JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Perbandingan Kompresi File Audio Menggunakan Algoritma Fibonacci Dan Algoritma Invert Elias Delta

Sartika Siahaan

Fakultas Ilmu Komputer & Teknologi Informasi, Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: sartikasiahaan@gmail.com

Abstrak - Perkembangan teknologi informasi yang pesat telah menjadi peran yang sangat penting untuk pertukaran informasi atau pengiriman data dari satu terminal komputer ke terminal komputer yang lain. Kecepatan pengiriman sangat bergantung pada ukuran dari informasi atau data tersebut. Data dengan ukuran besar akan memakan waktu pengiriman yang lebih lama dibandingkan dengan data yang memiliki ukuran data yang lebih kecil, karena tidak dapat tertampung pada media penyimpanan. Format file audio adalah format file untuk menyimpan data audio digital pada sistem komputer. Salah satuformat file audio adalah MP3 yang bisa membuat seseorang mendengarkan lagu secara digital, tidak perlu menggunakan kaset atau CD seperti zaman dulu dan memungkinkan jutaan orang diseluruh dunia untuk saling bertukar rekaman musik melalui komputer yang terhubung jaringan internet. Penyimpanan data tersebut, tidak bisa hanya dilakukan dengan format file yang tersedia. Maka dari itu diperlukan proses kompresi file. Dalam kompresi data terdapat 3 faktor penting yang perlu diperhatikan yaitu: ratio commpression, compression ratio, dan space saving. Setelah dilakukan implementasi dan pengujian sistem dapat diketahui bahwa algoritma Invert Elias Delta memiliki kinerja yang lebih baik dibandingkan dengan algoritma Fibonacci pada kompresi file audio dengan hasil akhir menggunakan metode eksponensial, algoritma Invert Elias Delta dengan nilai akhir 10,217 % sedangkan algoritma Fibonacci 11,074% . Sehingga semakin tinggi total nilai yang diperoleh maka akan semakin tinggi jumlah usaha yang dilakukan oleh algoritma tersebut

Kata Kunci: Perbandingan, Algoritma, Kompresi, Fibonacci, Invert Elias Delta

Abstract - *The rapid development of information technology has played a very important role in exchanging information or sending data from one computer terminal to another. Delivery speed really depends on the size of the information or data. Data with a large size will take longer to send than data with a smaller data size, because it cannot be accommodated on the storage media. Audio file format is a file format for storing digital audio data on a computer system. One audio file format is MP3 which can allow someone to listen to songs digitally, no need to use cassettes or CDs like in the past and allows millions of people throughout the world to exchange music recordings via computers connected to the internet network. Storing this data cannot only be done using the available file formats. Therefore, a file compression process is needed. In data compression, there are 3 important factors that need to be considered, namely: compression ratio, compression ratio, and space saving. After implementing and testing the system, it can be seen that the Invert Elias Delta algorithm has better performance compared to the Fibonacci algorithm in audio file compression with the final result using the exponential method, the Invert Elias Delta algorithm with a final value of 10.217% while the Fibonacci algorithm is 11.074%. So the higher the total value obtained, the higher the amount of effort carried out by the algorithm.*

Keywords: *Comparison, Algorithms, Compression, Fibonacci, Invert Elias Code*

1. PENDAHULUAN

Kompresi adalah pengubahan data yang berupa kumpulan karakter menjadi bentuk kode dengan tujuan untuk menghemat kebutuhan tempat penyimpanan dan waktu transmisi data.[1] Dalam kompresi data terdapat 4 faktor penting yang perlu diperhatikan yaitu: *time process* (waktu yang dibutuhkan dalam menjalankan proses), *completness* (kelengkapan data setelah dikompresi), *ratio compress* (ukuran data setelah dikompresi), *optimality* (perbandingan ukuran data sebelum dikompresi dengan yang telah dikompresi. Banyak algoritma kompresi yang dapat digunakan, diantaranya adalah algoritma *Fibonacci* dan algoritma *Invert Elias Delta*. Penggunaan algoritma *Fibonacci* dan algoritma *Invert Elias Delta* dalam kompresi *file* audio dimaksudkan untuk memberikan manfaat yang sangat besar dalam proses pengiriman maupun penyimpanan serta membutuhkan ruang memori yang lebih sedikit dbandingkan dengan audio yang tidak dikompresi. Sehingga dalam pengiriman ataupun penyimpanan audio yang telah dikompresi membutuhkan waktu pengiriman dan ruang simpan yang lebih singkat dan sedikit dibandingkan dengan audio yang belum dikompresi. Setelah melakukan kompresi kedua algoritma tersebut maka, kita

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

membandingkannya menggunakan metode eksponensial. Membandingkan kedua algoritma tersebut bertujuan untuk mengetahui algoritma mana yang lebih efisien dalam melakukan kompresi, sehingga kita bisa memilih algoritma mana yang akan kita gunakan saat akan mengkompresi *file* audio.

Algoritma *Fibonacci* akan memberikan ukuran kompresi yang lebih baik dan memberikan penghematan ruang yang lebih baik juga. Dengan menggunakan metode tersebut, hasil kompresi dari nilai *Fibonacci* mempunyai hasil yang berbeda-beda dari setiap nilainya, dan hasil kompresi akan menguntugkan dalam melakukan pengiriman, pengunduhan dan pemindahan *file audio* akan semakin mudah. *Elias delta code* adalah kode universal yang mengkodekan bilangan bulat pusat, yang dikembangkan oleh Peter Elias. *Elias Delta Code* adalah salah satu dari tiga *Elias Code* yang dipopuleri oleh Peter Elias. Didalam kode ini, setiap karakter diakili dengan memetakan kode sumber kesejumlah variabel bit.[2]

Penelitian yang dilakukan oleh Krisna Monita R Hutahaean pada tahun 2022 tentang "Penerapan Algoritma Inverted Elias Delta Untuk Kompresi Konten Pada Aplikasi Psikologi Berbasis Android" dapat disimpulkan bahwa prosedur dari kompresi konten pada aplikasi psikologi dengan menggunakan algoritma inverted elias deltadapat ditarik kesimpulan bahwa prosedur tersebuttelah berhasil untuk melakukan proses kompresi recorddatabase pada aplikasi psikologi. Berdasarkan hasil penerapan algoritma Inverted Elias Delta, ukuran data sebelum dikompresi adalah 648 bit dan setelah di kompresi adalah 400bit. Dari hasil perhitungan compression ratio dapat disimpulkan bahwa setelah dikompres ukuran file adalah 62% dari data sebelum dikompres [3]. Penelitian yang dilakukan oleh Dita Anggraini, dengan menggunakan algoritma *Fibonacci* pada tahun 2021 yang berjudul "Penerapan Algoritma *Fibonacci Code* Pada *File Transfer* Berbasis Android" dapat disimpulkan bahwa pada aplikasi file transfer dalam mengirim sebuah file yaitu dengan memilih *file* yang akan dikompresi, setelah dipilih maka lakukan proses kompresi dan *file* hasil kompresi tersebut yang akan dikirim melalui jaringan, lalu proses dekompresi dilakukan oleh si penerima setelah file telah diterima[4]. Penerapan algortima Fibonacci Code untuk mengkompresi file yang akan dikirim telah berhasil mmembuat ukuran ukuran file menjadi lebih kecil dengan rasio 48% sehingga dapat memberikan proses transmisi yang cepat serta membantu mengurangi pemakaian paket data internet.

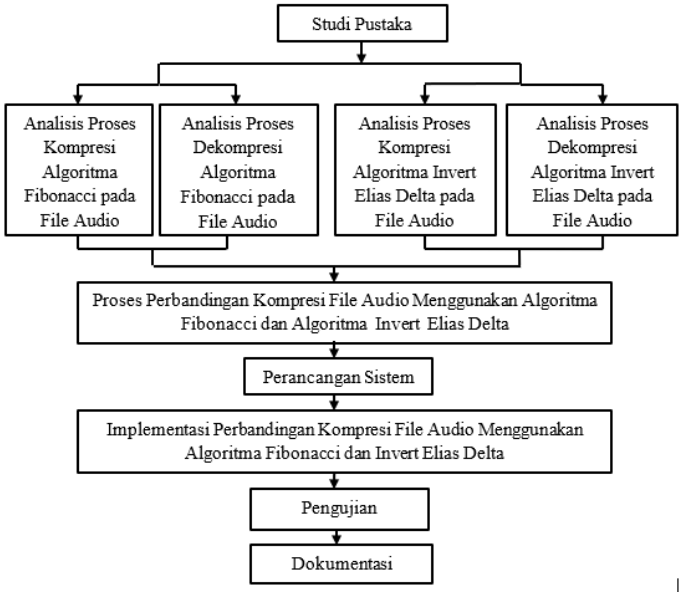
Penelitian-penelitian terdahulu tersebut menjadi acuan dalam penelitian ini tentang kompresi file audio, dengan menggunakan algoritma Fibonacci dan algoritma Invert Elias Delta. Pengguna algoritma ini diharapkan dapat mengurangi byte data dalam file audio dan menghemat ruang penyimpanan. Dari uraian latar belakang masalah di atas, penulis dapat menguraikan tahap-tahap pemberian solusi yang dituangkan dalam skripsi dengan judul "Perbandingan Kompresi File Audio Menggunakan Algoritma Fibonacci dan Algoritma Invert Elias Delta".

2. METODE PENELITIAN

Metode penelitian tentang perbandingan kompresi file audio menggunakan algoritma fibonacci dan algoritma invert elias delta. Kerangka penelitian adalah suatu bentuk kerangka kerja yang dapat digunakan sebagai pendekatan untuk pemecahan masalah. Kerangka kerja ini merupakan prosedur yang akan diambil untuk memecahkan masalah yang dibahas. Adapun kerangka kerja penelitian dapat digambarkan pada gambar:

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin



Gambar 1. Metode Penelitian

2.1 Kompresi

Kompresi adalah memperkecil file yang berukuran besar menjadi lebih kecil dan mengurangi ruang penyimpanan. Kompresi merupakan proses mendekati minimalisasi jumlah bit untuk representasi digital seperti video, gambar, maupun audio yang memiliki ukuran data lebih kecil namun tetap menjaga kualitas informasi pada data tersebut [6]. Kompresi data sangat penting karena bisa memperkecil ukuran bytes data, menghemat ruang penyimpanan, mempercepat waktu pengiriman data. Teknik kompresi bisa dilakukan terhadap data teks/biner, gambar (JPEG, PNG, TIFF), *audio* (MP3, AAC, RMA, WMA), dan *video* (MPEG, H261, H263).

Terdapat beberapa teknik untuk membuktikan nilai pada suatu metode kompresi yaitu:

1. *Compression Ratio (CR)*
   Compression ratio adalah presentasi data yang pernah di kompresi dari hasil perselisihan antara ukuran data yang belum dikompresi serta data yang sudah dikompres.
   $$CR = \frac{ukuran\ data\ sesudah\ dikompres}{ukuran\ data\ sebelum\ dikompres} \quad (1)$$
2. *Ratio of Compression (RC)*
   Ratio of Compression (RC) merupakan perselisihan antara ukuran data sebelum dikompresi dengan ukuran data telah dikompresi[9].
   $$RC = \frac{ukuran\ data\ sebelum\ dikompres}{ukuran\ data\ sesudah\ dikompres} \times 100\% \quad (2)$$
3. *Space Saving* (SS)
   Space Savings (SS) adalah persentase selisih antara data yang belum dikompresi dengan besar data yang dikompresi.
   SS = 100% - *Ratio of Compression* (3)

2.2 Algoritma Fibonacci

Kode Fibonacci adalah *variable length code*, dimana bilangan bulat yang lebih kecil mendapatkan kode pendek. Kode berakhir dengan dua buah bit satu, dan nilai yang didapatkan adalah jumlah dari nilai-nilai Fibonacci yang sesuai untuk bit yang ditetapkan (kecuali bit terakhir, yang merupakan akhir dari kode). Barisan fibonacci merupakan sebuah bilangan positif yang dimana bilangan tersebut ditentukan dari hasil penjumlahan kedua suku sebelumnya sehingga diperoleh barisan bilangan dengan pola dibawah ini:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 … dan seterusnya.

Langkah-langkah pembentukan sebuah kode Fibonacci adalah sebagai berikut:

1. Tentukan sebuah bilangan bulat positif n.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

2. Temukan bilangan Fibonacci f terbesar yang lebih kecil atau sama dengan n, kurangkan nilai n dengan f dan catat sisa pengurangan nilai n dengan f.
3. Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret Fibonacci F(i), tambahkan angka "1" pada i-2 dalam kode Fibonacci yangakan dibentuk.
4. Ulangi langkah 2, tukar nilai n dengan sisa pengurangan nilai n dengan fsampai sisa pengurangan nilai n dengan f adalah 0.
5. Tambahkan angka "1" pada posisi paling kanan kode Fibonacci yang akandibentuk.

Melakukan *decode* sebuah kode Fibonacci, hilangkan angka "1" paling kanan, kemudian substitusi dan jumlahkan kode Fibonacci yang tersisadenganmenggunakan deret Fibonacci.

Tabel 1. Pembentukan Kode *Fibonacci*

Urutan bilangan *Fibonacci*	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)	F(10)
Bilangan *Fibonacci*	0	1	1	2	3	5	8	13	21	34	55
Kode *Fibonacci* sementara	-	-	0	1	0	0	1	0	0	0	1

Sumber : Dita Anggraini, 2021 [10].

2.3 Algoritma Invert Elias Delta

Langkah-langkah untuk mengkodekan sebuah bilangan dengan menggunakan Invert Elias Delta Code adalah sebagai berikut [12]:

1. Tuliskan n dalam bentuk bilangan biner. Biner n diumpamakan dengan b.
2. Hitung jumlah bit pada b nya, kemudian tuliskan dalam bentuk bilangan biner. Jumlah bit pada b diumpamakan dengan M.
3. Tuliskan M dalam bentuk bilangan biner. Biner M diumpamakan dengan Mb.
4. hapus satu bit yang paling kiri dari biner n atau yang disebut dengan b, kemudian tuliskan b yang sudah dihapus satu bit dari sebelah kiri tersebut. b yang sudah dihapus satu bit dari sebelah kiri tersebut diumpamakan dengan L.
5. Kemudian lakukan perhitungan dengan menggunakan rumus di bawah ini [11]: Δ = (length(Mb) – 1) * "0" + Mb + L Keterangan : b = decimal to biner (n) M = len (b) Mb = decimal to biner (M) Len (Mb) = jumlah digit (Mb) L = b yang sudah dihapus satu digit nilai biner yang paling kiri.

2.4 Metode Eksponensial

Metode perbandingan *eksponential* adalah satu metode untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak. Dalam menggunakan metode perbandingan *eksponential* ada beberapa tahap yang dilakukan, yaitu [13], [14]:

1. Menyusun dan menentukan alternatif-alternatif keputusan yang dipilih
2. Menentukan kriteria atau perbandingan keputusan yang penting untuk dievaluasi.
3. Menentukan tingkat kepentingan dari setiap kriteria keputusan.
4. Menentukan penilaian terhadap semua alternatif pada setiap kriteria.
5. Menghitung skor atau nilai total setiap alternatif.
6. Menetukan urutan prioritas keputusan didasarkan pada skor atau nilai total masing-masing alternatif.

Metode perbandingan Exponential adalah satu metode untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak. Adapun rumus dari metode Exponential sebagai berikut:

Total Nilai (TNi) $= \sum_{J=1}^{m} (ViJ)BJ$ ..... (4)

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

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.

3. HASIL DAN PEMBAHASAN

Analisa dan permasalahan yang dibahas dalam penelitian ini adalah bagaimana kinerja antara algoritma *Fibonacci* dan algoritma *Invert Elias Delta* dalam mengkompresi file audio dan perbandingan kinerja kedua algoritma tersebut berdasarkan parameter *Ratio of Compression* (RC), *Compression Ratio* (C_R), *Redudancy* (RD), *Space Saving* (S_S) Kompresi data menerapkan algoritma Fibonacci dan algoritma Invert Elias Delta terhadap file audio yang berformat mp3. Kedua algoritma tersebut merupakan teknik kompresi *losseless* dimana data yang telah dikompresi dapat dikembalikan ke data asli.

Proses awal dalam analisa file *audio* yang akan dikompresi lalu diubah menjadi nilai *hexadecimal* dengan menggunakan aplikasi *Binery Viewer*. Setelah diketahui nilai *hexadecimal* maka dilakukan proses kompresi dengan algoritma *Fibonacci* dan algoritma *Invert Elias Delta*. Kemudian kedua algoritma tersebut akan dibandingkan untuk mendapatkan algoritma yang terbaik dalam kompres *file* audio. Prosedur perbandingan algoritma *Fibonacci* dan algoritma *Invert Elias Delta*

3.1 Analisa Proses Kompresi Algoritma Fibonacci

Proses analisa kompresi *file* audio dengan menggunakan algoritma Fibonacci. Algoritma Fibonacci merupakan suatu cara atau langkah-langkah sistematis yang digunakan untuk menghasilkan deret angka Fibonacci. Deret angka Fibonacci adalah deret bilangan yang setiap angka selanjutnya merupakan hasil penjumlahan dari dua angka sebelumnya dalam deret tersebut. Deret dimulai dengan angka 0 dan 1. Langkah-langkah pembentukan kode Fibonacci adalah sebagai berikut:

1. Tentukan sebuah bilangan bulat positif n yang lebih besar atau sama dengan 2.
2. Temukan bilangan Fibonacci f terbesar yang lebih kecil atau sama dengan n, kurangkan nilai n dengan f dan catat sisa pengurangan nilai n dengan f.
3. Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret Fibonacci F(i), tambahkan angka “1” dalam kode Fibonacci yang akan dibentuk.
4. Ulangi langkah 2, tukar nilai n dengan sisa pengurangan nilai n dengan f sampai sisa pengurangan nilai n dengan f adalah 0.
5. Tambahkan angka “1” pada posisi paling kanan kode Fibonacci yang akan dibentuk.

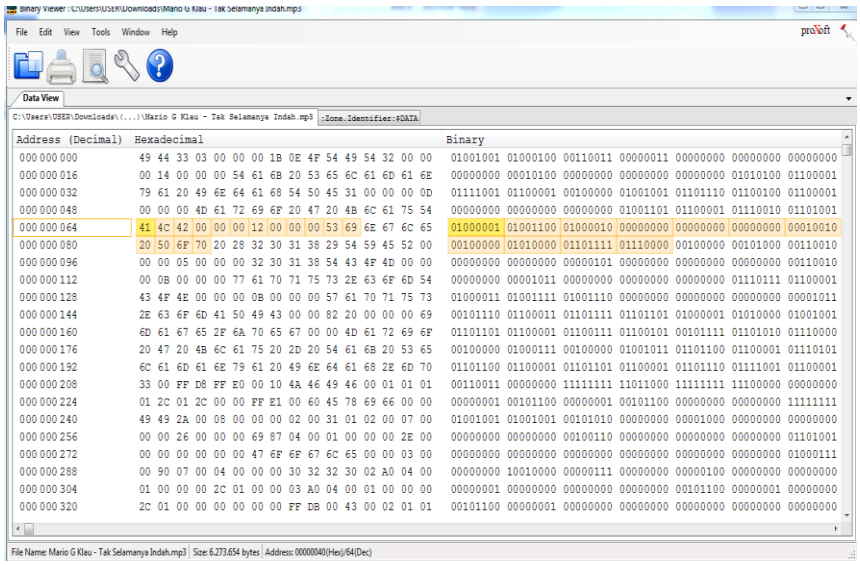
Terlebih dahulu cari *file* audio yang akan dikompresi, berikut ini contoh *file* audio yang akan dikompresi dan dekompresi.

Tabel 2. Sampel *file* yang akan dikompresi

Nama *File*	**Tak Selamanya Indah**
Jenis item	File MP3 (mp3)
Ukuran	6,1 MB

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin



Gambar 3. Nilai *Hexadecimal* Sampel *File Audio*

Berdasarkan sampel nilai *hexadecimal* diatas, maka akan diambil 20 sampel nilai *hexadecimal* dalam pengoperasian *file audio*.

Tabel 3. Tampilan Bilangan *Hexadecimal*

41	4C	42	00	00	00	12	00
00	00	53	69	20	50	6F	70

Berdasarkan tabel diatas, maka diperoleh nilai bilangan *hexadecimal file audio* mp3 yang akan dihitung dengan cara manual. Nilai bilangan *hexadecimal* diurutkan berdasarkan frekuensinya, nilai frekuensi paling banyak yang akan berada diurutan paling atas, dapat dilihat pada tabel berikut:

Tabel 4. Nilai *hexadecimal* yang belum dikompresi

Nilai		Bit	Frekuuensi	Bit x Frekuensi
Hexadecimal	Biner			
00	00000000	8	6	48
41	01000001	8	1	8
4C	01001100	8	1	8
42	01000010	8	1	8
12	00010010	8	1	8
53	01010011	8	1	8
69	01101001	8	1	8
20	00100000	8	1	8
50	01010000	8	1	8
6F	01101111	8	1	8
70	01110000	8	1	8
Total				128 bit

Berdasarkan tabel 4 di atas, satu nilai *hexadecimal* terdapat delapan bit bilangan biner. Maka 16 nilai *hexadecimal* mempunyai nilai biner sebanyak 128 bit. Selanjutnya proses kompresi untuk algoritma Fibonacci dapat dilihat pada tabel 5 berikut:

Tabel 5. Nilai *Hexadecimal* yang terkompresi dengan Fibonacci

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

N	Hexa	Freq	Pengurangan nilai n	Fibonacci code sementara	Codeword	Bit	Freq x Bit
1	00	6	1-1	1	11	2	12
2	41	1	2-2	01	011	3	3
3	4C	1	3-3	001	0011	4	4
4	42	1	4-3-1	101	1011	4	4
5	12	1	5-5	0001	00011	5	5
6	53	1	6-5-1	1001	10011	5	5
7	69	1	7-5-2	0101	01011	5	5
8	20	1	8-8	00001	00011	6	6
9	50	1	9-8-1	10001	100011	6	6
10	6F	1	10-8-2	01001	010011	6	6
11	70	1	11-8-3	00101	001011	6	6
Total Bit							62 bit

Tahap selanjutnya, menyusun kembali nilai *hexadecimal* sebelum dikompresi yaitu " **41, 4C, 42, 00, 00, 00, 12, 00, 00, 00, 53, 69, 20, 50, 6F, 70"**, seperti yang disajikan pada tabel berikut:

Tabel 6. *String Bit* Hasil Kompresi dengan Fibonacci

41	**4C**	**42**	**00**	**00**	**00**	**12**	**00**
011	0011	1011	11	11	11	00011	11
00	**00**	**53**	**69**	**20**	**50**	**6F**	**70**
11	11	10011	01011	000011	100011	010011	001011

Berdasarkan pada tabel 4 diatas, string bit yang dihasilkan dari pengkompresian dengan algoritma Fibonacci dapat ditulis sebagai berikut:

Tabel 7. *String bit* hasil Kompresi

011	0011	1011	11
11	11	00011	11
11	11	10011	01011
000011	100011	010011	001011

Selanjutnya dilakukan penambahan string bit yaitu *padding* dan *flagging* dengan mengacu pada sisa jumlah *bit* dibagi 8. Jumlah dari hasil *string bit* yaitu 62 tidak habis dibagi 8 kemudian akan dibentuk variabel penambahan *bit* data. Rumus *padding* yaitu 7-n + "1" dan rumus *flagging* yaitu 9 – n.

62 mod 8 = 6 = n
Padding
7 – 6 + "1" = 01
Flagging
9 – n
9 – 6 = 3 = 00000011

Gambar 4.Perhitungan Penambahan *Bit*

Dari penambahan *padding* dan *flagging* maka diperoleh *string bit* baru yaitu:

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

"01100111011111111000111111111001101011000011100011010011001011**0100000011**" sehingga jumlah total panjang *bit* keseluruhan setelah ada penambahan *padding* dan *flagging* adalah 72 bit. Selanjutnya lakukan pembagian *string bit* menjadi per 8 *bit* seperti dibawah ini:

Setelah diketahui hail kompresi, maka dapat dihitung kinerja algoritma Fibonacci yaitu:

a. Ratio of Compression (RC)
   RC = $\frac{128}{72}$
   RC = 1,7
b. Compression Ratio (CR)
   CR = $\frac{72}{128}$ × 100%
   CR = 56,25%
c. Space Saving (SS)
   SS = 100% - CR
   SS = 100% - 56,25%
   SS = 43,74%

Dari perhitungan diatas dapat diketahui berapa ukuran audio setelah dikompresi menggunakan Algoritma Fibonacci adalah sebagai berikut:

= ukuran audio awal x Compression Ratio
= 6,1 MB x 56,75%
= 3,4 MB

3.2 Analisa Proses Kompresi Algoritma Invert Elias Delta

Pada penelitian ini membahas tentang pengkompresian sebuah file audio dengan menerapkan algoritma Invvert Elias Delta dengan nilai *Pixel* karakter sebagai berikut, 41, 4C, 42, 00, 00, 00, 12, 00, 00, 00, 53, 69, 20, 50, 6F, 70. Nilai *Pixel* yang di dapat dimasukkan ke dalam tabel, dan mencari nilai *Pixel* yang sama untuk dilakukan pembacaan total frekuensi. Adapun pencarian nilai *Pixel* yang sama dan pembacaan total frekuensi dapat dilihat pada tabel di bawah ini:

Tabel 13. Pembacaan Nilai *Pixel*

Pixel	Frekuensi
41	1
4C	1
42	1
00	6
12	1
53	1
69	1
20	1
50	1
6F	1
70	1

Berdasarkan pada tabel diatas, terdapat beberapa nilai *Pixel* yang sama. Sebelum proses kompresi, langkah awal adalah membaca nilai *Pixel* citra kemudian membuat tabel nilai *Pixel* yang diurutkan dari nilai frekuensi terbesar hingga ke terkecil atau pengurutan secara *descending*. Urutan nilai *Pixel* dapat dilihat pada tabel di bawah ini:

Tabel 14. Ukuran *Pixel* sebelum dikompre Bentuk *Descending*

Nilai *Pixel*	Biner	*Bit*	Frekuensi	*Bit* x Frekuensi

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

00	00000000	8	6	48
41	01000001	8	1	8
4C	01001100	8	1	8
4E	01001110	8	1	8
12	00010010	8	1	8
53	01010011	8	1	8
69	01101001	8	1	8
20	00100000	8	1	8
50	01010000	8	1	8
6F	01101111	8	1	8
70	01110000	8	1	8
Total Bit				128 bit

Setelah dilakukan pengurutan secara *descending*, selanjutnya masuk ketahap proses kompresi, dapat dilihat pada tabel dibawah ini.

Tabel 14. Ukuran *Pixel* Setelah dikompresi

N	*Pixel*	Frekuensi	*Invert Elias Delta*	*Bit*	*Bit* x Frekuensi
1	00	6	10	2	12
2	41	1	1011	4	4
3	4C	1	1010	4	4
4	4E	1	10011	5	5
5	12	1	10010	5	5
6	53	1	10001	5	5
7	69	1	10000	5	5
8	20	1	1011111	7	7
9	50	1	1011110	7	7
10	6F	1	1011101	7	7
11	70	1	1011100	7	7
Total Bit					68 bit

Tahap selanjutnya, menyusun kembali nilai *hexadecimal* sebelum dikompresi yaitu " **41, 4C, 42, 00, 00, 00, 12, 00, 00, 00, 53, 69, 20, 50, 6F, 70"**, seperti yang disajikan pada tabel berikut:

Tabel 15. *String Bit* Hasil Kompresi dengan Fibonacci

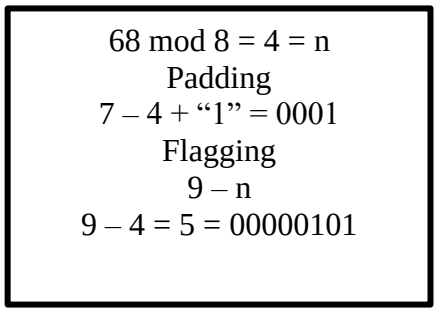
41	**4C**	**42**	**00**	**00**	**00**	**12**	**00**
1011	1010	10011	10	10	10	10010	10
00	**00**	**53**	**69**	**20**	**50**	**6F**	**70**
10	10	10001	10000	1011111	1011110	1011101	1011100

Dari tabel diatas dapat dibentuk *string bit* dari *string* sebelum dikompresi yaitu: "10111010100111010101001010101010001100001011111101111010111011011100".

Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan string *bit* itu sendiri yaitu *padding bit* dan *flagging bit*. Selanjutnya dilakukan penambahan string bit yaitu *padding* dan *flagging* dengan mengacu pada sisa jumlah *bit* dibagi 8. Jumlah dari hasil *string bit* yaitu 68 tidak habis dibagi 8 kemudian akan dibentuk variabel penambahan *bit* data. Rumus *padding* yaitu 7-n + "1" dan rumus *flagging* yaitu 9 – n.

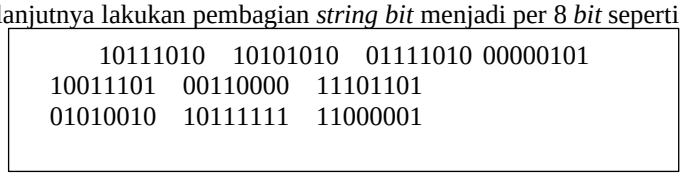
Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin



Gambar 5. Perhitungan Penambahan *Bit*

Dari penambahan *padding* dan *flagging* maka diperoleh *string bit* baru yaitu: "10111010100111010101001010101010001100001011111101111010111011011100000100000101" sehingga jumlah total panjang *bit* keseluruhan setelah ada penambahan *padding* dan *flagging* adalah 80 bit. Selanjutnya lakukan pembagian *string bit* menjadi per 8 *bit* seperti dibawah ini:



Gambar 6. Kelompok *Bit* yang sudah dibagi 8

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 10 kelompok nilai biner baru yang sudah terkompresi. Setelah pembagian dilakukan, maka nilai yang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari string bit tersebut menggunakan kode ASCII untuk megetahui nilai yang sudah terkompresi. Adapun simbol yang sudah terkompresi dapat dilihat pada tabel di bawah ini :

Tabel 16. Hasil Kompresi yang sudah menjadi Simbol

Biner	Nilai Desimal Terkompresi	Simbol
10111010	186	°
10011101	157	.
01010010	82	R
10101010	170	A
00110000	48	0
10111111	191	¿
01111010	122	z
11101101	237	Í
11000001	193	Á
00000101	5	

Setelah diketahui hail kompresi, maka dapat dihitung kinerja algoritma Invert Elias Delta yaitu:

a. Ratio of Compression (RC)

$RC = \frac{128}{80}$

$RC = 1{,}6$

b. Compression Ratio (CR)

$CR = \frac{80}{128} \times 100\%$

$CR = 62{,}5\%$

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

c. Space Saving (SS)
SS = 100% - CR
SS = 100% - 62,5%
SS = 37,5%

Dari perhitungan diatas dapat diketahui berapa ukuran audio setelah dikompresi menggunakan Algoritma Invert Elias Delta adalah sebagai berikut:
= ukuran audio awal x Compression Ratio
= 6,1 MB x 62,5%
= 3,8 MB

3.3 Penerapan Metode Eksponential

Dalam menghitung dan membandingkan proses pencariandari kedua algoritma tersebut adalah sebagai berikut:

a. Menentukan kriteria penguji 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 Saving* (SS).

1. *Ratio of Compression (RC)*
$$RC = \frac{ukuran\ data\ sebelum\ dikompresi}{ukuran\ data\ setelah\ dikompresi}$$
$$F = \frac{128}{72}$$
F = 1,7
$$IED = \frac{128}{80}$$
IED = 1,6
2. Compression Ratio (CR)
$$CR = \frac{ukuran\ data\ setelah\ dikompresi}{ukuran\ data\ sebelum\ \ dikompresi} \times 100\%$$
$$F = \frac{72}{128} \times 100\%$$
F = 56,25%
$$IED = \frac{80}{128} \times 100\%$$
IED = 62,5%
3. Space Saving (SS)
SS = 100% - CR
F = 100% - 56,25%
F = 43,75%
IED = 100% - 62,5%
IED = 37,5%

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 19. Pembobotan Kriteria

Kriteria	Presentase pengaruh kriteria	Bobot range (0-1)
Ratio of Compression	20%	0.2
Compression Ratio	30%	0.3
Space Saving	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 Fibonacci dan algoritma Invert Elias Delta sebelumnya.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Tabel 20. Pemberian Nilai pada setiap kriteria

Alternatif	kriteria				
	RC		CR		SS
	SB	SD	SD	SB	
Algoritma Fibonacci	128 (bit)	72 (bit)	72 (bit)	128 (bit)	1- CR* 100%
Algoritma Invert Elias Delta	128 (bit)	80 (bit)	80 (bit)	128 (bit)	1- CR* 100%

Keterangan :
F : Fibonacci
IED : Invert Elias Delta
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 21. Hasil Perhitungan menggunakan metode perbandingan eksponential

Kriteria	Bobot	Algoritma Fibonacci	Algoritma Invert Elias Delta
RC	0,2	1,7	1,6
CR	0,3	56,25	62,5
SS	0,5	43,75	37,5
Total nilai	1	101,7	101,6

Nilai algoritma Fibonacci = $(1,7)^{0,2} + (56,25)^{0,3} + (43,75)^{0,5}$
= 1,111 + 3,349 + 6,614
= 11,074

Nilai algoritma Invert Elias Delta = $(1,6)^{0,2} + (37,5)^{0,3} + (37,5)^{0,5}$
= 1,098 + 2,966 + 6123
= 10,217

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 22. Prioritas Keputusan

Alternatif	Total Nilai	Rangking
Algoritma Fibonacci	14,544	2
Algoritma Invert Elias Delta	14,143	1

Pada tabel diatas dapat disimpulkan bahwa nilai total dari alternatif terendah mendapatkan peringkat pertama, hal tersebut dikarenakan semakin tinggi nilai ttal tang diperoleh maka semakin tinggi pula usaha algoritma tersebut dalam mengkompresi file audio. Berdasarkan analisis tersebut, algoritma Invert Elias Delta merupakan algoritma dengan performa terbaik saat mengkompresi file audio.

Pengujian sistem telah selesai dilakukan, selanjutnya akan dilihat algoritma mana yang lebih efisien di antara Algoritma Fibonacci dan Algoritma Invert Elias Delta dalam mengkompresi file audio yang berekstensi *.MP3**.** Hal ini dilihat berdasarkan hasil pengujian terhadap beberapa

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 32-44
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

parameter yang telah ditentukan dan akan dibandingkan dari proses kompresi maupun dekompresi yang telah di uji menggunakan file audio. Hasil pengujian dengan menggunakan algoritma Fibonacci dan algoritma Invert Elias Delta terhadap Ratio of Compression (RC), Compressio Ratio (CR), dan Space Saving (SS) dapat dilihat pada tabel berikut:

Tabel 23. Pengujian Perbandingan

Algoritma	Nama File	Ukuran Awal	Ukuran Kompresi	RC	CR	SS
Fibonacci	Tak Selamanya Indah	6,1 MB	3,4 MB	1,7	56,25%	43,74%
Inveret Elias Delta	Tak Selamanya Indah	6,1 MB	3,8 MB	1,6	62,5%	37,5%

4. KESIMPULAN

Hasil dan pembahasan dari penelitian Perbandingan Kompresi File Audio Menggunakan Algoritma Fibonacci dan Algoritma Invert Elias Delta, menghasilkan poin kesimpulan yaitu Mempraktikkan dan menerapkan algoritma Fibonacci serta algoritma Invert Elias Delta dalam mengkompresi file bacaan sangat baik, hasilnya dari jumlah karakter tidak menurun dari file aslinya ataupun tidak terdapat pengurangan (isi file nya). Terkait dengan aplikasi, aplikasi yang dirancang sudah sanggup melaksanakan kompresi file bacaan dengan memakai algoritma Fibonacci serta algoritma Invert Elias Delta, serta melaksanakan dekompresi terhadap hasil kompresi dengan algoritma Fibonacci serta algoritma Invert Elias Delta jadi file asli.Dari hasil ujicoba perbandingan kompresi file audio antara algoritma Fibonacci serta algoritma Invert Elias Delta mempunyai hasil berbeda, dalam pelaksanaan Tata Cara Perbandingan Eksponential dengan 3 kriteria ialah Ratio of Compression (RC), Compression Ratio (CR), Ruang Saving (SS) dengan hasil akhir algoritma Fibonacci 11,074 sebaliknya algoritma Invert Elias Delta 10,217. Sehingga semakin besar total nilai yang diperoleh maka semakin besar pula jumlah usaha yang dicoba oleh algoritma tersebut.

REFERENCES

[1] S. D. Nasution, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Goldbach Codes," *J. Ilm. INFOTEK*, vol. 1, no. 1, pp. 104–109, 2013.

[2] A. Tanjung, "Perbandingan Kinerja Algoritma Elias Delta Code Dan Algoritma Punctured Elias Code Dalam Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 182–190, 2023.

[3] K. M. R. Hutahaean, "Penerapan Algoritma Inverted Elias Delta Untuk Kompresi Konten Pada Aplikasi Psikologi Berbasis Android," *J. Ris. Tek. Inform. dan Data Sains*, vol. 1, no. 1, pp. 26–37, 2022.

[4] D. Anggraini, "Penerapan Algoritma Fibonacci Code Pada File Transfer Berbasis Android," *J. Informatics Manag. Inf. Technol.*, vol. 1, no. 3, pp. 91–94, 2021.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**