

Analisa Perbandingan Proses Algoritma Elias Gamma Code dengan algoritma Elias Delta Code Dalam Kompresi File Audio Video Interleave

Pengalaman Ndruru^{1*} Eferoni Ndruru²

¹ Fakultas Ilmu Komputer, Program Stud Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

² Fakultas Ilmu Komputer, Program Stud Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: pengalamanndruru97@gmail.com, ronindruru@gmail.com

Abstrak– Kompresi data ialah suatu cara dalam ilmu komputer untuk pengecilan ukuran data ataupun memampatkan data. Data yang akan dimampatkan berukuran lebih kecil dari data sebelumnya dengan tujuan agar terjadi penghematan penyimpanan. Jika kompresi data dilakukan, maka ruang penyimpanan yang dibutuhkan kecil. Selain lebih efisien, kompresi juga dapat menyebabkan lebih cepatnya waktu pertukaran data. Seiring berkembangnya teknologi pada zaman sekarang ini, data memiliki peranan yang sangat penting, besar jumlah data yang disimpan pada memori atau hardisk akan terjadi penambahan besar kapasitas terpakai media penyimpanan.

Terdapat dua teknik yang dapat dilakukan dalam melakukan kompresi data yaitu *lossy Compression* dan *lossless compression*. *Lossy Compression* adalah kompresi file data dimana hasil dekompresi dari data yang terkompresi tidak sama dengan data aslinya karena ada informasi yang hilang, tetapi masih dapat ditolerir oleh persepsi mata. Kompresi data *lossy* akan menjadi efektif apabila diaplikasikan pada gambar, film, dan suara digital. Sedangkan, *Lossless compression* merupakan kompresi data dimana hasil dekompresi dari data yang terkompresi sama dengan data aslinya dan tidak ada informasi yang hilang. Salah satu algoritma kompresi yang dapat digunakan adalah Algoritma *Elias Delta Code* yang merupakan sebuah algoritma kompresi yang dibuat oleh *Peter Elias*. Kode *Elias Gamma Code* menambahkan panjang kode dalam unary (α). Kode berikutnya, δ (delta), kemudian ditambahkan pada kode biner (β). *Elias Delta Code* untuk blangan bulat positif, sedikit lebih kompleks untuk dibangun. Algoritma ini juga dapat mengompresikan suatu file yang berukuran besar hingga memperkecil ukuran file tersebut.

Kata Kunci: Compress, Comparison, Analysis,

Abstract– Data compression is a method in computer science for reducing data size or compressing data. The data to be compressed is smaller in size than the previous data with the aim of saving storage. If data compression is carried out, the storage space required is small. Apart from being more efficient, compression can also result in faster data measurement times. As technology develops nowadays, data has a very important role, the large amount of data stored on memory or hard disk will result in a large increase in the used capacity of the storage media.

There are two techniques that can be used to compress data, namely *lossy compression* and *lossless compression*. *Lossy Compression* is data file compression where the decompression results of the compressed data are not the same as the original data because there is information lost, but it can still be tolerated by eye perception. *Lossy* data compression will be effective when applied to images, films and digital sound. Meanwhile, *Lossless compression* is data compression where the decompression results of the compressed data are the same as the original data and no information is lost. One of the compression algorithms that can be used is the *Elias Delta Code Algorithm* which is a compression algorithm created by *Peter Elias*. The *Elias Gamma Code* code adds the length of the code in unary (α). The next code, δ (delta), is then added to the binary code (β). *Elias Delta Code* for positive integers, a bit more complex to construct. This algorithm can also compress large files to reduce the size of the file.

Keywords: Compress, Comparison, Analysis,

1. PENDAHULUAN

Kompresi data ialah suatu cara dalam ilmu komputer untuk pengecilan ukuran data ataupun memampatkan data. Data yang akan dimampatkan berukuran lebih kecil dari data sebelumnya dengan tujuan agar terjadi penghematan penyimpanan. Jika kompresi data dilakukan, maka ruang penyimpanan yang dibutuhkan kecil. Selain lebih efisien, kompresi juga dapat menyebabkan lebih cepatnya waktu pertukaran data. Seiring berkembangnya teknologi pada zaman sekarang ini, data memiliki peranan yang sangat penting, besar jumlah data yang disimpan pada memori atau hardisk akan terjadi penambahan besar kapasitas terpakai media penyimpanan.

Terdapat dua teknik yang dapat dilakukan dalam melakukan kompresi data yaitu *lossy Compression* dan *lossless compression*. *Lossy Compression* adalah kompresi file data dimana hasil dekompresi dari data yang terkompresi tidak sama dengan data aslinya karena ada informasi yang hilang, tetapi masih dapat ditolerir oleh

persepsi mata. Kompresi data lossy akan menjadi efektif apabila diaplikasikan pada gambar, film, dan suara digital. Sedangkan, Lossless compression merupakan kompresi data dimana hasil dekompresi dari data yang terkompresi sama dengan data aslinya dan tidak ada informasi yang hilang. Salah satu algoritma kompresi yang dapat digunakan adalah Algoritma *Elias Delta Code* yang merupakan sebuah algoritma kompresi yang dibuat oleh *Peter Elias*. Kode *Elias Gamma Code* menambahkan panjang kode dalam unary (α). Kode berikutnya, δ (delta), kemudian ditambahkan pada kode biner (β). *Elias Delta Code* untuk blangan bulat positif, sedikit lebih kompleks untuk dibangun. Algoritma ini juga dapat mengkompresikan suatu file yang berukuran besar hingga memperkecil ukuran file tersebut.

Ukuran file AVI tergolong sangat besar dan terjadi pengaruh pada ruang penyimpanan serta waktu berkirim data juga tergolong lama dengan ukuran file yang begitu besar. Oleh karenanya perlunya dilakukan kompresi dalam dimampatkannya isi file video dengan ekstensi AVI. Ketika proses kompresi dilakukan harus adanya algoritma pendukung dalam kompresi[1].

Pada penelitian kali dilakukan perbandingan algoritma untuk mengetahui algoritma manakah yang akurat dalam proses kompresi file AVI. Algoritma yang akan dibandingkan yaitu algoritma *Elias Gamma Code* dan *Elias Delta Code*. Kedua algoritma tersebut mereka algoritma untuk proses kompresi. Variabel perbandingan kedua algoritma adalah rasio kompresi dan space saving[2].

Penelitian yang dilakukan oleh Ari Pradana pada tahun 2022 dengan judul penelitian “Analisa Perbandingan *Elias Gamma Code* Dan Algoritma *Golbach Code* Pada Kompresi File Dokumen” mengatakan bahwa perbandingan hasil kedua algoritma ini untuk mengetahui kinerja dari kedua algoritma diatas, dari kinerja algoritma diatas ada beberapa parameter, yaitu Ratio Of Compression(RC), Compression Ratio(CR). file dokumen yang akan dikompresi kemudian mengkonversi file tersebut ke heksadesimal menggunakan software HxD[3]

Penelitian lainnya yang dilakukan oleh Hikka Satria pada tahun 2019 yang berjudul “Perancangan Dan Implementasi Algoritma *Elias Gamma Code* Untuk Mengkompresi Record Database Pada Aplikasi Rangkuman Pengetahuan Umum Lengkap” dari penelitian ini disimpulkan bahwa Algoritma *Elias Gamma Code* menjadi salah satu solusi yang dapat dilakukan untuk meningkatkan performa Aplikasi RPUL, Penerapan Algoritma kompresi ini dapat menurunkan ukuran database[4].

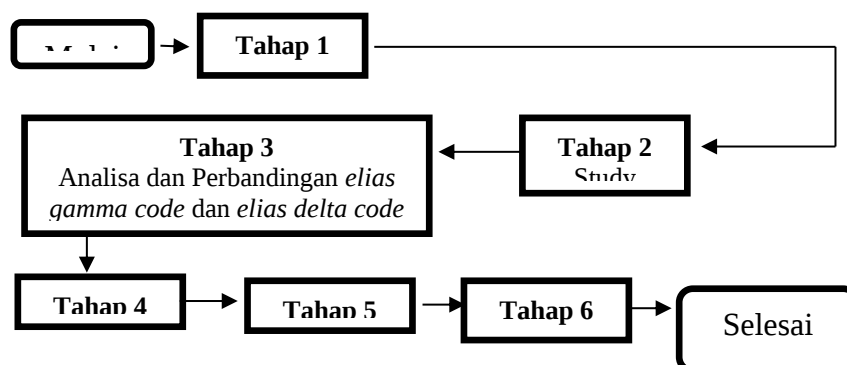
Penelitian lainnya yang dilakukan oleh Ledi Varia Simanjuntak pada tahun 2020 dengan judul penelitian “Perbandingan Algoritma *Elias Delta Code* dengan *Levenstein* Untuk Kompresi File Teks” mengatakan bahwa nilai rasio kompresi dipengaruhi oleh isi dari file yang dikompresi dan semakin banyak pengulangan karakter pada suatu file yang dikompresi maka semakin tinggi pula rasio kompresinya[5]

Penelitian lainnya juga dilakukan oleh Michael Simangunsong pada tahun 2020 dengan judul penelitian “Perbandingan Algoritma *Elias Delta Code* Dan *Unary Coding* Dalam Kompresi File Citra Forensik” mengatakan bahwa algoritma *Elias Delta Code* dan *Unary Coding* adalah algoritma yang dapat mengkompresi citra forensik[6].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Dibawah ini merupakan alur sederhana proses pengumpulan data dari penelitian yang dilakukan.

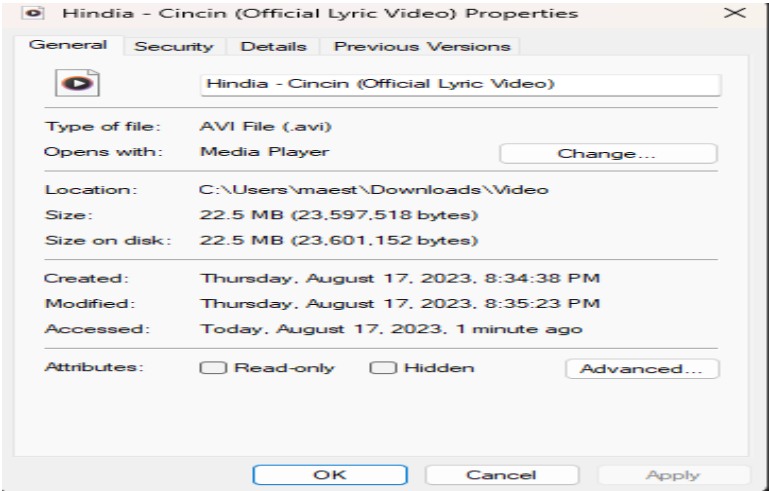


Kompresi adalah teknik memperkecil ataupun memadatkan *file* yang berukuran besar menjadi lebih kecil dan mengurangi kebutuhan ruang penyimpanan. Proses kompresi merupakan proses yang mengarah pada minimasi jumlah bit untuk representasi digital seperti gambar, audio dan video, yang menghasilkan ukuran data menjadi lebih kecil namun kualitas informasi tetap terjaga[3]. Manfaat yang akan didapatkan dari proses kompresi antara lain[4], yaitu:

1. Dengan adanya proses kompresi terhadap file AVI maka akan terjadi pemampatan terhadap data sehingga dapat menghemat penyimpanan.
2. Ruang penyimpanan memori lebih sedikit dari pada representasi file AVI yang tidak dikompresi.

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini data yang diambil berupa sampel data *file* AVI yang nantinya digunakan dalam proses kompresi perbandingan algoritma *elias gamma code* dan *elias delta code*. Dan nilai heksadesimal dari sample data dicari dengan software HxD. Sampel data yang digunakan adalah nilai heksadesimal dari file AVI. Adapun sample data yang digunakan dapat dilihat pada tabel di bawah ini.

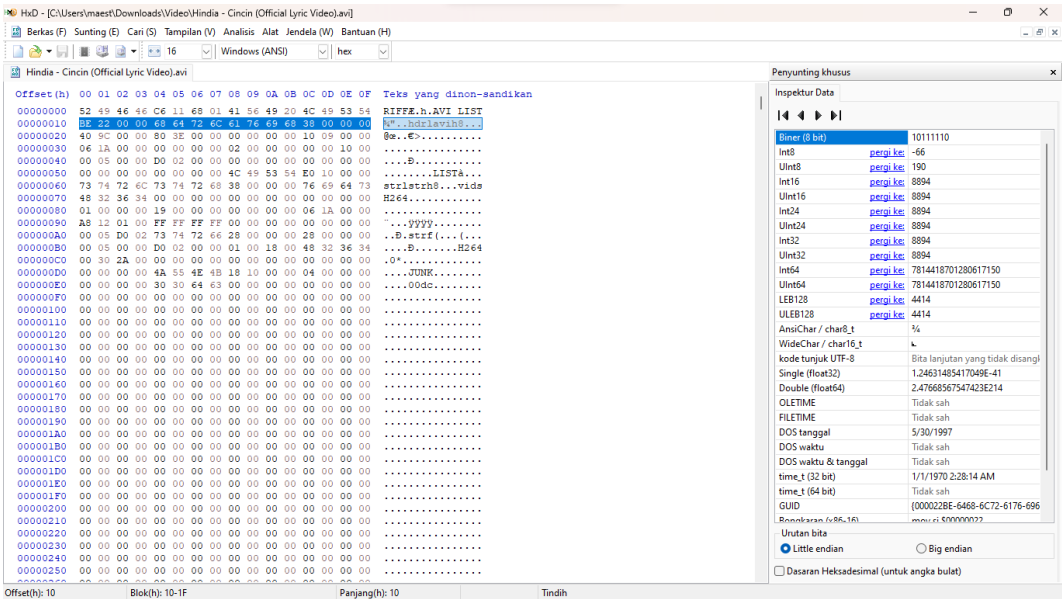


Gambar 1 yang akan dikompresi
Sebelum *file* AVI dikompres, lebih dulu dilakukan pembacaan *file* AVI agar didapatkan nilai hexadesimal menggunakan aplikasi HxD. Berikut contoh *file* AVI yang dikompres serta didekompresi. Adapun contoh dapat dilihat pada tahapan di bawah ini :

- 1. Analisa proses kompresi file AVI memakai algoritma *elias gamma code*
 - a. Memasukkan File

Tabel 1 Sampel File AVI yang akan dikompresi

Nama File	Hindia - Cincin (Official Lyric Video)
Extension File	.AVI
Size	22,5 MB



Gambar 2 Isi File AVI dari Hasil HxD

Dari gambar di atas didapat nilai hexadesimal *file* AVI sampel. Untuk keperluan perhitungan manual, maka hanya akan diambil sampel nilai sebanyak 16 karakter nilai *hexadesimal file* AVI sampel. Nilai *hexadesimal* diambil dari sisi kiri ke kanan.

b. Melakukan Pembacaan Isi *File*

Nilai hexadesimal *file* AVI ini ialah BE, 22, 00, 00, 68, 64, 72, 6C, 61, 76, 69, 68, 38, 00, 00, 00. Nilai data ini dimasukkan dalam tabel agar dilakukan pembacaan frekuensi. Frekuensi dibaca dengan perhitungan jumlah nilai yang sama disetiap nilai yang muncul. Pembacaan frekuensi dapat dilihat pada table di bawah ini:

Tabel 2 Nilai *File*

Nilai	Frekuensi
00	5
68	2
BE	1
22	1
64	1
72	1
6C	1
61	1
76	1
69	1
38	1

c. Dilakukan pengurutan karakter yang memiliki frekuensi paling besar (banyak nilai yang sama) ke frekuensi paling kecil. Urutan nilai dapat dilihat dalam tabel di bawah:

Tabel 3 Nilai *Bit File* AVI Sample

Nilai		Bit	Frek	Bit x Frek
Hexa	Biner			
00	00000000	8	5	40
68	011010001	8	2	16
BE	10111110	8	1	8
22	00100010	8	1	8
64	01100100	8	1	8
72	00000111	8	1	8
6C	01101100	8	1	8
61	01100001	8	1	8
76	01110110	8	1	8
69	01101001	8	1	8
38	00111000	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 dengan jumlah 128 bit. Dalam pengubahan satuan menjadi byte maka jumlah seluruh bit dibagi 8. Maka dihasilkan $128/8 = 16$.

d. Membentuk Tabel *Elias Gamma Code*

Aturan pada pembentukan kode bilangan dengan memakai *elias gamma code* dapat dilihat dalam bab sebelumnya. Kode *elias gamma code* dapat dilihat pada tabel di bawah.

Tabel 4 Kode *Elias Gamma Code*

N	Kode Elias Gamma
1	1
2	010
3	011
4	00100
5	00101
6	00110
7	00111
8	0001000
9	0001001
10	0001010

Proses berikutnya ialah dilakukan kompresi nilai pada sample menggunakan kode *elias gamma code* yang didapat pada tabel diatas. Proses kompresi *file* AVI sample dapat di lihat pada tabel dibawah:

Tabel 5 Kompresi Nilai *File* AVI Dengan *Elias Gamma Code*

N	Nilai Hexa	Elias Gamma Code	Bit	Frek	Bit x Frek
1	00	1	1	5	5
2	68	010	3	2	6
3	BE	011	3	1	3
4	22	00100	5	1	5
5	64	00101	5	1	5
6	72	00110	5	1	5
7	6C	00111	5	1	5
8	61	0001000	7	1	7
9	76	0001001	7	1	7
10	69	0001010	7	1	7
11	38	0001011	7	1	7
Total Bit					62

Dari perhitungan tabel di atas setelah dikompresi memakai *elias gamma code* adalah 62 bit. Untuk dilakukan pengubahan menjadi satuan byte maka dibagikan 8 yaitu $62/8 = 7,75$.

- e. Melakukan hasil string bit *elias gamma code*

Sebelum dilakukan hasil string bit *elias gamma code* menjadi nilai *file*, terlebih dulu lakukan pemeriksaan pada panjang *string* bit. Dibentuknya nilai bit baru hasil kompresi pada susunan nilai heksadesimal sebelum kompresi yaitu BE, 22, 00, 00, 68, 64, 72, 6C, 61, 76, 69, 68, 38, 00, 00, 00 (tanpa tanda koma dan spasi) menjadi nilai bit biner :

“011 00100 1 1 010 00101 00110 00111 0001000 0001001 0001010 010 0001011 1 1 1” Kemudian sebelum didapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *string bit* itu sendiri yaitu *padding* dan *flag bit*. Jika sisa bagi panjang string bit terhadap 8 adalah 0 maka tambahkan 0000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi panjang *string* bit terhadap 8 adalah $n(1,2,3,4,5,6,7)$ maka 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 62 tidak habis dibagi 8 dan sisanya 6 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.

011001001101000101001100011100010000001001000101001000
010111110100000011

Gambar 2 Perhitungan Penambahan Bit

01100100 11010001 01001100 01110001 00000010 01000101
00100001 01111101 00000011

Gambar 3 String bit yang telah dilakukan penambahan

Total panjang bit keseluruhan setelah ada penambahan bit adalah $62+2+8 = 72$. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti gambar di bawah ini

$7 - n + 1$
 $7 - 6 + 1 = 01$
 Bit Akhir 9 – n
 Bit Akhir = $9 - 6 = 3 = 00000011$

Gambar 4 Pembagian String bit

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 9 kelompok nilai biner baru (9 byte) yang sudah terkompresi beserta nilai biner penambahan bit. Setelah pembagian dilakukan, maka nilai yang sudah dibagi diubah ke dalam suatu karakter dengan terlebih dahulu mencari nilai *hexadecimal* dan *string* bit tersebut menggunakan kode ASCII untuk mengetahui nilai yang sudah terkompresi. Adapun nilai yang sudah terkompresi dapat dilihat pada tabel di bawah ini.

Tabel 6 Nilai *Hexadecimal* Terkompresi

Urutan Nilai Terkompresi	Nilai <i>Hexadecimal</i> Terkompresi
1	64
2	D1
3	4C
4	71
5	2
6	45

7	21
8	7D
9	3

Persentase ukuran data yang dikompresi dan dapat dilihat dari hasil perbandingan antara ukuran data setelah dikompresi dengan ukuran data sebelum dikompresi.

Ukuran data sebelum dikompresi = $128/8 = 16$ byte

Ukuran data sesudah dikompresi = $72/8 = 9$ byte

Berdasarkan data tersebut dapat dihitung kinerja kompresinya yaitu :

- *Compression Ratio (Cr)*

$$Cr = \frac{\text{Ukuran data Sesudah Dikompresi}}{\text{Ukuran data Sebelum Dikompresi}} \times 100\%$$

$$Cr = \frac{9}{16} \times 100\%$$

$$Cr = 56,25\%$$

- *Space Saving*

SS = ukuran data sebelum dikompresi – ukuran data setelah dikompresi

$$SS = 128 - 72$$

$$SS = 56$$

2. Analisa proses dekompresi file AVI menggunakan Elias Gamma Code

Pada proses dekompresi, langkah awalnya sistem akan membaca string hasil kompresi, setelah mendapatkan string bit hasil kompresi, tabel hexadesimal dan kodennya, kemudian sistem melakukan proses dekompresi dengan mengembalikan string bit tersebut ke bentuk semula berdasarkan tabel karakter dan tabel kode yang telah didapat dari hasil kompresi.

Dilakukan analisa keseluruhan bit dari hasil kompresi sebelumnya untuk melakukan proses dekompresi. Adapun keseluruhan bit dan gabungan hasil kompresi, yaitu :

“01100100 11010001 01001100 01110001 00000010 01000101 00100001 01111101 00000011”

- Proses awal dekompresi yaitu melakukan pembacaan nilai *flag* bit dari keseluruhan bit tersebut dengan cara merubah nilai 8 bit terakhir ke dalam nilai desimal seperti 00001000 = 8. Nilai 8 tersebut merupakan 8 buah nilai bit yang pada saat *padding* sebelum ditambahkan *flag* bits.
- Selanjutnya hapus *flag* bit dan *padding* dari nilai keseluruhan bit sehingga menjadi “0110010011010001010011000110001000000100100010100100001 011111”
- Pengecekan bit yaitu dengan cara melakukan cek bit dari bit pertama dengan tabel kode *elias gamma code*. Jika ditemukan bit yang sesuai dengan tabel kode *elias gamma code* maka ubah nilai string yang sesuai sehingga akan mendapatkan hasil tabel seperti dibawah ini.

Tabel 7 Hasil dekompresi menggunakan Elias Gamma Code

Elias Gamma Code	Heks
011	BE
00100	22
1	00
1	00
010	68
00101	64
00110	72
00111	6C
0001000	61
0001001	76
0001010	69
010	68
0001011	38
1	00
1	00
1	00

Berdasarkan hasil dekompresi di atas didapatkan karakter awal sebelum dikompresi sebagai berikut “BE, 22, 00, 00, 68, 64, 72, 6C, 61, 76, 69, 68, 38, 00, 00, 00”.

3. Kompresi File AVI dengan algoritma elias delta code

- Dilakukan pengurutan karakter yang memiliki frekuensi paling besar (banyak nilai yang sama) ke frekuensi paling kecil. Urutan nilai dapat dilihat dalam tabel di bawah:

Tabel 8 Nilai Bit File AVI Sample

Nilai		Bit	Frek	Bit x Frek
Hexa	Biner			
00	00000000	8	5	40
68	011010001	8	2	16
BE	10111110	8	1	8
22	00100010	8	1	8
64	01100100	8	1	8
72	00000111	8	1	8
6C	01101100	8	1	8
61	01100001	8	1	8
76	01110110	8	1	8
69	01101001	8	1	8
38	00111000	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 dengan jumlah 128 bit. Dalam pengubahan satuan menjadi byte maka jumlah seluruh bit dibagi 8. Maka dihasilkan $128/8 = 16$.

b. Membentuk Tabel Elias Delta Code

Aturan pada pembentukan kode bilangan dengan memakai *elias delta code* dapat dilihat pada bab sebelumnya. Adapun kode *elias delta code* dapat dilihat pada tabel di bawah ini.

Tabel 9 Kode *Elias Delta Code*

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

Proses berikutnya ialah dilakukan kompresi nilai pada sample dengan kode *elias delta code* yang didapat pada tabel 10 di atas. Adapun proses kompresi *file AVI sample* dapat di lihat pada tabel berikut :

Tabel 10 Kompresi Nilai *File AVI Sample* Dengan *Elias Delta Code*

N	Nilai Hexa	Kode Elias Delta	Bit	Frek	Bit x Frek
1	00	1	1	5	5
2	68	0100	4	2	8
3	BE	0101	4	1	4
4	22	01100	5	1	5
5	64	01101	5	1	5
6	72	01110	5	1	5
7	6C	01111	5	1	5
8	61	00100000	8	1	8
9	76	00100001	8	1	8
10	69	00100010	8	1	8
11	38	00100011	8	1	8
Total Bit					69

Pada perhitungan tabel di atas setelah dikompresi dengan memakai *elias delta* ialah 69 bit. Agar dilakukan pengubahan menjadi satuan byte harus dibagi 8 yaitu $69/8 = 8,625$.

c. Melakukan hasil string bit elias delta code

Sebelum dilakukan hasil string bit elias delta code menjadi nilai file, terlebih dulu lakukan pemeriksaan pada panjang string bit. Pembentukan nilai bit baru hasil kompresi pada susunan nilai heksadesimal sebelum kompresi yaitu BE, 22, 00, 00, 68, 64, 72, 6C, 61, 76, 69, 68, 38, 00, 00, 00 (tanpa tanda koma dan spasi) menjadi nilai bit biner :

“0101 01100 1 1 0100 01101 01110 00100000 00100001 00100010 0100 00100011 1 1 1”. Lalu sebelum didapat hasil semua akhir kompresi dilakukan penambahan *string bit* yaitu *padding* serta *flag bit*. Jika sisa bagi panjang string bit terhadap 8 ialah 0 maka ditambahkan 0000001. Dinyatakan dengan bit

akhir. Sedangkan jika sisa bagi panjang *string* bit terhadap 8 adalah $n(1,2,3,4,5,6,7)$ maka 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 69 tidak habis dibagi 8 dibagi 8 dan memiliki sisa 5, 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.

$$\begin{aligned} &7 - n + "1" \\ &7 - 5 + "1" = \mathbf{001} \\ &\text{Bit Akhir } 9 - n \\ &\text{Bit Akhir} = 9 - 5 = 4 = \mathbf{00000100} \end{aligned}$$

Gambar 10 Perhitungan Penambahan Bit

010101100110100011010111001111001000000010000
10010001001000010001111100100000100

Gambar 11 String bit yang telah dilakukan penambahan

Total panjang bit keseluruhan setelah ada dilakukan tambah nilai bit ialah $69+3+8 = 80$. Lalu lakukan pemisahan bit ke beberapa kelompok. Perkelompok terdiri atas 8 bit seperti gambar di bawah

01010110 01101000 11010111 00111100 10000000 10000100
10001001 00001000 11111001 00000100

Gambar 12 Pembagian String Bit

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 10 kelompok nilai biner baru (10 byte) yang telah dikompresi nilai biner penambahan bit. Lalu dilakukan pembagian, maka nilai yang telah dibagikan diubah dalam suatu karakter dengan terlebih dahulu dilakukan pencarian nilai *hexadecimal* serta *string* bit ini memakai kode ASCII dalam diketahui nilai yang sudah dikompresi. Adapun nilai yang telah terkompresi dapat di lihat dalam tabel di bawah.

Tabel 9 Nilai *Hexadecimal* Terkompresi

Urutan Nilai Terkompresi	Nilai <i>Hexadecimal</i> Terkompresi
1	56
2	68
3	D7
4	3C
5	80
6	84
7	89
8	8
9	F9
10	4

Persentase ukuran data yang dikompresi dapat dilihat pada hasil perbandingan antara ukuran data sesudah dikompresi menggunakan ukuran data sebelum dikompresi.

Ukuran data sebelum dikompresi = $128/8 = 16$ byte

Ukuran data sesudah dikompresi = $80/8 = 10$ byte

Dari data ini dapat dilakukan perhitungan kinerja kompresinya yaitu :

- *Compression Ratio* (Cr)

$$Cr = \frac{\text{Ukuran data Sesudah Dikompresi}}{\text{Ukuran data Sebelum Dikompresi}} \times 100\%$$

$$Cr = \frac{10}{16} \times 100\%$$

$$Cr = 62,50\%$$

- *Space Saving*

SS = ukuran data sebelum dikompresi – ukuran data setelah dikompresi

$$SS = 128 - 80$$

$$SS = 48$$

4. Analisa proses dekompresi file AVI menggunakan elias delta code

a. Memasukkan hasil file kompresi

Nilai *hexadecimal* dari hasil kompresi yaitu : 56, 68, D7, 3C, 80, 84, 89, 8, F9, 4.

- b. Hasil *string* bit *elias delta code* yang dijadikan nilai *file* diubah ke dalam bentuk biner. Bit keseluruhan hasil kompresi dapat dilihat pada tabel berikut:

Tabel 10 Nilai *Hexadecimal* dan Biner Kompresi

Urutan Nilai Terkompresi	Nilai <i>Hexadecimal</i>	Nilai Biner
1	56	01010110
2	68	01101000
3	D7	11010111
4	3C	00111100
5	80	10000000
6	84	10000100
7	89	10001001
8	8	00001000
9	F9	11111001
10	4	00000100

Berdasarkan tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi 01010110 01101000 11010111 00111100 10000000 10000100 10001001 00001000 11111001 00000100

- c. Dikembalikan *binary* menjadi *string* bit awal dengan melakukan penghilangan biner ditebalkan. Dalam pengembalian *binary* menjadi *string* bit awal dapat dilakukan dengan langkah berikut. Dilakukan pembacaan pada 8 bit terakhir, hasil pembacaan dalam bentuk bilangan desimal. Dinyatakan hasil yang dibaca dengan n . Hilangkan bit dalam bagian akhir dengan jumlah $7 + n$. Setelah hitungan dilakukan pembacaan bit akhir. Nilai biner yang dihilangkan berjumlah 8 bit pada akhir $n=1$. Dari yang dijelaskan diatas ditunjukkan jika bit akhir harus dihilangkan. Hasil bagi *binary* menjadi *string bit* semula adalah 01010110 01101000 11010111 00111100 10000000 10000100 10001001 00001000 11111. Dari hitungan diatas menggunakan algoritma *elias delta code*, *string bit* di atas bernilai 69 bit seperti diawal hingga dilakukan pembacaan *string* bit awal. Adapun tabel hasil perhitungan diatas ialah :

Tabel 11 Hasil dekomposisi menggunakan *Elias Delta Code*

<i>Elias Delta Code</i>	Heks
0101	BE
01100	22
1	00
1	00
0100	68
01101	64
01110	72
01111	6C
00100000	61
00100001	76
00100010	69
0100	68
00100011	38
1	00
1	00
1	00

Berdasarkan hasil dekomposisi di atas didapatkan karakter awal sebelum dikompresi sebagai berikut “BE, 22, 00, 00, 68, 64, 72, 6C, 61, 76, 69, 68, 38, 00, 00, 00”.

5. Perbandingan Algoritma *elias delta code* dan *elias gamma code*

Perbandingan algoritma *elias delta code* dan *elias gamma code* ialah membandingkan rasio kompresi dari kedua algoritma. Dimana setelah dilakukan perhitungan dengan algoritma *elias delta code* dan *elias gamma code* maka hasil dari rasio kompresi dan *space saving* didapatkan. Hasil dari rasio kompresi algoritma *elias gamma code* sebesar 56,25% dan hasil rasio dari algoritma *elias delta code* sebesar 62,50%. Dari hasil rasio kedua algoritma dapat diketahui bahwa algoritma *elias delta code* lebih akurat dalam melakukan kompresi file AVI. Adapun tabel perbandingan dapat di lihat pada tabel di bawah ini.

Tabel 4.14 Nilai Perbandingan

<i>Elias Delta Code</i>	<i>Elias Gamma Code</i>
Rasio Kompresi 62,50%	Rasio Kompresi 56,25%

4. KESIMPULAN

Ada beberapa kesimpulan yang dapat ditarik setelah dilakukan pengerjaan pada bab-bab sebelumnya. Kesimpulan tersebut antara lain:

1. Kompresi *file* AVI dapat dilakukan dengan mengubah kedalam bentuk heksadesimal untuk dapat melakukan perhitungan. Bilangan heksadesimal yang didapat diolah menggunakan algoritma *additive code* dan *stout code* sehingga menghasilkan nilai yang baru yang akan merubah ukuran *file* AVI.
2. Algoritma *additive code* dan *stout code* dapat dilakukan perbandingan setelah dilakukan perhitungan. Dengan skala perbandingan yaitu rasio kompresi dan *space saving*. Dengan perbandingan yang dilakukan maka diketahui bahwa persentase rasio kompresi algoritma *additive code* lebih besar daripada algoritma *stout code*.
3. Rasio kompresi dan *space* dapat diketahui dengan melakukan mengubah nilai awal *file* AVI dengan nilai heksadesimal. Didapatkan nilai rasio kompresi algoritma *additive code* sebesar 81,25% dan hasil rasio kompresi dengan algoritma *stout code* sebesar 56, 25%.

REFERENCES

- [1] D. Pranata and D. K. Marisa, "RANCANG BANGUN WEBSITE JURNAL ILMIAH BIDANG KOMPUTER (STUDI KASUS : PROGRAM STUDI ILMU KOMPUTER UNIVERSITAS MULAWARMAN)," 2015.
- [2] K. Tampubolon, H. Saragih, B. Reza, K. Epicentrum, A. Asosiasi, and A. Apriori, "Implementasi Data Mining Algoritma Apriori Pada Sistem Persediaan Alat-Alat Kesehatan," pp. 93–106, 2013.
- [3] E. Ndruru and E. N. Purba, "Penerapan Metode ARAS Dalam Pemilihan Lokasi Objek Wisata Yang Terbaik Pada Kabupaten Nias Selatan," *METHOMIKA J. Manaj. Inform. Komputerisasi Akunt.*, vol. 3, no. 2, pp. 151–159, 2019.
- [4] R. Damanik, C. P. D. Sinurat, and K. M. Rajaguguk, "Pelatihan Pembuatan Aplikasi Mobile Berbasis Android pada SMA Swasta Deli Murni Sibolangit," *ULEAD J. E-Pengabdian*, vol. 1, pp. 40–42, 2022, doi: 10.54367/ulead.v2i1.2045.
- [5] D. Jurusan and S. Rupa, "PERAN DESAIN GRAFIS PADA LABEL DAN KEMASAN PRODUK MAKANAN UMKM Oleh: Rahina Nugrahani," 2015.
- [6] M. Simangunsong, "Perbandingan Algoritma Elias Delta Code Dan Unary Coding Dalam Kompresi Citra Forensik," vol. 12, no. 1, pp. 18–26, 2020.
- [7] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: C.V Andi Offset, 2010.