

Kompresi File Advanced Audio Coding (AAC) Menggunakan Metode Lempel Ziv Oberhumer (LZO)

Ginta Enjelita Purba

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: gintaenjelitapurba@gmail.com

Abstrak—Dalam era teknologi yang semakin canggih seperti saat ini. Kebutuhan terhadap kapasitas penyimpanan yang semakin besar merupakan penyebab munculnya berbagai teknik Kompresi. Semakin banyak data yang tersimpan maka akan semakin besar pula kapasitas yang dibutuhkan. Ukuran data yang besar mengakibatkan pemborosan memori dan lambatnya proses pemindahan data. Penyimpanan data yang semakin lama akan mengakibatkan menumpuknya data-data yang semakin banyak pula. Seiring perkembangan teknologi saat ini. Maka semakin berkembangnya kemampuan dalam mengolah data yaitu dengan memanfaatkan metode kompresi yang diharapkan dapat menjawab masalah tersebut. Berdasarkan permasalahan tersebut maka perlu untuk meminimalisir penyimpanan dengan menggunakan metode Lempel Ziv Oberhumer digunakan untuk mengkompresi file Advanced Audio Coding (AAC) agar memperkecil ukuran bit. Sehingga dengan menerapkan metode Lempel Ziv Oberhumer untuk mengkompresi file dapat mengurangi kapasitas penyimpanan secara optimal.

Kata Kunci: Kompresi; AAC; Lempel Ziv Oberhumer

Abstract—In an age of increasingly sophisticated technology such as today. The need for greater storage capacity is responsible for the emergence of various compression techniques. The more file is stored, the greater the capacity. The enormous size of file results in an inefficient memory and slow-motion process of file transfer. Getting older file storage results in more and more file piling up. As technology progresses today. And thus, a growing ability to process file is utilizing a compression method that is expected to address the problem, it will be necessary to minimize storage by using Lempel Ziv Oberhumer methods to compress Advanced Audio Coding (AAC) files to narrow the size of the beets. So by applying an Lempel Ziv Oberhumer to compression the file could reduce storage capacity optimally.

Keywords: Compression; AAC; Lempel Ziv Oberhumer

1. PENDAHULUAN

Kompresi data adalah proses mengubah sebuah aliran data input menjadi aliran data baru yang memiliki ukuran lebih kecil. Aliran yang dimaksud adalah berupa file atau pun buffer dalam memori. Terdapat banyak metode untuk kompresi data. Lossy dan Lossless Compression adalah pengelompokan metode kompresi berdasarkan keutuhan data. Kompresi file dapat dilakukan dengan cara memenuhi isi file yang berukuran besar sehingga ukurannya menjadi lebih kecil tanpa menghilangkan keaslian dari data tersebut [1].

Advanced Audio Coding (AAC) merupakan standart format berkas audio terkompresi. AAC umumnya memiliki kualitas suara yang lebih baik dibandingkan dengan format populer MP3 dalam bitrate yang sama khususnya pada bitrate dibawah 100 kbit/s. AAC merupakan format yang umum digunakan ketika melakukan kompresi CD audio pada Apple iPod dan iTunes (ekstensi.aac). Format ini merupakan bagian standart Motion Picture Experts Group (MPEG).

Permasalahan yang timbul adalah File audio perlu dikompresi. Pada saat ingin mengirimkan sebuah file, sangat banyak kendala yang dihadapi dalam pengiriman file seperti kualitas audio, kapasitas atau pun ukuran, sehingga file yang hendak dikirim akan mengalami kendala, oleh karena itu diperlukan suatu metode yaitu Metode Lempel-Ziv-Oberhumer (LZO) yang berfungsi mengecilkan atau pun mengurangi ukuran file tanpa mengurangi kualitas suara yang dapat diterima, tetapi sangat hemat bitrate atau bandwidth. Sehingga memudahkan dalam pengiriman data. Salah satu teknik sederhana yang saat ini dibutuhkan untuk masalah diatas adalah kompresi file. Kompresi file adalah proses mereduksi suatu data dengan meminimalkan ukuran dari data tersebut dengan tujuan memampatkan ukuran data tanpa menghilangkan keaslian dari data tersebut. Penyimpanan file audio yang memiliki ukuran lebih besar dapat dikompres menjadi ukuran yang lebih kecil dan mengurangi redundansi dari data-data yang terdapat dalam suatu file sehingga dapat disimpan atau ditransmisikan secara efisien. Salah satu metode yang digunakan dalam melakukan kompresi atau pemampatan data yaitu metode Lempel Ziv Oberhumer untuk mengompresi file audio tersebut. LZO adalah metode kompresi lossless yang memberikan kecepatan lebih dari kompresi diharapkan dengan memperkecil ukuran dari file yang ada dapat mengefisienkan penyimpanan yang tersedia.

Berdasarkan penelitian terdahulu yang dilakukan oleh Ronal Watrianthos, Marnis Nasution dengan judul “Analisa Kemampuan Transver Data Vpn Berbasis Open Source Pada Kondisi Encripsi-Dekripsi dan Kompresi-Dekompresi Kecepatan” transfer ini sangat di pengaruhi oleh bentuk data yang di transfer. Hal ini juga akan berlaku pada data yang sudah dikompresi dan tidak dikompresi. Analisa untuk mengetahui perbedaan transfer rate ini sangat di perlukan untuk menentukan VPN yang terbaik. Pengujian untuk mencari peforma transfer rate terbaik akan di lakukan terhadap ke semua data yang di uji yaitu data yang dienkripsi, data yang tidak dienkripsi (dekripsi), data yang dikompresi dan data yang tidak di kompresi (dekompresi) [2].

Penelitian juga dilakukan oleh Maria Roslin Apriani Neta yang judul “Perbandingan Algoritma Kompresi Terhadap Objek Citra Menggunakan JAVA pemampatan data atau kompresi data adalah sebuah cara untuk memadatkan data sehingga hanya memerlukan ruangan penyimpanan lebih kecil sehingga lebih efisien dalam menyimpannya atau mempersingkat waktu pertukaran data tersebut. Kebalikan dari proses kompresi adalah dekompresi, yang merupakan

proses pengembalian data yang telah dihasilkan oleh proses kompresi citra. Untuk memperoleh hasil uji sebagai perbandingan kompresi objek digital, karena ada data yang hilang atau dianggap tidak terlalu penting mengubah informasi, maka disebut lossy compression [3].

Selain itu penelitian juga dilakukan oleh Akshay S dan Apoorva P dengan judul “Algoritma Multicast Routing Dioptimalkan Bandwidth Berdasarkan Hybrid Mesh dan Struktur Pohon dengan Kontrol Tabrakan di Manet menggunakan Metode Lempel-Ziv-Oberhumer” menunjukkan peran penting untuk komunikasi sementara dan cepat yang memberikan jaminan dan keberhasilan pengiriman. Karena keberadaan hanya satu tautan antara dua node berbasis Pohon memberikan konsumsi bandwidth yang rendah, efisiensi penerusan yang tinggi. Protokol routing multicast berbasis pohon paling cocok untuk jaringan yang ringan. Kualitas Layanan dan kinerja terdegradasi ketika beban jaringan meningkat. Dengan menggabungkan manfaat struktur pohon dan struktur mesh efek beban jaringan dapat ditingkatkan [4].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Adapun tahapan penelitian pada penelitian ini sebagai berikut:

1. Identifikasi Masalah
Pada tahap ini masalah yang akan diteliti adalah kompresi File Advanced Audio Coding (AAC)
2. Studi Literatur
Pada tahap ini penulis mengumpulkan referensi yang diperlukan dalam penelitian, yang mana dilakukan untuk memperoleh data-data atau informasi yang dibutuhkan dalam penelitian ini. Pada tahap studi literatur dilakukan pengumpulan buku, jurnal, e-book, artikel, makalah, maupun situs internet yang membahas metode LZO (Lempel Ziv Oberhumer) untuk dipelajari lebih lanjut.
3. Analisis
Pada tahap ini dilakukan analisis terhadap bahan referensi yang telah dikumpulkan pada tahap sebelumnya untuk mendapatkan pemahaman mengenai metode Lempel Ziv Oberhumer untuk mengkompresi file audio. Pada tahap ini dilakukan proses kompresi file audio dengan menerapkan metode Lempel Ziv Oberhumer.
4. Perancangan Sistem
Pada tahap ini dilakukan perancangan aplikasi seperti pembuatan flowchart, pembuatan UML (Unified Modelling Language) meliputi usecase diagram, activity diagram serta sequence diagram dan rancangan untuk desain antarmuka aplikasi.
5. Pengujian
Pengujian sistem yang sudah dikembangkan dengan sistematis yang sudah dirancang sedemikian rupa untuk melihat kompresi file AAC memberikan hasil yang diinginkan.
6. Implementasi
Pada tahap implementasi sistem ini dilakukan implementasi terhadap hasil perancangan aplikasi kompresi file advanced audio coding. Dan untuk melihat apakah metode (LZO) dalam mengkompresi file ACC dapat menyelesaikan masalah.
7. Dokumentasi
Dalam tahap dokumentasi dilakukan penyusunan laporan dari hasil perancangan sistem dalam format penulisan penelitian.

2.2 Kompresi Data

Kompresi data dalam bidang ilmu komputer, ilmu pengetahuan dan seni adalah sebuah penyajian informasi ke dalam bentuk yang lebih sederhana. Kompresi data atau source coding dalam ilmu komputer dan teori informasi adalah proses mengencode informasi dengan menggunakan lebih sedikit bit dari suatu sumber yang belum di-encode melalui penggunaan skema pengkodean yang spesifik. Kompresi data dapat diartikan juga sebagai proses yang dapat mengubah sebuah aliran data masukan (sumber atau data asli) ke dalam aliran data yang lain (keluaran atau data yang dimampatkan) yang memiliki ukuran yang lebih kecil. Kompresi data sudah ada sejak puluhan tahun yg lalu. Kompresi data menjadi salah satu cabang Teori Informasi karena kompresi data berkecimpung dengan masalah redundancy dalam suatu informasi[5][6].

Untuk mengetahui kecepatan dari hasil suatu kompresi dengan menggunakan metode tertentu ada tiga parameter yang digunakan dalam mengukur hasil kompresi dapat dilihat sebagai berikut [7].

1. Rasio Kompresi
Rasio kompresi adalah suatu parameter kompresi untuk membandingkan hasil data sebelum di kompresi dengan data yang setelah di kompresi. Rasio hasil kompresi sebagai indikator untuk mengetahui hasil dari sebuah kompresi. Berikut formula penulisan rasio kompresi.

$$Rc = \frac{\text{Jumlah Bit sebelum dikompresi}}{\text{Jumlah Bit sesudah dikompresi}}$$
2. Compression Ratio (CR)

Compression Ratio (CR) merupakan persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.

$$CR = \frac{\text{Jumlah Bit sebelum dikompresi}}{\text{Jumlah Bit sesudah dikompresi}} \times 100\%$$

3. Redudancy (Rd)

Redudancy (Rd) merupakan kelebihan yang terdapat didalam data sebelum dikompresi. Jadi setelah dikompresi dapat dihitung Redudancy data yaitu persentasi dari hasil selisih antara ukuran data sebelum dikompresi dengan data setelah dikompresi.

$$Rd = 100\% - CR$$

2.3 Metode Lempel Ziv Oberhumer

Metode Algoritma LZO yang menggunakan hanya satu tread untuk kompresi, LZO adalah algoritma kompresi lossless yang memberikan kecepatan lebih dari dekompresi. Rasio kompresi didefinisikan sebagai $C=CDS/UCDS$.

Metode LZO merupakan pustaka data lossless portable yang ditulis dalam ANSI C yang menawarkan kompresi yang cukup cepat dan dekompresi yang sangat cepat. Dikembangkan oleh Markus FXJ diluncurkan pertama pada 20 Mei 1996. Memori tambahan untuk dekompresi tidak diperlukan oleh format dan algoritma LZO karena hanya membutuhkan 64KB untuk kompresi. Berikut Rumus Kompresi dengan menggunakan metode LZO.

Rumus:

$$D_T(I) = C * R * S / (1-i) * 120\% \quad (1)$$

Di mana,

C = adalah rasio kompresi (kompresi LZO),

R = adalah faktor replikasi,

S = adalah jumlah awal data dan saya disebut faktor data menengah.

120% = adalah 1,2 kali ukuran total data.

1. Encode LZO Dapatkan Ukuran untuk melakukan struktur penyandian, ukuran buffer memori perangkat seluler dihitung,

$$M(E)(S)() = IL(DEID \rightarrow (32 * V)) \quad (2)$$

$D_{M(E)(S)}(SAYA)$ = ukuran perangkat seluler yang disandikan

D = metode data tertentu

IL = menandakan panjang input data perangkat seluler, $32 * V$ (variabel input). Fungsi ini menggunakan metode struktur status LZO perangkat seluler dan mengompresi data sumber.

2. Menyandikan LZO

Struktur penyandian perangkat seluler dalam buffer eksternal diinisialisasi oleh fungsi ini.

$$(I)(E)(M)() = \text{saya } IL(DEID \rightarrow (8 * V)) \quad (3)$$

$D_{M(E)(I)}(SAYA)$ = menginisialisasi yang dikodekan dalam buffer eksternal,

$8 * V$ = variabel input pointer

2.4 File Audio AAC

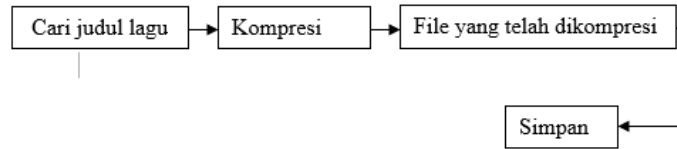
Audio atau suara merupakan gelombang yang merambat melalui medium tertentu. Audio atau suara merupakan gelombang yang mengandung sejumlah komponen penting (amplitudo, panjang gelombang dan frekuensi) yang dapat menyebabkan suara yang satu berbeda dari suara lain. AAC adalah singkatan dari Advanced Audio Coding merupakan standar format berkas audio terkompresi. AAC umumnya memiliki kualitas suara yang lebih baik dibandingkan dengan format populer MP3 dalam bitrate yang sama khususnya pada bitrate di bawah 100 kbit/s. AAC merupakan format yang umum digunakan ketika melakukan kompresi CD audio pada Apple iPod dan iTunes (eksensi .m4a). AAC memiliki keterbatasan diakibatkan lebih berat, lebih rumit, dan mempunyai dukungan yang minim dari para pengembang. AAC Bukan untuk Penyimpanan. Dari hasil pengujian terlihat bahwa AAC sangat superior dalam menghasilkan audio yang "masih bisa didengar" pada bit rate yang sangat kecil (15 Kbps). Kualitas suara jauh dari memuaskan, tetapi pada bit rate minimal masih mampu memberikan kualitas suara yang "cukup baik". Pada bit rate satu lagu hanya membutuhkan sekitar 500 KB. Sehingga dengan demikian mempunyai efisiensi yang sangat tinggi. AAC tidak terlalu istimewa pada bit rate tinggi. Berdasarkan pengamatan masih memiliki kualitas yang kurang lebih sama dengan format-format yang sudah ada (MP3, WMA, ataupun OGG)[9].

3. HASIL DAN PEMBAHASAN

Analisa dan perancangan yang dilakukan dalam penelitian ini yaitu mengompresi *file advanced audio coding(aac)* dengan menggunakan metode *lempel ziv oberhumer(lzo)* tujuannya untuk mengetahui ukuran audio setelah dikompresi . Pada saat ini banyak orang-orang mengoleksi maupun mengumpulkan *file* audio 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 ruang dalam memori. *File* audio yang akan di kompresi adalah *file* advanced audio coding (AAC) hasil download dari sebuah situs internet yang mempunyai format AAC.

Kompresi dilakukan untuk mengurangi kapasitas dari suatu file dimana ukuran dari sebuah file sangat mempengaruhi tempat penyimpanan, semakin besar ukuran file maka semakin besar pula ruang penyimpanan yang dibutuhkan karena itu diperlukan kompresi agar ukuran file bisa menjadi lebih kecil, sehingga media penyimpanan dapat menjadi lebih efisien. Adapun diagram kompresi *file* AAC dapat dilihat pada gambar di bawah:

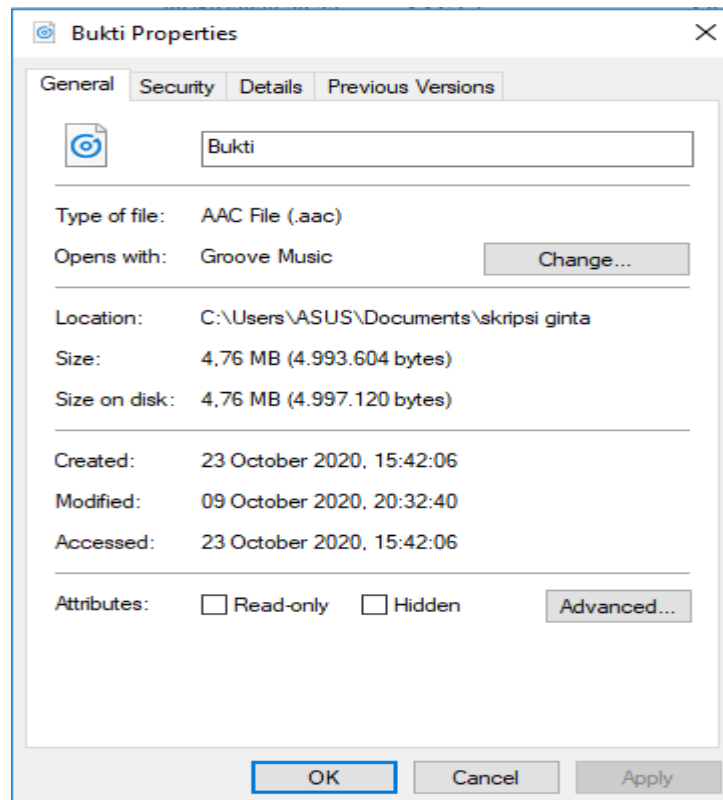


Gambar 1. Skema Analisa Yang Digunakan

3.1 Penerapan Lempel Ziv Oberhumer

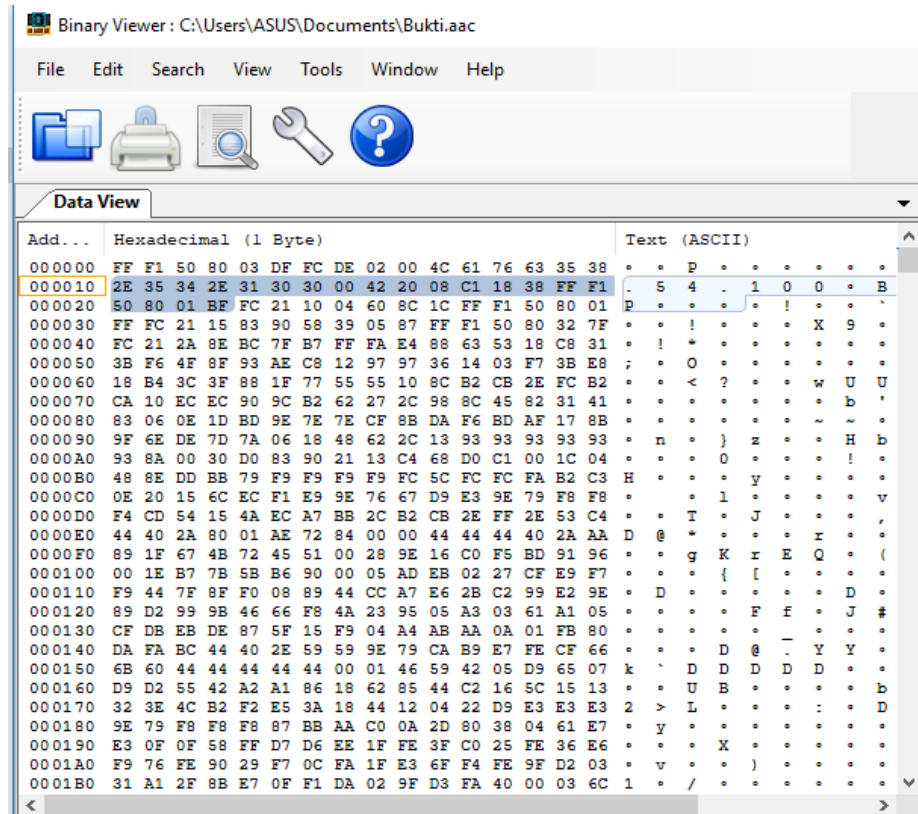
Berdasarkan analisa, dengan melakukan kompresi data, data yang berukuran besar akan dikompresi menjadi ukuran lebih kecil dan akan mengurangi tempat penyimpanan. Dalam menganalisa *file* audio harus dilakukan pengambilan *sample file* untuk mendapatkan nilai data pada sebuah *file* audio yang berupa nilai hexadecimal. Berikut adalah langkah untuk mengkompresi *file* audio.

- Sebelum perancangan, masukkan tabel hasil dari sebuah aplikasi. File audio yang digunakan adalah *file* Advanced Audio Coding (AAC) yang dihasilkan dari sebuah aplikasi. Berikut adalah informasi objek *file* audio yang akan diambil sampelnya sebelum dilakukan kompresi:



Gambar 2. Informasi *File* audio sebelum dikompresi

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



Gambar 3. Nilai Hexadecimal File audio Sample

Berdasarkan pada gambar 3 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 2E,35,34,2E,31,30,30,00,42,20,08,C1,18,38,FF,F1,50,80,01,BF. Nilai data ini dimasukkan kedalam tabel untuk dilakukan pembacaan frekuensi. Pembacaan frekuensi dilakukan dengan menghitung jumlah nilai yang sama di setiap nilai data yang muncul. Tahapan kompresi ditunjukkan pada tabel 1.

Tabel 1. Tahapan Kompresi

Langkah	Posisi, [kode] & karakter (P)	Gabungan Posisi & Karakter (Q)	Dictionary (P+Q)	Kode	Output
1	inisialisasi		2E	[1]2E	
2			35	[2]35	
3			34	[3]34	
4			31	[4]31	
5			30	[5]30	
6			00	[6]00	
7			42	[7]42	
8			20	[8]20	
9			08	[9]08	
10			C1	[10]C1	
11			18	[11]18	
12			38	[12]38	
13			FF	[13]FF	
14			F1	[14]F1	
15			50	[15]50	
16			80	[16]80	
17			01	[17]01	
18			BF	[18]BF	

Tabel 2. Tahapan Kompresi Selanjutnya

Langkah	Posisi,[kode] & karakter (P)	Gabungan posisi & Karakter (Q)	Dictionary (P + Q)	Kode	Output
1	[1]2E	[35]1+2	2E,35		[1]
2	[2]35	[34]2+3	35,34		[2]
3	[3]34	[31]3+4	34,31		[3]
4	[4]31	[30]4+5	31,30		[4]
5	[5]30	[00]5+6	30,00		[5]
6	[6]00	[42]6+7	00,46		[6]
7	[7]42	[20]7+8	42,20		[7]
8	[8]20	[08]8+9	20,08		[8]
9	[9]08	[C1]9+10	08,C1		[9]
10	[10]C1	[18]10+11	C1,18		[10]
11	[11]18	[38]11+12	18,38		[11]
12	[12]38	[FF]12+13	38,FF		[12]
13	[13]FF	[F1]13+14	FF,F1		[13]
14	[14]F1	[50]14+15	F1,50		[14]
15	[15]50	[80]15+16	50,80		[15]
16	[16]80	[01]16+17	80,01		[16]
17	[17]01	[BF]17+18	01,BF		[17]
18	[18]BF	Habis			[18]

Berdasarkan perhitungan kompresi diatas maka didapat hasil kompresi adalah sebagai berikut ini.

Tabel 3. Hasil Kompresi

Dictionary	Output
2E,35	[1]
35,34	[2]
34,31	[3]
31,30	[4]
30,00	[5]
00,46	[6]
42,20	[7]
20,08	[8]
08,C1	[9]
C1,18	[10]
18,38	[11]
38,FF	[12]
FF,F1	[13]
F1,50	[14]
50,80	[15]
80,01	[16]
01,BF	[17]

Apalagi dilakukan kompresi menggunakan metode LZ0, maka hasilnya akan menjadi : [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Output [1] artinya di dictionary dikodekan menjadi 2E,35, *output* [2] Total awal bit disimpan tanpa kompresi = Total *input* * bit dictionary

$$= 20 * 8$$

$$= 160 \text{ bit}$$

Besar *file* setelah dikompresi = Total *output* * bit dictionary

$$= 17 * 8$$

$$= 136 \text{ bit}$$

Hasil Rasio Kompresi :

$$Rasio = 100 - \frac{\text{ukuran file terkompresi}}{\text{ukuran file asli}} \times 100\%$$

$$Rasio = 100 - (136/160) \times 100\%$$

$$Rasio = 100 - (0,85) \times 100\%$$

$$Rasio = 100 - 85\%$$

$$Rasio = 15\%$$

Berdasarkan perhitungan di atas didapat rasio 15% yang artinya memori berhasil menghemat memori sebesar 15 persen. 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 = $136/8=17$ byte

Maka hasil dari kompresi dapat diukur kinerjanya berdasarkan metode *lempel ziv oberhumer*:

a. *Ratio of Compression (R_c)*

$$R_c = \frac{\text{Jumlah Bit sebelum dikompresi}}{\text{Jumlah Bit sesudah dikompresi}}$$

$$R_c = \frac{160}{136}$$

$R_c = 1.17$ Hasil dari pembagian data sebelum dan sesudah dikompresi

b. *Compression Ratio (C_R)*

$$C_R = \frac{\text{Jumlah Bit sesudah dikompresi}}{\text{Jumlah Bit sebelum dikompresi}} \times 100\%$$

$$C_R = \frac{136}{160} \times 100\%$$

$C_R = 0.85\%$ Hasil dari pembagian data sesudah dan sebelum dikompresi lalu di kali 100 %

c. *Redundancy (R_d)*

$$R_d = 100\% - C_R$$

$$R_d = 100\% - 0,85\%$$

$R_d = 0,15\%$ Hasil kelebihan atau pemborosan dari hasil CR diatas

4. KESIMPULAN

Berdasarkan pembahasan dan analisa terhadap kompresi file advanced audio coding menggunakan metode lempel ziv oberhumer, dimana proses kompresi pada file audio dilakukan dengan proses yang terdapat pada metode lempel ziv oberhumer(lzo). Metode Lempel Ziv Oberhumer dapat diterapkan untuk melakukan kompresi karakter yang terdapat pada file audio sehingga ukuran file menjadi lebih kecil. Aplikasi kompresi file audio telah selesai dirancang dengan menggunakan Eclipse dimana file audio didalamnya telah terkompresi dengan menggunakan metode lempel ziv oberhumer.

REFERENCES

- [1] A. A. Guci, M. Syahrizal, and P. B. Simangunsong, "ANALISA DAN IMPLEMENTASI KOMPRESI CITRA CT SCAN MENGGUNAKAN METODE DELTA MODULATION," vol. 6, no. April, pp. 369–371, 2018.
- [2] S. D. Nasution, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Goldbach Codes," *Jurnal Infotek STIEKOM*, vol. 1, no. Februari, pp. 2008–2010, 2016.
- [3] H. Sartika, T. Zebua, and R. Parapat, "PERANCANGAN DAN IMPLEMENTASI ALGORITMA ELIAS GAMMA CODE UNTUK MENGKOMPRESI RECORD DATABASE PADA APLIKASI RANGKUMAN," vol. 3, pp. 259–265, 2019, doi: 10.30865/komik.v3i1.1600.
- [4] D. T. Informasi, D. T. Informasi, and B. Amman, "Retrieval data besar menggunakan hdfs dengan kompresi lzo," 2020.
- [5] D. Jurusan, T. Elektronika, F. Teknik, and U. N. Makassar, "TEKNOLOGI KOMPRESI AUDIO DENGAN ADVANCED AUDIO CODING (AAC) Tasri Ponta," pp. 665–673.
- [6] Sukanto dan and Shalahuddin(2015:29), *Shalahuddin, M. Rosa A.S 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Edisi REVISI*. 2015.
- [7] K. K. Budaya and T. Samosir, "APLIKASI EDUKASI BUDAYA TOBA SAMOSIR BERBASIS ANDROID," vol. 9, no. 1, pp. 9–18, 2016.
- [8] Sri Sulistiyani, *Step by Step Menjadi Programmer android*. Yogyakarta: Andi, Wahana KOMputer, 2013.
- [9] S. K. Alfa Satyaputra, M.Sc, Eva Maulina Aritonang, *JAVA for Beginners with eclipse 4.2 Juno*. Jakarta: PT Elex Media Komputindo Kelompok Gramedia, Anggota IKAPI, 2012.