection of strings or characters that make up a single whole. One of the media that we usually find in the use of text is analog or paper media that is commonly used in writing a text or characters. As of now, there are quite a lot of methods/algorithms used in data compression/compression. So in this case we need to learn about the technique of compressing the text data. Compression is carried out to reduce the data capacity of the original data (Yonathan et al., 2021). Compression of text data can be done with various algorithms, such as the Huffman algorithm method and the Unary Code Algorithm. Unary Code is generally represented in a string of n and bit 1 followed by one bit 0 which terminates which is defined as n-1 bit 1 followed by one bit 0 or vice versa as an alternative it can also equivalently start from n bit 0 followed by bit 1 which ends which is defined as n-1 bit 0 followed by one bit 1. In the Unary Code, there is no frequency division of symbols on the string (Michael et al., 2020).

While the Huffman Algorithm uses coding standards such as morse code, it means that each character (symbol) is encoded with only a series of several bits, where the usual characters that appear are encoded with a short bit pool and the characters that rarely appear are encoded with a longer string of bits (Kharisma Mahesa, 2018).

In this research, the author wants to develop previous research by compressing data files in the form of text files with the research title "Application of Huffman Algorithms and Unary Codes for Text File Compression".

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The expected result is to be able to analyze which algorithm is most efficient in compressing text files, in terms of speed and file size of compressed results.

METHOD

In these cases the type of research carried out is experimental research. This experimental research is a study that recognizes causation with a specific purpose or in other words, tests a hypothesis.

Data

The following data was used in this study: The data format used is .txt; Data in the form of text containing articles obtained from the internet and news as much as 10 data as a sample data.

Work Procedures

Literature Studies is a collection of activities related to the method of collecting library data, recording, reading, and processing research topics. The data collected from the two methods that are the topic of discussion in this study will later be processed to produce a comparison of effectiveness. Stages of Application Creation, Analysis In this stage, an analysis of the needs needed in making applications is carried out with the application of the Huffman algorithm and Unary Codes. Coding, the activities carried out in this stage are translating data or solving problems that have been analyzed into programming languages. In making this application, the programming language used is java language. Testing, at this stage, the author will test the application and see a comparison of the compression results between the Huffman algorithm and the Unary Codes Algorithm. Maintenance, is the final stage where the completed application undergoes changes and adjustments from errors that appear during testing.

Usecase Diagram

The following usecase diagram design is taken from the application built, namely the Application of Huffman Algorithms and Unary Codes for Text File Compression, can be seen in figure 1. below.

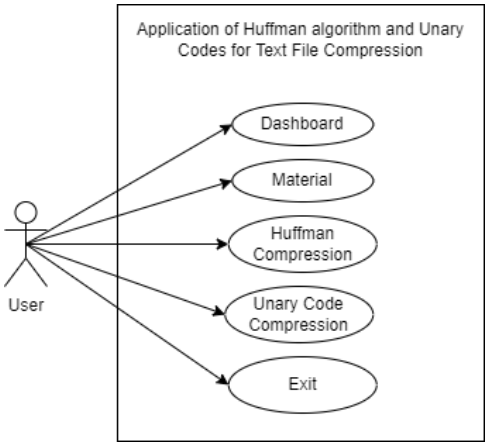

Fig. 1 Usecase Diagram Compression File

In the usecase picture above, we can understand that users who use the application can access several menus in the application. Among them are the following: Users can open the dashboard page menu to see the dashboard page on the built application; Users can look at the material page menu to see the basic theory of the algorithm used in building file compression applications with huffman algorithms and unary codes; Users can access the huffman compression page on the built application, to perform the file compression process with the application of the huffman algorithm; Furthermore, the user can access the compression of the file by implementing the unary codes algorithm in the built application.

Flowchart Huffman and Unary Codes

As for flowcharts in the process of Huffman and unary codes in the research application of huffman algorithms and unary codes for compression of text files. To find out in detail in each process can be seen in the figure 2 below.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

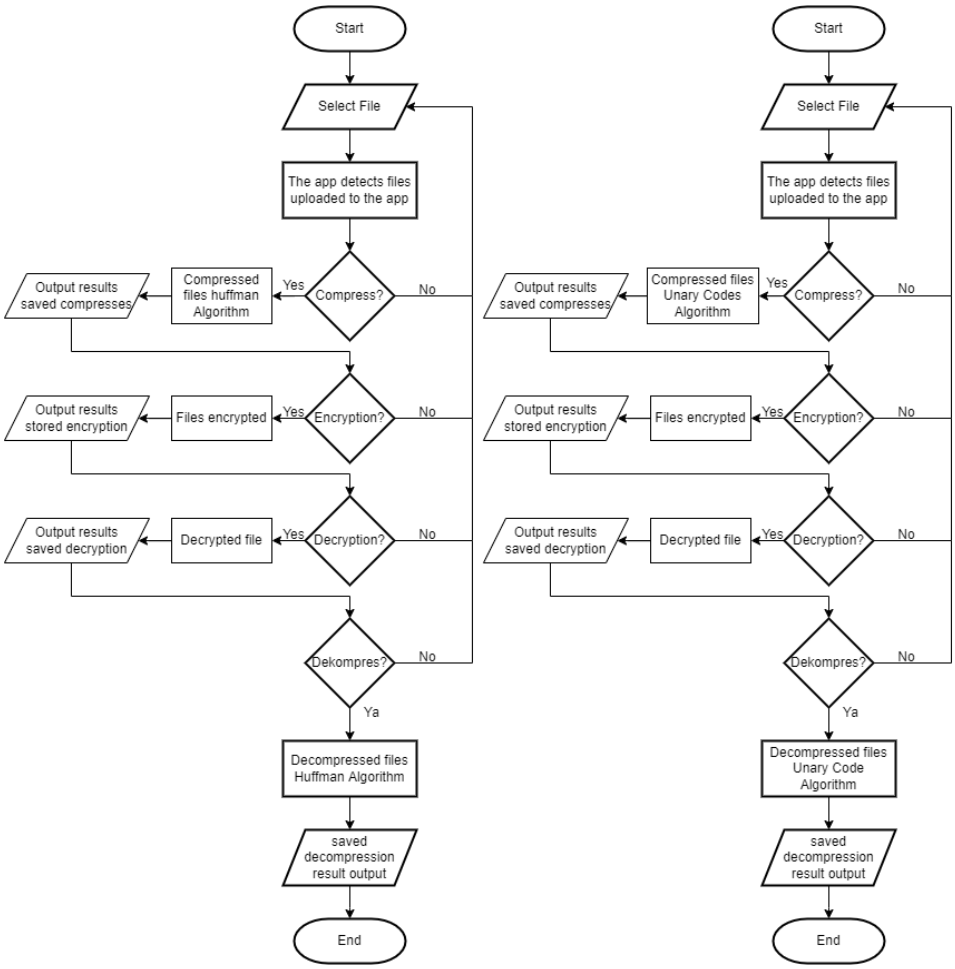

Fig. 2 Flowchart Huffman and Unary Codes

RESULT

System Implementation

In this discussion is a presentation of the results of applications that have been built using netbeans 8.2 software and the Java programming language. Among them are the following:

Dashboard Page

The appearance of the application built in the dashboard page menu can be seen in figure 3 below.

Fig. 3 Dashboard Page

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Material Page

The appearance on the application built in the material page menu can be seen in figure 4 below.

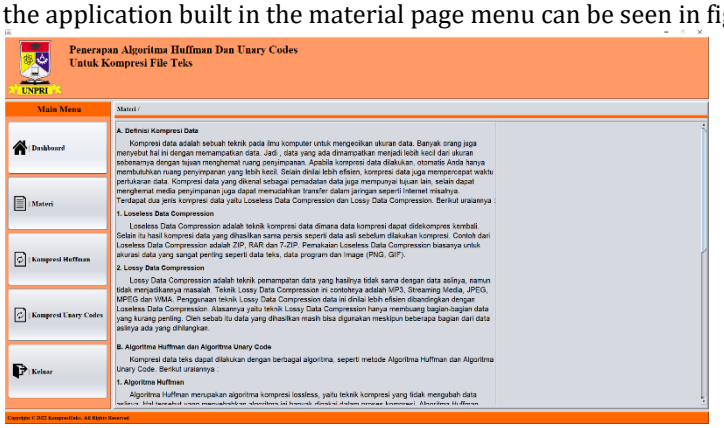

Fig. 4 Material Page

Huffman Compression Page

As for the display on the application built in the huffman compression page menu, it can be seen in the following figure 5.

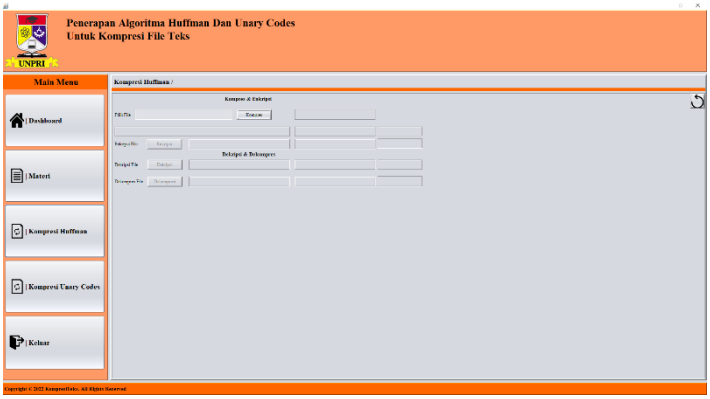

Fig. 5 Huffman Compression Page

Unary Codes Compression Page

The appearance on the application built in the menu of the unary codes compression page can be seen in the following figure 6.

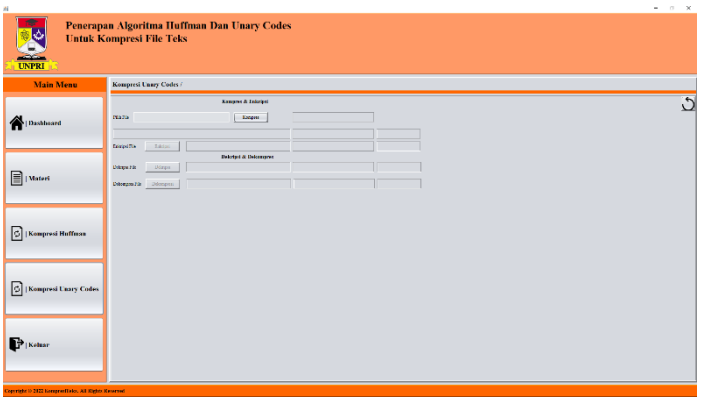

Fig. 6 Unary Codes Compression Page

DISCUSSIONS

Process of Huffman Algorithm

The following is an example of the text compression process in the huffman algorithm, is as follows. In this test example, the compression stage of a text file containing 100,000 strings, including 45,000 characters "A", 13,000 characters "B", 12,000 characters "C", 16,000 characters "D", 9,000 characters "E", 5,000 characters "F".

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

This test will be carried out in 3 ways, namely by using ASCII (American Standart Code For Information Interchange) codes and huffman. The comparison can be seen in the following description.

Stage 1 ASCII Code

Table 1.
ASCII codes A, B, C, D, E, F

Character	ASCII	Binary
A	065	01000001
B	066	01000010
C	067	01000011
D	068	01000100
E	069	01000101
F	070	01000110

Then the encode stages for each character are as many as:

Character "A"
45,000 x 8 bits (01000001) = 360,000 bits
Character “B”
13.000 x 8 bits (01000010) = 104.000 bits
Character “C”
12.000 x 8 bits (01000011) = 96.000 bits
Character “D”
16.000 x 8 bits (01000100) = 128.000 bits
Character “E”
9.000 x 8 bits (01000101) = 72.000 bits
Character “F”
5000 x 8 bits (01000110) = 40.000 bits
Total count = 800,000 bits

Stage 2 of the Huffman Code

Table 2.
Huffman Codes A, B, C, D, E, F

Character	Frequency	Chance	Huffman Code
A	45000	3/13	0
B	13000	3/13	101
C	12000	1/13	100
D	16000	1/13	111
E	9000	1/13	1101
F	5000	1/13	1100

Then the encode stages for each character are as many as:

Character "A"
45,000 x 1 bits (01000001) = 45,000 bits
Character "B"
13,000 x 3 bits (01000010) = 39,000 bits
Character "C"
12,000 x 3 bits (01000011) = 36,000 bits
Character "D"
16,000 x 3 bits (01000100) = 48,000 bits
Character "E"
9,000 x 4 bits (01000101) = 36,000 bits
Character "F"
5000 x 4 bits (01000110) = 20,000 bits
Total number = 224,000 bits

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Therefore, we can know that compression using the huffman code can produce a lower bit result of 70% better than we use ASCII codes.

Process of Unary Codes Algorithm

The following is an example of the text compression process in the unary code algorithm, which is as follows.

Stage 1

Is to provide input in the form of characters contained in the document .txt as an example is:
HAI APA KABAR RIA?
Number of strings = 18.

Stage 2

Namely the process of providing string inputs, is sorted starting from the largest into the frequency table, which can be seen in table 3 and greatest frequency can be seen in table 4.

Table 3.
Frequency Unary Codes

Character	Frequency
H	1
A	6
I	2
(space)	3
P	1
K	1
B	1
R	2
?	1

Table 4.
Greatest Frequency Unary Codes

Character	Frequency
A	6
(space)	3
I	2
R	2
H	1
P	1
K	1
B	1
?	1

After that, find frequency of binary code from the cases. Witch can be seen in table 5.

Table 5.
Binary Code Frequency

Character	ASCII	Binary	Frequency	Bit * Frequencies
A	65	01000001	6	48
(space)	32	00100000	3	24
I	73	01001001	2	16
R	82	01010010	2	16
H	72	01001000	1	8
P	80	01010000	1	8
K	75	01001011	1	8
B	66	01000010	1	8
?	63	00111111	1	8
			Total	**144**

(n) is an arrangement of the most to the least frequencies, if there are several symbols with the same frequency then the symbols will be sorted to appear on the string to be compressed. In unary code non-negative is defined as n-1 bit 1 followed by bit 0. the last n is obtained from a modification, which is n-1 bit 1 without being followed by bit 0. The following is the unary code obtained from the example above.

Table 6.
Frequency Unary Codes

Character	ASCII	Unary Code	Number of Bits	Freq	Bit * Frequencies
A	65	0	1	6	6
(space)	32	10	2	3	6
I	73	110	3	2	6
R	82	1110	4	2	8
H	72	11110	5	1	5

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

P	80	111110	6	1	6
K	75	1111110	7	1	7
B	66	11111110	8	1	8
?	63	111111110	9	1	9
				Total	**61**

Stage 3

Then the results of compression are obtained as follows:

HI HOW ARE YOU RIA?

1111001101001111100101111110011111110011101011101100111111110

Stage 4 Binary division divided by 8 bits:

11110011 (ó)
11110011 (ó)
01001111 (O)
10101110 (®)
10010111 (–)
11001111 (Ï)
11100111 (ç)
00011110 (>)

Stage 5

Obtain the text that has been compressed in the document into:

Compressed = ó O–ç ó ® Ï >

From Binary Code = 1111001101001111100101111110011111110011101011101100111111110

Display of Test Process Result On The System

After the implementation of the built system, it can be seen the results of testing on the system. Among them are the following; Display of Compression Result Huffman and Unary Codes. Can be seen in figure 7.

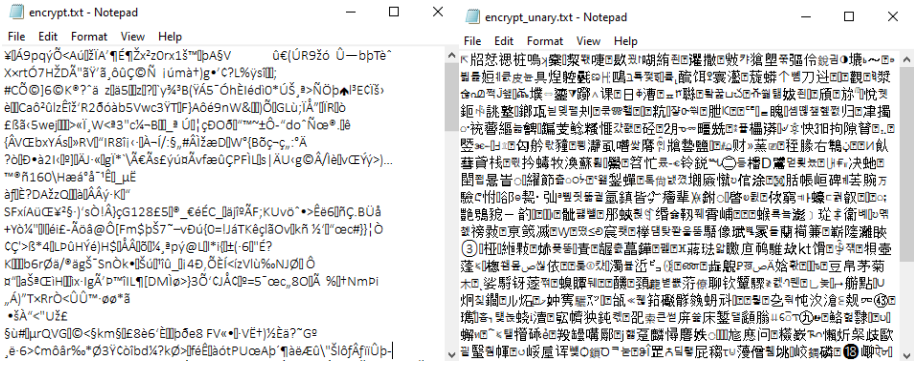

Fig. 7 Huffman and Unary Codes Compression Result

Test Result On The System

At this stage is a discussion of testing. The test was conducted 10 times with 10 article data from different sources. Each article has a different file size and content for testing in the system.

From the tests that have been carried out, the compression and decompression process goes as desired and can restore the text of the article that was inputted. The final result of this discussion is the testing carried out on the system being built. From the initial file size and after the file goes through the compression stage, for the results of each data input can be seen in the following table 7 below.

Table 7.
Huffman And Unary Codes File Compression Results

No.	File Name	Initial File Size	Huffman Final Size	Final Size Unary Codes
1.	input1.txt	4,61 KB	3,35 KB	1,88 KB
2.	input2.txt	5,12 KB	3,43 KB	1,99 KB
3.	input3.txt	7,01 KB	5,05 KB	2,86 KB
4.	input4.txt	4,60 KB	3,48 KB	1,81 KB

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

5.	input5.txt	3,46 KB	2,62 KB	1,34 KB
6.	Input6.txt	5,92 KB	4,06 KB	2,25 KB
7.	input7.txt	4,65 KB	3,74 KB	1,98 KB
8.	input8.txt	4,05 KB	3,2 KB	1,79 KB
9.	input9.txt	5 KB	3,3 KB	1,89 KB
10.	input10.txt	3,57 KB	3,01 KB	1,46 KB

CONCLUSION

There are conclusions that have been drawn from the research carried out and described above: For the stage performed on the huffman algorithm and unary codes in the file compression process is actually relatively the same. The result of file compression of the two algorithms is sorted by character based on their frequency, then binary formation and ending as code formation; The compressed file size can return to its original state after the decompression stage and can be reopened according to the original data file; The file size does not display the percentage results of the comparison, the comparison is only done manually by the author taken from the data of the compression results of each algorithm; From the results of the comparison, it can be concluded that the file compression carried out by the unary codes algorithm is better and faster the process depends also on the speed of performance of the data testing hardware.

REFERENCES

Chulkamdi, M., Sholeh Pramono, and Erni Yudaningtyas. (2015). "Kompresi Teks Menggunakan Algoritma Huffman Dan Md5 Pada Instant Messaging Smartphone Android." *Jurnal EECCIS* 9(1): 103–8.

Fitri Angelina, Siahaan. (2018). "Mengkompresi Teks Terenkripsi El Gamal." 2(1): 26–34.

Kharisma Mahesa, Karpen. (2018). "Rancang Bangun Aplikasi Kompresi Dan Dekompresi Pada Citra Digital Menggunakan Metode Huffman." *Jurnal Processor* 12(1): 948–63. http://ejournal.stikom-db.ac.id/index.php/processor/article/view/367.

Lamsah, and Dito Putro Utomo. (2020). "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel." *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)* 4(1): 311–14.

M. A. Budiman, D. Rachmawati and S. W. Jannah. (2021). "The Comparison of Rice Codes and Even-Rodeh Codes Algorithms for Text Compression". *International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*. pp. 117-120, doi: 10.1109/DATABIA53375.2021.9650288.

Meisiria Sitorus. (2020). "Penerapan Algoritma Golomb Pada Aplikasi Kompresi Short Message Service (SMS)." *Journal of Computer System and Informatics ...* 1(4): 159–65. https://ejurnal.seminar-id.com/index.php/josyc/article/view/73.

Mhd. Rajani, Pane. (2017). "Perancangan Aplikasi Kompresi Menggunakan Metode Shannon Fano Dan Unary Coding Pada File Teks." *Informasi dan Teknologi Ilmiah (INTI)* 12(September): 306–11.

Michael, Simangunsong. (2020). "Perbandingan Algoritma Elias Delta Code Dan Unary Coding Dalam Kompresi Citra Forensik." *Resolusi: Rekayasa Teknik Informatika dan ...* 1(1): 18–26. http://djournals.com/resolusi/article/view/9.

Rahmat, Limbong. (2020). "Implementasi Algoritma Unary Coding Pada Kompresi Citra Ultrasonografi." *Pelita Informatika: Informasi dan ...* 8: 367–70. http://ejurnal.stmik-budidarma.ac.id/index.php/pelita/article/view/2424.

Siahaan, Andysah Putera Utama. (2016). "JURNAL INFORMATIKA Vol. 10, No. 2, Jul 2016 IMPLEMENTASI TEKNIK KOMPRESI TEKS HUFFMAN." *J. Informatika* 10(2): 1251–61.

Sitio, A. S. (2018). TEXT MESSAGE COMPRESSION ANALYSIS USING THE LZ77 ALGORITHM. *INFOKUM*, *7*(1, Desember), 16-21. Retrieved from http://infor.seaninstitute.org/index.php/infokum/article/view/20

Wijaya, B. A., Nugraha, A., Juandry, J., Okinawa, J. and Kinoto, J. (2020) "Film Recommendation System with Social-Union Algorithm: Film Recommendation System with Social-Union Algorithm", *Jurnal Mantik*, 4(2), pp. 1278-1284. doi: 10.35335/mantik.Vol4.2020.932. pp 1278-1284.

Wijaya, B. A., Manalu, A. J. . ., Tarigan, B. A. . . and Silitonga, L. S. . . (2021) "Steganography Text Message Using LSB and DCT Methods", *Jurnal Mantik*, 5(3), pp. 1825-1832. Available at: https://iocscience.org/ejournal/index.php/mantik/article/view/1767 (Accessed: 17July2022).

Yonathan, Febri Dwinata, Helfi Nasution, and Heri Priyanto. (2021). "Aplikasi Pengaman Dokumen Digital Menggunakan Algoritma Kriptografi Hybrid Dan Algoritma Kompresi Huffman." *Jurnal Edukasi dan Penelitian Informatika (JEPIN)* 7(2): 181.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11567
e-ISSN : 2541-2019
p-ISSN : 2541-044X

*name of corresponding author