

Analisis Perbandingan Kompresi File Audio Menggunakan Algoritma Shannon Fano Dengan Algoritma Fibonacci Code

Sari Magdalena Simanjuntak

Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: sarysimanjuntak1999@gmail.com

Abstrak

Kompresi merupakan sebuah teknik yang dilakukan untuk mengurangi ukuran data yang berfungsi untuk mengurangi kapasitas memori dan mempercepat proses pemindahan data. Dalam penelitian ini metode yang digunakan adalah dengan menggunakan metode Shannon Fano dan Fibonacci Code. Pada metode *Shannon Fano* proses kompresi dilakukan dengan mengetahui frekuensi kemunculan setiap symbol yang kemudian akan diurutkan berdasarkan frekuensi yang paling besar hingga terkecil, sedangkan pada metode *Fibonacci Code* menggunakan deret integer *Fibonacci* untuk mengkodekan nilai bit dari data atau *file* yang akan dikompresi. *File* yang akan dikompresi merupakan *file* audio yang berekstensi MP3, untuk melakukan kompresi dalam memperkecil ukuran *file* audio teknik yang digunakan yaitu *lossless*. Teknik *lossless* merupakan teknik kompresi yang dapat mengembalikan *file* yang dikompresi diubah kebentuk aslinya (dekompresi). Algoritma yang digunakan dalam pengompresian *file* sama-sama menggunakan teknik *lossless*. *File* yang sudah dikompresi menggunakan metode *Shannon Fano* dan *Fibonacci Code* akan dilakukan *ratio* perbandingan untuk mengetahui metode yang lebih efisien dalam pengompresian *file* audio. Hasil penelitian menunjukkan bahwa metode *Shannon Fano* dan *Fibonacci* dapat mengkompresi *file* audio. Berdasarkan analisa hasil *ratio* kompresi yang dihasilkan menggunakan algoritma *Shannon Fano* mengalami perubahan 69% dengan menggunakan 48 sampel data, sedangkan hasil *compression ratio* yang dihasilkan menggunakan algoritma *fibonacci* mengalami perubahan 66% menggunakan 48 sampel data yang sama dengan algoritma *Shannon Fano*.

Kata Kunci: Kompresi; Dekompresi; Lossless; Shannon Fano; Fibonacci Code

Abstract

Compression is a technique used to reduce the size of the data that serves to reduce memory capacity and speed up the data transfer process. In this study, the method used is the Shannon Fano method and the Fibonacci Code. In the Shannon Fano method the compression process is carried out by knowing the frequency of occurrence of each symbol which will then be sorted from the largest to the smallest frequency, while the Fibonacci Code method uses the Fibonacci integer series to encode the bit value of the data or file to be compressed. The file to be compressed is an audio file with an MP3 extension, to perform compression in reducing the size of the audio file, the technique used is lossless. Lossless technique is a compression technique that can restore compressed files converted to their original form (decompression). The algorithms used in file compression both use lossless techniques. Files that have been compressed using the Shannon Fano and Fibonacci Code methods will be compared to find out which method is more efficient in compressing audio files. The results showed that the method of Shannon Fano and Fibonacci can compress audio files. Based on the analysis of the compression ratio results generated using the Shannon Fano algorithm, there was a 69% change using 48 data samples, while the compression ratio results generated using the Fibonacci algorithm experienced a 66% change using the same 48 data samples as the Shannon Fano algorithm.

Keywords: Compression; Decompression; Lossless; Shannon Fano; Fibonacci Code

1. PENDAHULUAN

Teknologi komunikasi yang semakin maju mempermudah masyarakat untuk memperoleh informasi dengan cepat. Banyak fasilitas teknologi yang dikembangkan dalam bentuk teks, audio dan gambar. *File* audio termasuk media informasi yang sering digunakan, namun memiliki ukuran sangat besar sehingga proses transmisi data dan media penyimpanan menjadi besar. Maka dilakukan proses kompresi data, untuk mengatasi masalah terkait dengan ruang penyimpanan dan proses transmisi data yang cukup lama. *File* audio adalah tempat penyimpanan berkas audio pada sistem komputer. *File* audio memiliki beberapa format yaitu MP3, AAC, WAV, AIFF dan sebagainya. MP3 merupakan salah satu format berkas pengodean suara yang memiliki kompresi yang baik. [1]

Kompresi merupakan teknik mengubah ukuran data awal menjadi ukuran yang lebih kecil. Proses kompresi terdapat dua teknik yaitu *lossy* dan *lossless*, dimana teknik *lossy* merupakan sebuah teknik mengubah ukuran *file* menjadi lebih kecil dengan cara menghilangkan data sehingga kualitas dari *file* audio berkurang dan data *file* audio yang dikompresi tidak dapat dikembalikan ke ukuran semula. Sedangkan teknik *lossless* merupakan teknik mengubah ukuran *file* audio dengan cara mengurangi perulangan pada *file* audio tanpa mempengaruhi mutu *file* audio tersebut. *File* audio yang dikompresi dengan teknik *lossless* dapat kembali ke ukuran semula (dekompresi). [2]

Proses kompresi mempunyai beberapa metode, dimana setiap metode memiliki hasil rasio kompresi yang berbeda namun pada penelitian ini penulis menggunakan dua metode untuk membandingkan, diantaranya algoritma *Shannon Fano* dan *Fibonacci Code*. Algoritma *Shannon Fano* merupakan algoritma yang memiliki sebuah kode dengan pembentukan pohon biner dan menggunakan teknik *lossless*, sedangkan *Fibonacci Code* merupakan *variable length code* dimana bilangan bulat yang lebih kecil mendapatkan kode pendek dan berakhir dengan dua bit angka 1 yang sama dan menggunakan teknik *lossless*. Dengan demikian maka dilakukan perbandingan antara algoritma *Shannon fano* dan *Fibonacci code* agar diketahui metode yang menghasilkan *ratio* kompresi yang lebih kecil dan seberapa efisien untuk kompresi *file* audio dan mempermudah proses pengiriman karena kapasitas semakin kecil.

Penelitian terdahulu terkait kompresi *file* audio adalah penelitian oleh Tetti Purnama Sari pada tahun 2018 yang berjudul Penerapan Algoritma Levenstein Pada Aplikasi Kompresi *File* MP3. Penelitian ini menyimpulkan MP3 yang berukuran besar akan dikompresi menjadi lebih kecil dan proses transmisi lebih cepat serta memperkecil lokasi penyimpanan[3]. Penelitian Aldi Medana Pratama pada tahun 2017 yang berjudul “Penerapan Algoritma *Huffman* dan *Shannon Fano* dalam Pemampatan File Teks” menyimpulkan bahwa *Shannon Fano* merupakan algoritma yang baik untuk pemampatan *file-file* lain[4]. Penelitian lainnya dilakukan oleh Jesica Martina Panjaitan yang berjudul “Penerapan Algoritma *Fibonacci Code* pada tahun 2020 yang berjudul Kompresi Aplikasi Audio Berbasis Dekstop” menyimpulkan bahwa algoritma *Fibonacci Codes* dapat mengurangi penggunaan media penyimpanan lebih dari 30% untuk sebuah *file*[5]. Penelitian Bobby Ramadhan pada tahun 2021 yang berjudul Implementasi Kombinasi Algoritma *Fibonacci Codes* dan *Levenstein Codes* Untuk Kompresi *File* pdf. Penelitian ini menyimpulkan bahwa hasil anaisa dengan dua proses kompresi didapatkan *space saving* kompresi *file* pdf mencapai hingga 68,75%, sedangkan rasio kompresi sebanyak 31,25%[6]

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi merupakan sebuah ilmu komputer yang melakukan pemampatan data dari data yang asli ke data yang lebih kecil dengan tujuan untuk efisiensi tempat penyimpanan dan proses pengiriman data lebih cepat. Kompresi dapat dilakukan untuk jenis data seperti gambar, audio, video maupun teks. Namun tidak semua algoritma kompresi sesuai untuk semua jenis data, melainkan disesuaikan dengan jenis data yang akan di kompresi.

2.2 Algoritma Shannon Fano

Algoritma *Shannon Fano Code* mengkodekan tiap karakter yang ada dalam serangkaian data input dengan menggunakan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang lebih pendek dibandingkan karakter yang jarang muncul. Pada dasarnya metode ini menggantikan setiap simbol dengan sebuah *alternative* kode biner yang panjangnya berdasarkan probabilitas dari simbol tersebut. *Shannon Fano Code* adalah sebuah teknik untuk membangun sebuah kode awalan didasarkan pada simbol dan probabilitas dan untuk membuat sebuah kode prefiks/awalan berdasarkan beberapa set simbol [1]. Contoh kode prefiks dapat dilihat pada tabel 1 berikut:

Tabel 1. Kode Prefiks

Non Prefiks	Prefix
00	10
001	110
00011	111

Langkah-langkah percobaan *Shannon Fano Code* sebagai berikut:

- Menentukan jumlah probabilitas huruf
- Mengurutkan huruf pada suatu tabel
- Meletakkan pada baris pertama untuk huruf yang memiliki probabilitas tinggi
- Melakukan pembagian terhadap tabel atas dua kelompok dengan probabilitas yang sama atau saling mendekati
- Memberikan kode 0 pada kelompok pertama dan kode 1 pada kelompok kedua
- Mengulangi langkah empat dan lima sampai tidak ada lagi kemungkinan untuk membagi kelompok-kelompok tersebut dalam dua sub kelompok.

2.3. Algoritma Fibonacci Code

Fibonacci Code merupakan salah satu algoritma kompresi *lossless*. *Fibonacci Code* adalah *variable length code* yang sangat berhubungan erat dengan bilangan bulat *Fibonacci*. Bilangan bulat *Fibonacci* merupakan urutan angka tertentu yang didapatkan dengan persamaan.

Fibonacci Code berkaitan dengan bilangan bulat, kode *Fibonacci* dari bilangan bulat positif n adalah *Fibonacci* yang merepresentasikan n dengan tambahan 1 ditambahkan ke ujung kanan. Dengan demikian *Fibonacci* kode 5 adalah 0001|1 dan dari 33 adalah 101010|1. Hal ini jelas bahwa kode tersebut berakhir dengan sepasang 11, seperti pada tabel berikut :

Tabel 2. Kode Fibonacci

1	11
2	011
3	0011
4	1011
5	00011
6	10011
7	01011

8	000011
9	100011
10	010011
11	001011
12	101011

Langkah-langkah pembentukan *Fibonacci Code*:

- Tentukan sebuah bilangan bulat positif n .
- Temukan bilangan *Fibonacci* f terbesar hingga terkecil atau sama dengan n , kurangkan nilai n dengan f dan catat sisa pengurangan nilai n dengan f .
- Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret *Fibonacci* $F(i)$, tambahkan angka 1 dalam kode *Fibonacci* yang akan dibentuk.
- Ulangi langkah 2, tukar nilai n dengan sisa pengurangan nilai n dengan f sampai sisa pengurangan nilai n dengan f adalah 0.
- Tambahkan angka 1 pada posisi paling kanan kode *Fibonacci* yang akan dibentuk sehingga diujung kode berakhir dengan sepasang angka 11.

2.4 Parameter Analisis Kompresi

Teknik dalam proses kompresi data terdapat beberapa faktor atau variabel yang biasa digunakan untuk mengukur kualitas dari suatu teknik data yaitu [6] :

- Ratio of Compression (RC)*

Ratio of Compression (R_c) adalah perbandingan antara ukuran data sebelum dikompresi dengan ukuran data setelah dikompresi.

$$RC = \frac{\text{ukuran data setelah dikompresi}}{\text{ukuran data sebelum dikompresi}} \quad (1)$$

- Compression Ratio (CR)*

Compression Ratio (CR) adalah persentase besar data yang telah dikompresi, yang didapat dari hasil perbandingan antara ukuran data setelah dikompresi dengan ukuran data sebelum dikompresi.

$$CR = \frac{\text{ukuran data setelah dikompresi}}{\text{ukuran data sebelum dikompresi}} \times 100 \% \quad (2)$$

- Space Savings (SS)*

Space Savings merupakan persentase isi antara data yang belum dikompresi dengan besar data yang telah di kompresi.

$$SS = 1 - \frac{\text{ukuran data setelah dikompresi}}{\text{ukuran data sebelum dikompresi}} \quad (3)$$

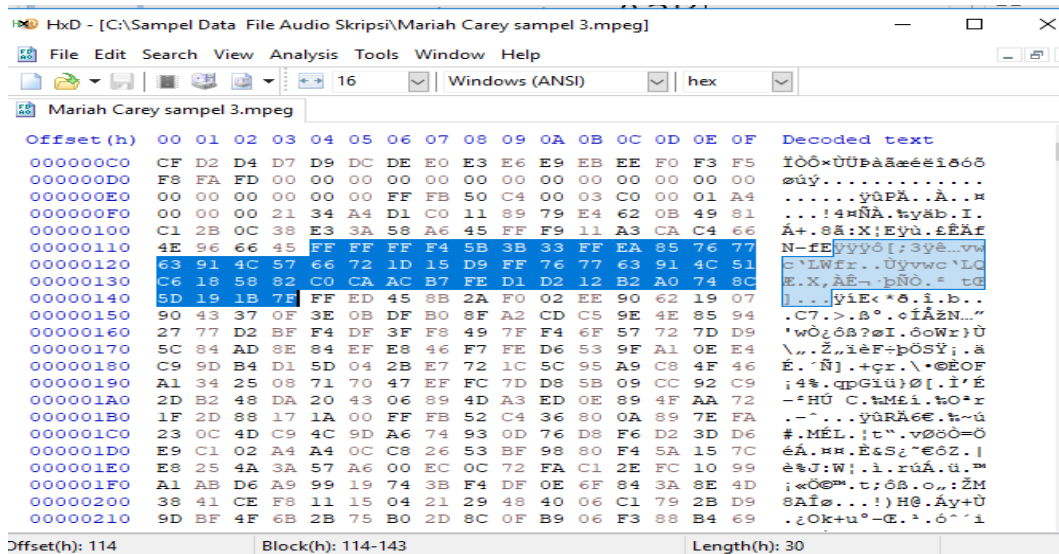
3. HASIL DAN PEMBAHASAN

File audio merupakan *file* suara yang dapat diputar dan salah satu *file* audio yang sering di gunakan adalah dengan ekstensi *file* Mp3. Analisa perbandingan adalah suatu proses untuk mengamati kinerja metode yang digunakan untuk menyelesaikan sebuah permasalahan. Penelitian ini menggunakan dua metode untuk mengetahui metode yang lebih efisien digunakan untuk memecahkan masalah yang sedang diteliti agar berjalan dengan baik. Metode yang digunakan adalah algoritma *Shannon Fano* dan *Fibonacci*.

Sebelum melakukan perbandingan kompresi terlebih dahulu *file* MP3 diubah ke heksadesimal. Untuk mengetahui nilai heksadesimalnya maka digunakan aplikasi HxD untuk dijadikan sampel kompresi. *File* yang sudah dikompresi menggunakan kedua metode akan dilakukan perbandingan menggunakan *ratio* perbandingan.

3.1 Penerapan Algoritma Shannon Fano Codes

Kompresi file audio dengan menggunakan algoritma *Shannon Fano* menggunakan sampel data yang berukuran 1.98 mb. Sampel data mp3 didapat nilai heksadesimal menggunakan bantuan software hex editor seperti pada gambar berikut ini:

**Gambar 1.** Nilai Hexadesimal File Mp3

Berdasarkan gambar diatas dapat dilihat nilai heksadesimal dari file audio MP3. Sampel hitungan menggunakan 48 karakter nilai heksadesimal audio MP3. Sampel dapat dilihat pada tabel 1 berikut:

Tabel 3. Sampel Data

FF	FF	FF	F4	5B	3B	33	FF
EA	85	76	77	63	91	4C	57
66	72	1D	15	D9	FF	76	77
63	91	4C	51	C6	18	58	82
C0	CA	AC	B7	FE	D1	D2	12
B2	A0	74	8C	5D	19	1B	7F

Dalam melakukan kompresi menggunakan algoritma Shannon Fano ada beberapa tahap yang dilakukan yaitu:

- Menentukan karakter, frekuensi dan kode ASCII

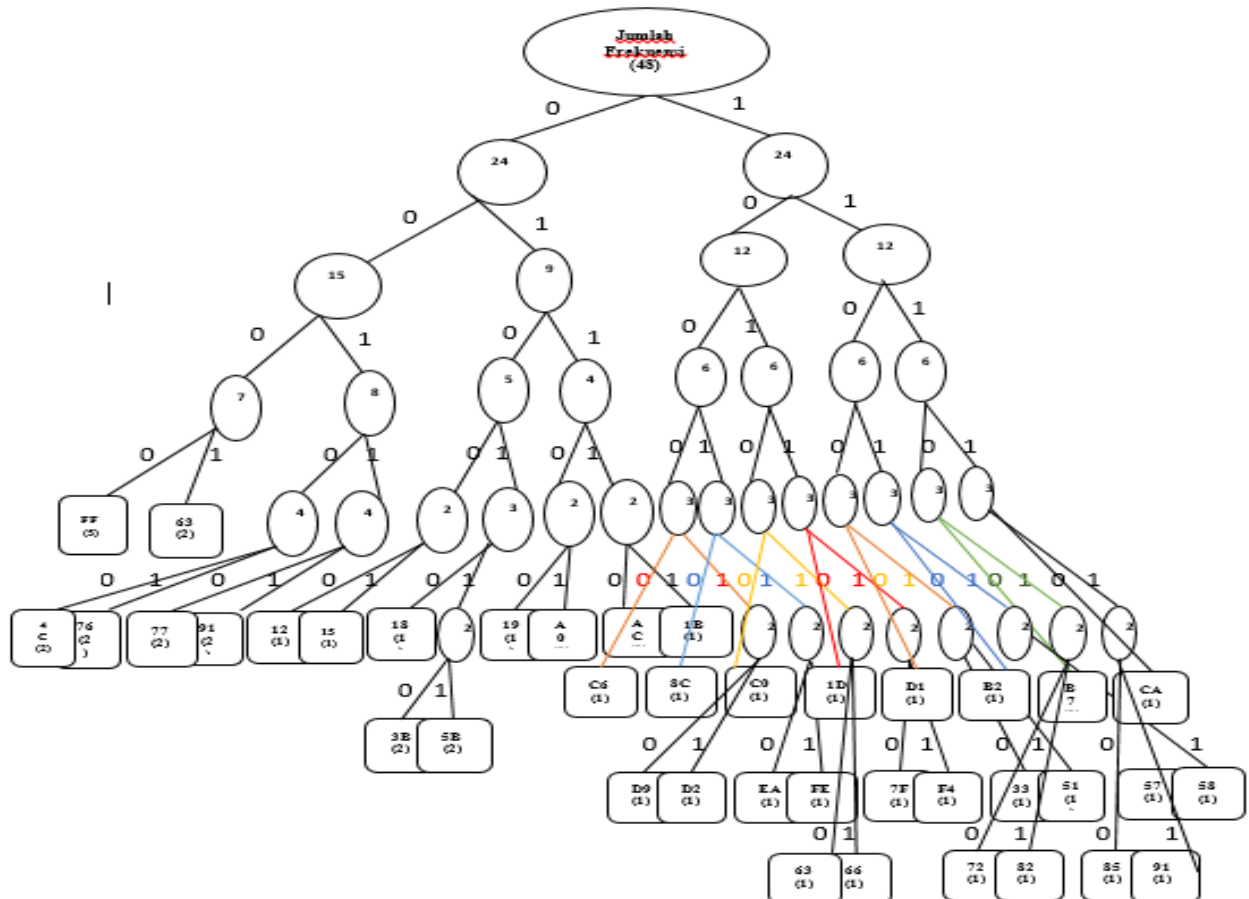
Kode ASCII memiliki 8 kode bit dan setiap karakter memiliki kode yang berbeda-beda seperti tabel berikut ini:

Tabel 4. Kode Karakter Dalam ASCII

Heksadesimal (Karakter)	Biner	Frekuensi
FF	11111111	5
63	01100011	2
4C	01001100	2
76	01110110	2
77	01110111	2
91	10010001	2
12	00010010	1
15	00010101	1
18	00011000	1
19	00011001	1
A0	10100000	1
AC	10101100	1
1B	00011011	1
3B	00111011	1
5B	01011011	1
C6	11000110	1
8C	10001100	1
C0	11000000	1
1D	00011101	1
D1	11010001	1
B2	10110010	1
B7	10110111	1
CA	11001010	1
D9	11011001	1

D2	11010010	1
EA	11101010	1
FE	11111110	1
7F	01111111	1
F4	11110100	1
33	00110011	1
51	01010001	1
57	01010111	1
58	01011000	1
	01100011	1
66	01100110	1
72	01110010	1
82	10000010	1
85	10000101	1
91	10010001	1
Total bit :		48

- b. Setelah karakter diurutkan berdasarkan frekuensi kemunculannya, kemudian dibagi menjadi bagian dengan jumlah probabilitas yang sama atau mendekati kemudian membentuk pohon biner, setelah terbentuk pohon biner kemudian berikan kode 0 dan 1 dimana pada bagian kiri terdapat kode 0 sedangkan bagian kanan kode 1. Lakukan pemberian kode tersebut sampai selesai dan membentuk kode karakter. Proses pemberian kode seperti gambar 2 berikut:



Gambar 2. Pemberian Kode 0 dan 1

Pembentukan pohon biner dilakukan agar mengetahui kode Shannon Fano dengan mengurutkan karakter dan pemberian kode 0 dan 1. Setelah pemberian kode 0 dan 1, maka terbentuklah kode Shannon Fano yang kemudian akan diubah ke bentuk heksadesimal.

Hasil kode Shannon Fano yang terbentuk setelah dikompresi sebagai berikut dan urutannya disesuaikan dengan susunan nilai heksadesimal atau karakter sebelum dikompresi adalah :

FF FF FF F4 5B 3B 33 FF EA 85 76 77 63 91 4C 57 66 72 1D 15 D9 FF 76 77 63 91 4C 51 C6 18 58 81 C0 CA AC B7 FE D1 D2 12 B2 A0 74 8C 5D 19 1B 7F menjadi nilai bit biner :

“000000000001011110101110101100100000100110110010101111000110001110000100100110110101011111010111001010101010011000100000001010100001001110010011001110000010101101111101111000100011010101101001101001101100110011000110010111101111011101101110”

Dengan total bit = 251 bit.

c. Penambahan Padding dan Flagging

String bit yang dihasilkan menggunakan *Shannon Fano* adalah 251 bit, karena 251 tidak habis dibagi 8 dan menyisakan 3, atau dengan kata lain sebagai berikut:

$$251 \bmod 8 = 3$$

nyatakan hasil bagi tersebut dengan “n”

Rumus *Padding*

$$7 - n + “1”$$

$$7 - 3 + “1” = 00001$$

Rumus *Flagging*

$$9 - n$$

$$9 - 3 = 6 = 00000011$$

Dengan penambahan *Padding* dan *Flagging* string biner menjadi seperti berikut:

“00000000000101111010111010110010000010011011001010111100011000111000010010011011010101111101011100101010101001100010000000101010000100111001001100111000001010110111110111100010001101010110100110100110110011000110010111101111011101101110000100000011”

Total bit keseluruhan setelah ada penambahan bit adalah: $251 + 5 + 8 = 264$

Adapun nilai yang sudah terkompresi dapat dilihat pada tabel berikut:

Tabel 5. Nilai Heksadesimal Terkompresi

No	Biner	Hexa
1	00000000	0
2	00001011	B
3	11010111	D7
4	01011011	5B
5	00100000	20
6	10011011	9B
7	00101011	2B
8	11100011	E3
9	00011100	1C
10	00100100	24
11	11011010	DA
12	10111110	BE
13	10101110	AE
14	01010101	55
15	01001100	4C
16	01000000	40
17	01010110	56
18	00010011	14
19	10010011	93
20	00111000	38
21	00101011	2B
22	01111110	7E
23	11110001	F1
24	00011010	1A
25	10101101	AD
26	00110100	34
27	11011001	D9
28	00110001	31
29	10010111	97
30	10111110	BE
31	11101101	ED
32	11000001	C1
33	00000011	3

Dari perhitungan tabel diatas setelah dikompresi menggunakan Shannon Fano code adalah 264 bit. Untuk diubah menjadi satuan byte maka dibagi 8.

a. Ratio of Compression (RC)

$$RC \text{ Shannon Fano} = \frac{384}{264} = 1,45 \text{ byte.}$$

b. Compression Ratio (CR)

$$CR \text{ Shannon Fano} = \frac{264}{384} \times 100\% = 68,75\% = 69\%$$

c. *Space Saving* (SS)

$$SS \text{ Shannon Fano} = \frac{384 - 264}{264} \times 100\% = 45,45\%$$

3.2 Penerapan Algoritma Fibonacci Code

Dalam melakukan kompresi menggunakan algoritma Fibonacci ada beberapa tahap yang dilakukan yaitu:

a. Mengurutkan Karakter

Melakukan kompresi terlebih dahulu file MP3 diubah kedalam bentuk hexa dan biner, kemudian diurutkan dengan memulai dari frekuensi terbesar sampai terkecil. Adapun urutan sampel nilai file MP3 dapat dilihat pada tabel 4 berikut:

Tabel 6. Urutan Karakter

N	Heksadesimal (Karakter)	Biner	Frekuensi
1	FF	11111111	5
2	63	01100011	2
3	4C	01001100	2
4	76	01110110	2
5	77	01110111	2
6	91	10010001	2
7	12	00010010	1
8	15	00010101	1
9	18	00011000	1
10	19	00011001	1
11	A0	10100000	1
12	AC	10101100	1
13	1B	00011011	1
14	3B	00111011	1
15	5B	01011011	1
16	C6	11000110	1
17	8C	10001100	1
18	C0	11000000	1
19	1D	00011101	1
20	D1	11010001	1
21	B2	10110010	1
22	B7	10110111	1
23	CA	11001010	1
24	D9	11011001	1
25	D2	11010010	1
26	EA	11101010	1
27	FE	11111110	1
28	7F	01111111	1
28	F4	11110100	1
30	33	00110011	1
31	51	01010001	1
32	57	01010111	1
33	58	01011000	1
34	63	01100011	1
35	66	01100110	1
36	72	01110010	1
37	82	10000010	1
38	85	10000101	1
39	91	10010001	1
Total bit :			48

b. Pembentukan Kode *Fibonacci*

Langkah-langkah untuk pembentukan sebuah kode Fibonacci untuk sampel yang berjumlah 48 karakter sebagai berikut :

01 01 01 01 01 FF C0 00 11 08 00 DC 00 DC 03 01 22 00 02 11 01 03 11 01 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B FF C4

Dalam pembentukan kode *Fibonacci* perlu diketahui nilai n yang akan diubah menjadi kode *Fibonacci* (biner). Nilai n diambil dari pengurutan karakter heksadesimal nilai n dapat dilihat pada tabel 6 diatas. Proses pembentukan nilai n sebagai berikut:

Nilai $n = 1$: Bilangan bulat positif

Deret *Fibonacci* yang mendekati nilai n adalah : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34....dst.

Proses pengurangan nilai n dengan setiap bilangan *Fibonacci* sampai sisa pengurangan nilai n adalah 0 :

$1 - 1 = 0$: Nilai n dikurangkan dengan satu bilangan *Fibonacci*, pengurangan nilai n dapat dibentuk dengan deret *fibonacci* yang lebih kecil atau sama dengan nilai n dan catat sisa pengurangan nilai n adalah 0.

Tabel 7. Proses Pembentukan Kode *Fibonacci* Untuk $N = 1$

Urutan Bilangan <i>Fibonacci</i>	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)
Bilangan <i>Fibonacci</i>	0	1	1	2	3	5	8	13
Kode <i>Fibonacci</i> Sementara	-	-	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, lakukan proses pembentukan langkah diatas hingga selesai.

Hasil perhitungan dari tabel proses pembentukan kode *Fibonacci* diatas terdapat hasil kompresi terdapat pada tabel 8 berikut:

Tabel 8. Hasil Akhir Kode *Fibonacci*

Nilai N	Kode Fibonacci	Codeword			Frekuensi	Bit * Frekuensi
	Hex	Sementara	Bit			
1	FF	1	11	2	5	10
2	63	01	011	3	2	6
3	4C	001	0011	4	2	8
4	76	101	1011	4	2	8
5	77	0001	00011	5	2	10
6	91	1001	10011	5	2	10
7	12	0101	01011	5	1	5
8	15	00001	000011	6	1	6
9	18	10001	100011	6	1	6
10	19	01001	010011	6	1	6
11	A0	00101	1011	4	1	4
12	AC	10101	101011	6	1	6
13	1B	01	011	7	1	7
14	3B	100001	1000011	7	1	7
15	5B	010101	0101011	7	1	7
16	C6	001001	0010011	7	1	7
17	8C	101001	1010011	7	1	7
18	C0	000101	000101	7	1	7
19	1D	100101	1001011	7	1	7
20	D1	10101	101011	6	1	6
21	B2	1	11	3	1	2
22	B7	10001	100011	6	1	6
23	CA	100001	1000011	7	1	7
24	D9	01001	010011	6	1	6
25	D2	101001	1010011	7	1	7
26	EA	1001	10011	5	1	5
27	FE	1001001	10010011	8	1	8
28	7F	101001	1010011	7	1	7
29	F4	101	1011	4	1	4
30	33	100101	1001011	7	1	7
31	51	101001	1010011	7	1	7
32	57	10101	101011	6	1	6
33	58	100101	1001011	7	1	7
34	63	1	11	2	1	2
35	66	1000001	10000011	8	1	8
36	72	100001	1000011	6	1	6
37	82	10001	100011	5	1	5
38	85	101001	1010011	7	1	7
39	91	10001	100011	5	1	5
		Total				247

Dari tabel diatas menunjukkan kode dari setiap karakter yang ada menggunakan algoritma *Fibonacci Code* adalah 247 bit. Untuk diubah menjadi satuan byte maka dibagi 8 yaitu $247/8 = 30.87$ byte.

c. Proses Padding dan Flagging

String bit yang dihasilkan menggunakan algoritma Fibonacci adalah 255 bit, karena 247 tidak habis dibagi 8 dan menyisakan 7, atau dengan kata lain sebagai berikut:

$$247 \bmod 8 = 7$$

Nyatakan hasil bagi tersebut dengan "n"

Rumus *Padding* Rumus *Flagging*

$$7 - n + "1"$$

$$9 - n$$

$$7 - 7 + "1" = 1$$

$$9 - 1 = 8 = \mathbf{00000101}$$

Total bit keseluruhan setelah ada penambahan bit adalah: $247 + 1 + 8 = 256$.

Dari perhitungan tabel diatas setelah dikompresi menggunakan Shannon Fano code adalah 256 bit. Untuk diubah menjadi satuan byte maka dibagi 8.

1. *Ratio of Compression* (RC)

$$\text{RC Fibonacci Code} = \frac{384}{256} = 1.5 \text{ byte.}$$

2. *Compression Ratio* (CR)

$$\text{CR Fibobnacci Code} = \frac{256}{384} \times 100\% = 66.6 \%$$

3. *Space Saving* (SS)

$$\text{SS Fibonacci Code} = \frac{384 - 256}{256} \times 100\% = 50 \%$$

Berdasarkan proses kompresi file audio menggunakan metode Shannon Fano dan Fibonacci code yang telah dilakukan, maka diperoleh rasio perbandingan dengan menggunakan rumus. *Ratio* perbandingan kedua metode setelah melakukan perhitungan dapat dilihat pada tabel 9 berikut :

Tabel 9. *Ratio* Perbandingan

Parameter Analisis Kompresi	Algoritma Shannon Fano	Algoritma Fibonacci Code
<i>Ratio of Compression</i>	1.45 byte	1.5 byte
<i>Compression Ratio</i>	69 %	66 %
<i>Space Saving</i>	45.45%	50 %

Setelah mengetahui hasil ratio perbandingan dari kedua metode pada proses kompresi berdasarkan analisa yang dilakukan, maka dapat diketahui kelebihan dan kekurangan pada kedua metode tersebut. Kelebihan dan kekurangan akan dijelaskan pada tabel 10 berikut:

Tabel 10. Kelebihan dan Kekurangan

Kelebihan	Kekurangan
<ul style="list-style-type: none"> - Proses kompresi file audio MP3 berhasil dilakukan menggunakan dua algoritma yaitu Shannon Fano dan Fibonacci Code - Pada file audio MP3 hasil kompresi mengalami perubahan yang signifikan. Rata-rata <i>Compression Ratio</i> yang didapatkan lebih dari 60% untuk sampel 48. - Waktu yang dibutuhkan untuk proses kompresi file audio MP3 bervariasi. Hal ini dipengaruhi oleh panjangnya karakter file yang diinputkan. 	<ul style="list-style-type: none"> - Kompresi pada algoritma Fibonacci cukup efektif dalam penginputan kode file audio, namun jika nilai n lebih besar maka otomatis kode binernya akan lebih panjang dan lebih banyak. - Proses kompresi menggunakan algoritma Shannon Fano akan lebih optimal jika variasi karakter tidak terlalu banyak meskipun frekuensinya. Jika karakternya lebih sedikit maka tinggi dan subpohon yang akan terbentuk akan lebih singkat untuk mewakili karakter.

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan oleh penulis tentang Analisis Perbandingan Kompresi File Audio Menggunakan Algoritma *Shannon Fano* Dengan Algoritma *Fibonacci Code* maka dapat diambil kesimpulan Dimana proses dalam melakukan kompresi dan dekompresi dengan menggunakan algoritma *Shannon Fano Code* dan *Fibonacci Code* dilakukan dengan cara mengubah sampel data menjadi bilangan heksadesimal menggunakan aplikasi Hxd. Kemudian dilakukan kompresi menggunakan algoritma *Shannon Fano Code* dan *Fibonacci Code*. Proses kompresi yang dilakukan menggunakan algoritma *Shannon Fano Code* dan *Fibonacci Code* menghasilkan rasio kompresi. Hasil *ratio* kompresin yang dihasilkan menggunakan algoritma *Shannon Fano* adalah 69% perubahan dengan menggunakan 48 sampel data, sedangkan hasil *compression ratio* yang didapatkan menggunakan algoritma *Fibonacci* adalah 66% perubahan menggunakan 48 sampel data yang sama dengan algoritma *Shannon Fano*.

REFERENCES

- [1] Amri, Yahya Fathoni. (2012). "Kompresi File Audio Menggunakan Arithmetic Coding," Univesitas Sebelas Maret, pp. 63-73.
- [2] H. Ibrahim, Tommy, & S. Irsa Nenna. (2021). "Analisis Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital," Algoritma. Jurnal. Ilmu Komputer. dan Informatika, vol. 6341.
- [3] T. P. Sari, S. D. Nasution, & R. K. Hondro. (2018). "Penerapan Algoritma Levenstein Pada Aplikasi Kompresi File Mp3," KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), vol.2, no. 1.
- [4] A. M. Pratama, N. A. Hasibuan, and E. Buulolo, (2017) "Penerapan Algoritma Huffman Dan Shannon-Fano Dalam Pemampatan File Teks," Majalah Ilmiah Informasi dan Teknologi., vol. 12, no.1, pp. 312–317.
- [5] J. Martina and B. Panjaitan, (2021) "Penerapan Algoritma Fibonacci Codes Pada Kompresi Aplikasi Audio Mp3 Berbasis Dekstop," pp. 27–33.
- [6] B. Ramadhana, (2021) "Implementasi Kombinasi Algoritma Fibonacci Codes Dan Levenstein Codes Untuk Kompresi File Pdf," vol. 8, no. 2, pp. 67–71.
- [7] M. R. Prasetyo, (2019) "Analisis Perbandingan Kinerja Alogaritma Shannon Fano dan Levenstein Code pada Kompresi File Video," Repos Intitusi Universitas Sumatera Utara, pp. 23.
- [8] A. Ilham, (2019) "Implementasi Fibonacci Code Dalam Kompresi dan Pengamanan File" J. Pembang. Wil. Kota, vol. 1, no. 3, pp. 82–91.
- [9] D. H. Gulo, (2017) "Pengamanan File MP3 Dengan Menggunakan Metode Triple Data Encryption Standar (Triple Des)," Jurnal Teknik Informatika UNIKA St. Thomas, vol. 02, pp. 7–13.
- [10] M. R Anggara (2018) "Perbandingan Algoritma Huffman Code dan Fibonacci Kompresi Audio," vol. 1, pp.33-35.
- [11] K. Heri Abijono, (2019) "Implementasi Algoritma Analytical Untuk Kompresi File Teks," vol. 3, no. 1, pp. 21–30.
- [12] H. S. Mangiri, (2018) "Pembelajaran Kompresi Text Menggunakan Metode Shannon Fano" Journal Of Informatics Education, vol. 1, pp. 44 - 54.
- [13] M. R. Pane, (2017) "Perancangan Aplikasi Kompresi Menggunakan Metode Shannon Fano dan Unary Coding pada File Teks," Majalah Ilmiah INTI, vol.5, Nomor 1, pp. 306–311, 2017.