

## ANALISIS DAN PERANCANGAN APLIKASI ALGORITMA HUFFMAN DAN DYNAMIC MARKOV COMPRESSION PADA FILE TEXT PDF

Suendri<sup>1)</sup>, Abdul Halim Hasugian<sup>2)</sup>, Mhd. Nanda Khairul Rijal<sup>3)</sup>

<sup>1</sup> Sistem Informasi, Universitas Islam Negeri Sumatera Utara, Medan, Sumatera Utara

<sup>2,3</sup> Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Medan, Sumatera Utara

Co Responden Email: rijalmnkr@gmail.com

### Abstract

#### Article history

Received 27 Oct 2024

Revised 19 Nov 2024

Accepted 12 Dec 2024

Available online 31 Jan 2025

#### Keywords

Huffman,  
Compression,  
Document,  
PDF

*Everybody wants to employ efficient and effective storage medium; document size (both big and little) and transmission speed (both up and down) are major factors in this. It is possible to decrease the size of a document by using several compression techniques. There are a number of methods for compressing and decompressing text files, including the Dynamic Markov Compression technique and the Huffman method. This study will compare and contrast the two methods using.pdf files as examples. For successfully compressed files of the same size, the study findings reveal that the Huffman technique outperforms the Dynamic Markov Compression approach in terms of compression ratio, making it the ideal method for document compression. Huffman and Dynamic Markov Compression both come out at 57.97% and 57.41% in terms of time complexity and average compression ratio value, respectively. Reason being, compression methods work better with larger compression ratios. According to the findings, businesses involved in real-time data processing may benefit from this application's development as it compresses document files, which speeds up delivery and gives consumers a new experience.*

### Abstrak

#### Riwayat

Diterima 27 Okt 2024

Revisi 19 Nov 2024

Disetujui 12 Des 2024

Terbit online 31 Jan 2025

#### Kata Kunci

Huffman,  
Kompresi,  
Dokumen  
PDF

Semua orang ingin menggunakan media penyimpanan yang efisien dan efektif; ukuran dokumen (baik besar maupun kecil) dan kecepatan transmisi (baik ke atas maupun ke bawah) merupakan faktor utama dalam hal ini. Dimungkinkan untuk memperkecil ukuran dokumen dengan menggunakan beberapa teknik kompresi. Ada sejumlah metode untuk mengompresi dan mendekomposisi berkas teks, termasuk teknik Dynamic Markov Compression dan metode Huffman. Studi ini akan membandingkan dan mengontraskan kedua metode tersebut dengan menggunakan berkas .pdf sebagai contoh. Untuk berkas yang berhasil dikompresi dengan ukuran yang sama, temuan studi mengungkapkan bahwa teknik Huffman mengungguli pendekatan Dynamic Markov Compression dalam hal rasio kompresi, menjadikannya metode yang ideal untuk kompresi dokumen. Huffman dan Dynamic Markov Compression masing-masing menghasilkan 57,97% dan 57,41% dalam hal kompleksitas waktu dan nilai rasio kompresi rata-rata. Alasannya, metode kompresi bekerja lebih baik dengan rasio kompresi yang lebih besar. Menurut temuan tersebut, bisnis yang terlibat dalam pemrosesan data waktu nyata dapat memperoleh manfaat dari pengembangan aplikasi ini karena mengompresi berkas dokumen, yang mempercepat pengiriman dan memberikan pengalaman baru kepada konsumen.

## PENDAHULUAN

Gambar, video, audio, teks, dan jenis presentasi lainnya digunakan untuk menyampaikan data dan informasi (Manurung, 2020). Multimedia mengacu pada metode apa pun untuk menyajikan fakta atau informasi. Pertumbuhan eksponensial teknologi

informasi dan permintaannya membuat hampir mustahil untuk melambat. Data yang disimpan dalam media digital dapat diakses secara instan karena kebutuhan masyarakat akan penyebaran informasi yang cepat (Ulfah, 2020). Akan selalu ada permintaan untuk ruang penyimpanan yang lebih besar karena jumlah data yang dibutuhkan meningkat.

Untuk mengurangi jumlah ruang penyimpanan yang dibutuhkan dan jumlah waktu yang dibutuhkan untuk mengirimkan data, metode penyimpanan data alternatif diperlukan untuk mengurangi ukuran file dan dokumen (Permana, 2022).

Untuk memanfaatkan ruang penyimpanan dengan lebih baik, kompresi data merupakan opsi yang layak (Tanjung & Mesran, 2021). Mengompresi file dan data ke dalam format yang lebih mudah dikelola menggunakan kode bit dapat membantu menghemat waktu dan ruang (Reswan & Tri Anggoro, 2024). Secara umum, metode kompresi data menggunakan berbagai pendekatan untuk mengidentifikasi dan menghapus data duplikat. Kompresi lossy dan kompresi lossless adalah dua bentuk utama kompresi data (Mansyuri, 2021). Sangat sulit, jika tidak mustahil, untuk mengembalikan data terkompresi ke keadaan aslinya, tidak terkompresi menggunakan Kompresi Lossy. Sebaliknya, kompresi lossless memungkinkan pemulihan lengkap data terkompresi ke keadaan aslinya. Karena kompresi lossless dapat mengembalikan materi terkompresi ke bentuk aslinya tanpa memengaruhi informasi asli apa pun, kompresi ini sering digunakan untuk data teks (Tarigan & Matondang, 2019).

Dalam studi sebelumnya tentang kompresi data teks, Lazuardi Imani et al. (2021) yang didokumentasikan dalam jurnal Implementasi Algoritma Huffman untuk Kompresi Data Teks, file berhasil dikompresi lebih dari 45% dari ukuran aslinya menggunakan metode Huffman. Rasio pengujian mungkin mengambil nilai serendah 46,53% dan setinggi 47,08%. Ditemukan bahwa dari lima file \*.docx dan \*.odt yang dikompres, file yang paling besar belum tentu memiliki rasio kompresi yang paling besar, hal ini berdasarkan penelitian yang dilakukan oleh (Nainggolan, 2019) dari Universitas Sumatera Utara dengan menggunakan Algoritma Dynamic Markov Compression (DMC) pada Berkas Teks. Perusahaan media berita daring yang bergerak di bidang pendidikan salah satunya adalah PT. Media Delegasi Nusantara. Beralamat di Jl. Sei Batang Gadis No. 22 Medan merupakan kantor redaksi perusahaan ini. Ketepatan waktu dan ketepatan berita tertulis tidak dapat dipisahkan dari media daring. Pengunggahan langsung peristiwa atau kejadian di lapangan dapat dilakukan dalam

hitungan menit atau detik (Aditya & Wiyono, 2023), sehingga mempercepat penyampaian informasi ke media secara signifikan. Keterlambatan pemuatan informasi di situs web PT. Media Delegasi Nusantara masih sering terjadi.

Hal ini disebabkan oleh kombinasi beberapa faktor, termasuk, namun tidak terbatas pada, keterlambatan pengiriman naskah berita melalui email, penumpukan data yang disebabkan oleh tingginya volume aktivitas pengiriman file, dan kecepatan pengiriman yang lambat yang disebabkan oleh ukuran file yang besar (Mahyuddin et al., 2021). Mengingat bahwa kedua algoritma tersebut menggunakan Lossless Compression untuk mengompresi data teks, akan sangat membantu jika pengguna memahami kelebihan dan kekurangan masing-masing serta seberapa besar dampak algoritma kompresi ini terhadap komputer untuk memfasilitasi transfer data yang lebih cepat (Roring et al., 2022).

## **METODE PENELITIAN**

Setelah menguraikan masalah menjadi beberapa bagian, penulis melakukan penelitian untuk mengidentifikasi penyebab dan solusi yang mungkin. Periode Penelitian:

### **A. Studi Literatural**

Peneliti menggunakan tinjauan pustaka untuk menyusun karya-karya sebelumnya tentang topik atau kasus yang diteliti dari berbagai sumber cetak dan digital, termasuk buku, jurnal, dan artikel laporan penelitian (Mahanum, 2021). Temuan penelitian akan dapat dijelaskan dengan merujuk ke sumber-sumber ini di masa mendatang.

### **B. Observasi**

Pada penelitian Analisis dan Desain Aplikasi Komparatif Metode Kompresi Huffman dan Dynamic Markov dalam Kompresi Berkas Teks PDF (Suhardi et al., 2023), observasi merupakan metode yang digunakan untuk mengumpulkan informasi dan pengetahuan yang relevan dengan penelitian (Prawiyogi et al., 2021).

### **C. Pengumpulan data**

Pengumpulan informasi dilakukan dengan mengekstrak tujuh berkas teks PDF yang dikompresi. Di sini kita akan melihat berkas yang akan ditampilkan di halaman situs web Media Delegasi: teks berita.

#### D. Perancangan Sistem

Pada tahap ini, desain ditampilkan kepada pengguna di layar desktop sebagai kerangka kerja perangkat lunak. Berikut ini adalah persyaratan perangkat keras: Aplikasi desktop yang membandingkan algoritma Huffman dan Dynamic Markov Compression dirancang dengan persyaratan input, output, format file, program, proses, perangkat keras, dan perangkat lunak (Asdini & Utomo, 2022).

#### E. Implementasi Sistem

Algoritma Huffman dan Dynamic Markov Compression digunakan untuk membangun sistem.

#### F. Pengujian Sistem

Uji kinerja sistem dan akurasi hasil kompresi dan dekompresi file teks PDF menggunakan algoritma Huffman dan Dynamic Markov Compression sekarang sedang dilakukan.

### HASIL DAN PEMBAHASAN

#### 1. Analisa Data

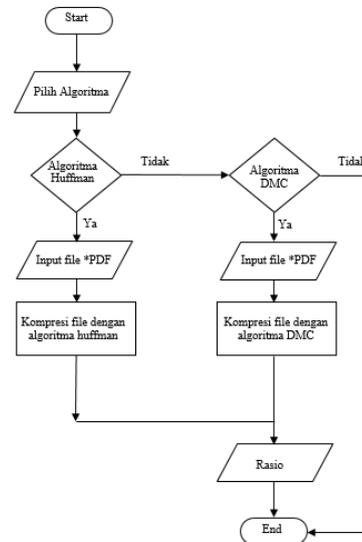
Informasi yang dikumpulkan untuk penelitian ini disertakan dalam berkas PDF yang berisi konten berita yang bersumber dari PT. Media Delegasi Nusantara. Berkas tersebut kemudian akan melalui beberapa tahap penyuntingan dan seleksi sebelum dipublikasikan.

Tabel 1. File Percobaan

Perco baan	Nama File (.pdf)	Ukuran file asli (Byte)
1	Berita Medan	6292
2	Berita Toba	6815
3	Berita Humbahas	6869
4	Berita Nias	6966
5	Berita JawaBarat	7298
6	Berita Deliserdang	14241
7	Berita Dairi	30591

Untuk mengompresi file teks, pertama-tama Anda harus menganalisis file teks itu sendiri, yang merupakan file dokumen dengan ekstensi pdf. Prosedur untuk mengompresi file teks dalam format PDF menggunakan metode Huffman dan Dynamic Markov Compression (DMC), dimulai dengan pemilihan jenis metode dan dilanjutkan dengan input file \*.pdf, pemrosesan kompresi menggunakan metode Huffman atau Dynamic Markov Compression,

dan akhirnya, hasil kompresi disimpan dalam format yang ditentukan oleh algoritme, menghasilkan rasio akhir. Lihat Gambar 1, diagram alir sistem, untuk penjelasan terperinci tentang proses sistem kompresi dan dekompresi file teks:



Gambar 1. Flowchart Sistem

#### 2. Penerapan Metode Huffman

Kompresi data menggunakan metode Huffman merupakan proses yang terdiri dari beberapa langkah. Kita perlu menentukan peluang kemunculan setiap karakter dalam teks sebelum kita bisa mendapatkan kode Huffman. Berikut ini adalah langkah-langkah yang terlibat dalam prosedur menurut kode Huffman:

##### a) Menuliskan pesan terlebih dahulu

Tentukan jumlah semua karakter yang ada dalam suatu pesan yang akan dikodekan. Susunan kalimat data teks sebagai berikut: SAINS DAN TEKNOLOGI

Tabel 2. Nilai Kerapatan

Huruf	S	A	I	N	D	T	E	K	O	L	G
Kerapatan	2	2	2	3	1	1	1	1	2	1	1

Dari pesan tersebut bobot hurufnya = 17 huruf

##### b) Menentukan peluang

Yaitu berapa kali huruf tersebut muncul di dalam kalimat/string yang akan dikodekan. dari bobot-bobot huruf tersebut penulis dapat menentukan probabilitas masing-masing karakter adalah sebagai berikut:

Tabel 3. Nilai Kerapatan dan Peluang

Huruf	Kerapatan	Peluang
S	2	$\frac{2}{17}$
A	2	$\frac{2}{17}$
I	2	$\frac{2}{17}$
N	3	$\frac{3}{17}$
D	1	$\frac{1}{17}$
T	1	$\frac{1}{17}$
E	1	$\frac{1}{17}$
K	1	$\frac{1}{17}$
O	2	$\frac{2}{17}$
L	1	$\frac{1}{17}$
G	1	$\frac{1}{17}$

$$P(A) = \frac{x}{n}$$

Keterangan:

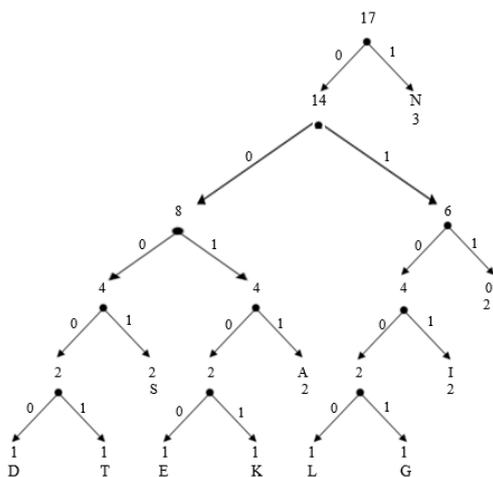
P(A) = Huruf

x = Jumlah kerapatan

n = Total jumlah kerapatan

### c) Membuat pohon biner

Untuk mendapatkan kode Huffman, pertama-tama kita harus menentukan kerapatan huruf dalam teks pesan dengan membuat pohon biner dengan cara berikut:



Gambar 2. Pohon Biner

Jumlah semua akar yang telah dihasilkan hingga titik ini adalah 17. Bagi cabang akar kiri dengan menggunakan bit 0 dan cabang akar kanan dengan menggunakan bit 1.

Jadi, berikut ini adalah hasil pembuatan pohon biner untuk setiap karakter dari gambar 2 ketika dikonversi ke kode biner:

Tabel 4. Kode Biner Setiap Karakter

Huruf	Kerapatan	Kode Biner
S	2	0001
A	2	0011
I	2	0101
N	3	1
D	1	00000
T	1	00001
E	1	00100
K	1	00101
O	2	011
L	1	01000
G	1	01001

### d) Menghitung banyak bit yang digunakan

Kode yang benar-benar ringkas dihasilkan dengan melakukan pengkodean yang disebutkan di atas. Karena ada 17 karakter dalam string di atas, dan setiap karakter memiliki 8 bit saat dihitung menggunakan ASCII, jumlah RAM yang dibutuhkan untuk menyimpan teks adalah:

Sebelum:  $17 \times 8 = 136 \text{ bit}$

Sedangkan, bila menggunakan *Huffman Code* maka dijabarkan:

- 1 bit: 1 karakter (N)  
 $N = 3 \times 1 = 3$   
Total = **3 bit**
- 3 bit: 1 karakter (O)  
 $O = 2 \times 3 = 6$   
Total = **6 bit**
- 4 bit: 3 karakter (S,A,I)  
 $S = 2 \times 4 = 8$   
 $A = 2 \times 4 = 8$

$$I = 2 \times 4 = 8$$

$$\text{Total} = 24 \text{ bit}$$

d. 5 bit: 6 karakter (D,T,E,K,L,G)

$$D = 1 \times 5 = 5$$

$$T = 1 \times 5 = 5$$

$$E = 1 \times 5 = 5$$

$$K = 1 \times 5 = 5$$

$$L = 1 \times 5 = 5$$

$$G = 1 \times 5 = 5$$

S	A	I	N	S	D
001	0011	0101	1	0001	00000
A	N	T	E	K	N
0011	1	00001	00100	00101	1
O	L	O	G	I	
011	01000	011	01001	0101	

$$\text{Total} = 30 \text{ bit}$$

Dengan begitu penulis mendapatkan hasil total akhir dari string di atas yaitu:

Sesudah:  $3+6+24+30 = 63 \text{ bit}$

0	1	2	3	4	5	6	7
0	1	4	9	20	41	83	167
0	1	4	9	20	41	83	168
0	1	4	9	19	40	82	166
0	1	4	9	19	39	80	161
0	1	4	9	19	40	81	163
0	1	3	8	17	35	71	143

Sehingga kapasitas memori yang diperlukan:

$$\frac{63}{136} \times 100\% = 46,32\%$$

Berdasarkan dari proses perhitungan di atas, didapatkan nilai *string* dengan pesan SAINS DAN TEKNOLOGI sebelumnya bernilai 136 bit, setelah dilakukan proses pemampatan menggunakan *Huffman* menjadi 63 bit.

Tabel 5. String Bit Hasil Kompresi

Perco baan	Nama File (.pdf)	Ukuran file asli (Byte)	Ukuran Setelah Dekomp resi (Byte)	Rasio kompresi (%)	Kece patan Komp resi (Detik)
1	Berita Medan.dm c	6226	10686	171.635 1 %	0.035 1 detik
2	Berita Toba.dmc	6742	11610	172.204 1 %	0.034 9 detik
3	Berita Humbahas .dmc	6793	11995	176.578 8 %	0.039 1 detik
4	Berita Nias.dmc	6896	11932	173.027 8 %	0.035 3 detik

5	Berita JawaBarat. dmc	7219	12573	174.165 4 %	0.039 4 detik
6	Berita Deliserdan g.pdf	14137	24952	176.501 4 %	0.070 3 detik
7	Berita Dairi.pdf	30385	53262	175.290 4 %	0.132 0 detik

### 3. Penerapan Metode DMC

Misalkan kata yang akan dimampatkan adalah PRODI ILKOMP. maka dilakukan pendataan nilai ASCII dimana P=80, R=82, O=79, D=68, I=73, Spasi=32, L=76, K=75, M=77, A=65.

Berikut adalah proses perhitungan untuk kompresi file.

$$\text{Bit Set} = (\text{mChar And } (2^{(7-i)})) \text{ And } \&\text{HFF}$$

Tabel 6. Hasil Perhitungan Bit Set

	0	1	2	3	4	5	6	7
ASCII	128	64	32	16	8	4	2	1
P	80	0	64	0	16	0	0	0
R	82	0	64	0	16	0	0	2
O	79	0	64	0	0	8	4	2
D	68	0	64	0	0	0	4	0
I	73	0	64	0	0	8	0	0
	32	0	0	32	0	0	0	0
I	73	0	64	0	0	8	0	0
L	76	0	64	0	0	8	4	0
K	75	0	64	0	0	8	0	2
O	79	0	64	0	0	8	4	2
M	77	0	64	0	0	8	4	0
P	80	0	64	0	16	0	0	0

Selanjutnya mencari Index =  $(2^{(8-i)} - 1) + \text{Int}(\text{mChar} / (2^{(8-i)}))$

Tabel 7. Hasil Index

0	1	4	9	19	40	81	163
0	1	4	9	19	40	82	165
0	1	4	9	19	40	81	164
0	1	4	9	19	40	82	166
0	1	4	9	19	40	82	165
0	1	4	9	20	41	83	167

Setelah itu mencari/menampilkan MaxValue dan MinValue dengan memanfaatkan fitur *debug* di Visual Studio 2019.

0	1	2	3	4	5	6	7
16777215	8388608	8388608	8388611	8388611	8388619	8388629	8388645
8388677	11184947	11184947	8699432	8699432	13601819	10656022	9069659
10283857	16639983	16639983	14430633	9459596	8567911	8567911	8567911
8567911	8402210	8402210	13818943	11348723	16436379	16095543	8855914
10472199	9265594	9265594	8427674	8554279	8554279	12535255	13898233
13898233	13100213	8996109	9720031	11158595	11372869	11801413	12658501
14372677	13004636	13004636	11978606	9340242	9340242	15947527	13133271
9489327	8655473	8655473	16013841	12608938	8440661	8440661	8914551
10810107	10003018	10003018	9357347	15271115	13765015	8845612	8845612
8845612	8789710	8789710	8743972	8620479	14839807	12902399	12902399
9027583	16737901	16737901	15640179	12646391	12646391	8515567	9579929
9579929	9452558	9452558	9344782	13820383	11449319	9671021	11237083

Gambar 3 Hasil Debug MaxValue

0	1	2	3	4	5	6	7
0	0	4194304	0	4194306	8	16	32
64	128	3728401	3728401	5385411	4764428	4764428	1214196
2428392	4856784	7802584	7802584	7802584	1939856	5253884	6910898
7739404	7739404	7871965	244640	489280	1957120	1616284	1616284
3232568	3232568	4238072	4238072	175072	4364676	1362976	2725952
8312092	8312092	8312092	4247872	214272	428544	857088	1714176
3428352	3428352	5822423	5822423	5822423	7229551	7504760	7504760
1984644	1984644	3467050	6934100	6934100	874210	5414081	4879108
2739216	2739216	4191976	4191976	8383952	3926210	3926210	7615762
8230687	8230687	8332328	8332328	8332328	1008512	1047002	6974700
1123984	2247968	4662957	4662957	4662957	6437053	2306228	4612456
7924105	7924105	8159252	8159252	4336128	4336128	4336128	567296

Gambar 4. Hasil Debug MinValue

Kemudian dengan mendapatkan nilai MaxValue dan MinValue setelah itu mencari RangValue dengan cara:

$$\text{RangValue} = \text{MaxValue} - \text{MinValue}$$

0	1	2	3	4	5	6	7
16777215	8388608	4194304	8388611	4194305	8388611	8388613	8388613
8388613	11184819	7456546	4971031	3314021	8837391	5891594	7855463
7855465	11783199	8837399	6628049	1657012	6628055	3314027	1657013
828507	662806	530245	13574303	10859443	14479259	14479259	7239630
7239631	6033026	5027522	4189602	8379207	4189603	11172279	11172281
5586141	4788121	684017	5472159	10944323	10944325	10944325	10944325
10944325	9576284	7182213	6156183	3517819	2110691	8442767	5628511
7504683	6670829	5188423	9079741	5674838	7566451	3026580	4035443
8070891	7263802	5811042	5165371	6887163	9838805	4919402	1229850
614925	559023	457382	411644	288151	13831295	11855397	5927699
7903599	14489933	12074944	10977222	7983434	6209338	6209339	4967473
1655824	1528453	1293306	1185530	9484255	7113191	5334893	10669787

Gambar 5. Hasil RangValue

Setelah itu mencari MidValue dengan cara seperti berikut.  $\text{MidValue} = \text{MinValue} + (\text{RangValue} * (\text{Zero}(\text{Index}) / (\text{One}(\text{Index}) + \text{Zero}(\text{Index})))$

0	1	2	3	4	5	6	7
8388608	4194304	6291456	4194306	6291458	4194314	4194322	4194338
5592473	3728401	8699432	5385411	7594758	10656022	8692157	5141928
8319991	7802584	14430633	9459596	8631090	5253884	6910898	7739404
8402210	7871965	8296161	5674361	4109094	9196750	8855914	5236099
9265594	4238072	8427674	6332873	4364676	5761210	6949116	8312092
13100213	8996109	8654100	6983952	5686434	5900706	6329250	7186338
13004636	5822423	11978606	9340242	7229551	8284896	13133271	9380930
8655473	3467050	8006920	12608938	8825713	5414081	6422941	6896830
10003018	4191976	9357347	7635557	10351713	8845612	7615762	8230687
8789710	8332328	8743972	8620479	8404366	8912109	6974700	8950600
8368950	4662957	15640179	12646391	6437053	9541722	4789964	7924105
9452558	8159252	9344782	9048400	11449319	9671021	7003574	7680487

Gambar 6. Hasil MidValue

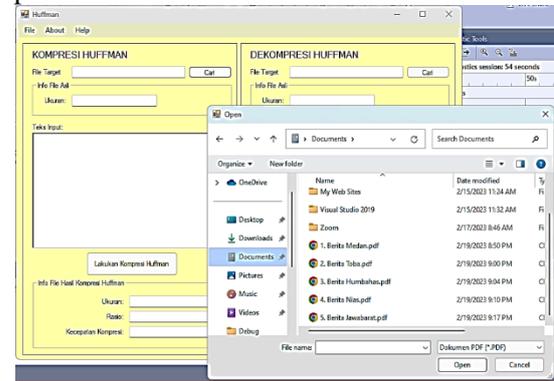
Nilai minvalue diganti dengan nilai midvalue jika bitset lebih besar dari 0, Nilai maxvalue diganti dengan nilai midvalue jika bitset lebih kecil sama dengan 0. Setelah itu dapatlah nilai – nilai ASCII akhir yaitu: 80 98 60 74 171 162 134 189 99 109 Akhirnya nilai ASCII ini berubah menjadi karakter – karakter simbol : 80=P, 98=b, 60=<, 74=B, 171=«, 162=ç, 134=†, 189=½, 99=c, 109=m Hasil akhir adalah: Pb<JB«ç†½cm

#### 4. Hasil Pengujian

Tujuan pengujian sistem adalah untuk memverifikasi bahwa sistem telah berfungsi sebagaimana diharapkan berdasarkan spesifikasi desain. Performa sistem dan pengembangan di masa mendatang akan

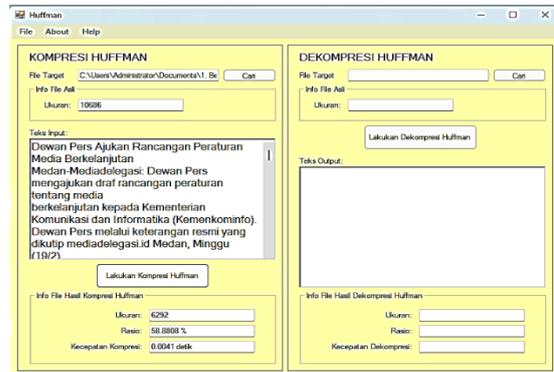
ditingkatkan dengan bantuan temuan pengujian.

Setelah Anda mengklik opsi kompresi pada menu utama, formulir akan ditampilkan seperti yang ditunjukkan di bawah ini. Dari sana, Anda dapat memilih file yang ingin dikompresi dengan mengklik tombol pencarian.



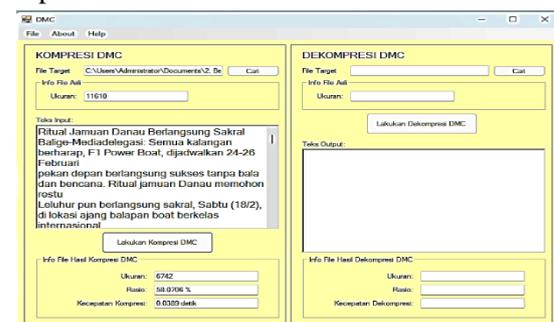
Gambar 7. Cari File .pdf Kompresi Huffman

Setelah file dipilih, langkah selanjutnya adalah menerapkan kompresi algoritma Huffman. Sistem kemudian akan memberikan hasil perhitungan, seperti yang terlihat pada gambar di bawah ini:



Gambar 8. Tampilan Kompresi Huffman

Kemudian untuk melakukan kompresi dengan algoritma Dynamic Markov Compression kembali ke menu utama klik Metode DMC dan terlihat hasil kompresi seperti dibawah ini.



Gambar 9. Tampilan Kompresi DMC

Setelah file berhasil dikompresi dengan Dengan membandingkan kedua pendekatan tersebut, kita dapat menentukan algoritma mana yang menghasilkan keluaran kompresi terkecil. Langkah selanjutnya adalah menyimpan file yang dikompresi menggunakan algoritma pilihan Anda; setelah itu, Anda dapat menggunakannya untuk mendekomposisi file tersebut.

Rasio kinerja dan kecepatan kompresi dan dekomposisi data dalam format .pdf diukur dalam pengujian sistem analisis penelitian. Hasil pengujian kompresi dan dekomposisi kategori teks ditunjukkan pada tabel di bawah ini.

Tabel 8. Hasil Pengujian Kompresi Huffman

Perco baan	Nama File (.pdf)	Ukura n file asli (Byte )	Ukura n Setela h Komp resi (Byte)	Rasio kompr esi (%)	Kecepat an Kompr esi (Detik)
1	Berita Medan.pdf	10686	6292	58.880 8 %	0.0019 detik
2	Berita Toba.pdf	11610	6815	58.699 4 %	0.0017 detik
3	Berita Humbahas.p df	11995	6870	57.273 9 %	0.0021 detik
4	Berita Nias.pdf	11932	6966	58.380 8 %	0.0023 detik
5	Berita JawaBarat.pd f	12573	7299	58.053 0 %	0.0018 detik
6	Berita Deliserdang. pdf	24952	14241	57.073 6 %	0.0058 detik
7	Berita Dairi.pdf	53262	30591	57.434 9 %	0.0096 detik

Hasil pengujian kompresi data teks berformat \*.pdf dengan metode Huffman menunjukkan keefektifannya, seperti yang ditunjukkan pada tabel 8 di atas. Setelah sampel Medan News berukuran 10,4 KB dikompresi menjadi 6,1 KB, menghasilkan rasio sebesar 59% dan durasi 1,9 ms. Karena banyaknya karakter yang berulang, maka menghasilkan rasio yang paling besar. Setelah dikompresi dari 11,3 KB menjadi 6,6 KB, menghasilkan rasio sebesar 59% dan durasi 1,7 ms pada Toba News. Setelah dikompresi dari 11,7 KB menjadi 6,7 KB, menghasilkan rasio sebesar 57% dan runtime 2,1 ms pada Humbahas News. Setelah dikompresi dari 11,6 KB menjadi 6,8 KB, menghasilkan rasio sebesar 58% dan durasi 2,3 ms pada Nias News. Setelah dikompresi dari 12,2 KB menjadi 7,1 KB, rasionya menjadi 58% dan membutuhkan waktu 1,8 ms untuk dimuat di

West Java News. Setelah dikompresi dari 24,3 KB menjadi 13,9 KB, rasionya menjadi 57% dan membutuhkan waktu 5,8 ms untuk dimuat di DeliSerdang News. Dan di Dairi News, berkasnya berubah dari 52 KB menjadi 29,8 KB setelah dikompresi, menghasilkan rasio 57% dan waktu proses 9,6 ms.

Tabel 9. Hasil Pengujian Dekompresi Huffman

Perco baan	Nama File (.pdf)	Ukura n file asli (Byte )	Ukura n Setela h Dekom presi (Byte)	Rasio kompre si (%)	Kecepat an Kompr esi (Detik)
1	Berita Medan.huf	6292	10686	169.834 7 %	0.0081 detik
2	Berita Toba.huf	6815	11610	170.359 5 %	0.0044 detik
3	Berita Humbahas. huf	6869	11995	174.625 1 %	0.0030 detik
4	Berita Nias.huf	6966	11932	171.289 1 %	0.0031 detik
5	Berita JawaBarat. huf	7298	12573	172.280 1 %	0.0033 detik
6	Berita Deliserdang. g.huf	14241	24952	175.212 4 %	0.0176 detik
7	Berita Dairi.huf	30591	53262	174.110 0 %	0.0187 detik

Dapat dilihat pada tabel 9 di atas bahwa hasil pengujian dekomposisi data teks format \*.pdf dengan algoritma Huffman membuktikan bekerja dengan baik. Pada sample Berita Medan dengan ukuran awal 6,2 KB setelah didekomposisi menjadi 10,6 KB menghasilkan rasio 170 % dengan durasi 8.1 ms. Pada Berita Toba dengan ukuran awal 6,8 KB setelah didekomposisi menjadi 11,6 KB menghasilkan rasio 170 % dengan durasi 4.4 ms. Pada Berita Humbahas dengan ukuran awal 6,8 KB setelah didekomposisi menjadi 11,9 KB menghasilkan rasio 175% dengan durasi 3 ms. Pada Berita Nias dengan ukuran awal 6,9 KB setelah didekomposisi menjadi 11,9 KB menghasilkan rasio 171% dengan durasi 3.1 ms. Pada Berita JawaBarat dengan ukuran awal 7,2 KB setelah didekomposisi menjadi 12,5 KB menghasilkan rasio 172% dengan durasi 3.3 ms. Pada Berita DeliSerdang dengan ukuran awal 14,2 KB setelah didekomposisi menjadi 24,9 KB menghasilkan rasio 175% dengan durasi 17.6 ms. Dan pada Berita Dairi dengan ukuran awal 30 KB setelah didekomposisi menjadi 53 KB menghasilkan rasio 174% dengan durasi 18.7 ms.

Tabel 10. Hasil Pengujian Kompresi DMC

Perco baan	Nama File (.pdf)	Ukuran file asli (Byte)	Ukuran Setelah Kompresi (Byte)	Rasio kompresi (%)	Kecepatan Kompresi (Detik)
1	Berita Medan.pdf	10686	6226	58.2631 %	0.0410 detik
2	Berita Toba.pdf	11610	6742	58.0706 %	0.0326 detik
3	Berita Humbahas .pdf	11995	6793	56.6319 %	0.0395 detik
4	Berita Nias.pdf	11932	6896	57.7942 %	0.0416 detik
5	Berita JawaBarat. pdf	12573	7219	57.4167 %	0.0483 detik
6	Berita Deliserdan g.pdf	24952	14137	56.6568 %	0.0863 detik
7	Berita Dairi.pdf	53262	30385	57.0482 %	0.1688 detik

Dapat dilihat pada tabel 10 di atas bahwa hasil pengujian kompresi data teks format \*.pdf dengan algoritma *Dynamic Markov Compression* membuktikan bekerja dengan sangat baik.

Pada sample Berita Medan dengan ukuran awal 10,6 KB setelah dikompresi menjadi 6,2 KB menghasilkan rasio 58 % dengan durasi 41 ms. Menghasilkan rasio paling terbanyak dikarenakan terdapat banyak pengulangan karakter didalamnya.

Pada Berita Toba dengan ukuran awal 11,6 KB setelah dikompresi menjadi 6,7 KB menghasilkan rasio 58 % dengan durasi 32.6 ms. Pada Berita Humbahas dengan ukuran awal 11,9 KB setelah dikompresi menjadi 6,7 KB menghasilkan rasio 57% dengan durasi 39.5 ms.

Pada Berita Nias dengan ukuran awal 11,9 KB setelah dikompresi menjadi 6,8 KB menghasilkan rasio 58% dengan durasi 41.6 ms.

Pada Berita JawaBarat dengan ukuran awal 12,5 KB setelah dikompresi menjadi 7,2 KB menghasilkan rasio 57% dengan durasi 48.3 ms.

Pada Berita DeliSerdang dengan ukuran awal 24,9 KB setelah dikompresi menjadi 14,1 KB menghasilkan rasio 57% dengan durasi 86.3 ms.

Dan pada Berita Dairi dengan ukuran awal 53,2 KB setelah dikompresi menjadi 30,3 KB menghasilkan rasio 57% dengan durasi 168.8 ms.

Tabel 11. Hasil Pengujian Dekompresi DMC

Perco baan	Nama File (.pdf)	Ukuran file asli (Byte)	Ukuran Setelah Dekompresi (Byte)	Rasio kompresi (%)	Kecepatan Kompresi (Detik)
1	Berita Medan.dmc	6226	10686	171.6351 %	0.0351 detik
2	Berita Toba.dmc	6742	11610	172.2049 %	0.0349 detik
3	Berita Humbahas .dmc	6793	11995	176.5788 %	0.0391 detik
4	Berita Nias.dmc	6896	11932	173.0278 %	0.0353 detik
5	Berita JawaBarat. dmc	7219	12573	174.1654 %	0.0394 detik
6	Berita Deliserdan g.dmc	14137	24952	176.5014 %	0.0703 detik
7	Berita Dairi.dmc	30385	53262	175.2904 %	0.1320 detik

Dapat dilihat pada tabel 11 di atas bahwa hasil pengujian dekompresi data teks format \*.pdf dengan algoritma *Dynamic Markov Compression* membuktikan bekerja dengan baik.

Pada sample Berita Medan dengan ukuran awal 6,2 KB setelah didekompresi menjadi 10,6 KB menghasilkan rasio 172 % dengan durasi 35.1 ms.

Pada Berita Toba dengan ukuran awal 6,7 KB setelah didekompresi menjadi 11,6 KB menghasilkan rasio 172 % dengan durasi 34.9 ms.

Pada Berita Humbahas dengan ukuran awal 6,7 KB setelah didekompresi menjadi 11,9 KB menghasilkan rasio 177% dengan durasi 39.1 ms.

Pada Berita Nias dengan ukuran awal 6,8 KB setelah didekompresi menjadi 11,9 KB menghasilkan rasio 173% dengan durasi 35.3 ms.

Pada Berita JawaBarat dengan ukuran awal 7,2 KB setelah didekompresi menjadi 12,5 KB menghasilkan rasio 174% dengan durasi 39.4 ms. Pada Berita Deli Serdang dengan ukuran awal 14,1 KB setelah didekompresi menjadi 24,9 KB menghasilkan rasio 177% dengan durasi 70.3 ms. Dan pada

Berita Dairi dengan ukuran awal 30,3 KB setelah didekompresi menjadi 53,2 KB menghasilkan rasio 175% dengan durasi 132.0 ms.

## KESIMPULAN

Temuan-temuan berikut diperoleh dari analisis, desain, implementasi, dan pengujian sistem, serta dari tinjauan pustaka:

Algoritma Huffman mencapai rasio kompresi rata-rata 57-97%, sedangkan metode DMC mencapai 57,31% dari file \*.pdf yang dikompresi. Dari file \*.pdf yang didekompresi, ditemukan bahwa rasio kompresi terbesar tidak selalu dikaitkan dengan file terbesar. Ketika membandingkan waktu kompresi rata-rata algoritma, algoritma Huffman mengungguli metode DMC.

Kami berharap PT. Media Delegasi Nusantara dapat menggunakan informasi yang dikumpulkan dari penelitian ini untuk mempercepat transmisi data mereka. Mentransfer file yang dikompresi juga menggunakan lebih sedikit bandwidth dan membutuhkan waktu lebih sedikit. Hal ini memiliki manfaat ganda, yaitu mengurangi biaya dan meningkatkan output.

## REFERENSI

- Aditya, C. S. K., & Wiyono, B. S. (2023). Pengembangan Fitur Peringkasan Artikel Otomatis pada Media Online Satukanal. *BAKTIMAS: Jurnal Pengabdian Pada Masyarakat*, 5(1), 60–67.
- Asdini, D., & Utomo, D. P. (2022). Analisis Perbandingan Kinerja Algoritma Huffman dan Algoritma Levenstein Dalam Kompresi File Dokumen Format .RTF. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, 6(November), 87–99.
- Mahanum, M. (2021). Tinjauan Kepustakaan. *ALACRITY: Journal of Education*, 1–12.
- Mahyuddin, M., Rangan, P. R., Nur, K. N., & Halim, H. (2021). *Perancangan bandar udara*. Yayasan Kita Menulis.
- Mansyuri, U. (2021). Kompresi Data Teks Dengan Metode Run Length Encoding. *Jurnal Ilmiah Sistem Informasi*, 1(2), 102–109.
- Manurung, P. (2020). Multimedia Interaktif Sebagai Media Pembelajaran Pada Masa Pandemi Covid 19. *Al-Fikru: Jurnal Ilmiah*, 14(1), 1–12.
- Nainggolan, S. (2019). Analisa Perbandingan Algoritma Goldbach Codes Dengan Algoritma Dynamic Markov Compression (DMC) Pada Kompresi File Teks Menggunakan Metode Eksponensial. *Majalah Ilmiah INTI*, 6(3), 395–399.
- Permana, R. (2022). *PENERAPAN ALGORITMA HUFFMAN UNTUK PEMAMPATAN FILE/Rendi Permana/15180036/TEKNIK INFORMATIKA/Pembimbing I: Adiat Pariddudin/Pembimbing II: Arif Harbani*.
- Prawiyogi, A. G., Sadih, T. L., Purwanugraha, A., & Elisa, P. N. (2021). Penggunaan Media Big Book untuk Menumbuhkan Minat Membaca di Sekolah Dasar. *Jurnal Basicedu*, 5(1), 446–452.
- Reswan, Y., & Tri Anggoro, L. (2024). Implementasi Metode Cepat Kompresi File Document Otomatis Pada Aplikasi Berbasis Web Menggunakan Algoritma Additive Code. *Jurnal Media Informatika*, 20(1), 211–215.
- Roring, R. S., Gunawan, T. I., & Samponu, Y. B. (2022). *Dasar dan Teori Sistem Multimedia*. JIU Press.
- Suhardi, Lubis, A. H., Aprilia, A., & Ningrum, I. A. (2023). Penerapan Metode Simple Multi Attribute Rating Technique pada Pemilihan Cafe Terfavorit. *Sistem Pendukung Keputusan Dengan Aplikasi*, 2(1), 1–11.
- Tanjung, R. Y., & Mesran, M. (2021). Perancangan Aplikasi Kompresi File Dokumen Menggunakan Algoritma Adiiitive Code. *JURIKOM (Jurnal Riset Komputer)*, 8(4), 108–113.
- Tarigan, P., & Matondang, Z. A. (2019). Pengujian Metode Kuantitasi Untuk Melakukan Kompresi Citra RGB. *JUKI: Jurnal Komputer Dan Informatika*, 1(2), 77–82.
- Ulfah, M. (2020). *DIGITAL PARENTING: Bagaimana Orang Tua Melindungi Anak-anak dari Bahaya Digital?* Edu Publisher.