International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8956-8963

A Review of Data Compression Techniques

Luluk Anjar Fitriya
College Student, Faculty of Electrical Engineering,
Telkom University, Bandung, Indonesia.
Orcid ID: 0000-0002-3576-3855

Tito Waluyo Purboyo
Lecturer, Faculty of Electrical Engineering,
Telkom University, Bandung, Indonesia.
Orcid Id : 0000-0001-9817-3185

Anggunmeka Luhur Prasasti
Lecturer, Faculty of Electrical Engineering,
Telkom University, Bandung, Indonesia.
Orcid Id: 0000-0001-6197-4157

Abstract

This paper presents a review kind of data compression techniques. Data compression is widely used by the community because through a compression we can save storage. Data compression can also speed up a transmission of data from one person to another. In performing a compression requires a method of data compression that can be used, the method can then be used to compress a data. Data that can be compressed not only text data but can be images and video. Data compression technique is divided into 2 namely lossy compression and lossless compression. But which is often used to perform a compression that is lossless compression. A kind of lossless compressions such as Huffman, Shannon Fano, Tunstall, Lempel Ziv welch and run-length encoding. Each method has the ability to perform a different compression. This paper explains how a method works in doing a compression and explains which method is well used in doing a data compression in the form of text. The output generated in doing a can be known through the compression file size that becomes smaller than the original file.

Keywords: Data Compression, compression techniques, lossless compression, Huffman, Shannon Fano, Tunstall, RLE, LZW.

INTRODUCTION

With the rapid development of technology with the support of software and hardware that increasingly facilitate widespread information quickly through the internet around the world. Information obtained can be sent easily via the internet as a the medium of communication for information technology experts. However, not all information can be sent easily. There is a large size that can hinder data transmission quickly and save on existing storage in the computer. To overcome the problem of information or data to be transmitted or transmitted can be done quickly than required a compression that can save storage and transmission of data to be done.

Compression is the process of converting a data set into a code to save the need for storage and transmission of data making it easier to transmit a data.

With the compression of a can save in terms of time and storage that exist in memory (storage). Many compression algorithm techniques can be performed and function properly such as the Huffman, Lempel Ziv Welch, Run Length Encoding, Tunstall, And Shannon Fano methods. The data process of data compression is shown in figure 1.

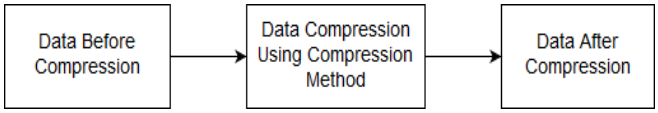

Figure 1: The data process of data compression

In figure 1, explain the process of data compression in general. how the data when not compressed then uncompressed data will be continued and processed by compression method that is lossless compression then the data has been compressed will produce a size smaller than the size of the file before it is compressed.

Compression is the reduction of a file size from a large size to a smaller file size. A compression will be done to facilitate the transmission of a file with a large size and contains many characters. The workings of a compression are by looking for patterns of repetition in the data and replace it with a certain sign. The type of compression has two methods, lossless compression and lossy compression. Lossless compression is the process of converting the original data with compressed

data becomes more concise without reducing the loss of information. Lossy compression is the prose converts the original data into the compression data there are different values, but the value of this difference is considered does not reduce the essential information from the original data.

Here is an explanation of the applicability that exists in a data compression

- Compression for audio
- Compression for text
- Compression for video
- Compression for image

Compression for audio

Audio compression is one form of data compression

option to shrink the size of the audio / video file by the method

- Overflow format: Vorbis, MP3;
- Unlimited format: FLAC; users: audio engineer, audiophiles

Compression at the time of the creation of the audio file and at the moment distribution of the audio file.

Audio compression constraints:

- The development of sound recording is fast and diverse
- Value of audio sample changes quickly

Endless audio codecs have no issues in sound quality,

usage can be focused on:

- the speed of compression and decompression
- The degree of compression
- Support hardware and software

Missing audio codecs available on:

- Audio quality
- compression factor
- the speed of compression and decompression
- Inherent latency of algorithm (essential for real-time streaming)
- Support hardware and software

Compression for text

The decompression process returns the compressed file to the beginning of the text. Decompression results depend on the nature of the compression used, namely Lossless Compression or Lossy Compression. If a lossless compression technique has been performed on a text, the original text can be recovered correctly from the decompressed file (Sayood, 2001). Arithmetic encoding is a compression technique that is lossless compression. Lossy Compression results in the loss of some information, and decompression results can not produce exactly the same text as the original text (Sayood, 2001).

Compression Ratio

The Compression ratio shows the percentage of compression made against the original file. The compression ratio is derived from the equation: Difference in size Compression ratio = x 100% (1) original file

The difference in size = original file - compression file (2) The higher the compression ratio the smaller the resulting compression file, the better compression result.

Compression for video

The video is a technology for capturing, recording, processing, transmitting, and rearranging moving images. Usually use celluloid film, electronic signal, or digital media.

To digitize and store full-motion video clips for 10 minutes into a computer, it must transfer data in large quantities in a short time. To reproduce one frame of a 24-bit digital video component, computer data required is almost 1 MB, video not converted with layer for 30 seconds will meet the hard disk charged gigabyte. Full-size video and full-motion requires a computer that can transmit data of approximately 30 MB per second. Major technological bottlenecks can be overcome using digital video conversion schemes or codecs (coders / decoders). Codecs are algorithms used to convert (code) a video to be transmitted, then decoded directly for fast playback. Different codecs are optimized for different delivery methods (for example, from hard drives, CD-ROMs, or via the Web). The purpose of compression / video conversion is: minimization of bit rate in the digital representation of video signal, maintaining the desired signal quality level, minimizing codec complexity (coder and decoder-encoding and decoding), and delay or delay

In other words video compression is one form of data compression that aims to shrink the video file size. Video compression refers to reducing the amount of data used to represent a digital video image, and is a combination of compression space of images and temporal compression of motion.

Compression for image

Graphic Interchange Format (GIF) created by Compuserve in 1987 to store multiple images with bitmap format into a file that is easy to change on a computer network. GIF is the oldest graphic file format on the Web. GIF supports up to 8-bit pixels, meaning a maximum number of colors 256 colors (28 = 256 colors), 4-pass interlacing, transparency and using variants of the Lempel-Ziv Welch (LZW) compression algorithm.

There are two types of GIFs, among others:

GIF87a: support with interlacing and capacity of multiple files. The technique was called GIF87 because in 1987 this standard was found and made standard.

GIF89a: is a continuation of the GIF87a specification and additions to transparency, text, and animation of text and graphics. Portable Network Graphic (PNG) format is designed to be better with the previous format that GIF has been legalized. PNG is designed for lossless algorithms for storing a bitmap image.PNG has a feature equation with GIF one of which is (multiple images), improving something eg (interlacing, compression) and adding the latest features. Support for Web where plug-ins can be made on web browsers. Joint Photographic Experts (JPEG, read jay-peg, [6]) are designed to compress some full-color or gray-scale of an original image, such as the original scene in the world. JPEGs work well on continuous tone images such as photographs or all the realm of art that permit the real; but not very good at the sharpness of images and the art of coloration such as writing, simple cartoons or drawings that use many lines. JPEGs already support for 24-bit color depth or equal to 16.7 million colors (224 = 16,777,216 colors) .The advantages of JPEG and type - they seem to be on the same steps as interlaced GIFs. JPEG 2000 is the most recent image compression technique. Jpeg 2000 is the development of Jpeg, which the number of bit errors are relatively low, rated, transmission and have a good quality compared with Jpeg. Jpeg 2000 applies lossy and lossless compression techniques. And the use of ROI coding (Region of interest coding). JPEG 2000 is designed for internet, scanning, digital photography, remote sensing, medical imagery, digital library and E-commerce.

Since the 80s we remember that the International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) have collaborated to make standardization for grayscale compression and image imaging, which we know by the name JPEG (Joint Photographic Experts). With the rapid development of multimedia technology requiring high-performance compression techniques, in March 1997 a new standard compression project for image was created, known as JPEG 2000. This project created a new coding system for several different image types (bi-level, grey level, Color, Multi component) with different characteristics (Natural Images, scientific, medical, remote sensing, text, etc.).

LITERATURE REVIEW

Shannon Fano Algorithm

Data Compression is a technique where a compression make a very useful compression technique which is uses in a implode compression method which are use in zip file or .rar format [2]. Shannon Fano algorithm can be implemented using VHDL coding using ModelSim SE 6.4 simulator and the data gets compression using the algorithm. If we want to find out how much amount of data get compression using these algorithm, then which can be done by following equation,

Amount Of Compression (Ratio Compression)

$$\frac{Amount\ Data\ Bits\ Before\ Compression}{Amount\ Data\ Bits\ After\ Compression}$$

We remark that using compression for data can to improved with encryption. Unfortunately, conventional Shannon Fano code usually relatively, this paper [1] proposed two algorithm to reduce the length of Shannon Fano code. In those applications the improved Shannon Fano Coding technique is very useful [1].

Run Length Encoding

RLE (Run Length Encoding) algorithm is one algorithm that can be used to compress data so that the size of the data produced is lower than the actual size. The example discussed this time is the cost and return of data from a sentence.

RLE (Run Length Encoding) is the easiest form of lossless data compression technique where a series of data with the same value in sequence will be saved into a data. This algorithm is very useful in data that has a lot of data with the same values in sequence like icon files, line drawings, and animations. This algorithm is not suitable for normal data because it will increase.

Flowchart Run Length Encoding Algorithm was given Figure 2.

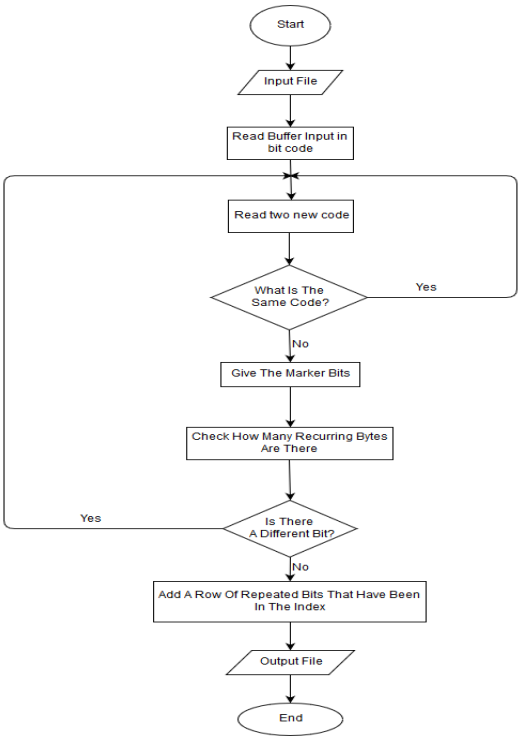

Figure 2: Flowchart Diagram Compression Data With RLE

Lempel Ziv Welch

In general the LZW algorithm is a lossless compression algorithm and uses a dictionary. LZW compression will form a dictionary during the compression process takes place.. LZW algorithm can be implemented in many compression applications. In this experiment [5] have provided a comparison between the conventional LZW coding and proposed MLZW coding [5]. Compression result in term of dictionary. Output from LZW algorithm is amount of bit or codeword compression result must be small than file before compression. Algorithm is adapted for Unicode standard, it may be used very easily for any Bangla compression text [5].

Concept from this algorithm is to find the new dictionary from a new character. LZW method use variable word width dictionary to balance the compression and decompression file.

Flowchart Lempel Ziv Welch Algorithm was given Figure 3.

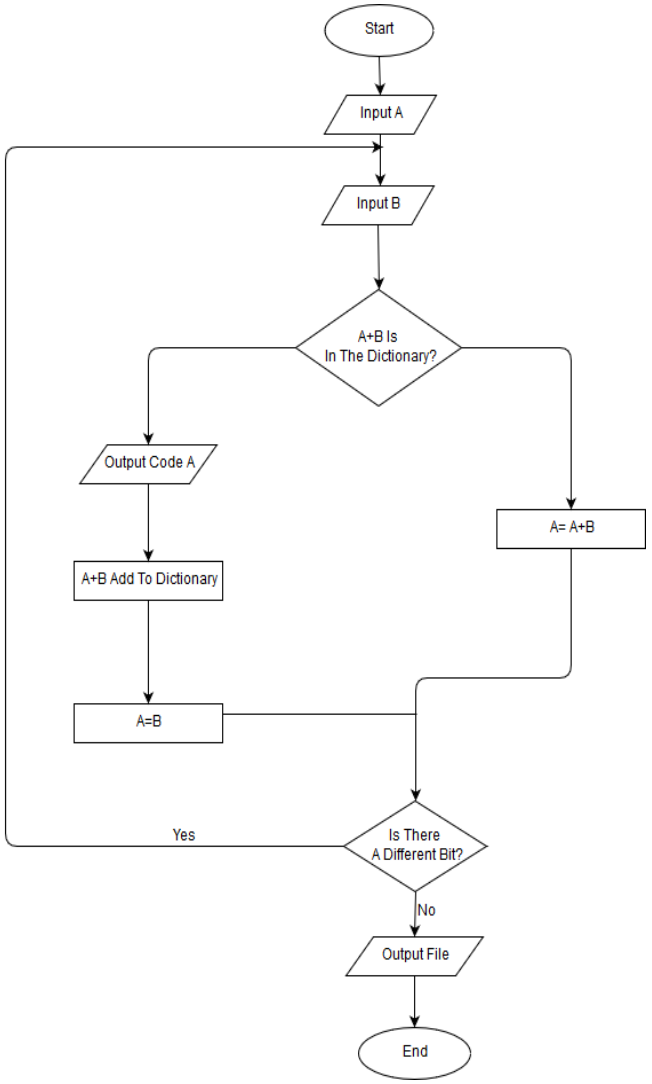

Figure 3: Flowchart Diagram Compression Data With LZW

Tunstall

The Tunstall algorithm was the subject of Brian Parker Tunstall's thesis in 1967 while at the Georgia Institute of Technology. The subject of this thesis is "Synthesis of noiseless compression codes." The Tunstall Algorithm is a code that maps the source symbol to a fixed number of bits as well as sorts the stochastic source with a variable length codeword.

Huffman

The Huffman method applied in this paper is a static type, which is done twice (two-pass) reading of a data / file to be compressed. To calculate the appearance of characters in the formation of a Huffman tree and encode with the Huffman code symbol.

This method encodes symbols or characters with the help of a binary tree by combining the two smallest frequencies to form a code tree.

Huffman codes are based on the number of character frequencies that often appear. The larger the Huffman code frequency the less number of bits produced. Conversely, the fewer character appearances the more number of bits produced during a compression.

This Huffman compression algorithm includes methods lossless compression. lossless compression is a compression technique that does not change the original data information to a smaller size.

The rationale of Huffman's algorithm is that each ASCII character is usually represented by 8bits. So for example a file contains a row of characters "AABCD" then the number of bits in the file 5 x 8 is 40 bits or 5 bytes. If each character is given a code eg A = 0, B = 10, C = 111, D = 110 then we only need a file with the size of 10bits (0010111110) .that note that the codes must be unique or in other words a code can not be formed from other codes.

To specify codes with code criteria must be unique and the characters that often appear are made small by the number of bits, we can use the Huffman algorithm to compress, and form the Huffman Tree.

Flowchart Huffman Algorithm was given Figure 4.

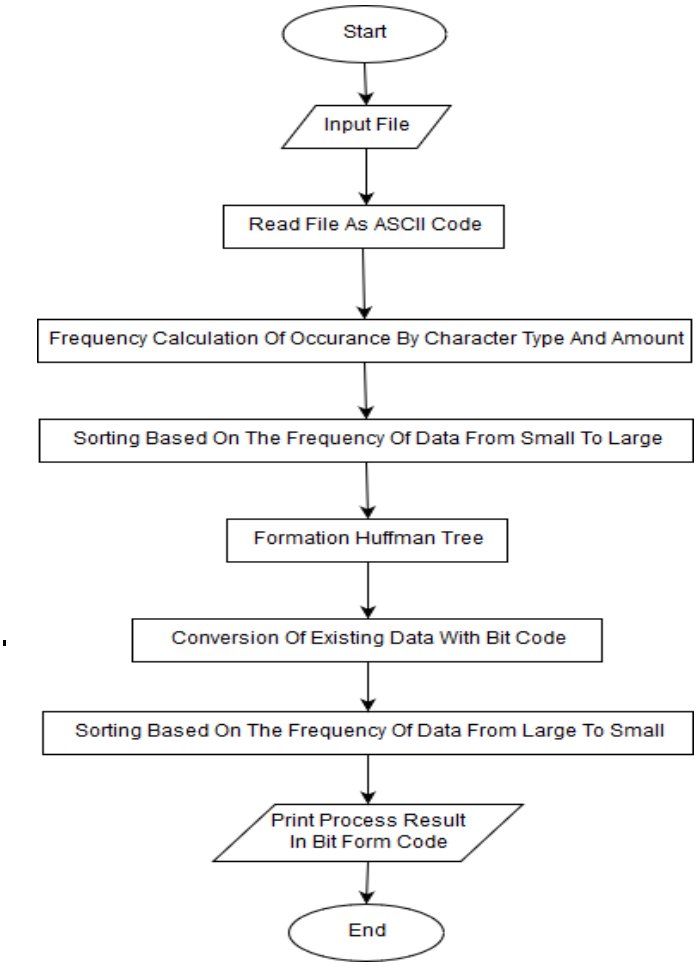

Figure 4: Flowchart Diagram Compression Data with Huffman

REVIEW AND DISCUSSIONS

No.	Methods	Author's Name	Description
1	Prefix Code	Yupeng Tai et al[27]	This paper describe about redundancy code problem for algorithm. This algorithm improved with more algorithm eg. Huffman. Propose this paper is could speed up the classical one for several times. Condition is usually satisfied before whole Huffman.
2	Tunstall	H.Hashempour et al[28]	This paper describe if Tunstall algorithm compressed test data streams. This algorithm requires a simple algorithm to compression algorithm. On benchmark circuit have been provided to substantiate the validity of this approach to an ATE environment.
3	Golomb Code	Mattew F.Brejza et al[29]	This paper have extended ang generalized the code of give to the propose another code. This also shown accrose a wide range in application scenario that their family of proposode schemes outperform several benchamarkers.
4	Lempel Ziv Welch	G.R.Gnana King et al[1]	This paper discusses compression and transmission on a compound image. This journal uses the decomposition technique and the modified image is then compressed using the Lempel Ziv Welch algorithm. The

			performance measures the PSNR value and the MSE value (Mean Square Error) is calculated. When compared with using embedded compression algorithm higher PSNR value using LZW algorithm.
		Linkon Barua et al[2]	In this paper shows the MLZW algorithm for Bangla compression text. The degree of compression depends on the frequency of the join character and the dependent vowel sign. The experimental results show that the proposed MLZW algorithm improves the compression level effectively than the conventional LZW algorithm. As the proposed algorithm adapted to the Unicode standard, it can be used very easily for Bangla compression text. The proposed algorithm can also be used on small memory devices and text communications. These results prove that the proposed MLZW algorithm can be an appropriate candidate for Bangla's text compression. The proposed MLZW algorithm can be applied to other languages that have similar characteristics such as Bangla text.
5	Huffman	Haoqi Ren[3]	This paper described a new test data compression technique based on Reversed Leading Bits Coding and Huffman Coding (RLBC-HC). RLBC-HC divides the test data into codeword segments with proper size, and achieves a high Huffinan coding efficiency consequently. The decoder hardware is very simply, and easy to be implemented in hardware. The performance of the proposed pattern generator has been demonstrated by experimental results and comparisons against other test data compression techniques.
		Djuned F et al[4]	The method we propose, especially the LZW slice bits proves that having a compression result is better than using 8 bit LZW. In addition to pre-processing techniques, this is done well to create a more monotonous grayscale image so it can be optimally compressed by Adaptive Huffman. The file output is a structured file containing the original image metadata and unstructured compression bit stream. This is lossy compression. LZW with bit storage problem when used for data compress or uncompressed random data. But by using the data slice bit technique can look more harmonic by LZW, because by using it then the data will make more repetitive bit stream.

CONCLUSION

Using the compression technique can reduce the number of file sizes. data that has a large size into a smaller size that can save storage in a computer. data compression can be implemented on a text, photo, and video data. various compression algorithm techniques have advantages and disadvantages in doing a compression.

REFERENCES

[1] Xiaoyu Ruan, Rajendra Katti, "Using Improved Shannon-Fano-Elias Codes Data Encryption", *Proceedings of ISIT Conference*, North Dakota State University Fargo, July 9-14, 2006.

[2] Mr.Mahesh Vaidya, Mr.Ekjot Singh Walia, , and Mr.Aditya Gupta, "Data Compression Using Shannon-Fano Algorithm Implemented By VHDL", *IEEE International Conference on Advances in Engineering & Technology Research*, August 01-02,2014**.**

[3] Lung-Jen Lee, Wang-Dauh Tseng, Rung-Bin Lin, and Cheng-Ho Chang, " Pattern Run-Length for Test Data compression", *IEEE Transaction on Computer-Aded Design of Integrated Circuits And System,* Vol.31, No.4,April,2012.

[4] Mohammad Arif, R.S.Anand, "Run Length Encoding for Speech Data Comprassion", *IEEE International Conference on Computional Intelligence and Computing Research*, 2012.

[5] Linkon Barua, Pranab Kumar Dhar, Lamia Alam, and Isao Echizen, " Bangla Text Compression Based on Modified Lempel-Ziv-Welch Algorithm", *International Conference on Electrical, Computer and Communication Engineering* (*ECCE*), Bangladesh, February 16-8,2017.

[6] G.R.Gnana King, C.Seldev Christoper, And N.Albert Singh, " Coumpound Image Compression Using Parallel Lempel Ziv-Welch Algorithm", *IET Chennati Fourth International Conference on Sustainable Energy and Intelligent System*,Chennati,December 12-14, 2013.

[7] Haoqi Ren, " A data Comprssion Technique based on Resersed Leading Bits Coding and Huffman Coding", *International Conference on Communication and Networking*, China, 2015.

[8] Djuned Fernando Djusdek, Hudan Studiawan, and Tohari Ahmad, " Adaptive Image Compression Using Adaptive Huffman and LZW", *International Conference on Information, Communication Technology and System*, 2016.

[9] Tsutomu Kawabata, " Enumerative Implementation of Lempel-Ziv-77 Algorithm", *ISIT*, Toronto, Canada, July 6-11, 2008.

[10] Adrian Traian Murgan, Radu Radescu, " A Comprison of Algorthm for Lossless Data Compression Using the Lempel-Ziv-Welch Type Methods", Bucharest.

[11] Victor Amrizal, " Implementasi Algoritma Kompresi Data Huffamn Untuk Memperkecil Ukuran File MP3 Player", 2-14,2010.

[12] Cut Try Utari, " Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi dan Dekompresi File Citra", *Jurnal TIMES, Vol.V No.2*, 24-31, 2016.

[13] M.VidyaSagar, J.S, Rose Victor, "Modified Run Length Encoding Scheme for High Data Compression Rate", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vijayawada, December 2013.

[14] K. Ashok Babu and V. Satish Kumar, "Implementation of Data Compression Using Huffman Coding", *International Conference on Methods and Models in Computer Science*, India, 2010.

[15] Harry Fernando, "Kompresi data dengan algoritma Huffman dan algoritma lainnya", ITB, Bandung.

[16] Mohammed Al-laham1 & Ibrahiem M. M. El Emary, "Comparative Study between Various Algorithms of Data Compression Techniques", *IJCSNS International Journal of Computer Science and Network Security*, Jordan, April 2007.

[17] S.R.Kodituwakku and U.S.Amarasinghe, "Comparison of Lossless Data Compression Algorithms for Text", *Indian Journal of Computer Science and Engineering, Sri Lanka*.

[18] Rhen Anjerome Bedruz and Ana Riza F. Quiros, "Comparison of Huffman Algorithm and Lempel-Ziv Algorithm for Audio, Image and Text Compression", *IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM)*. Philippines, 9-12 Decmber 2015.

[19] C. Oswald, Anirban I Ghosh and B.Sivaselvan, "Knowledge Engineering Perspective of Text Compression", *IEEE INDICON*, India, 2015.

[20] Ardiles Sinaga, Adiwijaya and Hertog Nugroho, "Development of Word-Based Text Compression Algorithm for Indonesian Language Document", *International Conference on Information and Communication Technology (ICoICT)*, Indonesia, 2015

[21] Manjeet Kaur, "Lossless Text Data Compression Algorithm Using Modified Huffman Algorithm", *International Journal of Advanced Research in Computer Science and Software Engineering*, india, July 2015

[22] Tanvi Patel, Kruti Dangarwala,Judith Angela, and Poonam Choudhary, "Survey of Text Compression Algorithms", *International Journal of Engineering Research & Technology (IJERT)*, India, March 2015

[23] Shmuel T. Klein and Dana Shapira, "On Improving Tunstall Codes", *Information Processing & Management*, Israel, September 2011.

[23] Mohammad Hosseini, "A Survey of Data Compression Algorithms and their Applications", Applications of Advanced Algorithms, At Simon Fraser University, Canada, January 2012

[24] Maria Roslin Apriani Neta, "Perbandingan Algoritma Kompresi Terhadap Objek Citra Menggunakan JAVA", Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013 (SEMANTIK 2013), Semarang, November 2013.

[25] Dr. Shabana Mehfz1, Usha Tiwad, "A Tunstall Based Lossless Compression Algorithm for Wireless Sensor Networks", *India Conference (INDICON)*, 2015 Annual IEEE, India, 2015.

[26] Dr. Ahmad Odat, Dr. Mohammed Otair and Mahmoud Al-Khalayleh, "Comparative Study between LM-DH Technique and Huffman Coding Technique", *International Journal of Applied Engineering Research*, India.

[27] Yupeng Tai, Haibin Wang, " A Fast Algorithm for Calculating Minumum Redudancy Prefix Codes with Unsorted Alphabet", *International CHINACOM*,China.

[28] H.Hashempour,L.Schiano, and F.Lombardi , "Error-Resilient Test Data Compression Using Tunstall Code", Boston Mass 02115.

[29] Matthew F.Brejza, Tao Wang, Wenbo Zhang, David Al-Khalili, Robert G. Maunder, Basher M. Al-Hashimi and Lajoz Hanzo, " Exponential Golomb and Rice Error Correction Codes for Generalized Near-Capacity Joint Source and Channel Coding, UK.