Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 110-119
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Perbandingan Algoritma Elias Omega Code Dan Elias Delta Code Dalam Mengkompresi File Video (Mp4)

Soraya Ainun Fitrya

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: sorayaainun0605@gmail.com

Abstrak−Ukuran file video tergolong sangat besar dimana semakin baik kualitas file video dan waktu file video cukup lama, maka ukuran file terbilang sangat besar dalam menyimpan file tersebut. Dengan ukuran file vidoe yang sangat juga mempengaruhi lamanya waktu pengiriman file video.Adapun solusi dalam permasalah ini adalah dengan melakukan kompresi file video. Ada banyak algoritma di dalam kompresi file video, contohnya algoritma elias omega code, elias delta code, stout code, punctured elias code dan masih banyak algoritma kompresi lainnya. Sehingga dengan banyaknya algoritma tersebut perlu dilakukan pengujian dari beberapa algoritma kompresi ataupun perbandingan algoritma. Perbandinagn algoritma bertujuan untuk mengertahui algoritma manakah yang lebih akurat dalam melakukan proses kompresi data. Algoritma yang akan dibandingkan yaitu elias omega code dan elias delta code. Sedangkan parameter perbandingan dari kedua algoritma adalah rasio kompresi dan space saving.

Kata Kunci: Kompresi; Perbandingan; Video Mp4; Elias Omega Code; Elias Delta Code

Abstract−The video file size is quite large where the better the video file quality and the longer the video file time, the file size is quite large in storing the file. With the video file size which greatly affects the length of time the video file is sent. The solution to this problem is to compress the video file. There are many algorithms in video file compression, for example the elias omega code algorithm, elias delta code, stout code, punctured elias code and many other compression algorithms. So with so many algorithms, it is necessary to test several compression algorithms or algorithm comparisons. Algorithm comparison aims to find out which algorithm is more accurate in carrying out the data compression process. The algorithms that will be compared are elias omega code and elias delta code. While the comparison parameters of the two algorithms are the compression ratio and space saving.

Keywords: Compression; Comparison; Mp4 Videos; Elias Omega Code; Elias Delta Code

1. PENDAHULUAN

Kompresi data merupakan sebuah teknik ilmu komputer untuk mengecilkan ukuran data atau memampatkan data. Jadi data yang akan dimampatkan menjadi lebih kecil dari ukuran sebenarnya dengan tujuan menghemat ruang penyimpanan. Apabila kompresi data dilakukan, maka tempat penyimpanan yang dibutuhkan tidak terlalu besar. Selain dinilai efisien, kompresi data juga mempercepat waktu pertukuran data[1][2]. Seiring berkembangnya teknologi pada zaman sekarang ini, data memiliki peranan yang sangat penting, besarnya jumlah data yang telah disimpan didalam memori ataupun hardisk akan mengakibatkan menambah besarnya kapasitas pemakaian pada media penyimpanan.

MPEG-4 atau yang lebih dikenal dengan Mp4 merupakan format kontainer (wadah). Mp4 dapat digunakan untuk menyimpan data audio dan video. Karena mp4 adalah format kontainer, maka ia memiliki metode standar pengkodean untuk informasi audio atau video. Mp4 menggunakan *code* yang menentukan bagaimana suatu audio atau video akan dikodekan. Ukuran file video dengan format Mp4 tergolong memiliki ukuran yang cukup besar.

Ukuran file video tergolong sangat besar dan mempengaruhi ruang penyimpanan serta waktu pengiriman data juga tergolong lama dengan ukuran file yang begitu besar. Oleh karena itu perlunya dilakukan kompresi untuk memampatkan isi file video (mp4). Ketika proses kompresi dilakukan harus adanya algoritma pendukung dalam kompresi. Algoritma mempunyai kelebihan dan kelemahan dalam melakukan proses kompresi data.

Elias omega code dimana karakter yang paling banyak tersimpan pada bit yang paling kecil sesuai dengan code_so_far ke-n. Memakai dirinya sendiri dengan cara rekursif agar mengkodekan prefix M, oleh karenanya terkadang juga dikatakan kode elias rekursif. Algoritma elias omega code melakukan pengurutan karakter yang langka ke bit paling besar. Sehingga, ukuran file dapat diminimalisir dari ukuran awalnya. Sedangkan elias delta code ialah salah satu elias code yang dicetus oleh Peter Elias. Didalam elias delta code ditambahkan pada binary. Elias delta code juga dipakai dalam melakukan kode pada bilangan bulati positif.

Penelitian terdahulu yang dilakukan oleh Jamiatul Sisca dengan judul penelitian "Penerapan Algoritma Elias Delta Code Untuk Kompresi File Video Pada Aplikasi Video Downloader" mendapatkan kesimpulan yaitu algoritma elias delta code dapat diterapkan untuk mengkompres ukuran file video sehingga ukuran file video tergolong kecil pada saat kompresi telah dilakukan[3]. Penelitian lainnya yang dilakukan oleh Nurul Rizka dengan judul penelitian "Penerapan Algoritma Elias Omega Code Untuk Kompresi File Video Pada Aplikasi Rekam Layar" mendapatkan kesimpulan proses kompresi dengan memakai algoritma elias omega code berhasil melaksanakan kompresi file video hasil rekam layar dengan ekstensi mp4 hingga proses kompresi berjalan sesuai dengan teknik kompresi[4]. Nadia Fariza Rizky juga melakukan penelitian dengan judul penelitian "Penerapan Algoritma Elias Delta Codes Dalam Kompresi File Teks" menyimpulkan bahwa proses pengompresian file teks telah terbukti bahwa proses penerapan tersebut file yang ukurannya besar dapat diperkecil dengan menerapkan algoritma *elias omega*

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 110-119
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

code[5]. Penelitian lain yang dilakukan oleh Michael Simangunsong dengan judul penelitian "Perbandingan Algoritma Elias Delta Code dan Unary Coding Dalam Kompresi Citra Forensik" menyimpulkan bahwa algoritma elias delta code dan unary coding adalah algoritma yang dapat mengkompresi citra forensik dan format citra yang paling baik untuk mengkompresi citra forensik ialah format bitmap .bmp yang hasilnya jauh lebih baik dari ekstensi lainnya[6]. Ibrahim Hasan juga melakukan penelitian pada algoritma *elias omega code* dengan judul penelitian "Analisa Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital" menyimpulkan bahwa perancangan aplikasi kompresi berhasil menerapkan algoritma elias omega code dan fibonacci code untuk kompresi file digital dengan mendapatkan ukuran hasil kompresi file digital rata-rata sebesar <50%[7].

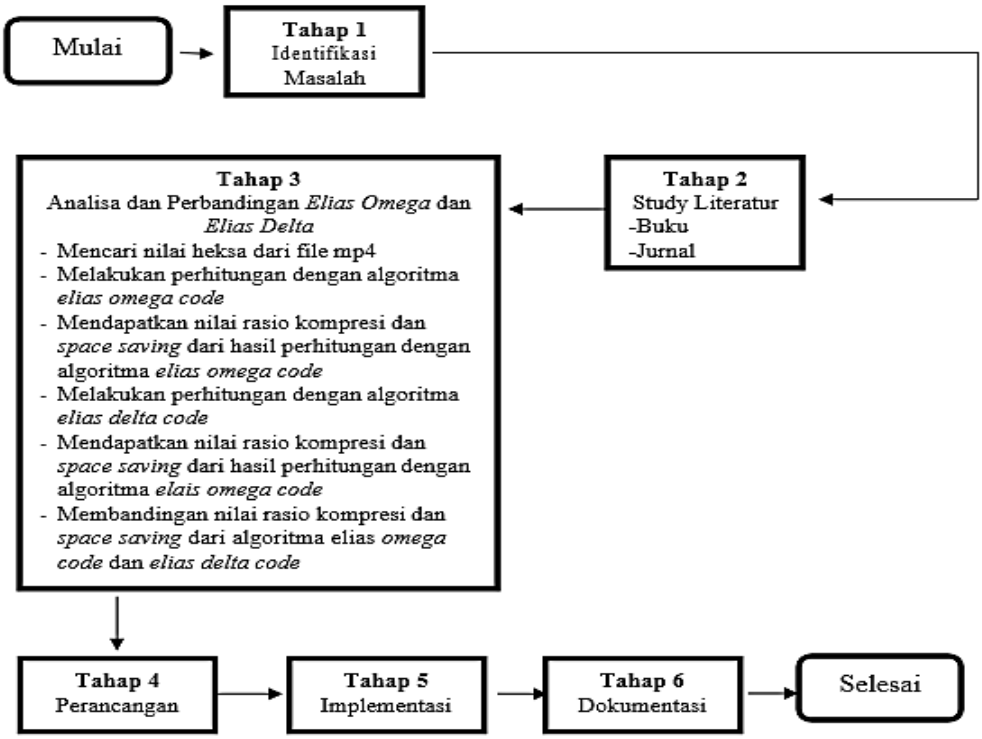
Penelitian lainnya juga dilakukan oleh Nur Aisyah dengan judul penelitian "Penerapan Algoritma Elias Omega Code Pada Kompresi File Audio Aplikasi Murottal Muzammil Hasbalah" menyimpulkan bahwa algoritma elias omega code dapat dipakai dalam proses kompresi karakter pada file audio hingga ukuran file berubah ke lebih kecil[8].

Jarangnya digunakan perbandingan algortima pada penelitian terdahulu maka sangat perlu dilakukan perbandingan algoritma kompresi agar dapat mengetahui algoritma yang lebih akurat dalam melakukan kompresi. Pada penelitian kali dilakukan perbandingan algoritma untuk mengetahui algoritma manakah yang akurat dalam proses komperesi file berfomat mp4. Algoritma yang akan dibandingkan ialah algoritma elias omega code dan elias delta code. Kedua algoritma tersebut mereka algoritma untuk melakukan kompresi. Dengan kriteria perbandingan yaitu rasio kompresi serta space saving dari algoritma elias omega code dan elias delta code.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pada metodologi penelitian dijabarkan langkah-langkah yang dilakukan pada penelitian. Metodologi penelitian terdiri atas beberapa langkah yang memiliki keterkaitan sistematis. Tahapan ini diperlukan dalam mempermudah melakukan penelitian. Sebelum membuat kerangka penelitian, penulis terlebih dahulu menganalisa topik yang akan diteliti.



Gambar 1. Tahapan Penelitian

2.2 Kompresi

Kompresi ialah cara memperkecil ataupun memadatkan file yang memiliki ukuran besar menjadi lebih kecil serta berkurangnya kebutuhan ruang penyimpanan. Proses kompresi merupakan tahapan yang mengarah pada minimasi jumlah bit dalam representasi digital seperti gambar, audio serta vidio, sehingga data memiliki ukuran data menjadi lebih kecil namun kualitas informasi tetap terjaga. Kompresi file video adalah proses pengecilan jumlah tiap bit yang merepresentasikan aplikasi hadist dengan data yang menjadi lebih kecil. Kompresi data juga dapat diartikan sebagai teknik memampatkan data sehinnga diperoleh ukuran data yang kecil dari ukuran awalnya dan menjadi efektif dalam penyimpanan atau waktu pertukaran data singkat[9][10].

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 110-119
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

2.3 Algoritma Elias Omega Code

Algoritma *elias omega code* ialah algoritma kompresi yang dipatenkan Peter Elias tahun 1975. Peter Elias memberi penjelasan tiga kode awalan (prefix) yang memiliki manfaat. Ide utama kode ialah agar awalan integral dikodekan meggunakan representasi pengkodean pada urutan terbesarnya. Agar dapat ditentukan panjang M diselesaikan dengan langkah berbeda dari berbagai kode *Elias*. Algoritma ini memakai dirinya dengan cara rekursif agar mengkodekan awalan (prefix) M oleh sebab itu kadang algoritma ini disebut dengan kode *Elias Rekursif*[11][4]. Kode engkoding bilangan bulat positif dilakukan dengan cara rekursif dengan langkah berikut [12][7]:

1. Inisialisasi kode dengan = 0
2. Jika n=1, yang dilakukan stop (proses berhenti) jika tidak, maka perlu penambahan representasi biner pada n ke kode yang ada. Asumsikan jika kita tengah melakukan penambahan bit sejumlah L (panjang digit bit).
3. Ulangi tahapan sampai representasi biner dari L – 1 untuk memakai n.
   Dimisal jika n=17 dikodekan dengan cara (1) diberikan angka 0, (2) tambahkan representasi biner dari 17, yaitu 10001 (terdiri dari 5 bit), (3)tambhakan dengan nilai 3 bit dari 5-1 = 100_2, dan (4) tambahkan nilai 2 bit dari 3-1 = 10_2. Maka hasilnya adalah 10|100|10001|0.

Kemudian pada decoding perlu dilakukan dengan tahapan-tahapan antara lain:

1. Inisialisasikan nilai n ialah 1
2. Baca bit selanjtunya, saat bernilai 0 maka berhenti (proses berhenti). Apabila tidak, baca bit untuk n, dan ulangi tahapan ini.
3. Pembacaan bit dilakukan berturut, dimulai hanya 0 atau diawali dengan 1 yang diikuti n digit bit. Jika grup adalah 0, maka nilai integernya berupa n yaitu 1. Dan jika grup dimulai dari 1, maka n menjadi nilai grup yang direpresentasikan ke dalam biner.

2.4 Algoritma Elias Delta Code

Elias delta code merupakan satu dari tiga *elias code* yang dipelopori oleh Peter Elias. *Elias delta code* ditambahkan pada binary (ß). *Elias delta code* juga digunakan untuk melakukan pengkodean pada bilangan bulat positif[6][13]. Adapun aturan untuk mengkodekan sebuah bilangan dengan menggunakan *elias delta code* adalah sebagai berikut[14][15][16]:

1. Tuliskan n dalam bilangan biner (binary). Bit yang paling kiri (paling signifikan) akan menjadi 1.
2. 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.
3. Kurangi 1 dari perhitungan langkah kedua dan tambahkan jumlah 0 ke kode. Ketika langkah-langkah ini diterapkan pada integer ke-17, hasilnya adalah: 17 = 10001_2 (lima bit). Hapus angka 1 yang paling kiri dan tambahkan 5 =101_2 sehingga hasilnya 101|0001. Tiga bit sudah ditambahkan , kemudian tambahkan 2 nol untuk mendapatkan kode delta 00|101|0001.
4. Ketika langkah-langkah ini diterapkan pada n=17, hasilnya adalah : 17 = 2N + L = 24 + 1. Kode gamma dari N + 1 = 5 adalah 00101, dan tambahkan L = 0001 sehingga hasilnya adalah 00101|0001.

3. HASIL DAN PEMBAHASAN

Tahap analisa ialah langkah yang begitu banyak pengaruhnya serta penentu terhadap tahap berikutnya. Analisa pada sistem ialah langkah yang amat penting agar dapat diketahui proses yang berlangsung pada aplikasi yang akan dibangun melakukan perbandingan kompresi, dalam hal ini peneliti menggunakan algoritma elias omega code dan elias delta code untuk kompresi file MP4. Perbandingan algoritma digunakan agar dapat diektaui algoritma yang lebih akurat ketika memperkecil ukuran file mp4. Pengompresan file mp4 berguna agar berkurangnya tempat penyimpanan dan proses suatu file yang dikirim menjadi lebih cepat. Dalam penelitian ini hal yang paling utama ialah membandingan besarnya rasio kompresi dan space saving dari algoritma elias omega code dan elias delta code. File video yang berekstensi mp4 mempunyai ukuran yang cukup besar, semakin lama waktu pada file video maka ruang penyimpanan yang dibutuhkan semakin besar, dan proses mengirim file tergolong cukup lama. Saat proses kompresi file mp4 dilakukan sebelumnya harus dilakukan analisa terhadap file mp4.

3.1 Penerapan Algoritma Elias Omega Code

Dengan melakukan kompresi *file* mp4, file yang memiliki ukuran lebih besar akan dikompresi ukuran lebih kecil serta ruang penyimpanan menjadi lebih hemat dan diketahui algoritma yang lebih akurat dalam melakukan kompresi *file* mp4 dengan algoritma *elias omega code* dan *elias delta code*. Sebelum *file* mp4 dikompresi, terlebih dahulu dilakukan pembacaan pada *file* mp4 untuk mendapatkan nilai hexadesimal menggunakan aplikasi HxD. Berikut adalah contoh *file* mp4 yang dikompresi dan dekompresi.

1. Memasukkan File

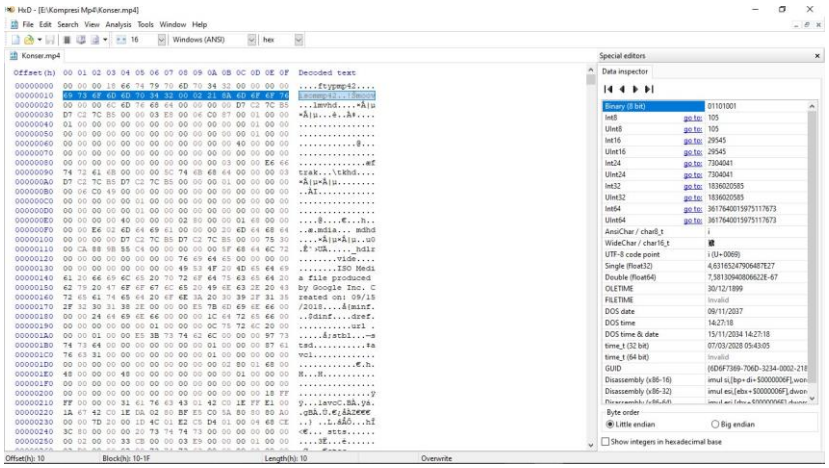
Tabel 1. Sampel File yang akan dikompres

Nama File	Konser

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 110-119
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Ekstensi File	Mp4
Ukuran	33.8 Mb

Gambar 2. File Mp4 yang akan dikompresi



Gambar 3. Isi File Mp4 dari Hasil HxD

Berdasarkan ambar di atas didapatkan nilai hexadesimal *file* mp4 sampel. Dalam hitungan manual, maka yang diambil untuk sampel nilai berjumlah 16 karakter nilai *hexadesimal file* mp4 sampel. Nilai *hexadesimal* diambil dari kiri ke kanan.

2. Melakukan Pembacaan Isi *File*
   Adapun nilai hexadesimal dari file mp4 tersebut ialah 69, 73, 6F, 6D, 6D, 70, 34, 32, 00, 02, 21, 8A, 6F, 6F, 6F, 76. Nilai data ini dimasukkan ke dalam tabel dalam membaca frekuensi. Dalam membaca frekuensi dilakukan dengan perhitungan jumlah nilai sama disetiap nilai yang ada. Pembacaan frekuensi dapat di lihat pada tabel di bawah ini:

Tabel 2. Nilai *File*

Nilai	Frekuensi
69	1
73	1
6F	3
6D	3
70	1
34	1
32	1
00	1
02	1
21	1
8A	1
76	1
Total Nilai	16

3. Karakter yang mempuntai frekuensi paling besar (banyak nilai yang sama) menuju frekuensi terkecil diurutkan. Urutan nilai dilihat ditabel di bawah ini :

Tabel 3. Nilai *Bit* Mp4 Sample

Nilai		Bit	Frek	Bit x Frek
Hexa	Biner			
6F	01101111	8	3	24
6D	01101101	8	3	24
69	01101001	8	1	8

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 110-119
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Nilai		Bit	Frek	Bit x Frek
Hexa	Biner			
73	00111011	8	1	8
70	01110000	8	1	8
34	00110100	8	1	8
32	00110010	8	1	8
00	00000000	8	1	8
02	00000010	8	1	8
21	00100001	8	1	8
8A	10001010	8	1	8
76	01110110	8	1	8
Total Bit				128

Dari tabel di atas, satu nilai hexadesimal (karakter) bernilai 8 bit bilangan biner. Hingga 16 bilangan *hexadesimal* memiliki nilai biner sejumlah 128 bit. Dalam pengubahan satuan kedalam byte maka jumlah semua bit dibagi dengan 8. Dan dihasilkan 182/8 = 16.

4. Membentuk Tabel *Elias Omega Code*
   Aturan pada pembentukan kode bilangan menggunakan *elias omega code* dapat di lihat di bab sebelumnya. Kode *elias omega code* dapat di lihat pada tabel di bawah ini.

Tabel 4. Kode *Elias Omega*

N	Kode Elias Omega
1	0
2	10 0
3	11 0
4	10 100 0
5	10 101 0
6	10 110 0
7	10 111 0
8	11 1000 0
9	11 1001 0
10	11 1010 0
11	11 1011 0
12	11 1100 0
13	11 1101 0
14	11 1110 0
15	11 1111 0
16	10 100 10000 0
17	10 100 10001 0
18	10 100 10010 0

Dari tabel *elias omega* di atas menunjukkan kode hingga nilai karakter ke 18. Bagaimana proses yang dilakukan dalam mendapatkan kode *elias omega*, apabila jumlah karakter yang dihasilkan lebih dari karakter 18 karakter. Misal pada nilai = 19, nilai desimal 19 dikodekan dengan (1) diberikan angka 0, (2) dilakukan penambahan dengan representasi biner dari 19, yaitu $10011_{(2)}$ (terdiri dari 5 bit), (3) dilakukan penambahan dengan nilai 3-bit dari 5-1 =100_2 serta(4) dilakukan penambahan nilai 2-bit dari 3-1 = 10_2. Hasilnya ialah 10|100|10011|0. Proses selanjunya ialah dilakukan kompresi nilai sample menggunakan kode *elias omega* yang didapat dari tabel 3.4 diatas. Proses kompresi *file* mp4sample dapat di lihat pada tabel berikut :

Tabel 5. Kompresi Nilai *File* mp4 *Sample* Dengan *Elias Omega Code*

N	Nilai Hexa	Kode Elias Omega	Bit	Frek	Bit x Frek
1	6F	0	1	3	3
2	6D	10 0	3	3	9
3	69	11 0	3	1	3
4	73	10 100 0	6	1	6
5	70	10 101 0	6	1	6
6	34	10 110 0	6	1	6
7	32	10 111 0	6	1	6
8	00	11 1000 0	7	1	7
9	02	11 1001 0	7	1	7
10	21	11 1010 0	7	1	7
11	8A	11 1011 0	7	1	7

N	Nilai Hexa	Kode Elias Omega	Bit	Frek	Bit x Frek
12	76	11 1100 0	7	1	7
		Total Bit			74

Pada perhitungan tabel di atas setelah dilakukan proses kompresi menggunakan *elias omega code* ialah 74 bit. Agar dirubah dalam satuan byte maka dibagi 8 yaitu 74/8 = 9,25.

5. Melakukan hasil string bit elias omega code
   Sebelum dilakukan hasil string bit elias omega code menjadi nilai file, maka dilakukan pemeriksaan pada panjang string bit. Melakukan bentuk nilai bit baru hasil kompresi pada susunan nilai heksadesimal sebelum dikompresi yaitu 69, 73, 6F, 6D, 6D, 70, 34, 32, 00, 02, 21, 8A, 6D, 6F, 6F, 76 (tanpa tanda koma serta spasi) dijadikan dalam nilai bit biner :
   11010100001001001010101011001011101110000111001011101001110110100001111000
   Lalu sebelum didapatkan seluruh hasil akhir kompresi dilakukan penambahan *string bit* itu sendiri yaitu *padding* serta *flag bit.* Jika sisa bagi panjang string bit terhadap 8 adalah 0 maka tambahakan 0000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi panjang *string* bit terhadap 8 adalah n(1,2,3,4,5,6,7) makan tambahkan 0 sebanyak 7-n+"1" diakhir *string bit*. Nyatakan dengan L, lalu tambahkan 0 bilangan biner dari 9 –n, nyatakan dengan bit akhir. karena jumlah string bit 74 tidak habis dibagi 8 dan sisanya 2 bit, nyatakan sisa bagi tersebut dengan nilai *n*. Maka tambahkan 0 sebanyak 7 – n + "1" akhir string bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari 9 – n. Nyatakan dengan bit akhir.
   1101010000100100101010101100101110111000011100101110100111011010000111100**00000100000111**
   Perhitungan Penambahan bit:

   7 – n + "1"

   7 – 2 + "1" = **000001**

   Bit Akhir 9 – n

   Bit Akhir = 9 – 2 = 7 = **00000111**

   Total panjang seluruh bit setelah dilakukan penambahan bit ialah 74+6+8 =88. Kemudian lakukan pemisahan bit beberapa kelompok. Setiap kelompok terdiri atas 8 bit berikut:
   11010100 00100100 10101010 11001011 10111000 01110010 11101001 11011010 00011110 00**000001 00000111**

   Dengan pembagian kelompok nilai biner, dihasilkan 11 kelompok nilai biner yang baru (11 byte) yang telah dikompresi berserta nilai biner yang telah ditambahkan bit. Lalu pembagian dilakukan, maka nilai yang telah dibagi diubah dalam suatu karakter dengan terlebih dahulu mencari nilai *hexadecimal* dan *string* bit tersebut menggunakan kode ASCII agar dapat diketahui nilai yang telah dikompresi. Adapun nilai yang telah dikompresi dapat dilihat pada tabel dibawah ini.

Tabel 6. Nilai *Hexadecimal* Terkompresi

Urutan Nilai Terkompresi	Nilai *Hexadecimal* Terkompresi
1	D4
2	24
3	2A
4	CB
5	B8
6	72
7	E9
8	DA
9	1E
10	1
11	7

Persentase ukuran data yang telah terkompresi serta dapat dilihat dari hasil yang dibandingkan antar ukuran data setelah kompresi dilakukan dan ukuran data sebelum kompresi dilakukan.
Ukuran data sebelum dikompresi = 128/8 = 16 byte Ukuran data sesudah dikompresi = 88/8 = 11 byte Dari data yang ada maka dihitung kinerja kompresi :

1. *Compression Ratio* (Cr)

   $$Cr = \frac{Ukuran\ Data\ Sesudah\ DIkompresi}{Ukuran\ Data\ Sebelum\ DIkimpresi} * 100\%$$

   $$Cr = \frac{11}{16} * 100\% = 68{,}75\%$$

Bulletin of Information System Research (BIOS)
Volume 1, No 3, August 2023 Page: 110-119
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

2. *Space Saving*

   SS = ukuran data sebelum dikompresi – ukuran data setelah dikompresi SS = 128- 88 = 40

3.2 Penerapan Algoritma *elias delta code*

1. Mengurutkan karakter yang dengan frekuensi terbesar (jumlah nilai yang memiliki kesamaan) menuju frekuensi terkecil. Urutan nilai dapat dilihat dari tabel di bawah:

Tabel 7. Nilai *Bit* Mp4 Sample

Nilai		Bit	Frek	Bit x Frek
Hexa	Biner			
6F	01101111	8	3	24
6D	01101101	8	3	24
69	01101001	8	1	8
73	00111011	8	1	8
70	01110000	8	1	8
34	00110100	8	1	8
32	00110010	8	1	8
00	00000000	8	1	8
02	00000010	8	1	8
21	00100001	8	1	8
8A	10001010	8	1	8
76	01110110	8	1	8
				128

Dari tabel diatas, satu nilai hexadesimal (karakter) memiliki nilai 8 bit bilangan biner. Kemudian 16 bilangan *hexadesimal* memiliki nilai biner dengan jumlah 128 bit. Untuk diubah satuan dijadikan dalam bentuk byte maka jumlah semua bit dibagi 8. Maka didapatkan hasil 182/8 = 16.

2. Membentuk Tabel *Elias Delta Code*
   Aturan pada pembentukan kode bilangan menggunakan *elias delta code* dapat dilihat di bab sebelumnya. Yang dijadikan kode *elias delta code* dapat dilihat pada tabel di bawah.

Tabel 8. Kode *Elias Delta*

N	Kode Elias Delta
1	1
2	0100
3	0101
4	01100
5	01101
6	01110
7	01111
8	00100000
9	00100001
10	00100010
11	00100011
12	00100100
13	00100101

Proses selanjunya ialah dilakukan kompresi nilai sample dengan kode *elias delta code* yang didapat dari tabel diatas. Adapun tahapan kompresi *file* mp4 sample dapat di lihat pada tabel berikut :

Tabel 9. Kompresi Nilai *File* mp4 *Sample* Dengan *Elias Delta Code*

N	Nilai Hexa	Kode Elias Omega	Bit	Frek	Bit x Frek
1	6F	1	1	3	3
2	6D	0100	4	3	12
3	69	0101	4	1	4
4	73	01100	5	1	5
5	70	01101	5	1	5
6	34	01110	5	1	5
7	32	01111	5	1	5
8	00	00100000	8	1	8

N	Nilai Hexa	Kode Elias Omega	Bit	Frek	Bit x Frek
9	02	00100001	8	1	8
10	21	00100010	8	1	8
11	8A	00100011	8	1	8
12	76	00100100	8	1	8
		Total Bit			79

Setelah perhitungan di atas dikompresi dengan memakai *elias delta code* ialah 79bit. Untuk dilakukan pengubahan jadi satuan byte harus dibagi 8 yaitu 79/8 = 9,87.

3. Melakukan hasil string bit *elias delta code*
Sebelum dilakukan hasil string bit *elias delta code* menjadi nilai *file*, terlebih pemeriksaan dilakukan pada panjang *string* bit. Pembentukan nilai bit baru hasil kompresi dari susunan nilai heksadesimal sebelum kompresi yaitu 69, 73, 6F, 6D, 6D, 70, 34, 32, 00, 02, 21, 8A, 6D, 6F, 6F, 76 (tanpa tanda koma dan spasi) menjadi nilai bit biner :
"0101011001010001000110101110011110010000000100001001000100010001101001100100100"

Lalu sebelum didapatkan hasil seluruh akhir kompresi dilakukan penambahan *string bit* itu sendiri yang biasa disebut *padding* serta *flag bit.* Jika sisa bagi string bit terhadap 8 ialah 0 maka tambahakan 0000001. Dinyatakan dengan bit akhir. Sedangkan apabila sisa bagi panjang *string* bit terhadap 8 ialah sn(1,2,3,4,5,6,7) makan tambahkan 0 sebanyak 7-n+"1" diakhir *string bit*. Nyatakan dengan L, lalu tambahkan 0 bilangan biner dari 9 –n, nyatakan dengan bit akhir. karena jumlah string bit 74 tidak habis dibagi 8 dan sisanya 2 bit, nyatakan sisa bagi tersebut dengan nilai *n.* Maka tambahkan 0 sebanyak 7 – n + "1" akhir string bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari 9 – n. Nyatakan dengan bit akhir.
01010110010100010001101011100111100100000001000010010001000100011010011001001001000000010
Perhitungan Penambahan bit

7 – n + "1"

7 – 7 + "1" = **1**

Bit Akhir 9 – n

Bit Akhir = 9 – 7 = 2 = **00000010**

Total seluruh panjang bit yang telah dilakukan penambahan bit yaitu 74+6+8 =88. Lalu dilakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti berikut.
01010110 01010001 00011010 11100111 10010000 00010000 10010001 00010001 10100110 01001001 **00000010**

Berdasarkan pada pembagian kelompok nilai biner, maka hasilnya berjumlah 11 kelompok nilai biner yang baru (11 byte) yng telah dikompres berserta nilai biner yang telah ditambahkan bit. Setelah proses pembagian, maka nilai yang telah dibagi diubah kedalam suatu karakter dengan lebih dulu dicari nilai *hexadecimal* serta *string* bit tersebut memakai kode ASCII agar dketahui nilai yang telah terkompresi. Adapun nilai yang sudah terkompresi dapat dilihat pada tabel di bawah ini.

Tabel 10. Nilai *Hexadecimal* Terkompresi

Urutan Nilai Terkompresi	Nilai *Hexadecimal* Terkompresi
1	56
2	51
3	1A
4	E7
5	90
6	10
7	91
8	11
9	A6
10	49
11	2

Persentase ukuran data yang telah dikompres dapat dilihat dari hasil perbandingan antar ukuran data setelah dikompresi dengan ukuran data sebelum dikompresi.
Ukuran data sebelum dikompresi = 128/8 = 16 byte Ukuran data sesudah dikompresi = 88/8 = 11 byte
Dari data diatas maka dapat dilakukan perhitungan kinerja kompresinya yaitu :

1. *Compression Ratio* (Cr)

$$Cr = \frac{Ukuran\ Data\ Sesudah\ DIkompresi}{Ukuran\ Data\ Sebelum\ DIkimpresi} * 100\%$$

$$Cr = \frac{11}{16} * 100\% = 68{,}75\%$$

2. *Space Saving*

SS = ukuran data sebelum dikompresi – ukuran data setelah dikompresi SS = 128- 88 = 80

3.3 Perbandingan Algoritma *Elias Omega Code* dan *Elias Delta Code*

Perbandingan algortima *elias omega code* dan *elias delta code* ialah membandingan rasio kompresi dan *space saving* dari kedua algoritma. Dimana setelah dilakukan perhitungan dengan algoritma *elias omega code* dan *elias delta code* maka hasil dari rasio kompresi dan *space saving* didapatkan. Hasil dari rasio kompresi algoritma *elias omega code* dan *elias delta code* memiliki kesamaan yaitu besar rasio kompresi 68,75% serta nilai ijuga memiliki kesamaan yaitu sebesar 40. Maka dari perhitungan tersebut dapat dikatakan bahwa hasil kompresi *file* mp4 dari algoritma idan *elias delta code* memiliki kesamaan.

Tabel 11. Perbadingan Hasil

Elias Delta Code	Elias Omega Code
Rasio Kompresi 68,75%	Rasio Kompresi 68,75%
Space Saving 40	Space Saving 40

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan, maka hasil akhir pada penelitian ini dapat diambil kesimpulan. Yang menjadi kesimpulan penelitian sebagai berikut Kompresi file video dapat dilakukan dengan mengubah ke dalam bentuk heksadesimal agar dapat dilakukan perhitungannya. Bilangan heksadesimal yang didapat diolah menggunakan algoritma elias omega code dan elias delta code sehingga akan menghasilkan nilai bilangan heksadesimal baru yang akan merubah ukuran dari file video. Setelah dilakukan proses kompresi file video maka didapatkan rasio kompresi dan space saving dari algoritma elias omega code dan elias delta code. Nilai rasio kompresi dan space saving dari kedua algoritma yang dijadikan sebagai pembanding dari algoritma elias omega code dan elias delta code. Rasio kompresi dari elias omega code sebesar 68,75% dan rasio kompresi dari algoritma elias delta code sebesar 68,75%. Dapat dikatakan bahwa kompresi file video dengan algoritma elias omega code dan elias delta code memiliki nilai rasio yang sama.

REFERENCES

[1] R. Fadillah, "Penerapan Algoritma Fibonacci Codes Dalam Aplikasi Kompresi File Citra Digital," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 225–233, 2023.
[2] R. Fadillah, A. S. Idris, D. M. lumban Gaol, G. Lubis, R. Meisisri, and M. Syahrizal, "Implementasi Algoritma Fast Encryption Algorithm (FEAL) Dan Algoritma Fibonacci Mengamankan File Teks," in *Prosiding Seminar Nasional Sosial, Humaniora, dan Teknologi*, 2022, pp. 295–300.
[3] J. Sisca, "Penerapan Algoritma Elias Delta Code Untuk Kompresi File Video Pada Aplikasi Video Downloader," *Resolusi Rekayasa Tek. Inform. dan Inf.*, vol. 1, no. 4, pp. 254–264, 2021.
[4] N. Rizka, S. D. Nasution, and K. Ulfa, "Penerapan Algoritma Elias Omega Code Untuk Kompresi File Video Pada Aplikasi Rekam Layar," vol. 9, no. 4, pp. 257–265, 2021.
[5] N. F. Rizky, S. D. Nasution, and F. Fadlina, "Penerapan Algoritma Elias Delta Codes Dalam Kompresi File Teks," *Build. Informatics, Technol. Sci.*, vol. 2, no. 2, pp. 109–114, 2020.
[6] M. Simangunsong, "Perbandingan Algoritma Elias Delta Code Dan Unary Coding Dalam Kompresi Citra Forensik," *Resolusi Rekayasa Tek. Inform. dan Inf.*, vol. 1, no. 1, pp. 18–26, 2020.
[7] I. Hasan, N. Irsa Syahputri, and U. Harapan Medan, "Analisis Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital," *Algoritm. J. Ilmu Komput. dan Inform.*, vol. 6341, no. April, p. 1, 2021.
[8] N. Aisyah and S. Aripin, "Penerapan Algoritma Elias Omega Code Pada Kompresi File Audio Aplikasi Murottal Muzzamil Hasbalah," *Pelita Inform.*, vol. 9, pp. 113–119, 2020.
[9] A. Bahrudin, P. Permata, and J. Jupriyadi, "Optimasi Arsip Penyimpanan Dokumen Foto Menggunakan Algoritma Kompresi Deflate (Studi Kasus: Studio Muezzart)," *J. Ilm. Infrastruktur Teknol. Inf.*, vol. 1, no. 2, pp. 14–18, 2020.
[10] W. Pramusinto, N. Wizaksono, and A. Saputro, "Aplikasi Pengamanan File Berbasis Web Dengan Metode Kriptografi Aes 192, Rc4 Dan Metode Kompresi Huffman," *Bit (Fakultas Teknol. Inf. Univ. Budi Luhur)*, vol. 16, no. 2, pp. 47–53, 2020.
[11] N. Aisyah and S. Aripin, "Penerapan Algoritma Elias Omega Code Pada Kompresi File Audio Aplikasi Murottal Muzzamil Hasbalah," *Pelita Inform. Inf. dan Inform.*, vol. 9, no. 2, pp. 113–119, 2020.
[12] F. R. Kesuma, "Implementasi Kombinasi Algoritma Elias Omega Codes Dan Golomb Rice Dengan Teknik High Compress Pada File Teks," *Inf. dan Teknol. Ilm.*, vol. 8, no. 2, pp. 76–81, 2021.
[13] K. Y. Sarumaha, M. Syahrizal, and E. R. Siagian, "Analisis Perbandingan Algoritma Elias Delta Code Dengan Algoritma Prefix Code Dalam Mengkompresi Data Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 460–469, 2023.
[14] 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. Syst. Res. Vol 1 No 3 April 2020*, vol. 1, 2020.
[15] E. Hariska, E. Yuliani, and S. D. Nasution, "Performance Comparison Analysis of the Elias Delta Code Algorithm with the

Even Rodeh Code Algorithm for Compressing Image Files," *IJICS (International J. Informatics Comput. Sci.*, vol. 5, no. 1, pp. 29–36, 2021.

[16] M. A. Budiman, D. Rachmawati, and A. Deviana, "A Comparative Performance Study of (s, c)-Dense Codes and Elias Delta Codes in Data Compression," in *2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 2021, pp. 121–124.