Bulletin of Information System Research (BIOS)
Volume 2, No 1, December 2023 Page: 30-40
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Analisa Perbandingan Kinerja Algoritma Arithmetic Coding Dan Fibonacci Codes Dalam Kompresi File Audio Menggunakan Metode Perbandingan Eksponensial

Arien Hotma Rotua Nababan

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Teknik Informatika, Universitas Budi darma, Medan, Indonesia
Email: arienHRN02@gmail.com

Abstrak−Kompresi merupakan proses mengubah sebuah aliran data input menjadi aliran data baru yang memiliki ukuran data yang lebih kecil dengan tujuan untuk merepresentasikan suatu data digital dengan sesedikit mungkin bit, tetapi tetap mempertahankan kebutuhan minimum untuk membentuk kembali data aslinya. Sebuah format file audio adalah format file untuk menyimpan data audio digital pada sistem komputer. Penyimpanan data tersebut, tidak bisa hanya dilakukan dengan format file yang tersedia. Maka dari itu diperlukan proses kompresi file. Arithmetic coding memiliki tingkat kompresi yang lebih baik atau minimal sama dengan Huffman coding, tetapi waktu kompresi dan dekompresinya lambat karena banyaknya proses perhitungan yang harus dilakukan. Arithmetic coding suatu bagian dari Entropy Encoding yang mengkonversi suatu data ke dalam bentuk data yang lain dengan lebih sering menggunakan sedikit bit dan jarang menggunakan lebih banyak bit karakter. Dalam penelitian sebelumnya teknik pengkodean ini memisahkan pesan masukan ke dalam simbol dan menukar masing–masing simbol dengan suatu floatingpoint. Fibonacci coding merupakan universal code yang dapat mengkodekan integer positif menjadi kode biner dalam bentuk code word. Dimana proses dari kompresi menggunakan Fibonacci code dilakukan dengan cara pembacaan file secara byte kemudian mengubah nilai byte menjadi suatu bilangan bulat dilanjutkan dengan mencari code word. Metode Perbandingan Exponensial (MPE) adalah salah satu metode dari Decision Suport System (DSS) yang digunakan untuk menentukan urutan prioritas alternatif dengan kriteria jamak. MPE sangat cocok untuk penilaian skala ordinal. Metode perbandingan eksponensial mempunyai keuntungan dalam mengurangi bias yang mungkin terjadi dalam analisis. Nilai skor yang menggambarkan urutan prioritas menjadi besar (fungsi eksponensial) ini mengakibatkan urutan prioritas alternatif keputusan lebih nyata.

Kata Kunci: Kompresi; File Audio; Arithmetic Coding; Fibonacci Codes; MPE

Abstract−Compression is the process of changing an input data stream into a new data stream that has a smaller data size with the aim of representing digital data with as few bits as possible, but still maintaining the minimum need to reshape the original data. An audio file format is a file format for storing digital audio data on a computer system. Storing this data cannot only be done using the available file formats. Therefore, a file compression process is needed. Arithmetic coding has a compression level that is better or at least the same as Huffman coding, but the compression and decompression times are slow because of the large number of calculation processes that must be carried out. Arithmetic coding is a part of Entropy Encoding which converts data into another form of data by using fewer bits more often and rarely using more character bits. In previous research, this coding technique separated the input message into symbols and exchanged each symbol with a floating point. Fibonacci coding is a universal code that can encode positive integers into binary code in the form of code words. Where the compression process using the Fibonacci code is carried out by reading the file in bytes then changing the byte value into an integer followed by searching for the code word. The Exponential Comparison Method (MPE) is one of the methods of the Decision Support System (DSS) which is used to determine the priority order of alternatives with multiple criteria. MPE is very suitable for ordinal scale assessment. The exponential comparison method has the advantage of reducing bias that may occur in the analysis. The score value that describes the order of priorities becomes large (exponential function) resulting in a more real priority order of decision alternatives.

Keywords: Compression; Audio Files; Arithmetic Coding; Fibonacci Code; MPE

1. PENDAHULUAN

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. Data yang dimaksud berupa text, gambar, audio, dan kombinasi dari ketiganya, seperti video.

Salah satu file yang memegang peranan penting sebagai bentuk informasi yaitu file audio. Audio (suara) adalah fenomena fisik yang dihasilkan oleh getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu yang disebut frekuensi. Sebuah format file audio adalah format file untuk menyimpan data audio digital pada sistem komputer. Penyimpanan data tersebut, tidak bisa hanya dilakukan dengan format file yang tersedia. Maka dari itu diperlukan proses kompresi file.

Kompresi (pemampatan) merupakan proses mengubah sebuah aliran data input menjadi aliran data baru yang memiliki ukuran data yang lebih kecil dengan tujuan untuk merepresentasikan suatu data digital dengan sesedikit mungkin bit, tetapi tetap mempertahankan kebutuhan minimum untuk membentuk kembali data aslinya. Dalam kompresi data, terdapat 4 (empat) factor penting yang perlu diperhatikan, yaitu: time process (waktu yang dibutuhkan dalam menjalankan proses), completeness (kelengkapan data setelah dikompresi), ratio compress (ukuran data setelah dikompresi), optimality (perbandingan ukuran data sebelum dikompresi dengan yang telah

dikompresi)[1]–[5]. Kompresi memiliki beberapa jenis algoritma untuk mengkompresi file, diantaranya adalah algoritma arithmetic coding dan Fibonacci codes.

Arithmetic coding memiliki tingkat kompresi yang lebih baik atau minimal sama dengan Huffman coding, tetapi waktu kompresi dan dekompresinya lambat karena banyaknya proses perhitungan yang harus dilakukan. Arithmetic coding suatu bagian dari Entropy Encoding yang mengkonversi suatu data ke dalam bentuk data yang lain dengan lebih sering menggunakan sedikit bit dan jarang menggunakan lebih banyak bit karakter. Dalam penelitian sebelumnya teknik pengkodean ini memisahkan pesan masukan ke dalam simbol dan menukar masing–masing simbol dengan suatu floatingpoint. Fibonacci coding merupakan universal code yang dapat mengkodekan integer positif menjadi kode biner dalam bentuk code word. Dimana proses dari kompresi menggunakan Fibonacci code dilakukan dengan cara pembacaan file secara byte kemudian mengubah nilai byte menjadi suatu bilangan bulat dilanjutkan dengan mencari code word[6]–[9].

Metode Perbandingan Exponensial (MPE) adalah salah satu metode dari Decision Suport System (DSS) yang digunakan untuk menentukan urutan prioritas alternatif dengan kriteria jamak. MPE sangat cocok untuk penilaian skala ordinal. Metode perbandingan eksponensial mempunyai keuntungan dalam mengurangi bias yang mungkin terjadi dalam analisis. Nilai skor yang menggambarkan urutan prioritas menjadi besar (fungsi eksponensial) ini mengakibatkan urutan prioritas alternatif keputusan lebih nyata.

Penelitian ini menguraikan analisa kinerja algoritma arithmetic coding dan Fibonacci codes dalam kompresi file audio untuk mengetahui perbandingan kinerja algoritma arithmetic coding dan Fibonacci codes dalam kompresi file audio. Agar proses perbandingan kedua metode dapat dianalisa, maka metode perbandingan eksponensial digunakan dalam menganalisa perbandingan kinerja algoritma *arithmetic coding* dan *Fibonacci codes*..

2. METODE PENELITIAN

2.1 Metodologi Penelitian

Metode yang digunakan penulis dalam menyelesaikan penelitian ini adalah:

1. *Library research* (Studi Pustaka)
   Yaitu pencarian pustaka yang berhubungan dengan algoritma pemapatan (kompresi) file audio.
2. Analisa
   Pada tahap ini dilakukan analisa perbandingan kinerja algoritma *arithmetic coding* dan *Fibonacci codes* dalam kompresi file audio menggunakan metode perbandingan eksponensial.
3. Pengujian
   Dalam tahap ini dilakukan pengujian terhadap sistem kinerja algoritma arithmetic coding dan Fibonacci codes yang telah dibangun.
4. Implementasi
   Pada tahap ini dilakukan implementsi terhadap hasil analisa perbandingan kinerja algoritma arithmetic coding dan Fibonacci codes dalam kompresi file audio menggunakan metode perbandingan eksponensial.
5. Dokumentasi
   Pada tahap ini dilakukan dokumentasi, yaitu penyusunan laporan dari penelitian yang dilakukan membuat laporan hasil analisa kinerja algoritma *arithmetic coding* dan *Fibonacci code*s dalam kompresi file audio menggunakan metode perbandingan eksponensial.

2.2 Kompresi

Proses kompresi merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi digital yang padat atau mampat (compact) namun tetap dapat mewakili kuantitas informasi yang terkandung pada data tersebut. Pada citra, audio, dan video, kompresi mengarah pada minimisasi jumlah bit rate untuk representasi digital[10]–[14]. Data dan informasi adalah dua hal yang berbeda. Pada data terkandung suatu informasi. Namun tidak semua bagian data terkait dengan informasi tersebut atau pada suatu data terdapat bagian-bagian data yang berulang untuk mewakili informasi yang sama. Bagian data yang tidak terkait atau bagian data yang berulang tersebut disebut dengan data berlebihan (redundancy data). Tujuan daripada kompresi data tiada lain adalah untuk mengurangi data berlebihan tersebut sehingga ukuran data menjadi lebih kecil dan lebih ringan dalam proses transmisi[15]–[17].

Kompresi data menjadi sangat penting karena memperkecil kebutuhanpenyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan bandwidth. Teknik kompresi bisa dilakukan terhadap data teks/biner, gambar(JPEG, PNG, TIFF), audio (MP3, AAC, RMA, WMA), dan video (MPEG, H261, H263). Pada suatu teknik yang digunakan dalam proses kompresi data terdapat beberapa faktor yang bisa digunakan untuk menganalisa kualitas dari suatu teknik kompresi data tersebut, yaitu[18]–[22]:

1. Ratio of Compression (RC)
   Ratio of compression adalah nilai perbandingan antara data sebelum dikompresi dengan data yang telah dikompresi. Secara matematis dapat dituliskan sebagai berikut :

$$\text{RC} = \frac{\text{Ukuran Data Sebelum Dikompresi}}{\text{Ukuran Data Setelah Dikompresi}} \quad (1)$$

2. Compression Ratio (CR)
   Compression ratio adalah persentase besar data terkompresi, hasil perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi. Secara matematis dapat dituliskan sebagai berikut :

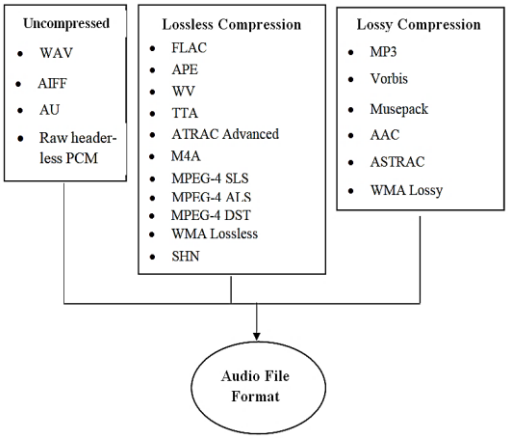
$$\text{CR} = \frac{\text{Ukuran Data Sebelum Dikompresi}}{\text{Ukuran Data Setelah Dikompresi}} \text{x } 100\% \quad (2)$$

3. Space Savings (SS)
   Space savings adalah persentase selisih antara data yang belum dikompresi dengan besar data yang dikompresi. Secara matematis dapat dituliskan sebagai berikut :

$$SS = 1 - Cr \quad (3)$$

2.3 File Audio

Terdapat tiga jenis format file audio berdasarkan sistem penyimpanannya, yaitu: *uncompressed, lossless compression*, dan *lossy compression.* Audio *uncompressed* disimpan sebagaimana adanyasesuai format *raw data*. *Lossless* dan *lossy compression*mengalami proses kompresi sebelum disimpan. Perbedaan antara kompresi *lossless* dan *lossy* adalah padahasil kompresinya; kompresi *lossless* menghasilkan duplikatdari alur audio asli dengan rasio kompresi antar 50 sampai 60 persen, kompresi *lossy* menghasilkan alur audio dengan rasiokompresi lebih tinggi namun dengan akurasi lebih rendah[14][23][24].



Gambar 1. Klasifikasi Format File Audio Berdasarkan Metode Kompresinya.

2.4 Algoritma arithmetic coding

Algoritma *arithmetic coding* adalah suatu bagian dari *Entropy Encoding* yang mengkonversi suatu data ke dalam bentuk data yang lain dengan lebih sering menggunakan sedikit *bit* dan jarang menggunakan lebih banyak *bit* karakter. Teknik aritmatika membuat *code word* untuk setiap simbol. *Code word* pada metode aritmatika berada dalam suatu interval bilangan antara 0 dan 1. Interval ini diberikan atas dasar probabilitas kehadiran setiap simbol. Batas atas dan batas bawah dari interval tersebut merupakan parameter yang sangat penting untuk proses *encoding* dan *decoding*. Semakin tinggi frekuensi kemunculan suatu karakter maka semakin luas interval nilainya dan semakin sedikit representasi *bit* yang dibutuhkan untuk mewakili nilai tersebut. Sebaliknya semakin rendah frekuensi kemunculan karakter maka semakin pendek intervalnya dan representasi *bit* yang dibutuhkan semakin besar[25][26][27].

Metode aritmatika diawali dengan pembentukan tabel probabilitas setiap simbol. Berdasarkan tersebut, batas atas dan batas bawah interval setiap simbol akan dibuat. Untuk melakukan proses *encoding* algoritma berikut dipakai:

1. Set low = 0,0 (kondisi awal)
2. Set high = 1.0 (kondisi awal)
3. While (simbol input masih ada) do
4. Ambil simbol input.
5. CR = high – low.
6. High = low + CR*high_range(simbol)
7. Low = low + CR*low_range(simbol)
8. End while

Pada algoritma diatas *low*, *high*, dan CR berturut-turut menyatakan batas bawah, batas atas, dan jangkauan interval dari setiap simbol. Pada tahap awal, LOW dan HIGH diinisialisasi dengan nilai 0 dan 1 kemudian pada proses selanjutnya kedua nilai tersebut diperbarui dengan rumus:

HIGH = LOW + CR * high range of CHARACTER
LOW = LOW + CR * low range of CHARACTER

Secara umum langkah–langkah yang dilakukanuntuk kompresi file audio dengan metode *Arithmetic Coding* adalah sebagai berikut :

1. Buka file audio untuk membaca header-header dan sample audio.
2. Baca file audio untuk mendapatkan data sample.
3. Ambil nilai sample audio ke 1 sampai ke n.
4. Susun nilai sample audio pada senarai berantai dengan karakter khusus pembatas antara data sample audio (#).
5. Mencari probabilitas masing-masing data dengan cara membagi jumlah data dengan jumlah total data.
6. Selanjutnya akan diperoleh tabel range probabilitas dengan menjumlahkan nilai range terendah data xi dengan probabilitas data xi.
7. Setelah itu menentukan code range (CR), kemudian menentukan nilai high(xi) dan low(xi).

2.5 Fibonacci Codes

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[28][8][9]:
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.
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.

Untuk melakukan *decode* sebuah kode Fibonacci, hilangkan angka “1” paling kanan, kemudian substitusi dan jumlahkan kode Fibonacci yang tersisadenganmenggunakan deret Fibonacci.

2.6 Metode Perbandingan Exponensial

Metode Perbandingan Exponensial (MPE) merupakan salah satu metode untuk menentukan urutan prioritas alternative keputusan dengan kriteria jamak. Dengan perhitungan secara eksponensial, perbedaan nilai antara kriteria dapat dibedakan tergantung kepada kemampuan orang yang menilai. Teknik ini digunakan sebagai pembantu bagi individu pengambilan keputusan untuk menggunakan rancang bangun model yang telah terdefinisi dengan baik pada tahapan proses. Untuk menggunakan metode MPE terdapatbeberpa langkah. Berikut ini adalah langkah-langkahyang perlu dilakukan dalam pemilihan keputusandengan menggunakan MPE adalah[29][30]:

1. Menyusun alternatif-alternatif keputusan yang akan dipilih.
2. Menentukan kriteria atau perbandingan relative kriteria keputusan yang penting untuk di evaluasidengan menggunkan skala konversi tertentusesuai dengan keinginan pengambil keputusan.
3. Menentukan tingkat kepentingan relatif darisetiap kriteria keputusan atau pertimbangankriteria. Penentuan bobot di tetapkan pada setiapkriteria untuk menunjukan tingkat kepentingansuatu kriteria.
4. Melakukan penilaian terhadap semua alternatif pada tiap kriteria dalam bentuk total skor tiapalternatif.
5. Menghitung skor atau nilai total setiap alternatif dan mengurutkannya. Semakin besar Total Nilai(TN) alternatif maka semakin tinggi urutanprioritasnya.Formulasi penghitungan MetodePerbandingan Eksponensial:

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

Keterangan:

TNi	= Total nilai alternatif ke-i
RKij	=Derajat kepentingan relatif kriteria ke-j pada pilihan keputusan ke-i
TKKj	= Derajat kepentingan kriteria keputusan ke-j TKKJ>0;bulat
n	= Jumlah pilihan keputusan
m	= Jumlah kriteria keputusan

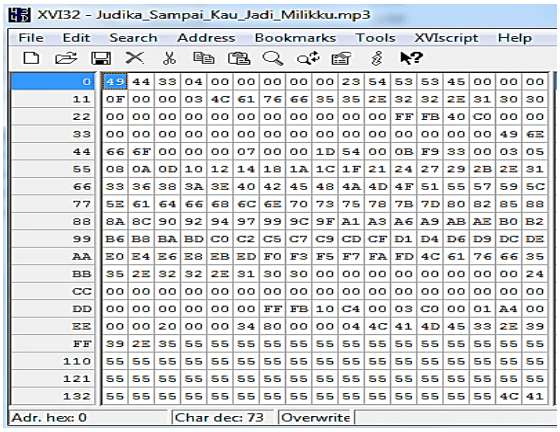
3. HASIL DAN PEMBAHASAN

Dalam sistem ini terdapat tiga proses utama, yaitu proses kompresi file audio mengunakan algoritma *arithmetic coding*, *Fibonacci codes* dan proses menghitung perbandingan rasio kompresi.

Bulletin of Information System Research (BIOS)
Volume 2, No 1, December 2023 Page: 30-40
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

3.1 Proses Kompresi File Audio Menggunakan Arithmetic coding

Struktur data pada file audio berbeda-beda tergantung format audionya, pada contoh berikut akan dikompresi file audio MP3 dengan ukuran 766 KB. Dari sampel MP3 didapat nilai hexadecimal menggunakan bantuan *software hex editor* seperti pada gambar dibawah ini:



Gambar 2. Nilai Hexadesimal File MP3

Berdasarkan pada gambar diatas didapat nilai hexadecimal file MP3. Untuk keperluan hitungan manual hanya diambil sampel nilai sebanyak 16 karakter nilai hexadecimal file MP3. Nilai hexadecimal diambil dari sisi kiri atas sampai dengan bilangan ke 16. Adapun bilangan hexadesimal file MP3 sampel tersebut adalah 49, 44, 33, 04, 00, 00, 00, 00, 00, 23, 54, 53, 53, 45, 00, 00. Langkah awal dari proses kompresi terlebih dahulu yaitu mencari probabilitas masing-masing data dengan cara membagi jumlah frekuensi bilangan hexadesimal dengan jumlah total frekuensi bilangan hexadesimal.

Tabel 1. Tabel Probabilitas

No	Nilai	Frekuensi	Probabilitas
1	49	1	1/16=0,06
2	44	1	1/16=0.06
3	33	1	1/16=0,06
4	04	1	1/16=0,06
5	0	7	7/16=0,43
6	23	1	1/16=0,06
7	54	1	1/16=0,06
8	53	2	2/16=0,12
9	45	1	1/16=0,06

Selanjutnya akan diperoleh tabel *range* probabilitas dengan menjumlahkan range terendah data dan probabilitas data.

Tabel 2. Range Probabilitas

No	Nilai	Frekuensi	Probabilitas	Range
1	49	1	1/16=0,06	0,00 < 49 > 0,06
2	44	1	1/16=0.06	0,06 < 44 >0,12
3	33	1	1/16=0,06	0,12 < 33 >0,24
4	04	1	1/16=0,06	0,24 < 04 >0,3
5	0	7	7/16=0,43	0,3 < 0 > 0,73
6	23	1	1/16=0,06	0,73 < 23 >0,79
7	54	1	1/16=0,06	0,79 < 54 >0,85
8	53	2	2/16=0,12	0,85 < 53 >0,97
9	45	1	1/16=0,06	0,97 < 45 >1,03

Selanjutnya mencari nilai Cr (Xi) atau nilai jarak interval karakter. Data yang diproses adalah data yang range awalnya adalah 0. Data yang pertama diolah adalah 49 dengan nilai low (X1) = 0,0 dan high (X1) =0,06. Setelah menentukan Cr (X2) = high (X1) – low (X1) =0,06 kemudian menentukan nilai high (X2) dan low (X2), yaitu.

Low(X2) = low(X1) + cr(X2) * low(X2)
= 0,0 + 0,06 * 0,06 = 0,0036

high(X2) = low(X1) + cr(X2) * high(X2)
= 0,0 + 0,06 * 0,12 = 0,0072

Bulletin of Information System Research (BIOS)
Volume 2, No 1, December 2023 Page: 30-40
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Low(X3) = low(X2) + cr(X3) * low(X3)
= 0,06 + 0,06 * 0,12 = 0,0672

high(X3) = low(X2) + cr(X3) * high(X3)
= 0,06 + 0,06 * 0,24 = 0,0744

Low(X4) = low(X3) + cr(X4) * low(X4)
= 0,12 + 0,12 * 0,24 = 0,1488

high(X4) = low(X3) + cr(X4) * high(X4)
= 0,12 + 0,12 * 0,3 = 0,156

Low(X5) = low(X4) + cr(X5) * low(X5)
= 0,24 + 0,06 * 0,3 = 0,258

high(X5) = low(X4) + cr(X5) * high(X5)
= 0,24 + 0,06 * 0,73 = 0,2838

Low(X6) = low(X5) + cr(X6) * low(X6)
= 0,3 + 0,43 * 0,73 = 0,6139

high(X6) = low(X5) + cr(X6) * high(X6)
= 0,3 + 0,43 * 0,79 = 0,6397

Low(X7) = low(X6) + cr(X7) * low(X7)
= 0,73 + 0,06 * 0,79 = 0,7774

high(X7) = low(X6) + cr(X7) * high(X7)
= 0,73 + 0,06 * 0,85 = 0,781

Low(X8) = low(X7) + cr(X8) * low(X8)
= 0,79 + 0,06 * 0,85 = 0,841

high(X8) = low(X7) + cr(X8) * high(X8)
= 0,79 + 0,06 * 0,97 = 0,8482

Low(X9) = low(X8) + cr(X9) * low(X9)
= 0,85 + 0,12 * 0,97 = 0,9664

high(X9) = low(X8) + cr(X9) * high(X9)
= 0,85 + 0,12 * 1,03 = 0,9736

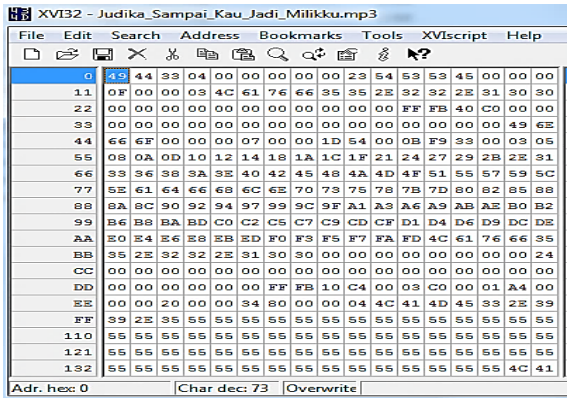
Tabel 3. Hasil *encoding* data sampel

No	Nilai	Low	High
1	49	0	0,06
2	44	0,0036	0,0072
3	33	0,0672	0,0744
4	4	0,1488	0,156
5	0	0,258	0,2838
6	23	0,6139	0,6397
7	54	0,7774	0,781
8	53	0,841	0,8482
9	45	0,9664	0,9736

Tabel 3 menunjukkan bahwa nilai *low* dan untuk data terakhir adalah nilai batas bawah yang akan digunakan untuk menggantikan sampel audio yang telah di-*encoding* yaitu nilai sampel audio 49, 44, 33, 04, 00, 00, 00, 00, 00, 23, 54, 53, 53, 45, 00, 00 diubah menjadi 0,9664 yang ketika dikonversi ke biner maka menjadi 0,10010111000000.

3.2. Proses Kompresi File Audio Menggunakan Fibonacci codes

Struktur data pada file audio berbeda-beda tergantung format audionya, pada contoh berikut akan dikompresi file audio dengan ukuran 766 KB. Dari sampel MP3 didapat nilai hexadecimal menggunakan bantuan *software hex editor* seperti pada gambar dibawah ini:



Gambar 3. Nilai Hexadesimal File MP3

Berdasarkan pada gambar diatas didapat nilai hexadecimal file MP3. Untuk keperluan hitungan manual hanya diambil sampel nilai sebanyak 16 karakter nilai hexadecimal file MP3. Nilai hexadecimal diambil dari sisi kiri atas sampai dengan bilangan ke 16. Adapun bilangan hexadesimal file MP3 sampel tersebut adalah 49, 44, 33, 04, 00, 00, 00, 00, 00, 23, 54, 53, 53, 45, 00, 00. Berdasarkan input nilai hexadecimal file audio MP3 tersebut maka setiap nilai akan dipetakan kedalam tabel frekuensi. Kemunculan data yang digambarkan dalam tabel 3.4.

Tabel 4. frekuensi data

Nilai	Biner	Bit	Frekuensi	Frekuensi*Bit
49	01001001	8	1	8
44	01000100	8	1	8
33	00110011	8	1	8
04	00000100	8	1	8
00	0	0	7	0
23	00100011	8	1	8
54	01010100	8	1	8
53	01010011	8	2	16
45	01000101	8	1	8
Total			16	72

Langkah-langkah pembentukan sebuah kode fibonacci untuk nilai sampel 49, 44, 33, 04, 00, 00, 00, 00, 00, 23, 54, 53, 53, 45, 00, 00 adalah sebagai berikut:
Nilai n = 49 : Bilangan bulat positif.
Deret Fibonacci yang mendekati nilai n adalah :
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
Proses pengurangan nilai n dengan setiap bilangan Fibonacci sampai sisa pengurangan nilai n adalah 0 :

1. 49 – 34 = 15 : Nilai n dikurangkan dengan bilangan fibonacci terbesar yang lebih kecil atau sama dengan nilai n,
   dan catat sisa pengurangan.
2. 15 – 13 = 2 : Tukarkan nilai n dari sisa pengurangan sebelumnya, lalu kurangkan nilai n dengan bilangan fibonacci yang lebih kecil atau sama dengan nilai n, dan catat sisa pengurangan.
3. 2 – 2 = 0 : Tukarkan nilai n dari sisa pengurangan sebelumnya, lalu kurangkan nilai n dengan bilangan fibonacci yang lebih kecil atau sama dengan nilai n.

Tabel 5. Proses pembentukan kode fibonacci nilai 49

Urutan Bilangan Fibonacci	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34
Kode Fibonacci Sementara	-	-	0	1	0	0	0	1	0	1

Keterangan :
Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret Fibonacci F(i), tambahkan angka “1” dan bilangan deret Fibonacci F(i) yang tidak dikurangkan tambahkan angka “0”.
Untuk nilai n berikutnya dilakukan dengan langkah-langkah diatas.
Nilai n = 44. Deret Fibonacci yang mendekati nilai n adalah :
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. 44 – 34 = 10
2. 10 – 8 = 2
3. 2 – 2 = 0

Tabel 6. Proses pembentukan kode fibonacci nilai 44

Urutan Bilangan Fibonacci	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34
Kode Fibonacci Sementara	-	-	0	1	0	0	1	0	0	1

Nilai n = 33. Deret Fibonacci yang mendekati nilai n adalah :
0, 1, 1, 2, 3, 5, 8, 13, 21
Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. 33 – 21 = 12
2. 12 – 8 = 4
3. 4 – 3 = 1
4. 1 – 1 = 0

Bulletin of Information System Research (BIOS)
Volume 2, No 1, December 2023 Page: 30-40
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Tabel 7. Proses pembentukan kode fibonacci nilai 33

Urutan Bilangan Fibonacci	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	-
Kode Fibonacci Sementara	-	-	1	0	1	0	1	0	1	-

Lakukan pencarian kode fibonacci hingga nilai 45 dengan melakukan proses seperti tabel 5, 6 dan 7. Berikut tabel 8 hasil akhir kode fibonacci setelah dialkukan perhitungan terhadap semua nilai.

Table 8. Hasil akhir kode fibonacci

Nilai	Fibonacci Codes	Bit	Frekuensi	Frekuensi*Bit
49	01000101	8	1	8
44	01001001	8	1	8
33	1010101	7	1	7
04	101	3	1	3
00	0	0	7	0
23	0100001	7	1	7
54	01010101	8	1	8
53	10010101	8	2	16
45	00101001	8	1	8
	Total		16	65

3.3 Analisa Proses Perhitungan Parameter Pembanding

1. Ratio of Compression (RC)

   RC Arithmetic Coding = $\frac{72}{75}$ = 0, 96

   RC Fibonacci Codes = $\frac{72}{65}$ = 1, 10

2. Compression Ratio (CR)

   CR Arithmetic Coding = $\frac{75}{72}$ x 100% = 1, 04%

   CR Fibonacci Codes = $\frac{65}{72}$ x 100% = 90, 27%

3. Space Saving (SS)

   SS Arithmetic Coding = 100% - 1, 04% = 98, 96%

   SS Fibonacci Codes = 100% - 90, 27% = 9, 73%

3.4 Analisa Metode Perbandingan Eksponensial

Langkah-langkah yang dilakukan dalam pemilihan keputusan dengan menggunakan MPE adalah :

1. Menentukan alternatif
   Untuk menganalisa perbandingan kecepatan antara algoritma arithmetic coding dengan fibonacci codes dalam kompresi file audio, adapun yang menjadi alternatif adalah algoritma arithmetic coding dan fibonacci codes.
2. Menentukan kriteria
   Untuk dapat membandingkan kedua alternatif diatas maka selanjutnya yang perlu dilakukan adalah menentukan kriteria dalam analisa ini, secara rinci kriteria dijelaskan pada tabel berikut :

Tabel 9. Penentuan Kriteria

No	Kriteria	Keterangan
1	Jumlah Time Proses	Jumlah waktu yang dikerjakan dalam algoritma
2	Jumlah Data	Jumlah data yang diproses

3. Menentukan bobot kriteria
   Penentuan bobot merupakan salah satu komponen yang sangat berpengaruh terhadap hasil analisa.

Tabel 10. Penentuan Bobot Kriteria

No	Kriteria	Bobot
1	Jumlah Time Proses	3
2	Jumlah Data	6

4. Pemberian nilai pada setiap kriteria

Bulletin of Information System Research (BIOS)
Volume 2, No 1, December 2023 Page: 30-40
ISSN 2963-2455 (media online)
https://journal.grahamitra.id/index.php/bios

Tahap ini adalah tahap dimana setiap kriteria yang telah terbentuk diberi nilai. Untuk dapat memberikan nilai, berikut adalah contoh hasil simulasi kompresi data yang diambil dari pembahasan analisa algoritma arithmetic coding dan fibonacci codes sebelumnya.

Tabel 11. Pemberian Nilai Pada Setiap Kriteria

Alternatif	Proses ke-	Kriteria	
		Jumlah Time Proses	Jumlah Data
Algoritma Arithmetic coding	1	3	8
	2	4	8
Algoritma Fibonacci Codes	1	5	7
	2	6	7

5. Menghitung skor
Setelah nilai pada setiap kriteria dimasukkan, maka tahapan selanjutnya adalah melakukan perhitungan dengan menggunakan rumus metode perbandingan eksponensial.

Perhitungan Nilai pada proses 1 :

Nilai Arithmetic Coding = $(3)^3 + (8)^3 = 539$

Nilai Fibonacci Codes = $(5)^3 + (7)^3 = 468$

Perhitungan Nilai pada proses 2 :

Nilai Arithmetic Coding = $(4)^6 + (8)^6 = 262.400$

Nilai Fibonacci Codes = $(6)^6 + (7)^6 = 164.305$

Tabel 12. Perhitungan analisa menggunakan MPE

Proses ke-	Kriteria						Total Nilai Arithmetic coding	Total Nilai Fibonacci Codes
	Jumlah Time Proses			Jumlah Data				
	B	Arithmetic coding	Fibonacci codes	B	Arithmetic coding	Fibonacci codes		
		N	N		N	N		
1	3	3	5	3	8	7	539	468
2	6	4	6	6	8	7	262.400	164.305
Total Nilai							262.939	164.773

Keterangan :
Arithmetic Coding = Algoritma Arithmetic Coding
Fibonacci Codes = Algoritma Fibonacci Codes
B = Nilai Bobot
N = Nilai dari Kriteria

6. Menentukan Prioritas Keputusan
Setelah total nilai dari setiap alternatif dihitung maka yang tahapan selanjutnya adalah tahapan terakhir yaitu menentukan prioritas keputusan berdasarkan total nilai dari setiap alternatif.

Tabel 13. Prioritas Keputusan

Alternatif	Total Nilai	Ranking
Algoritma Arithmetic coding	262.939	2
Algoritma Fibonacci codes	164.773	1

Pada tabel diatas terlihat bahwa total nilai dari alternatif terendah yang memperoleh rangking pertama, hal ini dikarenakan semakin tinggi total nilai yang diperoleh maka akan semakin tinggi jumlah usaha yang dilakukan oleh algoritma tersebut dalam kompresi data. Berdasarkan analisa tersebut maka algoritma fibonacci codes yang menjadi algoritma tercepat dalam penelitian kompresi data, serta permasalahan dalam penelitian ini dapat terpecahkan.

4. KESIMPULAN

Berdasarkan hasil penelitian tentang analisa perbandingan kinerja algoritma arithmetic coding dan fibonacci codes dalam kompresi file audio dengan menggunakan metode perbandingan eksponensial (MPE) maka penulis menyimpulkan bahwa pengujian yang dilakukan terhadap file audio menunjukkan Compression Ratio (CR), Ratio of Compression (RC), dan Space Saving (SS) yang dihasilkan dari kompresi algoritma fibonacci codes lebih baik dari algoritma arithmetic coding. Untuk dapat menganalisa perbandingan kinerja algoritma arithmetic coding dan

fibonacci codes , nilai perbandingannya dihitung dengan menggunakan metode perbandingan eksponensial (MPE). Dengan adanya total nilai perbandingan dari metode perbandingan eksponensial (MPE) yang dilakukan pada penelitian ini maka algoritma fibonacci codes merupakan algoritma yang tercepat dalam melakukan kompresi file audio dengan ekstensi MP3 (*.mp3).

REFERENCES

[1] A. Yasir and B. S. Hasugian, "Penggunaan Teknik Kompresi Jpeg Dalam Perancangan Kompresi Citra Digital Memakai Fungsi Gui Pada Matlab," *War. Dharmawangsa*, vol. 16, no. 4, pp. 1056–1066, 2022.
[2] C. A. N. Ginting, "Penerapan Algoritma Sequitur Pada Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 333–339, 2023.
[3] A. Nur, H. Yuana, and F. Febrinita, "Aplikasi Kompresi Citra dengan Menggunakan Algoritma Lempel Ziv Welch (LZW)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 684–695, 2022.
[4] D. Cahayati, A. M. H. Pardede, and H. Khair, "Implementasi Algoritma Elias Gamma Kompresi Pada File Teks," *Algoritm. J. ILMU Komput. DAN Inform.*, vol. 6, no. 1, 2022.
[5] A. N. Purnama, "Implementasi Algoritma Rice Code dan Ternary Comma Code Pada Kompresi File PDF," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 34–42, 2023.
[6] S. Karo-Karo, C. S. Daeli, and M. R. Manalu, "Application Of Coding Arithmetic Methods And Methods Lzw For Compression Files," *J. Tek. Indones.*, vol. 2, no. 01, pp. 1–5, 2023.
[7] N. H. Hussein and M. A. Ali, "Medical Image Compression and Encryption Using Adaptive Arithmetic Coding, Quantization Technique and RSA in DWT Domain," *Iraqi J. Sci.*, pp. 2279–2296, 2022.
[8] J. Martina and B. Panjaitan, "Penerapan Algoritma Fibonacci Codes Pada Kompresi Aplikasi Audio Mp3 Berbasis Dekstop," vol. 1, no. 1, pp. 27–33, 2021.
[9] 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.
[10] S. N. Ritonga, "Implementasi Algoritma Golmb Code Dalam Kompresi File Video," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 365–373, 2023.
[11] S. Simanjuntak, "Implementasi Metode Taboo Code Untuk Kompresi File Video," *Explorer (Hayward).*, vol. 2, no. 1, pp. 32–38, 2022.
[12] M. C. Aruan and W. Rahayu, "Analisis Performa Algoritma Kompresi Data dalam Penyimpanan dan Transfer Data," *LANCAH J. Inov. dan Tren*, vol. 1, no. 2, pp. 228–232, 2023.
[13] A. Pratiwi, "Perancangan Aplikasi Kompresi File Audio Dengan Menerapkan Algoritma Additive Code," *J. Glob. Technol. Comput.*, vol. 1, no. 3, pp. 92–100, 2022.
[14] C. B. Simbolon, "Penerapan Metode Self Delimiting Codes Dalam Kompresi File," *J. Comput. Informatics Res.*, vol. 1, no. 2, pp. 50–55, 2022.
[15] E. Suryana, A. Lestari, and K. Harto, "Teori Pemrosesan Informasi Dan Implikasi Dalam Pembelajaran," *J. Ilm. Mandala Educ.*, vol. 8, no. 3, 2022.
[16] I. Nola, "Penerapan Metode Delta Modulation dalam Kompresi File Video Mp4," *J. Comput. Informatics Res.*, vol. 1, no. 2, pp. 29–33, 2022.
[17] T. F. Silaban, "Kompresi File Teks dengan Menggunakan Kombinasi Squitur dan Elias Gamma Code," vol. 1, no. 3, pp. 82–87, 2022, doi: 10.47065/comforch.v1i3.346.
[18] R. Sihotang, "Perancangan Aplikasi Kompresi File Teks Dokumen Menggunakan Kombinasi Algoritma Bijective Burrows-Wheeler Transform (BBWT) Dan Algoritma Stout Code," *Inf. dan Teknol. Ilm.*, vol. 9, no. 3, pp. 60–63, 2022, [Online]. Available: http://www.stmik-budidarma.ac.id/ejurnal/index.php/inti/article/view/4638%0Ahttp://www.stmik-budidarma.ac.id/ejurnal/index.php/inti/article/download/4638/2864
[19] D. Asdini and D. P. Utomo, "Analisis Perbandingan Kinerja Algoritma Huffman Dan Algoritma Levenstein Dalam Kompresi File Dokumen Format . RTF," vol. 6, no. November, pp. 87–99, 2022, doi: 10.30865/komik.v6i1.5739.
[20] K. Y. Sarumaha, M. Syahrizal, and E. R. Siagian, "Analisis Perbandingan Algoritma Elias Delta Code Dengan Algoritma Prefix Code Dalam Mengkompresi Data Teks," vol. 6, no. November, pp. 460–469, 2022, doi: 10.30865/komik.v6i1.5698.
[21] Y. Sihura, T. Zebua, and H. Hutabarat, "Kinerja Algoritma Yamamoto ' s Recursive Code dan Algoritma Fix ed Length Binary Encoding pada Kompresi File PDF," vol. 3, no. 4, pp. 303–312, 2022, doi: 10.47065/josyc.v3i4.2109.
[22] A. Sihotang, "Implementasi Algoritma Prefix Codes untuk Kompresi File Video Hasil Ekstra Aplikasi Kinemaster," vol. 1, no. 1, pp. 22–29, 2022.
[23] R. J. Iskandar and E. Kantana, "RANCANG BANGUN APLIKASI KOMPRESI LEMPEL-ZIV 77 BERBASIS ANDROID," vol. 10, no. 1, pp. 30–39, 2019.
[24] 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.
[25] I. Syuhada, "Implementasi Algoritma Arithmetic Coding dan Sannon-Fano Pada Kompresi Citra PNG," *TIN Terap. Inform. Nusant.*, vol. 2, no. 9, pp. 527–532, 2022.
[26] W. Adinda, "Kompresi Citra Medis Sinar-X Dada Menggunakan Algoritma Fraktal dan Arithmetic Coding." Universitas Sumatera Utara, 2023.
[27] J. C. Simatupang, R. K. Hondro, S. Sarwandi, and A. Fau, "Penerapan Algoritma Aritmathic Coding Untuk Kompresi Record Pada Database Penjualan PT. Octa Putra Jaya," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 873–877, 2022.
[28] R. Fadillah, "Penerapan Algoritma Fibonacci Codes Dalam Aplikasi Kompresi File Citra Digital," vol. 6, no. November, pp. 225–233, 2022, doi: 10.30865/komik.v6i1.5681.
[29] R. Nuraini, "Pendukung Keputusan Pemilihan Vendor IT Menggunakan Metode Perbandingan Eksponensial (MPE) Sistem," vol. 2, 2022.

[30] I. I. Journal, "IJIS Indonesian Journal on Information System e- ISSN 2548-6438 p-ISSN 2614-7173," vol. 7, no. September, 2022.