



# InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Available online at : <http://bit.ly/InfoTekJar>

## Perbandingan Metode Huffman dan Run Length Encoding Pada Kompresi Document

Pujianto<sup>1</sup>, Mujito<sup>2</sup>, Basuki Hari Prasetyo<sup>3</sup>, Danang Prabowo<sup>1</sup>

<sup>1</sup> Teknik Informatika Universitas Muhammadiyah Metro, Jl. KH Dewantara No.116 Iringmulyo, Kota Metro, Lampung, 34381, Indonesia

<sup>2</sup> Sistem Informasi Universitas Muhammadiyah Metro, Jl. KH Dewantara No.116 Iringmulyo, Kota Metro, Lampung, 34381, Indonesia

<sup>3</sup> Teknik Informatika Universitas Budi Luhur, Jl.Ciledug Raya, Petukangan Utara, Jakarta, 12260, Indonesia

### KEYWORDS

Huffman, Run Length Encoding, Kompresi, Metode

### CORRESPONDENCE

Phone: 081288042180

E-mail: [pujianto@ummetro.ac.id](mailto:pujianto@ummetro.ac.id)

### A B S T R A C T

Penggunaan media penyimpanan yang efektif dan efisien sangat didambakan oleh semua orang, ukuran besar dan kecilnya sebuah document sangat mempengaruhi media penyimpanan dan juga ketika pengiriman document juga mempengaruhi kecepatan pengiriman. Dengan memanfaatkan beberapa metode Kompresi kita dapat memperkecil ukuran pada sebuah document. Beberapa diantaranya adalah metode Huffman dan Metode Run Length Encoding dimana pada penelitian ini akan membahas mengenai perbandingan dari metode Huffman dalam melakukan proses Kompresi dengan metode Run Length Encoding menggunakan beberapa jenis document yaitu document yang berjenis pdf, word, excell dan powerpoint. Dari hasil penelitian menunjukkan bahwa metode huffman lebih baik didalam melakukan proses compress document dibandingkan dengan metode Run Length Encoding. Hal ini dapat dilihat dari ukuran file setelah dilakukan proses Kompresi, ukuran file sesudah Kompresi pada metode huffman mengalami penyusutan sedangkan ukuran file sesudah Kompresi pada metode Run Length Encoding terdapat 1 file mengalami peningkatan.

### INTRODUCTION

Perkembangan teknologi yang sangat pesat berdampak pula pada metode-metode compressi file atau document. terdapat beberapa jenis metode didalam mengcompress sebuah document diantaranya adalah metode Huffman, Run Length Encoding, LZW, LZ77 dan lain sebagainya. Adapun tujuan dari metode-metode tersebut adalah memperkecil ukuran sebuah file sehingga tempat penyimpanan menjadi efektif serta efisien, juga ketika mengirimkan file tersebut menjadi lancar dan tidak mengalami hambatan karena ukuran sebuah document. penelitian mengenai berbagai macam jenis Kompresi juga telah dilakukan oleh beberapa penelitian yaitu Andysah Putera Utama Siahaan yang berjudul Impelementasu Teknik Kompresi Teks Huffman pada penelitian tersebut mengcompress file dengan jenis teks dengan beberapa tahapan atau fase yaitu Yang

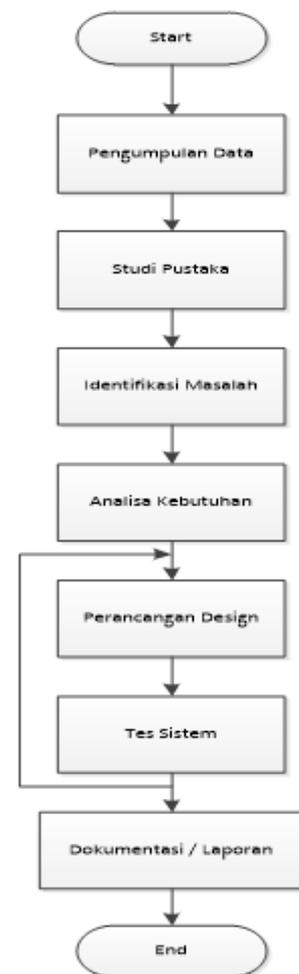
pertama adalah kelompok karakter. Yang kedua adalah untuk membangun pohon Huffman. Yang ketiga adalah encoding, dan yang terakhir adalah pembangunan bit kode.pada penelitian tersebut melakukan uji coba dengan panjang text 33. kemudian hasil compress menjadi 12 text sehingga tingkat kompresi menjadi 63% dari teks aslinya [1] penelitian serupa juga dilakukan oleh Akhmad Pahdi yang berjudul Algoritma Huffman Dalam Pemampatan Dan Enksripsi Data. pada penelitian tersebut menggunakan tiga tahapan utaman dalam penggunaan algoritma huffman pertama tahapan pembentukan pohon Huffman, kedua tahapan mengganti rangkaian karakter menjadi rangkaian bit yang lebih pendek sehingga mendapatkan ukuran file yang lebih kecil, tahapan ketiga merubah setiap 8 rangkaian bit menjadi rangkaian karakter yang baru, pada penelitiannya mencoba mengcompress file dengan jenis Excell, Word, Powerpoint dan file sql. hasil penelitiannya menghasilkan hasil compress sebesar 30% dari ukuran aslinya [2]. Penelitan

selanjutnya dilakukan oleh Suhermanto dan Rahmat Arief yang berjudul Optimasi Waktu Dekompresi Lossy Menggunakan Metode Pengelompokan Jumlah-Bit Kode Huffman Pada Data Lisa Satelit Lapan-A3 pada penelitian tersebut peneliti mengusulkan metode perbaikan algoritma dekomposisi data LISA real-time lossy menggunakan pengelompokan jumlah-bit pada algoritma dekoda huffman dan dari uji coba pada 6 buah contoh data menunjukkan ratio kompresi yang sangat baik [3] sedangkan penelitian yang membahas mengenai Run Length Encoding dilakukan oleh Ulfa Lu'luilmaknun dan Nilza Humaira Salsabila yang berjudul Penggunaan Metode Run Length Encoding Untuk Kompresi Data. Pada penelitiannya mengkompresi file dengan format jpg, png, bmp dan tiff. dan menggunakan image sebanyak 28 citra RGB (Red, Green, Blue) serta 28 Citra grayscale. dan penelitian tersebut menghasilkan kesimpulan bahwa metode RLE tidak efektif jika ratio kompresi lebih dari 100% karena memiliki sedikit perulangan warna pada piksel-pikselya. dan metode RLE hanya efektif pada 1 citra pada tipe RGB sedangkan untuk citra grayscale aktif efektif pada 6 Citra [4]. Kemudian penelitian oleh Yosia Adi jaya, Lukas Chrisantyo dan Willy Sudiarto Raharjo yang berjudul Pengembangan dan Analisis Kombinasi Run Length Encoding dan Relative Encoding Untuk Kompresi Citra. pada penelitian tersebut meneliti kompresi data dengan format bitmap 24 bit dan bitmap 8 bit. dengan hasil penelitian untuk pada kompresi bitmap 24 bit menghasilkan rata-rata rasio kompresi sebesar 100.847% sedangkan untuk bitmap 8 bit rata-rata rasio sebesar 97.713% [5] Penelitian lainnya juga dilakukan Damo Willfrid Midukta Simamora, Garuda Ginting dan Yasir Hasan yang berjudul Implementasi Algoritma Run Length Encoding pada kompresi file MP3. pada penelitiannya menggunakan file tipe hanya MP3 dengan ukuran awal sebuah file sebesar 4,382 kilobytes dan setelah berhasil dikompres menjadi 4,321 kilobytes dengan ratio kompres 99%. dan pada saat di dekompresi ukuran awal file dekompres sebesar 4,321 kilobytes dan setelah didekompresi ke ukuran awal sebesar 4,321 kilobytes. sehingga file tidak mengalami perubahan ketika dikembalikan ke kondisi semula [6] Adapun penelitian yang serupa membahas mengenai kompresi algoritma Huffman dan Run Length Encoding dilakukan oleh Eka Prayoga dan Kristien Margi Suryaningrum yang berjudul Implementasi Algoritma Huffman dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web. penelitiannya tersebut meneliti tentang kompresi menggunakan gabungan antara algoritma Huffman dan Run Length Encoding dimana file yang akan dikompres berjenis .txt pada penelitiannya menggunakan 5 data uji dan hasil penelitiannya menunjukkan ukuran file hasil dekompress tidak seperti semula karena proses kompresi yang bersifat lossy. dan perbedaan dari penelitian ini adalah file yang akan diujicoba tidak hanya .txt melainkan aplikasi office lainnya yakni microsoft word, excell, powerpoint dan lain sebagainya [7] kemudian penelitian sejenis yang membahas kompresi Huffman dan Run Length Encoding dilakukan oleh Helbert Sinaga, Poltak Sihombing dan Handrizal yang berjudul perbandingan Algoritma Huffman dan Run Length Encoding Untuk Kompresi File Audio. Penelitian tersebut dilakukan untuk menganalisis perbandingan hasil kompresi dan dekompresi file audio .mp3 dan .wav Pada file audio berformat \*.wav dan \*.mp3, algoritma Huffman dapat menghasilkan rasio kompresi dengan persentase

yang lebih kecil dibandingkan dengan algoritma Run Length Encoding dengan rata-rata rasio kompresi file .mp3 dengan lima kali percobaan menggunakan algoritma Huffman adalah 1.204% sedangkan menggunakan algoritma RLE adalah -94.44%, dan Rata-rata rasio kompresi file .wav dengan lima kali percobaan menggunakan algoritma Huffman adalah 28.954% sedangkan menggunakan algoritma RLE adalah -45.91%. Sehingga pada penelitian tersebut menarik kesimpulan bahwa algoritma Huffman lebih baik dari pada algoritma RLE dalam kompresi file audio .mp3 maupun .wav [8]

## METHOD

Metode penelitian digunakan untuk menggambarkan proses penelitian. pada gambar 1 menunjukkan metode yang digunakan pada penelitian.



Gambar 1. Metode Penelitian

Penjelasan dari Gambar 1 adalah pertama-tama kita lakukan Pengumpulan Data, Pada tahap ini adalah mengumpulkan data-data dari sebuah pokok permasalahan dari topik yang diangkat oleh penulis, yaitu dengan Observasi yang terdiri dari wawancara. Kemudian Studi Pustaka, Setelah data – data yang dibutuhkan terkumpul, selanjutnya adalah mencari data atau fakta yang real melalui studi pustaka. Selanjutnya Identifikasi

Masalah, Dari data nyata yang terkumpul maka selanjutnya dapat diidentifikasi suatu masalah dan permasalahannya yang ada dengan pembatasan. Kemudian Analisa Kebutuhan, Dari hasil identifikasi masalah yang diatas, selanjutnya baru dapat dilakukan analisa kebutuhan yang menunjang dalam perancangan system kompresi file huffman dan Run Length Encoding ini berdasarkan tinjauan pustaka, yaitu meliputi kebutuhan materi kompresi huffman dan Run Length Encoding, teori perancangan sistem atau program yang interaktif serta template atau platform dimana perancangan akan dilakukan. Kemudian Perancangan Desain, Pada tahap ini adalah merancang tampilan tatap muka pengguna yang mudah digunakan menurut kaidah interaksi manusia dengan komputer dan konten-konten yang ada didalamnya seperti, struktur menu, tombol. Kemudian proses Tes Sistem. Pada tahap ini adalah mengujikan apa saja yang telah diteliti kemudian dirancang kedalam bentuk model program. Jika belum sesuai dan atau masih ada kekurangan dalam perancangan model program ini dapat ditambah dalam rancangannya bahkan dirancang ulang pada tahap perancangan disain untuk mendapatkan hasil yang sesuai. Kemudian yang terakhir adalah Dokumentasi / pembuatan Laporan, Tahap dokumentasai atau pembuatan laporan adalah memaparkan hasil penelitian yang dilakukan dari tahap awal hingga akhir dan diimplementasikan kedalam bentuk laporan jurnal

## RESULTS AND DISCUSSION

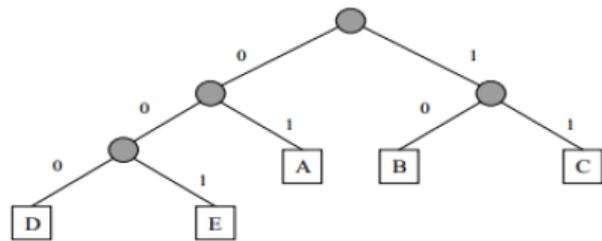
### A. Kompresi

Kompresi adalah suatu proses pengkodean informasi menggunakan bit-bit atau information-bearing pada unit lainnya yang lebih rendah jika dibandingkan dengan representasi data yang tidak terkodekan menggunakan sistem encoding tertentu. pada pihak pengirim data harus menggunakan algoritma kompresi data yang sudah distandarisasi dan pada pihak penerima pun harus menggunakan suatu teknik dekompresi yang sama dengan yang dikirimkan oleh pengirim hal ini dimaksudkan supaya data yang diterima oleh penerima dapat dibaca kembali dengan benar [9]. Pada kompresi data terdapat 2 buah tipe teknik kompresi yang pertama teknik lossless dan yang kedua adalah teknik Lossy. Pada teknik Lossless Metode Lossless merupakan sebuah metode kompresi dimana pada data yang akan dikompres adalah data kompresi yang memungkinkan data aslinya dapat disusun dan dikembalikan kembali seperti semula. penggunaan tipe kompresi lossless ini banyak digunakan pada aplikasi-aplikasi seperti winzip dan Gzip [10]. serta pada beberapa format gambar seperti PNG (Portable Network Graphics), BMP (Bitmap) dan GIF (Graphics Interchange Format) yang digunakan hanya kompresi dengan tipe kompresi lossless, tetapi pada tipe gambar TIFF (Tagged Image Format File) serta PNG bisa menggunakan tipe data kompresi lossy atau kompresi lossless ratio kompresi pada metode lossless ini masih relatif sangat rendah. pada metode ini data-data atau informasi-informasi penting yang mengakibatkan file rusak tidak ditemukan sehingga cocok untuk kompresi data. contoh metode yang menggunakan lossless adalah Run Length Encoding, Huffman, LZW (Lempel-Ziv-Welch) dan delta.. Sedangkan pada kompresi tipe Lossy merupakan suatu bentuk metode untuk kompresi data dimana hasil yang diperoleh

kemungkinan berbeda dari yang aslinya akan tetapi cukup dekat perbedaannya. penggunaan metode lossy ini kerap digunakan untuk kompresi data dengan jenis teks dan file. pada metode lossy juga terjadi generation loss yakni jika sebuah file atau data dilakukan proses kompresi dan dekompresi secara berulang-ulang maka kualitas akan berkurang secara signifikan hal ini berbeda dengan tipe lossless [11].

### B. Metode Huffman

Algoritma Huffman merupakan suatu algoritma yang terbilang klasik dan sangat populer digunakan. hal ini dikarenakan file yang dikompresi tidak mengalami perubahan ketika dilakukan proses dekompresi sehingga cocok digunakan untuk kompresi data citra digital, data teks, audio dan suara [12]. pada metode huffman karakter-karakter yang digunakan bentuk ASCII akan diubah kedalam bentuk bit-bit yakni 0 dan 1. dan pengkodeannya mirip dengan sandi morse. Prinsip pada algoritma kompresi huffman adalah data atau file yang terdapat perulangan karakter vokal ataupun konsonan akan dihitung banyaknya frekwensi kemunculan karakter tersebut [13]. sebagai contoh pada sebuah dokument terdapat karakter A, B, C, D dan E. kemudian mengurutkan data-data karakter dimulai dari tingkat frekwensi karakter yang paling sering muncul sampai dengan karakter yang jarang muncul. kemudian membentuk struktur pohon huffman untuk penggunaan karakter A sampai dengan karakter E.



Gambar 2. Struktur Pohon Huffman

dari gambar 2 pada pohon huffman dapat dibaca yaitu

1. untuk karakter A dikodekan menjadi kode biner yakni 01
2. Karakter B dikodekan menjadi kode biner 10.
3. Karakter C dikodekan menjadi kode biner 11.
4. Karakter D dikodekan menjadi kode biner 000.
5. Karakter E dikodekan menjadi kode biner 001.

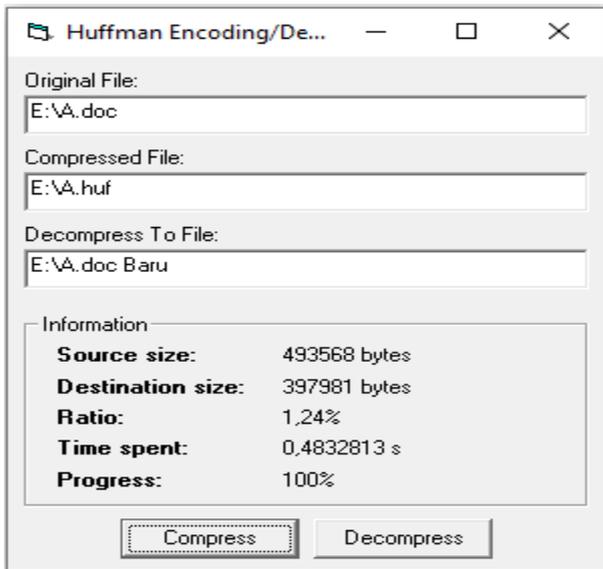
### C. Metode Run Length Encoding

Metode Run Length Encoding merupakan suatu bentuk algoritma kompresi data yang bersifat lossless [14]. Algoritma ini merupakan algoritma yang mudah untuk dipahami dan diterapkan untuk kompresi. Metode kompresi ini sangat sederhana, yaitu hanya memindahkan pengulangan byte yang sama berurut-turut secara terus-menerus dan teknik run length encoding ini bekerja berdasarkan sederetan karakter yang berurutan. cara kerja dari Algoritma Run Length Encoding yaitu langkah pertama pembacaan karakter data, kemudian kompresi data nilai jika ditemukan data nilai sampel yang sama secara berurutan lebih dari dua dengan cara: Berikan bit penanda # pada data kompresi kemudian Tambahkan deretan bit untuk menyatakan jumlah nilai data yang sama berurutan. Setelah itu

Tambahkan deretan bit yang menyatakan karakter yang berulang, kemudian Konversikan semua data kompresi ke dalam hexa [15]. Contoh kita akan melakukan proses kompresi karakter AAAACCB BBBBAACCB BDDDDDEEEEE Menggunakan metode Run Length Encoding maka didapatkan hasil sebagai berikut : (A, 4) karakter A muncul sebanyak 4 kali, (C, 2) karakter C muncul sebanyak 2 kali, (B, 4) karakter B muncul sebanyak 4 kali, (A, 2) karakter A muncul sebanyak 2 kali, (C, 2) karakter C muncul sebanyak 2 kali, (B, 3) karakter B muncul sebanyak 3 kali, (D, 5) karakter D muncul sebanyak 5 kali dan (E, 5) karakter E muncul sebanyak 5 kali. Hasil akhir kompresi karakter yang awalnya berjumlah 29 byte menjadi 16 byte. atau  $29 / 16 = 1,8125$  kali menjadi lebih kecil.

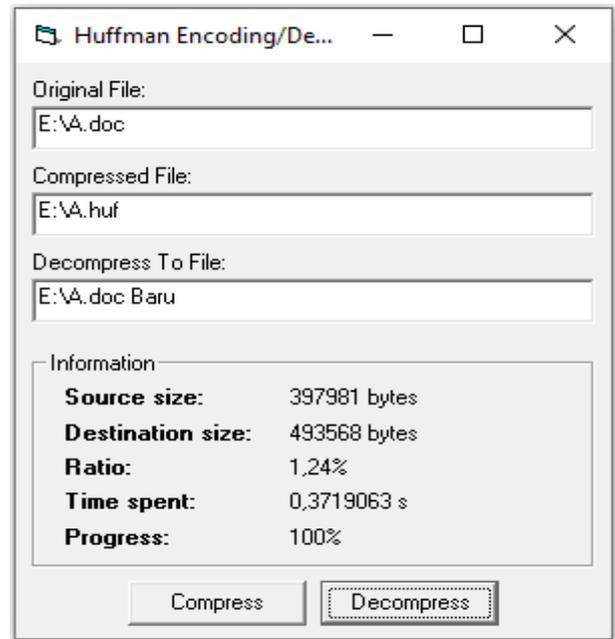
**D. Kompresi Data Menggunakan Metode Huffman**

Pada penelitian ini akan mencoba melakukan proses kompresi menggunakan algoritma huffman dengan file Word, Excell, Powerpoint dan PDF. Pertama-tama akan mencoba mengkompresi file word dengan nama A.doc



Gambar 3. Proses Kompresi File word

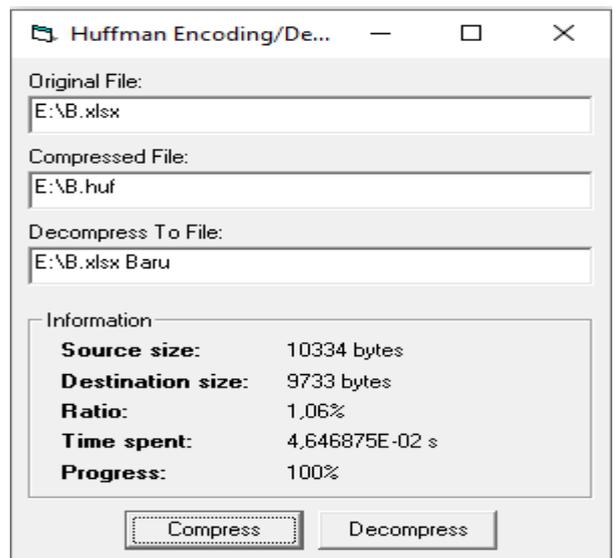
Pada proses kompresi file dengan nama A.doc dari ukuran file awal 493568 byte kemudian dilakukan proses dekomposisi menjadi 397981 byte dengan ratio kompresi sebesar 1,24% dan waktu yang diperlukan untuk kompresi 0,4 detik. Kemudian file kompresi tersebut akan kita coba kembalikan menjadi file word dengan nama A.doc.New



Gambar 4. Proses Dekomposisi File word

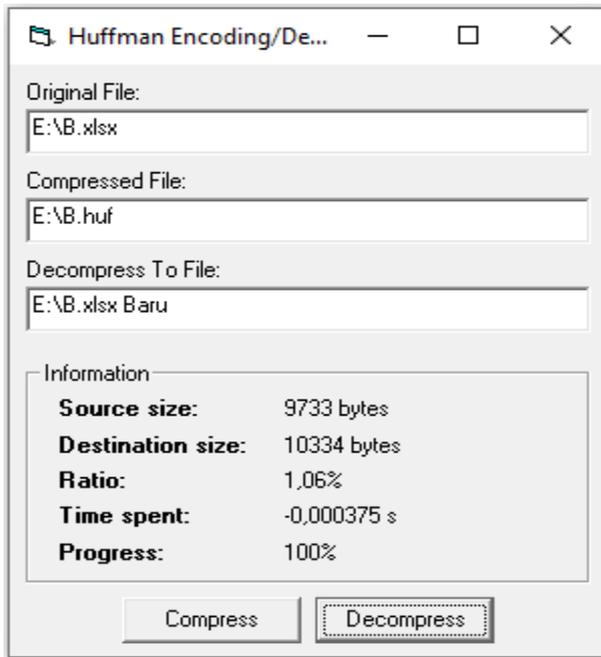
pada gambar 4 merupakan proses dekomposisi pada file word. terlihat hasil dekomposisi file awal yang berukuran 493568 kembali ke ukuran semula dan tidak ada perubahan. begitu juga dengan isi file document word tidak mengalami kerusakan.

File berikutnya adalah file dengan jenis Excell dengan nama B.xlsx.



Gambar 5. Proses Kompresi File excell

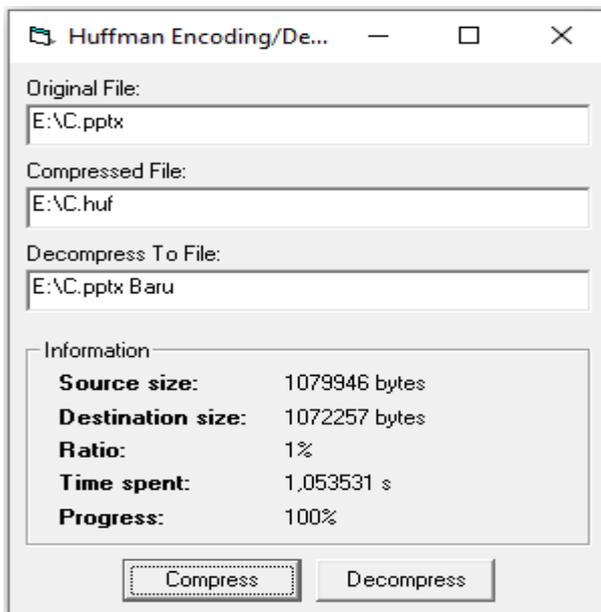
Pada proses kompresi file dengan nama B.xlsx dari ukuran file awal 10334 byte kemudian dilakukan proses dekomposisi menjadi 9733 byte dengan ratio kompresi sebesar 1,06% dan waktu yang diperlukan untuk kompresi 4 detik. Untuk proses dekomposisi terlihat pada gambar 6.



Gambar 6. Proses Dekompresi File excell

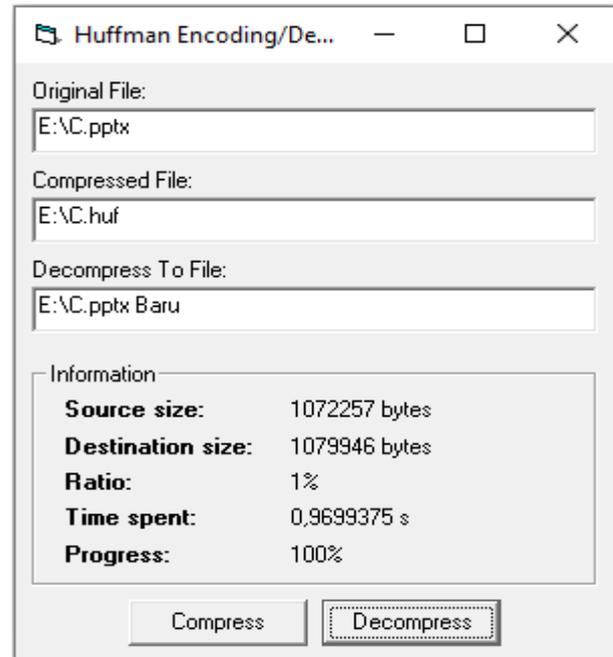
Dari hasil dekomposisi file excell terlihat ukuran file kembali ke ukuran semula yakni 10334 byte dengan ratio Dekompresi sebesar 1,06% dan waktu yang diperlukan untuk melakukan Dekompresi adalah 0,03 detik. dan isi dari file excell tersebut tidak mengalami perubahan.

File selanjutnya adalah file Powerpoint dengan nama file C.ppt proses kompresi dapat dilihat pada gambar 7.



Gambar 7. Proses Kompresi File powerpoint

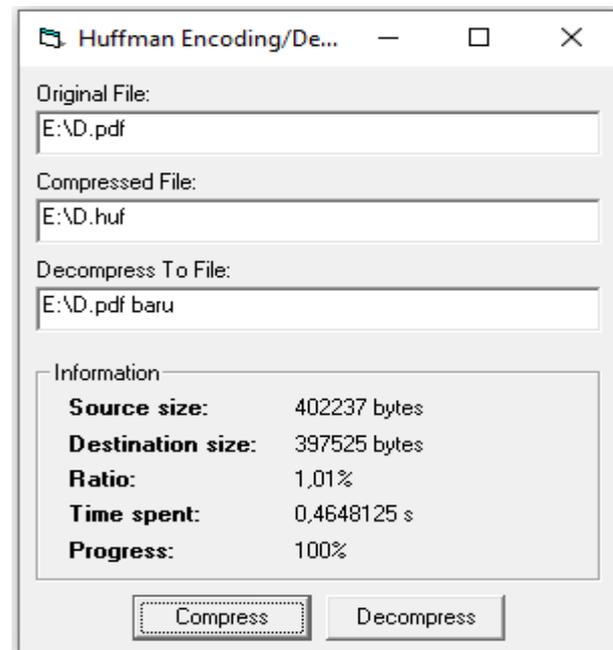
Pada proses kompresi file dengan nama C.pptx dari ukuran file awal 1079946 bytes kemudian dilakukan proses dekomposisi menjadi 1072257 bytes dengan ratio kompresi sebesar 1% dan waktu yang diperlukan untuk kompresi 1,05 detik.



Gambar 8. Proses Dekompresi File powerpoint

Hasil dekomposisi pada file dengan jenis powerpoint dapat terlihat yaitu ukuran file awal sebesar 1079946 bytes setelah dilakukan proses dekomposisi kembali ke ukuran semula dan isi dari file tidak mengalami perubahan dan waktu yang diperlukan sebesar 0,9 detik.

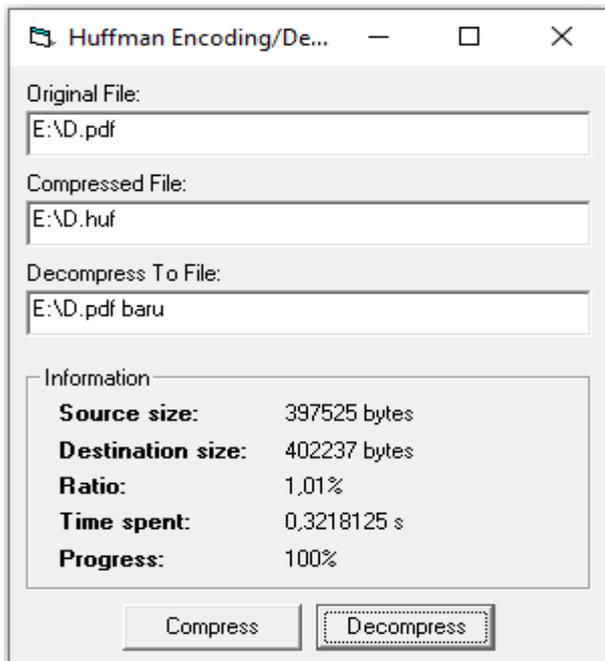
File yang terakhir adalah file dengan jenis PDF dengan nama file D.PDF



Gambar 9. Proses Kompresi File PDF

Proses kompresi file PDF dapat terlihat bahwa ukuran awal sebelum kompresi sebesar 402237 bytes kemudian setelah dilakukan proses Kompresi didapatkan hasil sebesar 397525 bytes. Dengan waktu kompresi sebesar 0,46 detik dan ratio

kompresi 1,01%. Kemudian dilakukan proses dekompresi yang terlihat pada gambar 10.

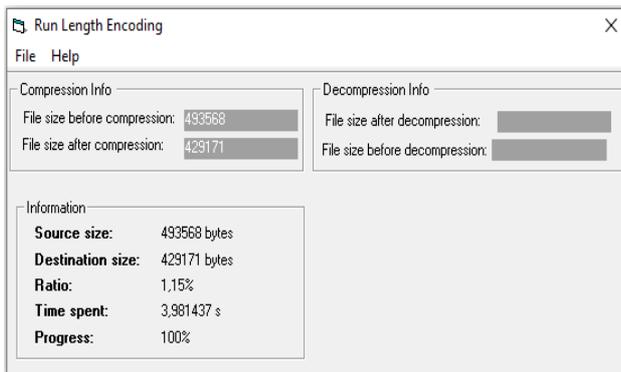


Gambar 10. Proses Dekompresi File PDF

Hasil dari dekompresi menunjukkan hasil bahwa tidak terdapat perbedaan dari ukuran file sebelum dan sesudah kompresi serta isi file juga tidak mengalami perubahan.

*E. Kompresi Data Menggunakan Metode Run Length Encoding.*

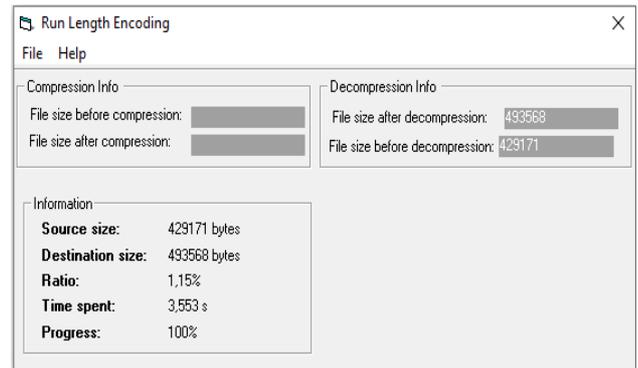
Setelah melakukan proses kompresi pada file word, excell, powerpoint dan PDF menggunakan metode Huffman maka selanjutnya akan melakukan proses kompresi menggunakan metode Run Length Encoding. File pertama yang akan dikompres menggunakan metode Run Length Encoding adalah file dengan jenis word dengan nama file A.doc hasil kompresi dapat terlihat pada gambar 11.



Gambar 11. Proses Kompresi File word

Pada file word yang dikompresi yang terlihat pada gambar 11 ukuran file awal sebesar 493568 bytes kemudian setelah didekompresi menjadi 429171 bytes. Dan waktu yang diperlukan untuk kompresi sebesar 3,9 detik dengan ratio 1,15%.

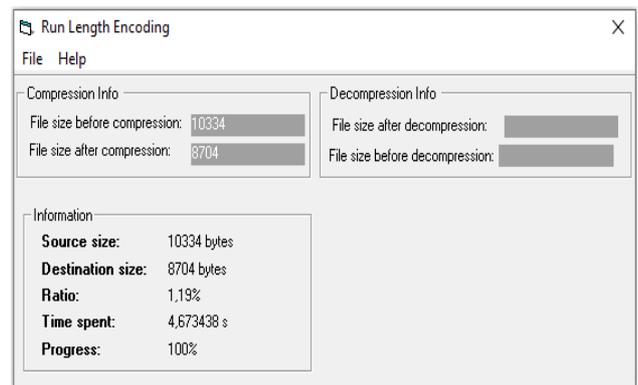
Kemudian dilakukan proses dekompresi yang dapat dilihat pada gambar 12.



Gambar 12. Proses Dekompresi File word

Pada proses dekompresi dari ukuran dekompresi sebesar 429171 bytes kemudian menjadi ukuran semula yaitu 493568 bytes. Waktu yang diperlukan untuk dekompresi sebesar 3.5 detik. Dan isi dari file juga tidak mengalami kerusakan dan dapat dibaca kembali.

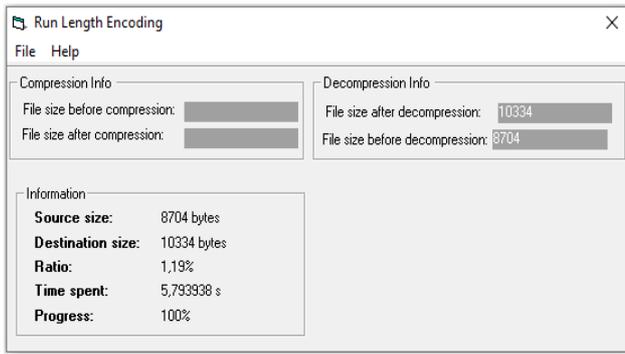
Kemudian akan dilakukan ujicoba kompresi data menggunakan data berjenis microsoft excell dengan nama B.xlsx hasil dari kompresi dapat dilihat pada gambar 13.



Gambar 13. Proses Kompresi File excell

Pada file excell yang telah dilakukan proses kompresi pada gambar 13 dapat dilihat bahwa ukuran file sebelum kompresi adalah sebesar 10334 bytes. Kemudian setelah dilakukan kompresi menjadi 8704 bytes. Dengan ratio kompresi sebesar 1.19 % dan waktu kompresi sebesar 4,6 detik.

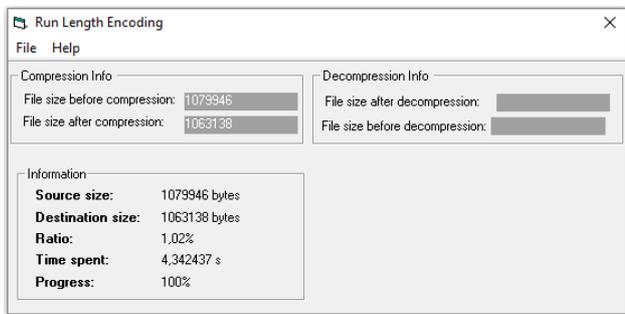
Kemudian akan dilakukan proses dekompresi yang dapat dilihat pada gambar 14.



Gambar 14. Proses Dekompresi File excell

Pada proses dekomposisi pada gambar 14. dari ukuran dekomposisi sebesar 8704 bytes kemudian menjadi ukuran semula yaitu 10334 bytes. Waktu yang diperlukan untuk dekomposisi sebesar 5,7 detik. Dan isi dari file juga tidak mengalami kerusakan dan dapat dibaca kembali.

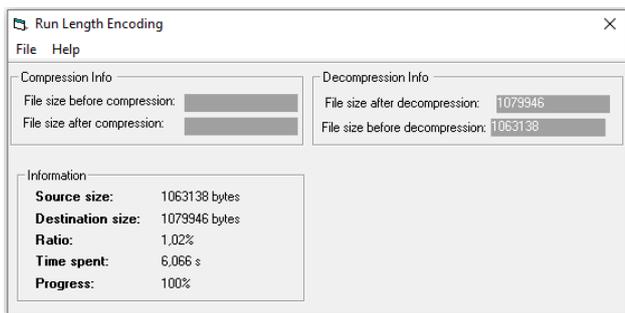
File berikutnya yang akan dicoba dikompresi adalah file berjenis powerpoint. Hasil kompresi dapat dilihat pada gambar 15.



Gambar 15. Proses Kompresi File powerpoint

Pada proses kompresi, ukuran file powerpoint sebelum kompresi adalah sebesar 1079946 bytes kemudian setelah dilakukan proses kompresi menjadi 1063138 bytes.dengan rasio kompresi 1,02% dan waktu kompresi sebesar 4,34 detik.

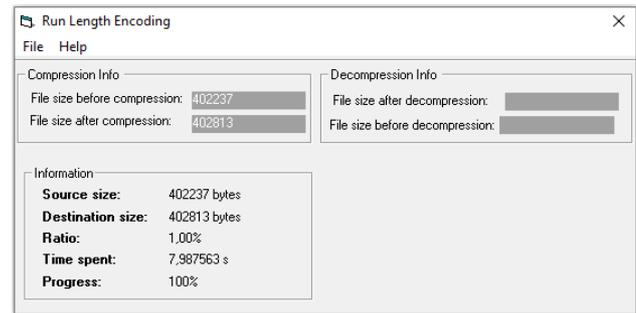
Adapun untuk proses dekomposisi dapat dilihat pada gambar 16 berikut.



Gambar 16. Proses Dekompresi File Powerpoint

Pada file powerpoint yang telah di dekomposisi pada gambar 16 ukuran file dekomposisi sebesar 1063138 bytes kemudian setelah didekomposisi kembali keukuran semula yaitu menjadi 1079946 bytes. Dan waktu yang diperlukan untuk Dekompresi sebesar 6,06 detik dengan ratio 1,02%.

Kemudian file berikutnya yang akan dikompresi adalah file berjenis PDF. Hasil dari kompresi data tersebut dapat dilihat pada gambar 17.



Gambar 17. Proses Kompresi File PDF

Setelah file dilakukan proses kompresi seperti yang terlihat pada gambar 17 maka ukuran file sebelum proses kompresi adalah sebesar 402237 bytes dan setelah dilakukan proses kompresi menjadi 402813 bytes. Dengan ratio 1 % dan waktu yang diperlukan untuk kompresi 7 detik.

Adapun perbandingan metode Huffman dan Metode Run Length Encoding dapat dilihat pada tabel 1.

Tabel 1. Hasil Kompresi dan Dekompresi metode Huffman

Nama File	Ukuran Awal	Metode Huffman		Ratio kompresi	Waktu kompresi
		Hasil Kompresi	Hasil Dekompresi		
A.doc	493568 Bytes	397981 Bytes	493568 Bytes	1,24%	0,48 Detik
B.xlsx	10334 Bytes	9733 Bytes	10334 Bytes	1,06%	4 Detik
C.pptx	1079946 Bytes	1072257 Bytes	1079946 Bytes	1%	1 Detik
D.pdf	402237 Bytes	397525 Bytes	402237 Bytes	1%	0,4 Detik

Tabel 2. Hasil Kompresi dan Dekompresi metode Run Length Encoding

Nama File	Ukuran Awal	Metode Run Length Encoding		Ratio kompresi	Waktu kompresi
		Hasil Kompresi	Hasil Dekompresi		
A.doc	493568 Bytes	429171 Bytes	493568 Bytes	1,15%	3,9 Detik
B.xlsx	10334 Bytes	8704 Bytes	10334 Bytes	1,19%	4,6 Detik
C.pptx	1079946 Bytes	1063138 Bytes	1079946 Bytes	1,02%	4,3 Detik
D.pdf	402237 Bytes	402813 Bytes	402237 Bytes	1%	7 Detik

Berdasarkan hasil kompresi pada metode Huffman dapat dilihat bahwa ukuran file setelah dilakukan proses kompresi maka ukuran file menjadi menurun. sedangkan pada metode Run Length Encoding terdapat 1 buah file berjenis PDF yang

mengalami kenaikan ukuran file yaitu yang semua 402237 Bytes menjadi 402813 Bytes.

## CONCLUSIONS

Setelah dilakukan pengujian terhadap beberapa jenis dokument menggunakan metode Huffman dan Run Length Encoding maka dapat disimpulkan yaitu pada kompresi metode Huffman seluruh dokument setelah proses kompresi maka ukuran file mengalami penurunan sedangkan pada metode Run Length Encoding terdapat 1 buah file berjenis PDF yang mengalami kenaikan ukuran file yaitu yang semula 402237 Bytes menjadi 402813 Bytes. Sehingga dapat disimpulkan metode Huffman lebih baik didalam melakukan proses kompresi jika dibandingkan dengan metode Run Length Encoding.

## REFERENCES

- [1] Putera A, Siahaan U. JURNAL INFORMATIKA Vol. 10, No. 2, Jul 2016 Implementasi Teknik Kompresi Teks Huffman. *J Inform.* 2016;10(2):1251-1261.
- [2] Pahdi A. Algoritma Huffman Dalam Pemampatan Dan Enkripsi Data. *IJNS - Indones J Netw Secur.* 2017;6(3):1-7. doi:10.2311/ijns.v6i3.1461
- [3] Suhermanto RA. Time Optimization for Lossy Decompression of the LISA Sensor Data on LAPAN A3 Satellite Using a Grouping Method of HUFFMAN Code Bit Number. *J Teknol Dirgant.* 2018;16(1):23. doi:10.30536/j.td.2018.v16.a2960
- [4] Lu U, Salsabila NH. Penggunaan Metode Run Length Encoding Untuk Kompresi Data. *Semin Mat DAN Pendidik Mat UNY 2017.* 2017;(1):273-280.
- [5] Jaya YA, Chrisantyo L, Raharjo WS. Pengembangan Dan Analisis Kombinasi Run Length Encoding Dan Relative Encoding Untuk Kompresi Citra. *J Inform.* 2016;12(2):141-150. doi:10.21460/inf.2016.122.488
- [6] Willfrid D, Simamora M, Ginting G, Hasan Y. Implementasi Algoritma Run Length Encoding Pada Kompresi File MP3. 2016;3(4):5-9.
- [7] Prayoga E, Suryaningrum KM. Implementasi Algoritma Huffman Dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web. *J Ilm Teknol Inf Terap.* 2018;IV(2):92-101.
- [8] Sinaga H, Sihombing P, Handrizal H. Perbandingan Algoritma Huffman Dan Run Length Encoding Untuk Kompresi File Audio. *Talent Conf Ser Sci Technol.* 2018;1(1):010-015. doi:10.32734/st.v1i1.183
- [9] Aria M, Sanjaya A. Teknik Kompresi pada Transmisi Data Citra Payload KOMURINDO. *Komputika J Sist Komput.* 2018;7(2):103-111. doi:10.34010/komputika.v7i2.1512
- [10] Lusiana V. Teknik Kompresi Citra Digital untuk Penyimpanan File menggunakan Format Data XML. 2014;19(2):112-119.
- [11] Mahesa K. Rancang Bangun Aplikasi Kompresi Dan Dekompresi Pada Citra Digital Menggunakan Metode Huffman. 2017;12(1):948-963.
- [12] Purba RA, Sitorus L, Coding A, Welch LZ. Analisis Perbandingan Algoritma Arithmetic Coding Dengan Algoritma Lempel Ziv Welch ( Lzw ) Dalam Kompresi Teks. 2018;03:158-165.
- [13] Frida O, Ilmu F, Program K, et al. Analisis Perbandingan Pemampatan Data Teks Dengan Menggunakan Metode Huffman Dan Half – Byte. *Algoritma J Ilmu Komput dan Inform.* 2018;6341(April):1-6.
- [14]. Saputra AK, Sutardi, Ningrum IP. Aplikasi kompresi. *semantik.* 2015;1(2):1-10.
- [15]. Utari CT. Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra. *J TIMES.* 2016;V(2):24-31. <http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/article/download/%0A553/12%0A>.

## AUTHOR(S) BIOGRAPHY



**Pujianto**

Lahir di Seputih Raman, pada tanggal 18 juli 1988, Pendidikan Terakhir S2 Magister Komputer pada Universitas Budi Luhur Jakarta dan lulus pada tahun 2016. Pendidikan S1 Sarjana Komputer di tempuh pada universitas yang sama dan lulus pada tahun 2011. Bidang Studi yang ditekuni adalah Sistem Informasi, Pemrograman Berorientasi Objek, Pemrograman Web dan Mobile Programming.



**Mujito**

Lahir di Metro, pada tanggal 15 Agustus 1985, Pendidikan Terakhir S2 Magister Komputer pada Universitas Budi Luhur Jakarta dan lulus pada tahun 2017. Pendidikan S1 Sarjana Komputer di tempuh pada universitas yang sama dan lulus pada tahun 2010. Bidang Studi yang ditekuni adalah Sistem Pendukung Keputusan, Sistem Informasi dan Data Base.



**Basuki Hary Prasetyo**

Lahir di Jakarta, pada tanggal 01 April 1981, Pendidikan Terakhir S2 Magister Komputer pada Universitas Budi Luhur Jakarta dan lulus pada tahun 2008. Pendidikan S1 Sarjana Komputer di tempuh pada universitas yang sama dan lulus pada tahun 2003. Bidang Studi yang ditekuni adalah Sistem Pendukung Keputusan, Sistem Informasi dan Data Mining



**Danang Prabowo**

Lahir di Sribhawono, pada tanggal 31 Oktober 1984, Pendidikan Terakhir S2 Pada Universitas Gajah Mada dan lulus pada tahun 2014. Pendidikan S1 Sarjana Komputer di tempuh pada universitas Darmajaya dan lulus pada tahun 2008. Bidang Studi yang ditekuni adalah Artificial Intelligence