KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)
Volume 6, Nomor 1, November 2022
DOI: 10.30865/komik.v6i1.5730
ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

Perbandingan Kinerja Algoritma Elias Delta Code Dan Algoritma Punctured Elias Code Dalam Kompresi File Teks

Arlansyah Tanjung

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: arlantanjoeng@gmail.com

Abstrak- Masalah ukuran data menjadi masalah penting dalam proses pertukaran data. Secara khusus, *file* teks adalah *file* yang relatif besar. Ukuran data yang besar menciptakan kendala untuk pengelolaan data, seperti kebutuhan ruang penyimpanan yang besar dan waktu transmisi yang lama. Untuk mengatasi permasalahan tersebut maka perlu diterapkan teknik kompresi. Pengguna dapat mengimpor *file* teks untuk dikompresi serta memilih algoritma kompresi. Dalam penelitian ini, algoritma *Elias Delta Code* dan *Punctured Elias Code* diterapkan untuk mengompresi *file* teks dan kemudian membandingkan kinerjanya berdasarkan parameter yang ditetapkan. Setelah dilakukan implementasi dan pengujian sistem dapat diketahui bahwa, algoritma *Elias Delta Code* memliki kinerja yang lebih baik dibandingkan algoritma *Punctured Elias Code* pada kompresi *file* teks dengan hasil nilai akhir menggunakan Metode Perbandingan *Eksponential*, algoritma *Elias Delta Code* dengan nilai akhir 6,526 sedangkan algoritma *Punctured Elias Code* 6,653, Sehingga semakin tinggi total nilai yang diperoleh maka akan semakin tinggi jumlah usaha yang dilakukan oleh algoritma tersebut. Berdasarkan analisa tersebut maka Algoritma *Elias Delta Code* yang menjadi algoritma memiliki kinerja terbaik dalam kompresi *file* teks.

Kata Kunci: Elias Delta Code; Punctured Elias Code; Kompresi; File Teks; Metode Perbandingan Eksponential

Abstract-The problem of data size is an important problem in the data exchange process. In particular, text files are relatively large files. Large data sizes create constraints for data management, such as the need for large storage space and long transmission times. To overcome these problems it is necessary to apply compression techniques. Users can import text files to be compressed as well as choose a compression algorithm. In this research, Elias Delta Code and Punctured Elias Code algorithms are applied to compress text files and then compare their performance based on the set parameters. After implementing and testing the system, it can be seen that the Elias Delta Code algorithm has better performance than the Punctured Elias Code algorithm in text file compression with the final value using the Exponential Comparison Method, the Elias Delta Code algorithm with a final value of 6.526 while the Punctured Elias Code algorithm 6.653, So the higher the total value obtained, the higher the amount of work done by the algorithm. Based on this analysis, the Elias Delta Code Algorithm which is the algorithm has the best performance in text file compression.

Keywords: Elias Delta Code; Punctured Elias Code; Compression; Text Files; Exponential Comparison Method

1. PENDAHULUAN

Berbagi dan berkirim informasi sudah menjadi hal yang lumrah dizaman sekarang ini, sehingga memudahkan banyak pekerjaan manusia apalagi di saat pandemi COVID-19 baru-baru ini terjadi sehingga sangat banyak kegiatan yang dilakukan secara *online*, baik pekerjaan rumah, surat menyurat, dll[1]. Ada banyak alternatif yang dapat digunakan untuk melakukan tugas tersebut, salah satunya adalah dengan menggunakan *file* teks yang merupakan dokumen atau *file* yang berisi teks, beberapa jenis *file* teks antara lain .rtf, .txt, .docx dan lain-lain[2]. Seperti diketahui, data teks lebih kecil dari ukuran gambar, dan *file* teks sering digunakan oleh banyak instansi untuk menyimpan data dan berbagi informasi, sehingga untu menyimpan *file* membutuhkan kapasitas yang banyak.

Proses pengiriman yang cepat merupakan hal yang sangat dibutuhkan dimasa saat ini, karena sangat menghemat waktu, namun terkadang kita mengalami masalah dalam proses pengiriman *file* yang memiliki ukuran yang cukup besar sehingga memperlambat proses pengiriman, dimana besar dan kecilnya ukuran sebuah *file* sangat mempengaruhi kecepatan pengiriman.

Ada banyak hal yang dapat dilakukan untuk menyelesaikan masalah tersebut dengan mengurangi ukuran *file* atau yang biasa disebut proses kompresi[3]. Ada banyak contoh algoritma yang telah digunakan untuk melakukan kompresi *file* diantaranya adalah algoritma *Sequitur Huffman, Goldbach, Elias Delta Code* dan *Punctured Elias Code* dan masih banyak algoritma lainnya.

Beberapa faktor yang biasanya dipertimbangkan ketika memilih algoritma kompresi yang tepat adalah kecepatan kompresi, sumber daya yang dibutuhkan (memori, kecepatan PC), ukuran *file* terkompresi, jumlah redundansi dan kompleksitas algoritma[4]. Tidak ada metode kompresi yang paling efektif untuk kompresi semua jenis *file*. Pada penelitian ini penulis akan membandingkan algoritma *Elias Delta Code* dengan *Punctured Elias Code*, masing-masing algoritma memiliki keunggulan dalam hal kompresi. Kedua algoritma ini telah diuji untuk proses kompresi dan dekompresi *file* teks. Setelah itu dilakukan analisis statistik untuk membandingkan kinerja masing-masing metode berdasarkan faktor hasil rasio kompresi, rasio kompresi, penghematan ruang dan waktu pemrosesan untuk mengetahui hasil kompresi *file* teks dengan format .docx terbaik sehingga dapat digunakan dan diadopsi oleh banyak instansi[5].

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi terdiri dari mengubah data dalam bentuk set karakter menjadi bentuk yang dikodekan dengan tujuan menghemat ruang penyimpanan dan waktu transmisi data. Beberapa faktor dipertimbangkan ketika memutuskan algoritma mana yang akan digunakan untuk mengompresi data, seperti sumber daya yang dibutuhkan (memori, kecepatan PC), kecepatan kompresi, ukuran hasil kompresi, jumlah redundansi dan kompleksitas algoritma[6].

2.2 *File* Teks

File teks adalah file yang berisi informasi dalam bentuk teks. Data yang diperoleh dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam *database* merupakan contoh *input* data teks yang terdiri dari karakter, angka, dan tanda baca. *Input* dan *output* data teks direpresentasikan sebagai kumpulan karakter atau sistem kode yang dikenali oleh sistem komputer[7].

2.3 Algoritma *Elias Delta Code*

Elias delta code adalah sebuah algoritma kompresi yang dibuat oleh Elias menggunakan kode yang telah dia buat sebelumnya, yaitu *Elias Gamma code*, sebagai *building block*. Kode *Gamma*, Elias menambah panjang kode dalam *unary* (α). dalam kode berikutnya, δ (delta), ditambahkan pada panjang kode dalam biner (β). Dengan demikian, *Elias Delta Code* yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun[8]. Adapun aturan untuk mengkodekan sebuah bilangan dengan menggunakan *elias delta code* adalah:

a. Tuliskan n dalam bilangan biner (*binary*). Bit yang paling kiri (paling signifikan) akan menjadi 1.
b. Hitung jumlah bit-nya, hapus bit paling kiri dari n dan tambahkan perhitungan dalam bilangan biner (*binary*) pada bagian kiri dari n setelah bit paling kiri dari n dihapus.
c. Hitung jumlah bit-nya, hapus bit paling kiri dari n dan tambahkan perhitungan dalam bilangan biner (*binary*) pada bagian kiri dari n setelah bit paling kiri dari n dihapus.

Kurangi 1 dari perhitungan pada langkah ke-2 dan tambahkan jumlah nol ke kode. Ketika langkah-langkah ini diterapkan pada integer ke-17, hasilnya adalah : 17 = 10001 (lima bit). Hapus angka 1 yang paling kiri dan tambahkan 5 = 101 sehingga hasilnya 101|0001. Tiga bit sudah ditambahkan, kemudian tambahkan 2 nol untuk mendapatkan kode delta 00|101|0001[9].

Tabel 1. Pembentukan Kode *Elias Delta Code*

$1 = 2^0 + 0 \Rightarrow \lvert L/ = 0 \Rightarrow 1$	$10 = 2^3 + 2 \Rightarrow \lvert L/ = 3 \Rightarrow 00100010$
$2 = 2^1 + 0 \Rightarrow \lvert L/ = 1 \Rightarrow 0100$	$11 = 2^3 + 3 \Rightarrow \lvert L/ = 3 \Rightarrow 00100011$
$3 = 2^1 + 0 \Rightarrow \lvert L/ = 1 \Rightarrow 0100$	$12 = 2^3 + 4 \Rightarrow \lvert L/ = 3 \Rightarrow 00100010$
$4 = 2^2 + 0 \Rightarrow \lvert L/ = 2 \Rightarrow 01100$	$13 = 2^3 + 5 \Rightarrow \lvert L/ = 3 \Rightarrow 00100101$
$5 = 2^2 + 1 \Rightarrow \lvert L/ = 2 \Rightarrow 01101$	$14 = 2^3 + 6 \Rightarrow \lvert L/ = 3 \Rightarrow 00100010$
$6 = 2^2 + 2 \Rightarrow \lvert L/ = 2 \Rightarrow 01110$	$15 = 2^3 + 7 \Rightarrow \lvert L/ = 3 \Rightarrow 00100111$
$7 = 2^2 + 3 \Rightarrow \lvert L/ = 2 \Rightarrow 01111$	$16 = 2^4 + 0 \Rightarrow \lvert L/ = 4 \Rightarrow 001010000$
$8 = 2^3 + 0 \Rightarrow \lvert L/ = 3 \Rightarrow 00100000$	$17 = 2^4 + 1 \Rightarrow \lvert L/ = 4 \Rightarrow 001010001$
$9 = 2^3 + 1 \Rightarrow \lvert L/ = 3 \Rightarrow 00100001$	$18 = 2^4 + 2 \Rightarrow \lvert L/ = 4 \Rightarrow 001010010$

2.4 Algoritma *Punctured Elias Code*

Punctured Elias Code untuk bilangan integer dirancang oleh Peter Fenwick dalam percobaan untuk meningkatkan kinerja *the Burrows–Wheeler transform*. Istilah *Punctured* berasal dari tempat pengawasan *Error Code-Code* (ECC). ECC terdiri dari data asli ditambah beberapa *check bit*. Jika beberapa *check bit* dihilangkan, untuk mempersingkat urutan kode, kode yang dihasilkan disebut *Punctured*[10]. Cara membangun Kode *Punctured Elias Code* adalah sebagai berikut:

a. Ambil bilangan biner dari n.
b. *Reversed* (balikkan bit-bitnya), dan siapkan *flag* untuk menunjukkan jumlah bit yang bernilai 1 di dalam n.
c. Untuk setiap bit 1 di dalam n kita siapkan *flag* dari 1 dan akhiri *flag* dengan 0.

Gabungkan *flag* dengan bilangan biner yang sudah dibalikkan (*reversed*)[11].

Tabel 2. Pembentukan Kode *Punctured Elias Code*

No.	Binary of n	Reserved	Flag	Flag I Reserved	P1
0	0	-	-	-	0
1	1	1	10	10 I 1	101
2	10	01	10	10 I 01	1001
3	11	11	110	110 I 11	11011
4	100	001	10	10 I 001	10001
5	101	101	110	110 I 101	110101

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

No.	Binary of n	Reserved	Flag	Flag I Reserved	P1
6	110	011	110	110 I 011	110011
7	111	111	1110	1110 I 111	1110111

2.5 Metode Perbandingan *Eksponential*

Metode perbandingan *eksponential* adalah satu metode untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak.

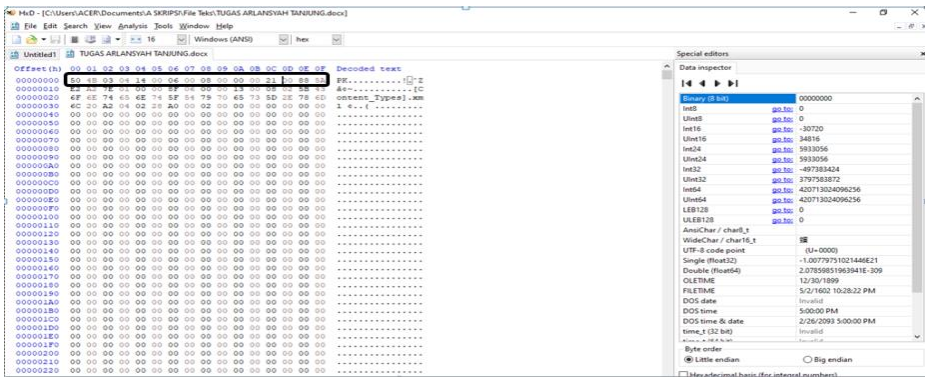
$$\text{Total Nilai } (TN_i) = \sum_{j=1}^{m} RK_{ij})^{TKKj} \quad (1)$$

Keterangan :
TNi = Total nilai alternatif ke-i.
RKij = Derajat kepentingan relatif kriteria ke-j pada keputusan ke-i, yang dapat dinyatakan dengan skala ordinal (1,2,3,4,5).
TKKj = Derajat kepentingan kriteria keputusan ke-j; TKKj > 0;
M = Jumlah kriteria keputusan.
n = Jumlah pilihan keputusan.
J = 1,2,3,.... m; m = jumlah kriteria.
I = 1,2,3.....n; n = jumlah pilihan alternatif[12]

3. HASIL DAN PEMBAHASAN

Analisa masalah merupakan tahap pengembangan sistem yang mendefinisikan proses penentuan sebab dan akibat dari dibangunnya sistem agar sistem yang dibangun dapat bekerja dengan baik sesuai dengan tujuan sistem. Masalah yang akan diangkat dari penelitian ini adalah bagaimana membandingkan kinerja algoritma *Elias Delta Code* dan *algoritma Punctured Elias Code* untuk mengetahui kinerja terbaik antara kedua algoritma tersebut dengan melakukan kompresi, mengukur waktu yang paling efektif dalam melakukan kompresi akan menjadi algoritma yang terbaik.

Untuk mendapatkan nilai *Hexadecimal* dari *file* “TUGAS ARLANSYAH TANJUNG.docx” dan untuk keperluan hitungan manual maka dibutuhkan aplikasi bantuan untuk melihat nilai *hexadecimal* dari *file* tersebut dengan menggunakan aplikasi *HxD*. Adapun hasil nilai *Hexadecimal* dari *file* “TUGAS ARLANSYAH TANJUNG.docx” berupa sampel dapat dilihat pada gambar dibawah ini:



Gambar 1. Nilai *Hexadecimal File* Teks

3.1 Penerapan Algoritma *Elias Delta Code*

Penerapan algoritma *Elias Delta Code* hanya menggunakan 16 sampel saja yaitu nilai *hexadecimal* dari *file* “TUGAS ARLANSYAH TANJUNG.docx” yang berisikan nilai *Hexadecimal* = 50, 4B, 03, 04, 14, 00, 06, 00, 08, 00, 00, 00, 21, 00, 88, 5A.

Tabel 3. String sebelum dikompresi degan algoritma EDC

No	*Hexadesimal*	*Frequency*	*ASCII Code (Decimal)*	*ASCII Code (Binary)*	Bit	Freq x Bits
1	00	6	00	00000000	8	48
2	50	1	80	01010000	8	8
3	4b	1	75	01001011	8	8
4	03	1	03	00000011	8	8
5	04	1	04	00000100	8	8

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

No	*Hexadesimal*	*Frequency*	*ASCII Code (Decimal)*	*ASCII Code (Binary)*	Bit	Freq x Bits
6	14	1	20	00010100	8	8
7	06	1	06	00000110	8	8
8	08	1	08	00001000	8	8
9	21	1	33	00100001	8	8
10	88	1	136	10001000	8	8
11	5A	1	90	01011010	8	8
Total						128

Setelah diurutkan secara *descending* dan kemudian memasuki fase kompresi, dapat dilihat tabel4.2 dibawah ini:

Tabel 4. String setelah dikompresi dengan algoritma EDC

No	*Character*	*Frequency*	*Elias Delta Code*	Bit	Freq x Bits
1	00	6	1	1	6
2	50	1	0100	4	4
3	4b	1	0101	4	4
4	03	1	01100	5	5
5	04	1	01101	5	5
6	14	1	01110	5	5
7	06	1	01111	5	5
8	08	1	00100000	8	8
9	21	1	00100001	8	8
10	88	1	00100010	8	8
11	5A	1	00100011	8	8
Total Bit					66

Hexadecimal yang telah di baca adalah "50, 4B, 03, 04, 14, 00, 06, 00, 08, 00, 00, 00, 21, 00, 88, 5A ", maka hasil string bit yang telah dikompresi adalah:
"01000101011000110101110101111100100000111001000011001000100010001100000100000111"

Tabel 5. Penambahan *Padding* dan *Flagging*

Padding	*Flagging*
$66\ mod\ 8 = 2 = 2$ $7 - n + "1"$ $7 - 2 + "1" = 000001$	$9 - n$ $9 - 2 = 7 = 00000111$

Setelah dilakukan proses *padding* dan *flagging*, maka total bit menjadi **80** bit.
"01000101011000110101110101111100100000111001000011001000100010001100000100000111"

3.1.1 Penerapan Metode *Punctured Elias Code*

Dalam melakukan analisa kompresi *file* teks dengan menerapkan metode *Punctured Elias Code* (PEC) untuk mengurangi ukuran pada *file* teks. Penulis menggunakan *file* "TUGAS ARLANSYAH TANJUNG.docx" dengan jumlah 16 sampel yang berisikan:
Nilai *Hexadecimal* = 50, 4B, 03, 04, 14, 00, 06, 00, 08, 00, 00, 00, 21, 00, 88, 5A.

Tabel 6. String sebelum dikompresi dengan algoritma PEC

No	Hexadesimal	Frequency	ASCII Code (Decimal)	ASCII Code (Binary)	Bits	Freq x Bits
1	00	6	00	00000000	8	48
2	50	1	80	01010000	8	8
3	4b	1	75	01001011	8	8
4	03	1	03	00000011	8	8
5	04	1	04	00000100	8	8
6	14	1	20	00010100	8	8
7	06	1	06	00000110	8	8
8	08	1	08	00001000	8	8
9	21	1	33	00100001	8	8

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

No	Hexadesimal	Frequency	ASCII Code (Decimal)	ASCII Code (Binary)	Bits	Freq x Bits
10	88	1	136	10001000	8	8
11	5A	1	90	01011010	8	8
Total						128

Setelah dilakukan pengurutan secara *descending*, selanjutnya masuk ke tahap proses kompresi, dapat dilihat pada tabel 4.4 dibawah ini:

Tabel 7. String setelah dikompresi dengan algoritma PEC P1

No	*Hexadecimal*	*Frequency*	*Punctured Elias Code*	Bits	Freq x Bits
0	00	6	0	1	6
1	50	1	101	3	3
2	4b	1	1001	4	4
3	03	1	11011	5	5
4	04	1	10001	5	5
5	14	1	110101	6	6
6	06	1	110011	6	6
7	08	1	1110111	7	7
8	21	1	100001	6	6
9	88	1	1101001	7	7
10	5A	1	1100101	7	7
Total Bit					62

Hexadecimal yang telah di baca adalah “50, 4B, 03, 04, 14, 00, 06, 00, 08, 00, 00, 00, 21, 00, 88, 5A ”, maka susunan string bit yang tercipta ialah:
“10110011101110001110101111001111110111111100001111010011100101”

Tabel 8. Penambahan *Padding* dan *Flagging* pada algoritma PEC

Padding	*Flagging*
$62\ mod\ 8 = 6 = 6$ $7 - n + "1"$ $7 - 6 + "1" = 01$	$9 - n$ $9 - 6 = 3 = 00000011$

Setelah ditambah *padding* dan *flagging* maka total bit menjadi **72** bit.
“101100111011100011101011110011111101111111000011110100111001010100000011”

3.1.2 Penerapan Metode *Exponential*

a. Menentukan kriteria pengujian kinerja algoritma
Untuk dapat membandingkan kedua algoritma tersebut, perlu ditentukan kriteria untuk menganalisis proses dan operasinya. Kriteria tersebut ialah *Ratio of Compression* (Rc), *Compression Ratio* (Cr), *Space Savings* (SS) dan *Time Processing* (TP)
1. *Ratio of Compression* (Rc)

$$\text{Rc} = \frac{\text{Ukuran data sebelum kompresi}}{\text{Ukuran data setelah kompresi}}$$

$$\text{EDC} = \frac{128\ bit}{80\ bit}$$

EDC = 1,6

$$\text{PEC} = \frac{128\ bit}{72\ bit}$$

EC = 1,7

2. *Compression Ratio* (Cr)

$$\text{Cr} = \frac{\text{Ukuran data setelah dikompresi}}{\text{Ukuran data sebelum dikompresi}}\ x\ 100\%$$

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

$$EDC = \frac{80\ bit}{128\ bit}\ x\ 100\%$$

$$EDC = 62{,}5\%$$

$$PEC = \frac{72\ bit}{128\ bit}\ x\ 100\%$$

$$PEC = 56{,}25\%$$

3. *Space Savings* (SS)

$$SS = \left(1 - \frac{\text{Ukuran data setelah dikompresi}}{\text{Ukuran data sebelum dikompresi}}\ x\ 100\%\right)$$

$$EDC = \left(1 - \frac{80}{128}\ x\ 100\%\right)$$

$$EDC = 37{,}5\%$$

$$PEC = \left(1 - \frac{72}{128}\ x\ 100\%\right)$$

$$PEC = 43{,}75\%$$

b. Menentukan Bobot Kriteria

Penentuan bobot merupakan faktor yang sangat mempengaruhi hasil analisis, sehingga penulis memberikan bobot kriteria sesuai dengan derajat pengaruhnya dalam menentukan kinerja terbaik dari algoritma, secara rinci. pembobotan kriteria dijelaskan pada tabel berikut:

Tabel 9. Pembobotan Kriteria

Kriteria	Pserentase Pengaruh Kriteria	Bobot Range (0-1)
Ratio of Compression	10%	0.1
Compression Ratio	10%	0.1
Space Saving	30%	0.3
Time Processing	50%	0.5

c. Pemberian nilai pada setiap kriteria

Pada kriteria yang telah dibentuk harus diberikan nilai. Nilai dapat dilihat dibawah ini dimana nilainya diambil berdasarkan analisa algoritma *Elias Delta Code* dan algoritma *Punctured Elias Codes* sebelumnya.

Tabel 10. Pemberian Nilai Pada Setiap Kriteria

Alternatif	Kriteria					
	TP	RC		CR		SS
		SB	SD	SD	SB	
Algoritma *Elias Delta Code*	1,001 ms	128 (bit)	80 (bit)	80 (bit)	128 (bit)	1 - CR * 1000%
Algoritma *Punctured Elias Code*	0,9956 ms	128 (bit)	72 (bit)	72 (bit)	128 (bit)	1 - CR * 1000%

Keterangan :
RC : *Ratio Of Compression*
CR : *Compression Ratio (*CR)
SS : *Space Saving*
SB : Sebelum Dikompresi
SD : Sesudah Dikompresi

d. Menghitung Skor

Setelah nilai masing-masing kriteria dimasukkan, langkah selanjutnya adalah melakukan perhitungan menggunakan rumus metode perbandingan eksponensial.

Tabel 11. Hasil Perhitungan Menggunakan Metode Perbandingan *Eksponential*

Kriteria	Bobot	Algoritma EDC	Algoritma PEC
RC	0.1	1,6	1,7
CR	0.1	62,5	56,25
SS	0.3	37,5	43,75

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

TP	0.5	1,001	0,9956
Total Nilai	1	6,526	6,653

Nilai Algoritma EDC $= (1,6)^{0,1} + (62,5)^{0,1} + (37,5)^{0,3} + (1,001)^{0,5}$

$= 1,048 + 1,512 + 2,966 + 1,000$

$= 6,526$

Nilai Algoritma PEC $= (1,7)^{0,1} + (56,25)^{0,1} + (43,75)^{0,3} + (0,9956)^{0,5}$

$= 1,054 + 1,496 + 3,106 + 0,997$

$= 6,653$

e. Menentukan hasil atau prioritas keputusan
Setelah mendapatkan nilai akhir atau nilai total dari masing-masing alternatif, langkah selanjutnya yang harus dilakukan adalah menentukan prioritas keputusan berdasarkan nilai dari masing-masing alternatif. Hasil keputusan penentuan prioritas dapat dilihat pada tabel di bawah ini:

Tabel 12. Prioritas Keputusan

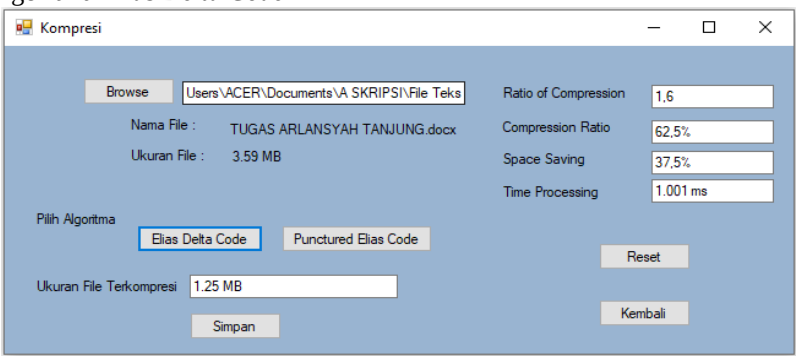
Alternatif	Total Nilai	Rangking
Algoritma *Elias Delta Code*	6,526	1
Algoritma *Punctured Elias Code*	6,653	2

Pada tabel di atas dapat kita lihat bahwa nilai total dari alternatif terendah mendapatkan peringkat pertama, hal tersebut dikarenakan semakin tinggi nilai total yang diperoleh maka semakin tinggi pula usaha algoritma dalam mengompresi *file* teks. Berdasarkan analisis tersebut, algoritma *Elias Delta Code*, merupakan algoritma dengan performa terbaik saat mengompresi *file* teks.

3.2 Implementasi

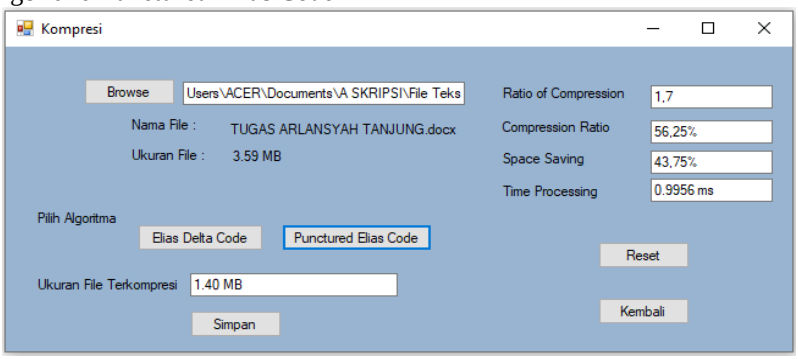
Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun dari fungsi-fungsi yang telah didefinisikan sebelumnya dan dimaksudkan untuk mengetahui performansi dari masing-masing algoritma. Pada tahap ini akan dilakukan pengujian pada *file* dengan ekstensi *.docx.

a. Hasil Kompresi Algoritma *Elias Delta Code*



Gambar 2. Hasil Proses Kompresi Menggunakan *Elias Delta Code*

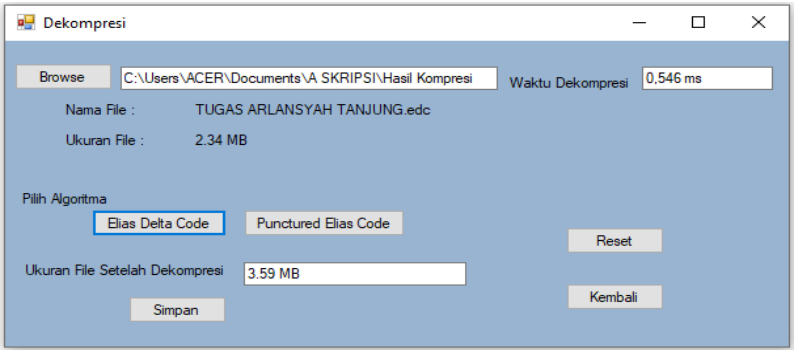
b. Hasil Kompresi Algoritma *Punctured Elias Code*



Gambar 3. Hasil Proses Kompresi Menggunakan *Punctured Elias Code*

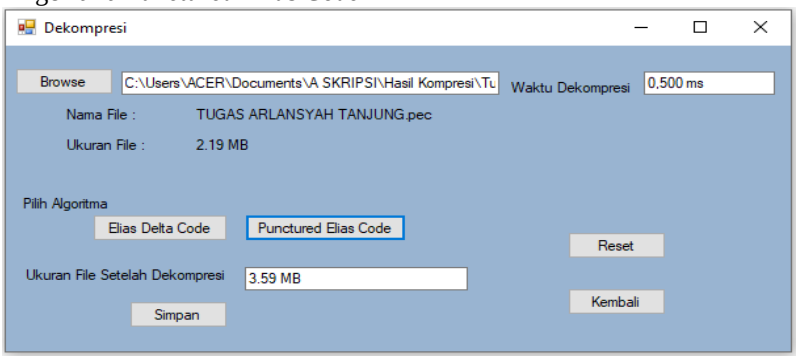
ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

c. Hasil Dekompresi Algoritma *Elias Delta Code*



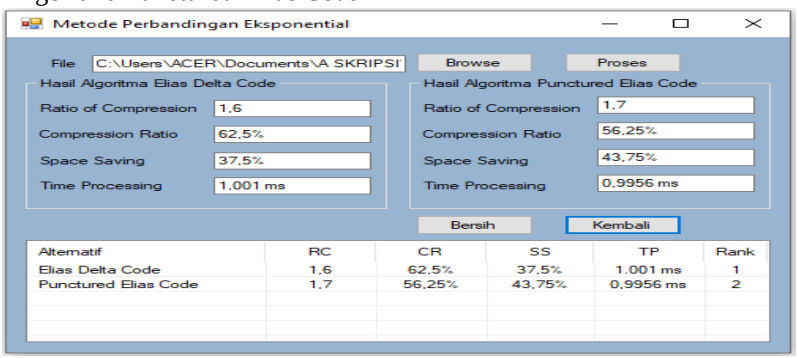
Gambar 4. Hasil Proses Dekompresi Menggunakan *Elias Delta Code*

d. Hasil Dekompresi Algoritma *Punctured Elias Code*



Gambar 5. Hasil Proses Dekompresi Menggunakan *Punctured Elias Code*

e. Hasil Dekompresi Algoritma *Punctured Elias Code*



Gambar 6. Hasil Perbandingan Menggunakan Metode Perbandingan *Exponential*

4. KESIMPULAN

Berdasarkan pembahasan dan evaluasi dari bab-bab sebelumnya dan analisis terhadap kompresi *file* teks yang berekstensi *.docx, maka dapat di tarik kesimpulan sebagai berikut: Dalam menerapkan algoritma *Elias Delta Code* dan algoritma *Punctured Elias Code* dalam mengkompresi *file* teks sangat baik, hasilnya dari jumlah karakter juga tidak berkurang dari *file* aslinya atau tidak ada pengurangan (isi *file* nya). Aplikasi yang dirancang telah mampu melakukan kompresi *file* teks dengan menggunakan algoritma *Elias Delta Code* dan algoritma *Punctured Elias Code*, dan melakukan dekompresi terhadap hasil kompresi dengan algoritma *Elias Delta Code* dan algoritma *Punctured Elias Code*, menjadi *file* asli. Hasil perbandingan kompresi *file* teks antara algoritma *Elias Delta Code* dan algoritma *Punctured Elias Code* memiliki hasil berbeda, dalam penerapan Metode Perbandingan Eksponential dengan empat kriteria yaitu *Ratio of Compression* (RC), *Compression Ratio* (CR), *Space Saving* (SS) dan *Time Processing* (TP) dengan hasil akhir algoritma *Elias Delta Code* adalah 6,527 sedangkan algoritma *Punctured Elias Code* adalah 6,653. Sehingga semakin tinggi total nilai yang diperoleh maka akan semakin tinggi jumlah usaha yang dilakukan oleh algoritma tersebut. Berdasarkan analisa tersebut maka Algoritma *Elias Delta Code* yang menjadi algoritma memiliki kinerja terbaik dalam kompresi *file* teks.

REFERENCES

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 182-190

[1] N. B. Argaheni, "Sistematik Review: Dampak Perkuliahan Daring Saat Pandemi COVID-19 Terhadap Mahasiswa Indonesia," *PLACENTUM J. Ilm. Kesehat. dan Apl.*, vol. 8, no. 2, pp. 99–108, 2020, doi: 10.20961/placentum.v8i2.43008.
[2] M. Subli, B. Sugiantoro, and Y. Prayudi, "Metadata Forensik Untuk Mendukung Proses Investigasi Digital," *Data Manag. dan Teknol. Inf.*, vol. 18, no. 1, pp. 44–50, 2017.
[3] P. Fitria, "Penerapan Algoritma Rice Codes Pada Aplikasi Kompresi File Gambar," *J. Comput. Syst. Informatics*, vol. 1, no. 3, pp. 158–165, 2020.
[4] S. D. Yunita and H. Sunandar, "Analisis Kombinasi Algoritma Knapsack dan Run Length Enconding (RLE) pada File Teks," vol. 01, pp. 6–13, 2019.
[5] E. Salsa Nugraha, A. R. Padri, O. Nurdiawan, A. Faqih, and S. Anwar, "Implementasi Aplikasi Pengaduan Masyarakat Berbasis Android Pada Gedung DPRD," *J. Ris. Komputer)*, vol. 8, no. 6, pp. 2407–389, 2021, doi: 10.30865/jurikom.v8i6.3679.
[6] D. A. Depika and S. D. Nasution, "Penerapan Algoritma Punctured Elias Codes Dalam Kompresi Citra," *Build. Informatics, Technol. Sci.*, vol. 2, no. 2, pp. 176–187, 2020, doi: 10.47065/bits.v2i2.301.
[7] Apijuddin, "Perancangan Aplikasi Kompresi File Teks Menggunakan Algoritma Stout Codes," vol. 9, pp. 183–188, 2021.
[8] L. V Simanjuntak, "Perbandingan Algoritma Elias Delta Code dengan Levenstein Untuk Kompresi File Teks," *J. Comput. Syst. ...*, vol. 1, no. 3, pp. 184–190, 2020, [Online]. Available: https://ejurnal.seminar-id.com/index.php/josyc/article/view/168.
[9] F. J. I. Harefa, "Penerapan Algoritma RC6 dan Algoritma Elias Delta Code Pada Aplikasi Pengamanan dan Kompresi Short Message Service (SMS) Berbasis Android," *J. Inf. Sist. Res.*, vol. 1, no. 3, pp. 155–161, 2020.
[10] F. Juleha, Guidio Leonarde Ginting, "Penerapan Algoritma Punctured Elias Code Pada Aplikasi Kompresi Short Message Service (SMS)," *J. Ris. Komput.*, vol. 7, no. 5, pp. 552–557, 2020, doi: 10.30865/jurikom.v7i5.2540.
[11] P. S. Hasmita and C. F. Sianturi, "Implementasi Algoritma Punctured Elias Code Untuk Kompresi File Audio Pada Aplikasi Lagu Rohani," vol. 10, pp. 46–50, 2021.
[12] Y. Devianto and S. Dwiasnati, "Aplikasi Pengambilan Keputusan Indeks Kepuasaan Masyarakat Dengan Metode Perbandingan Eksponensial (MPE) Pada Unit Pelayanan Masyarakat Dengan Alat Microcontroller Sebagai Alat Bantu Survey," *J. Ilm. FIFO*, vol. 10, no. 1, p. 13, 2018, doi: 10.22441/fifo.v10i1.2946.