RESOLUSI : Rekayasa Teknik Informatika dan Informasi
ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

Implementasi Algoritma Elias Gamma Code Untuk Kompresi File Audio Hasil Rekaman Dari Aplikasi Wesing Karaoke

Redita Manik

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: redita99manik@gmail.com

Abstrak-Dalam era teknologi yang semakin canggih seperti saat ini, tingginya aktivitas penyimpanan dan pengiriman data menimbulkan suatu masalah dimana dapat mengakibatkan pemborosan ruang penyimpanan dan lambatnya proses pengiriman data secara online ataupun pengiriman secara ofline. Penyimpanan data yang semakin lama akan mengakibatkan menumpuknya data-data yang semakin banyak pula. Penumpukan data ini menjadi masalah terutama dari ketersediaanya ruang penyimpanan. Seiring perkembangan teknologi saat ini maka semakin berkembangnya kemampuan dalam mengolah data yaitu dengan memanfaatkan metode kompresi yang diharapkan dapat menjawab masalah tersebut. Algoritma Elias Gamma Code digunakan untuk mengkompresi file audio hasil rekaman dari aplikasi wesing karaoke dengan memperkecil ukuran file dari semula. Sehingga dengan menerapkan algoritma elias gamma code untuk mengkompresi file audio dapat mengurangi kapasitas penyimpanan secara optimal pada smartphone (handphone).

Kata Kunci: Kompresi; File Audio; Wesing Karaoke; Elias Gamma Code.

Abstract-In an era of increasingly sophisticated technology, such as today, the high activity of data storage and transmission creates a problem which can lead to wasteful storage space and slow data transmission processes online or offline. The longer data storage will result in accumulating more and more data. This data accumulation becomes a problem, especially from the availability of storage space. Along with current technological developments, the ability to process data is increasingly developing, namely by utilizing compression methods which are expected to answer these problems. The Elias Gamma Code algorithm is used to compress audio files recorded from the Wesing Karaoke application by reducing the file size from the beginning. So that by applying the Elias Gamma Code algorithm to compress audio files, it can reduce the optimal storage capacity on smartphones (cellphones).

Keywords: Compression; Audio File; Wesing Karoke; Elias Gamma Codes.

1. PENDAHULUAN

Perkembangan teknologi informasi yang sangat pesat dewasa ini telah memasuki segala aspek kehidupan. Sepanjang manusia ingin berinovasi, selama itu juga perkembangan teknologi informasi dan ilmu pengetahuan senantiasa berkembang. Kemajuan teknologi ini sangat membantu aktivitas manusia, salah satunya dapat memudahkan setiap orang untuk mengakses berbagai bidang seni musik yang kini telah banyak di *play store* dalam bentuk aplikasi. Dimana dapat kita ketahui secara luas, banyak orang mempelajari teknik latihan vocal dengan cara berkaraoke bila tergoda untuk menjajal kemampuan bakat menyanyi ataupun menyimpan hasil rekaman lagu dari aplikasi Wesing.

Wesing adalah sebuah aplikasi android yang dapat digunakan untuk berkaraoke. Wesing sendiri dapat dijalankan dengan sistem online, dengan sistem online ini kita dapat berduet dengan teman-teman kita yang sesama menggunakan aplikasi Wesing dan kita juga dapat bernyanyi dan kemudian merekam lagu tersebut.

Pada aplikasi wesing banyak pilihan lagu dengan durasi rata-rata lebih dari dua menit, Selain menyimpan file lagu pengguna juga pasti menyimpan *file* video dan *file-file* lainnya sehingga membutuhkan ruang penyimpanan yang besar dan pengguna juga menginginkan data dengan kualitas terbaik dan kuantitas (ukuran) yang minimum. Maka untuk membuat banyak ruang kosong dan memiliki ukuran data yang tidak besar pada media penyimpanan diperlukan metode kompresi yang memiliki arti untuk mempersingkat ukuran *bit* yang diperlukan suatu data.

Salah satu inovasi yang akan dikembangkan adalah masalah mereduksi ukuran suatu kapasitas data atau sering disebut juga dengan kompresi. Kompresi data merupakan cara untuk memadatkan data, sehingga hanya memerlukan ruangan penyimpanan lebih sedikit dan lebih efisien dalam menyimpannya. kompresi data juga dapat mempersingkat waktu jika data tersebut ditransmisikan. Proses yang mengkonversi sebuah masukan berupa aliran data (the source atau data asli mentah) menjadi suatu aliran data lain (Output, aliran bit atau aliran sudah dikompres) yang memiliki ukuran lebih kecil. Meskipun ukurannya sudah diperkecil dari ukuran sebelummya namun tidak sedikitpun mengurangi kualitas dari *file* audio tersebut dan kompresi semacam ini bersifat *lossless. Losseles* adalah kompresi data dimana hasil kompresi dan dekompresinya sama dan yang mengalami perubahan hanyalah ukuran data saja.

Dalam proses kompresi data, ada beberapa hal yang harus diperhatikan. Yaitu, time process (waktu yang akan berjalan pada saat data dikompresi dan dekompresi), ratio (ukuran data setelah di kompresi dan dekompresi), completeness (kelengkapan data setelah file-file tersebut di kompresi dan dekompresi) space saving (persentase selisih ukuran data setelah dikompres dengan ukuran data sebelum di kompresi).

Untuk melakukan proses kompresi kita memerlukan algoritma yang tepat digunakan untuk mereduksi ukuran dari *file* audio ini. Salah satunya adalah dengan menggunakan algoritma *Elias Gamma Code* dimana algoritma *Elias Gamma Code* merupakan algoritma yang akan menghitung berdasarkan dari parameter *Ratio of Compression(RC), Compression Ratio(CR), Space Savingi (SS) dan waktu kompresi.*

ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

Dengan menerapkan algoritma *elias gamma code* dalam proses pengiriman *file* yang besar agar lebih cepat dan menghemat penyimpanan. *File* yang berukuran besar akan dikompresi menjadi ukuran yang lebih kecil.

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi adalah proses pengubahan sekumpulan data menjadi bentuk kode dengan tujuan untuk menghemat kebutuhan tempat penyimpanan data dan waktu untuk transmisi data.Pada saat ini kebutuhan akan informasi sangatlah diperlukan oleh masyarakat umum[1]. Dengan semakin banyaknya informasi yang perlu disimpan secara digital, secara automatis akan meningkatkan keperluan untuk menyediakan media penyimpanan data yang lebih besar lagi. Oleh karena itu diperlukan suatu alternatif mekanisme penyimpanan data sehingga dengan media penyimpanan yang ada, kita dapat menyimpan data sebanyak-banyaknya. Pemampatan atau kompresi data merupakan salah satu metode untuk memperkecil ruang penyimpanan data pada suatu media penyimpanan. Selain berguna dalam penyimpanan data, kompresi data dapat membantu memperkecil ukuran data yang ditransmisikan di dalam suatu media jaringan, seperti internet sehingga memperkecil waktu transfer data[2].

2.2 Algoritma Elias Gamma Code

Algoritma *Elias Gamma Code*merupakan sebuah algoritma kompresi yang dibuat oleh Peter *Elias.* Untuk membuat tabel kode *Elias Gamma*, *Elias* menambah panjang kode dalam *Unary* (u). Dalam kode berikutnya, Eγ ditambahkan pada panjang kode (M) dalam biner (β). Dengan demikian, *Elias Gamma Code*, yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun [2].

Adapun aturan untuk mengkodekan sebuah bilangan dengan menggunakan *Elias Gamma*[3] adalah sebagai berikut :

1. Tulis bilangan tersebut dalam bentuk biner.
2. Kurangkan 1 dari jumlah bit yang dipilih pada langkah pertama dan tambahkan sesuai dengan banyaknya bilangan nol. Proses yang ekivalen untuk menyatakan proses yang pada *point* nomor dua adalahsebagai berikut :
   a. Pisahkan *integer* menjadi pangkat 2 tertinggi (2^N) yang dapat dan ditampungnya sisakan digit biner N dari *integer* tersebut.
   b. Kodekan N dalam bentuk unary, jika N adalah nol maka diikuti oleh satu.
   c. Tambahkan sisa digit biner N untuk mempresentasikan N.

Proses kompresi/*encoding*suatu *integer* dengan menggunakan *Elias Gamma* dapat dilakukan dengan cara sebagai berikut :

1. Tentukan nilai N untuk pangkat yang paling mendekati nilai n yang dituliskan sebagai β(n). Nilai ini disebut sebagai *unary code*, dimana jumlah nilai N ditulis menjadi angka 0 dan diakhiri dengan angka 1
2. Dapatkan nilai L dengan mengurangi nilai n dengan nilai 2^N, nilai yang didapati diubah menjadi bilangan *biner*

Pembentukan kode *Elias Gamma Code* dapat diambil sebuah contoh n = 9 Lalu temukan bilangan bulat N terbesar sehingga $2^N \leq n < 2^{N+1} = 2^3 \leq 9 < 2^{3+1}$ setelah ditemukan kode N terbesar lalu rubah nilai n menjadi biner lalu hilangkan 1 bit paling kiri 9 = 1001 → 001, kodekan dalam bentuk unary N sebagai 0 diikuti oleh 1 sehingga dihasilkan unary 3 → 0001, lalu tambahkan sisa digit biner n dibelakang kode unary yang telah dihasilkan 0001001.Agar dapat lebih jelas berikut hasil pengkodean *Elias Gamma Code* dapat dilihat pada tabel 2.1 :

Tabel 1. Kode *Elias Gamma Code*

$1 = 2^0 + 0 = 1$	$10 = 2^3 + 2 = 0001010$
$2 = 2^1 + 0 = 010$	$11 = 2^3 + 3 = 0001011$
$3 = 2^1 + 1 = 011$	$12 = 2^3 + 4 = 0001100$
$4 = 2^2 + 0 = 00100$	$13 = 2^3 + 5 = 0001101$
$5 = 2^2 + 1 = 00101$	$14 = 2^3 + 6 = 0001110$
$6 = 2^2 + 2 = 00111$	$15 = 2^3 + 7 = 001111$
$7 = 2^3 + 3 = 00111$	$16 = 2^4 + 0 = 000010000$
$8 = 2^3 + 0 = 0001000$	$17 = 2^4 + 1 = 000010001$
$9 = 2^3 + 1 = 0001001$	$18 = 2^4 + 2 = 000010010$

Langkah berikutnya adalah mengkonversi karakter sesuai dengan kode *Elias Gamma*, sebelum dikonversi terlebih dahulu dilakukan pemeriksaan terhadap panjang *string bit* dengan langkah sebagai berikut :

1. Jika sisa bagi panjang *string bit* terhadap 8 adalah 0 maka tambahkan 00000001. Nyatakan dengan *bit* akhir.
2. Jika sisa bagi panjang *string bit* terhadap 8 adalah n (1, 2, 3, 4, 5, 6, 7) maka tambahkan 0 sebanyak 7 – n + "1" di akhir *string bit*. Nyatakan dengan L, lalu tambahkan bilangan biner dari 9 – n. Nyatakan dengan *bit* akhir.

Proses Dekompresi/*decoding* dalam kode *elias gamma code* dapat dilakukan langkah-langkah sebagai berikut [2]:

1. Lakukan pembacaan pada 8 *bit* terakhir, hasil pembacaan berupa bilangan decimal. Nyatakan hasil pembacaan dengan n.
2. Hilangkan *bit* pada bagian akhir sebanyak 7 + n.

ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

Berikut ini adalah algoritma dekompresi *Elias Gamma Code*[2]:

1. Lakukan pembacaan *string bit* dari awal sampai angka 1 ditemukan. Catat posisi angka 1 dan nyatakan sebagai p. Nyatakan jumlah angka 0 dengan n.
2. Lakukan pembacaan *string bit* setelah angka 1 sebanyak n
3. Ganti kode hasil pembacaan dengan karakter sesuai berdasarkan tabel *Elias Gamma Code*.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Analisa yang dilakukan dalam penelitian ini yaitu mengkompresi *file* audio dengan menerapkan algoritma *elias gamma code* hasil rekaman pada aplikasi *wessing. Wessing* karaoke adalah aplikasi untuk pecinta musik (khususnya penggemar karaoke). Koleksi lagu yang beragam sangat bagus untuk seluruh keluarga, dengan lagu 12 bahasa dari semua genre. Pada saat ini banyak orang-orang mengoleksi atau mengumpulkan *file* audio hasil rekaman dari aplikasi *wessing* karaoke ke dalam memori tanpa melihat seberapa besar kapasitas *file* audio tersebut dan tidak memperhatikan seberapa besar penyimpanan yang dimiliki, akibatnya dibutuhkan waktu yang lama untuk menampilkan audio. Sehingga orang-orang terkadang harus melakukan kompresi pada sebuah *file* audio yang dimilikinya agar dapat mempermudah dalam pengiriman *file* audio atau dengan mengurangi ukuran file tersebut sehingga dapat memenuhi ruangan dalam memori. *File* audio yang akan di kompresi adalah *file* audio hasil rekaman dari aplikasi *wessing* karaoke yang mempunyai format MP3.

Teknik yang digunakan dalam mengkompresi yaitu algoritma *Elias gamma code* ini ialah Teknik *lossless compression* yang memperkecil suatu data berdasarkan dengan frekuensi karakter pada objek yang akan dilakukan kompresi. untuk melakukan proses kompresi dengan menggunakan metode ini yaitu berdasarkan karakter yang sering muncul dan akan memiliki jumlah bit terkecil berdasarkan kode *elias gamma code*, sedangkan karakter paling sedikit muncul akan memiliki jumlah bit terpanjang. Tahapan analisa terhadap suatu sistem dilakukan sebelum tahapan perancangan dilakukan. Adapun tujuan dilakukan analisa terhadap suatu sistem adalah untuk mengetahui alasan mengapa sistem tersebut diperlukan, yaitu dengan merumuskan kebutuhan-kebutuhan dari sistem tersebut untuk meminimalisir sumber daya yang berlebih serta membantu merencanakan penjadwalan pembentukan sistem.

3.1.1 Penerapan Elias Gamma Code

Dengan melakukan kompresi data, data yang berukuran besar akan dikompresi menjadi ukuran yang kecil dan akan mengurangi alokasi penyimpanan. Dalam menganalisa *file* audio harus dilakukan mengambilan *sample file* untuk mendapatkan nilai dari data pada sebuah *file* audio yang berupa nilai *hexadecimal*. Berikut adalah langkah untuk mengkompresi dan mendekompresi *file* audio.

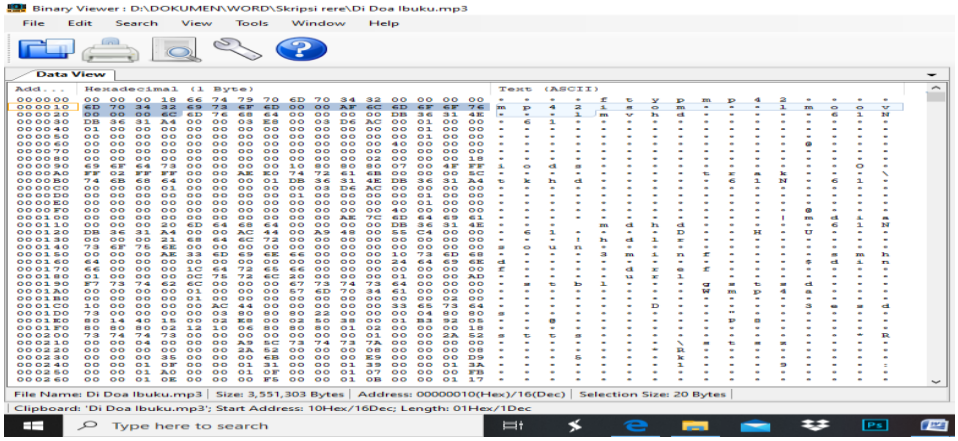
1. Analisa proses kompresi *file* audio dengan menggunakan *Elias Gamma Code*
   a. Sebelum perancangan masukkan hasil rekaman dari aplikasi *wesing* karaoke. *File* audio yang digunakan adalah *file* audio yang dihasilkan dari aplikasi hasil rekaman dari aplikasi *wessing* karaoke. Berikut adalah informasi objek *file* audio yang akan diambil sampelnya sebelum dilakukan kompresi:

Tabel 2. Informasi *file* audio sebelum dikompresi

Keterangan	
Jenis File	.MP3
Judul	Didoa Ibuku Ada Namaku Disebut
Ukuran	3, 38 MB
Durasi	4:11 Menit

Dari sample diatas didapatkan nilai *hexadecimal* menggunakan bantuan binary viewer seperti pada gambar dibawah ini:

RESOLUSI : Rekayasa Teknik Informatika dan Informasi
ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi



Gambar 1. Nilai *Hexadecimal File* audio *Sample*

Berdasarkan pada gambar diatas maka didapatkan nilai *hexadecimal* pada *file* audio sample. Untuk keperluan hitungan manual, maka hanya akan diambil *sample* nilai sebanyak 20 karakter nilai *hexadecimal file* audio *sample*. Nilai *hexadecimal* diambil dari sisi kiri sampai bilangan ke 20.

b. Melakukan Pembacaan Isi *File*
Adapun bilangan *hexadecimal file* audio *sample* tersebut adalah 6D, 70, 34, 32, 69, 73, 6F, 6D, 00, 00, AF, 6C, 6D, 6F, 6F, 76, 00, 00, 00, 6C. Nilai data ini dimasukan kedalam tabel untuk dilakukan pembacaan frekuensi. Pembacaan frekuensi dilakukan dengan menghitung jumlah nilai yang sama di setiap nilai data yang muncul. Adapun pembacaan frekuensi dapat dilihat pada tabel di bawah ini :

Tabel 3. Frekuensi data

NILAI	
HEXADESIMAL	**FREKUENSI**
6D	3
70	1
34	1
32	1
69	1
73	1
6F	3
00	5
AF	1
6C	2
76	1
TOTAL	20

c. Mengurutkan dari karakter yang memiliki frekuensi terbesar (banyak nilai yang sama) ke frekuensi terkecil. Urutan nilai dapat dilihat pada tabel berikut ini:

Tabel 4. Pembentukan Decoding

		Nilai			
No	**Hexadesimal**	**Binary**	**Frek**	**Bit**	**Bit x frek**
1	00	00000000	5	8	40
2	6D	01101101	3	8	24
3	6F	01101111	3	8	24
4	6C	01101100	2	8	16
5	70	01110000	1	8	8
6	34	00110100	1	8	8
7	32	00110010	1	8	8
8	69	01101001	1	8	8
9	73	01110011	1	8	8
10	AF	10101111	1	8	8
11	76	01110110	1	8	8
	Total				160

ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

Berdasarkan tabel diatas, satu nilai *hexadecimal* (karakter) bernilai 8 bit bilangan biner. Sehingga 20 bilangan *hexadecimal* mempunyai nilai biner sebanyak 160 bit. Untuk mengubah satuan menjadi byte maka jumlah keseluruhan bit dibagikan 8. Maka dihasilkan 160/8 = 20 byte.

d. Membentuk tabel *Elias Gamma Code*
Aturan dalam pembentukan kode bilangan dengan menggunakan *elias omega code* dapat dilihat pada sub landasan teori bab sebelumnya. Adapun kode *elias omega code* dapat dilihat pada tabel 5.:

Tabel 5. Kode *Elias Gamma*

N	Kode Elias Gamma
1	1
2	01 0
3	01 1
4	001 00
5	001 01
6	001 10
7	001 11
8	0001 000
9	0001 001
10	0001 010
11	0001 011
12	0001 100
13	0001 101
14	0001 110
15	001 111

Adapun proses kompresi *file* audio sampel dapat dilihat pada tabel berikut:

Tabel 6. Kompresi Nilai *File* audio *Sample* Dengan *Elias Gamma Code*

No	Nilai Hexa	Kode Elias Gamma	Bit	Frek	Bit x Frek
1	00	1	1	5	5
2	6D	01 0	3	3	9
3	6F	01 1	3	3	9
4	6C	001 00	5	2	10
5	70	001 01	5	1	5
6	34	001 10	5	1	5
7	32	001 11	5	1	5
8	69	0001 000	7	1	7
9	73	0001 001	7	1	7
10	AF	0001 010	7	1	7
11	76	0001 011	7	1	7
TOTAL					76

Dari perhitungan tabel diatas setelah dikompresi dengan menggunakan *elias gamma code* adalah 76 bit. Untuk diubah menjadi satuan byte maka dibagi 8 yaitu 76/8 = 9.5 byte.

e. Melakukan hasil string bit *elias gamma code* nilai *file*.
Sebelum melakukan hasil *string* bit *elias gamma code* menjadi nilai *file*, terlebih dahulu dilakukan pemeriksaan terhadap panjang *string* bit pembentukan nilai bit baru hasil kompresi dari susunan nilai *hexadecimal* sebelum dikompresi adalah 6D,70,34,32,69,73,6F,6D,00,00,AF,6C,6D,6F,6F,76,00,00,00,6C.(tanpa tanda koma dan spasi) menjadi nilai *bit* biner:

"0100010100110001110001000000100101101011000101000100010011011000101111100100 ".

Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *string bit* itu sendiri yaitu *padding* bit dan *flag* bit. Jika sisa bagi panjang *string* bit terhadap 8 adalah 0 maka tambahan 00000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi panjang *string* bit terhadap 8 adalah n (1,2,3,4,5,6,7) maka tambahkan 0 sebnyak 7 – *n* + "1" di akhir *string* bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari 9 – n. nyatakan dengan bit akhir. Karena jumlah string bit 76 tidak habis dibagi 8 dan sisanya 4 bit, nyatakan sisa bagi tersebut dengan nilai *n*. maka tambahkan 0 sebanyak 0 sebanyak 7 – n + "1" di akhir string bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari 9 – n. Nyatakan dengan bit akhir.

7 – n + "1"
7 – 4+ "1"=**"0001"**
Bit Akhir 9 – n
Bit Akhir = 9 – 4 = 5 = **00000101**

ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

0100010100110001110001000000100101101011000101000100010011011000101111100100**000100000101**

Total panjang bit keseluruhan setelah ada penambahan bit adalah 76+4+8=88. Selanjutnya lakukan pemisahan bit manjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti gambar di bawah ini

01000101	00110001	11000100	00001001	01101011	00010100	01000100
11011000	10111110	0100**0001**	**00000101**			

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 11 kelompok nilai biner baru (11 byte) yang sudah terkompresi beserta nilai biner penambahan bit. Setelah pembagian dilakukan, maka nilai yang sudah dibagi diubah kedalam suatu karakter dengan terlebih dahulu mencari nilai *hexadecimal* dari *string* bit tersebut menggunakan kode ASCII untuk mengetahui nilai yang sudah terkompresi. Adapun nilai yang sudah terkompresi

Tabel 7. Nilai *Hexadecimal* Terkompresi

Urutan Nilai Terkompresi	Nilai *Hexadecimal* Terkompresi
1	45
2	31
3	C4
4	9
5	6B
6	14
7	44
8	D8
9	BE
10	41
11	5

Persentase ukuran data yang telah dikompresi dan didapat dari hasil perbandingan antara ukuran data setelah dikompresi dengan ukuran data sebelum dikompresi.
Ukuran data sebelum dikompresi = 160/8=20 byte
Ukuran data sesudah dikompresi = 88/8=11 byte
Berdasarkan data tersebut dapat dihitung kinerja kompresinya yaitu :
Compression Ratio (Cr)

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

$$Cr=\frac{11}{20}x100\%$$

$$Cr=55\%$$

Tabel 8. hasil *file audio* setelah dikompresi

	Keterangan
Jenis File	.MP3
Judul	Didoa Ibuku Ada Namaku Disebut
Ukuran	1, 85 MB
Durasi	4:11 Menit

2. Analisa proses dekompresi *file* video dengan menggunakan *Elias Gamma*
   a. Memasukkan hasil *file* kompresi
      Adapun nilai hexadecimal dari hasil kompresi yaitu: 45, 31, C4, 9, 6B, 14, 44, D8, BE, 41, 5.
   b. Hasil string bit *elias Gamma code* yang menjadi nilai *file* diubah kembali ke dalam bentuk biner. Adapun bit keseluruhan hasil kompresi dapat dilihat pada tabel berikut :

Tabel 9. Nilai *Hexadecimal* dan Biner Kompresi

Urutan Nilai Terkompresi	Nilai *Hexadecimal*	Nilai Biner
1	45	01000101
2	31	00110001
3	C4	11000100
4	9	00001001
5	6B	01101011
6	14	00010100
7	44	01000100
8	D8	11011000
9	BE	10111110
10	41	01000001
11	5	00000101

ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi

0100010100110001110001000000100101101011000101000100010011011000101111100100**000100000101**

c. Mengembalikan *binary* menjadi *string* bit semula dengan menghilangkan biner yang ditebalkan. Untuk mengembalikan *binary* menjadi *string bit* semula dapat dilakukan melalui langkah berikut ini. Lakukan pembacaan pada 8 bit terakhir, hasil pembacaan berupa bilangan desimal. Nyatakan hasil pembacaan dengan n. Hilangkan bit pada bagian akhir sebanyak 7+n. Setelah dilakukan perhitungan pembacaan bit akhir. Hilangkan 7 + n atau 7+5 = 12. Penjelasan diatas menujukan bahwa bit akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula adalah

0100010100110001110001000000100101101011000101000100010011011000101111100100

Berdasarkan perhitungan dengan algoritma *elias gamma code*, *string bit* pada diatas berjumlah 76 bit seperti diawal sehingga dilakuka pembacaan *string bit* awal. Adapun tabel hasil perhitungan diatas adalah:

Tabel 10. Pengecekan Bit

Indeks	Nilai	Keterangan
1	0	Tidak Ada
2	01	Tidak Ada
3	010	Ada Pada Tabel
4	0	Tidak Ada
5	00	Tidak Ada
6	001	Tidak Ada
7	0010	Tidak Ada
8	00101	Ada Pada Tabel
9	0	Tidak Ada
10	00	Tidak Ada
11	001	Tidak Ada
12	0011	Tidak Ada
13	00110	Ada Pada Tabel
14	0	Tidak Ada
15	00	Tidak Ada
16	001	Tidak Ada
17	0011	Tidak Ada
18	00111	Ada Pada Tabel
19	0	Tidak Ada
20	00	Tidak Ada
21	000	Tidak Ada
22	0001	Tidak Ada
23	00010	Tidak Ada
24	000100	Tidak Ada
25	0001000	Ada Pada Tabel
26	0	Tidak Ada
27	00	Tidak Ada
28	000	Tidak Ada
29	0001	Tidak Ada
30	00010	Tidak Ada
31	000100	Tidak Ada
32	0001001	Ada Pada Tabel
33	0	Tidak Ada
34	01	Tidak Ada
35	011	Ada Pada Tabel
36	0	Tidak Ada
37	01	Tidak Ada
38	010	Ada Pada Tabel
39	1	Ada Pada Tabel
40	1	Ada Pada Tabel
41	0	Tidak Ada
42	00	Tidak Ada
43	000	Tidak Ada
44	0001	Tidak Ada

ISSN 2745-7966 (Media Online)
Vol 2, No 6, Juli 2022
Hal 265-273
https://djournals.com/resolusi

Indeks	Nilai	Keterangan
45	00010	Tidak Ada
46	000101	Tidak Ada
47	0001010	Ada Pada Tabel
48	0	Tidak Ada
49	00	Tidak Ada
50	001	Tidak Ada
51	0010	Tidak Ada
52	00100	Ada Pada Tabel
53	0	Tidak Ada
54	01	Tidak Ada
55	010	Ada Pada Tabel
56	0	Tidak Ada
57	01	Tidak Ada
58	011	Ada Pada Tabel
59	0	Tidak Ada
60	01	Tidak Ada
61	011	Ada Pada Tabel
62	0	Tidak Ada
63	00	Tidak Ada
64	000	Tidak Ada
65	0001	Tidak Ada
66	00010	Tidak Ada
67	000101	Tidak Ada
68	0001011	Ada Pada Tabel
69	1	Ada Pada Tabel
70	1	Ada Pada Tabel
71	1	Ada Pada Tabel
72	0	Tidak Ada
73	00	Tidak Ada
74	001	Tidak Ada
75	0010	Tidak Ada
76	00100	Ada Pada Tabel

Maka dari penjabaran diatas dapat dibentuk tabel *elias gamma code* dan nilai *file* awal.

Tabel 11. Nilai *file* dan *Elias Gamma Code*

N	*Elias Gamma Code*	Nilai *File*
1	010	6D
2	00101	70
3	00110	34
4	00111	32
5	0001000	69
6	0001001	73
7	011	6F
8	010	6D
9	1	00
10	1	00
11	0001010	AF
12	00100	6C
13	010	6D
14	011	6F
15	011	6F
16	0001011	76
17	1	00
18	1	00
19	1	00
20	00100	6C

Berdasarkan hasil dekompresi diatas, maka didapatkan nilai *hexadecimal* awal sebelum kompresi yaitu 6D, 70, 34, 32, 69, 73, 6F, 6D, 00, 00, AF, 6C, 6D, 6F, 6F, 76, 00, 00, 00, 6C.

4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. Adapun kesimpulan tersebut Tahapan proses pengkompresian *file* audio hasil rekaman dari aplikasi wesing karaoke yaitu, dengan memilih *file* audio kemudian dikompres, sehingga ukuran *fil*e audio tersebut menjadi lebih kecil dari ukuran semula. Penerapan algoritma *elias gamma code* dalam mengkompresi *file* audio hasil rekaman dari aplikasi wesing karaoke telah membuktikan bahwa *file* audio yang memiliki kapasitas yang besar dapat dikompresi menjadi lebih kecil

REFERENCES

[1] A. A. Zulen, "Penerapan Pohon Biner Huffman Pada Kompresi Citra AlvinnAndhikaaZulenn(13507037))," no. 10, pp. 1–8, 2010, [Online]. Available: informatika.stei.itb.ac.id.
[2] N. C. Rowe, *Digital Multimedia*. 2011.
[3] I. Algoritma and H. Dan, "Implementasi Algoritma Huffman Dan Lz78 Untuk Kompresi Data," *J. Ris. Komput.*, vol. 3, no. 6, pp. 42–44, 2016, [Online].Available:https://www.researchgate.net/publication/317671197_IMPLEMENTASI_ALGORITMA_HUFFMAN_DAN_LZ78_UNTUK_KOMPRESI_DATA.
[4] D. Salomon, *Data Compression*, 4th ed. California: Spinger Verlag London New York, 2007.
[5] D. Salomon, *Data Compression*, 4th ed. California: Spinger Verlag London New York, 2007.
[6] H. O. D. COMPRESSION, *David Salomon, Giovani Motta*, Fifth. Spinger London Dordrecht Heidelberg New York, 2010.
[7] G. G. Maulana, "PEMBELAJARAN DASAR ALGORITMA DAN PEMROGRAMAN MENGGUNAKAN EL-GORITMA BERBASIS WEB," *J. Tek. Mesin*, 2017, doi: 10.22441/jtm.v6i2.1183.
[8] R. A.S, *REKAYASA PERANGKAT LUNAK TERSTRUKTUR DAN BERORIENTASI OBJEK*. Bandung: Informatika Bnadung, 2016.
[9] A. Nugroho, *Pemograman JAAVA untuk Aplikasi Basis Data dengan Teknik XP Menggunakan IDE Eclipse*, 1st ed. Yogyakarta, 2007.
[10] M. S. Alfa Satyaputra, *JAVA for Beginners with eclipse 42 Juno*. Jakarta, 2012.
[11] M. S. Alfa Satyaputra, *JAVA for Beginners with eclipse 42 Juno*. Jakarta, 2012.
[12] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
[13] S. R. Saragih and D. P. Utomo, "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
[14] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.