

Analisis Kompresi dan Dekompresi Data Teks dan Audio dengan Algoritma *Run Length Encoding* (RLE)

Sylvia Rahma¹, Aditya Prapanca²

^{1,2}Jurusan Teknik Informatika/Teknik Informartika, Universitas Negeri Surabaya

¹sylvia.rahma16051204007@mhs.unesa.ac.id

²adityaprapanca@unesa.ac.id

Abstrak— Di era digital saat ini semakin berkembangnya dunia teknologi digital yang melakukan pengolahan data untuk pengiriman ataupun pembuatan data teks, audio, video, dan gambar yang berkualitas. Proses komunikasi data inilah yang sering mengalami hambatan dikarenakan ruang penyimpanan yang tidak cukup, *bandwidth* yang besar sehingga membutuhkan banyak waktu untuk menampilkan file dan kecepatan transmisi data multimedia. Maka dari itu diciptakanlah penyelesaian untuk memudahkan kendala tersebut dengan memampatkan suatu data atau disebut dengan kompresi data. Kompresi data adalah proses konversi data stream masukan menjadi aliran data lain dengan hasil file yang lebih kecil yang disebut dengan dekompresi. Proses ini dapat mengurangi ukuran asli file namun tidak menghilangkan data yang ada didalamnya. Salah satu metode yang akan digunakan peneliti pada penelitian ini untuk kompresi data teks dan audio yaitu menggunakan algoritma *Run Length Encoding* (RLE) dengan melalui dua metode yaitu proses kompresi dan dekompresi. Algoritma *Run Length Encoding* (RLE) merupakan kompresi data yang paling sederhana dan paling signifikan untuk digunakan apa saja, algoritma ini bekerja pada pengulangan karakter dalam string yang dimasukkan. Tujuan dari analisis penelitian ini yakni untuk mengukur kinerja rasio dan kecepatan kompresi dekompresi data pada format .docx, .txt, .pdf, .mp3, dan .wav. Hasil pengujian menunjukkan bahwa setiap ukuran data asli yang telah melalui proses kompresi menghasilkan angka yang berbeda-beda tergantung pada isi string karakter di dalamnya, semakin banyak string karakter berulang maka menghasilkan ukuran data yang lebih kecil. Begitu juga sebaliknya dengan proses dekompresi, dapat dikatakan berhasil jika ukuran file dan isi file dapat kembali seperti semula dan durasi yang dibutuhkan untuk melakukan kompresi dekompresi pun juga tergantung pada besar ukuran data asli.

Kata Kunci— *Run Length Encoding*, Kompresi, Dekompresi Audio, Teks.

I. PENDAHULUAN

Di era digital saat ini semakin berkembangnya dunia teknologi digital yang melakukan pengolahan data untuk pengiriman ataupun pembuatan data teks, audio, video, dan gambar yang berkualitas. Proses komunikasi data inilah yang sering mengalami hambatan dikarenakan ruang penyimpanan yang tidak cukup, *bandwidth* yang besar sehingga membutuhkan banyak waktu untuk menampilkan file dan kecepatan transmisi data multimedia.

Maka dari itu diciptakanlah penyelesaian untuk memudahkan kendala tersebut dengan memampatkan suatu data atau disebut dengan kompresi data. Adapun para ahli yang menyebutnya dengan *source coding*, kompresi *bandwidth*, dan kompresi sinyal [1].

Kompresi data adalah proses konversi data *stream* masukan menjadi aliran data lain dengan hasil file yang lebih kecil yang disebut dengan dekompresi sehingga dapat mengurangi waktu transmisi sewaktu data dikirimkan dan tidak banyak memenuhi ruang penyimpanan. Sebuah *stream* dapat berupa file yang berada di memori [2].

Pada umumnya teknik kompresi memiliki dua kriteria yaitu *lossy* dan *lossless* dengan memanfaatkan adanya redundansi informasi dengan pengkodean, pengenalan pola ataupun prediksi linear seperti pada kompresi video.

Pada kompresi *lossy*, hasil kompresi akan membuang sebanyak mungkin data ‘tidak relevan’ dari data asli sehingga menghasilkan data yang jauh lebih kecil. Sedangkan pada kompresi *lossless* hasil kompresinya tanpa kehilangan informasi atau tidak menurunkan kualitas sama sekali dan dapat dikembalikan seperti data asli tanpa ada perubahan. Maka rasio kompresi pun tidak terlalu besar.

Terdapat banyak sekali metode diantaranya yaitu seperti *Huffman Coding*, *Arithmetic Coding*, *Lempel-Zip-Welch* (LZW), *Run Length Encoding* (RLE) dan beberapa metode lainnya. Salah satu metode yang akan dianalisis oleh peneliti untuk kompresi data teks dan audio yaitu dengan menggunakan *Run Length Encoding* (RLE). Algoritma ini merupakan kompresi data yang paling sederhana dan paling signifikan untuk digunakan pada data apa saja. Metode ini bekerja pada pengulangan karakter dalam *string* yang dimasukkan.

Adapun hasil penelitian terdahulu yakni pada tahun 2013 oleh Aditya Rahandi, Dian Rachmawati, Sajadin Sembiring menganalisis dan mengimplementasikan kompresi file audio dengan menggunakan *algoritma Run Length Encoding* (RLE) kemudian berhasil memperoleh hasil pengujian sistem dengan 5 file audio. Kompresi file MP3 dengan rasio rata-rata 0.46% dan kompresi file WAV dengan rasio rata-rata 13.83% [3]. Hasil penelitian yang pernah dilakukan oleh Darno Willfrid Midukta Simamoral, Garuda Ginting, & Yasir Hasan pada tahun 2016 mencoba mengimplementasikan algoritma *Run Length Encoding* pada kompresi file mp3 dengan ukuran awal 4,321 kilobytes kemudian mendapatkan hasil dekompresi 4,382 kilobytes dengan rasio 99%. Kualitas mp3 berkurang sedikit tetapi tidak mengubah file aslinya [4]. Pada tahun 2017 hasil penelitian oleh Chetan R.Dudhagara dan Hasamukh B. Patel melakukan analisis performa dari kompresi data teks dengan menggunakan *Run Length Encoding* dengan hasil 3 case yang berbeda, case I mendapatkan kompresi dengan rasio 0.25%, case II rasio 0.5%, dan case III mendapatkan -2.0% [5]. Hasil penelitian oleh Sanjoy Mitra dan Debaprasad Das pada

tahun 2017 melakukan analisis pada aplikasi pengkodean teknik *Run Length Encoding* dalam skema pengkodean gabungan dengan hasil rasio Simple RLE 50%, rasio *Optimized RLE* 49.19% dan RLE in 40.63% [6]. Kemudian hasil penelitian yang dilakukan oleh Helbert Sinaga, Poltak Sihombing, Handrizal pada tahun 2018 membandingkan algoritma *Huffman* dan *Run Length Encoding* untuk kompresi file audio, dengan metode RLE mendapatkan rata-rata rasio kompresi *.mp3 -94.44% dan *.wav -45.91% [7].

A. Interface Aplikasi

Sistem yang telah digunakan untuk proses kompresi dan dekompresi ini yaitu dengan menggunakan desain GUI (*Graphical User Interface*) dari software Apache NetBeans IDE 12.0 dengan bahasa Java. Jadi dengan menggunakan desain GUI ini dapat memudahkan pengguna sebagai antarmuka berinteraksi dengan sistem.

B. Tujuan Pengujian Kompresi Dekompresi RLE

Pengujian kompresi algoritma ini dilakukan untuk melihat hasil kinerja dari algoritma *Run Length Encoding*, dapat dilihat dari hasil yang berupa rasio kompresi, ukuran awal, ukuran akhir data dan lama proses kompresi. Ukuran data berupa dalam satuan byte dan untuk waktu berupa dalam *millisecond* (ms).

Jenis data yang diuji terbagi menjadi 2 kategori. Untuk kategori teks terbagi menjadi 3 yaitu format .docx, .txt, dan .pdf. Sedangkan untuk kategori audio terbagi menjadi 2 yaitu format .mp3 dan .wav dengan masing-masing 5 data.

C. Kompresi Dekompresi

Di dalam ilmu komputer, kompresi data biasa disebut dengan pemampatan data yang merupakan suatu proses kompresi-dekompresi yang hanya memerlukan ruangan penyimpanan lebih kecil sehingga dapat mempersingkat waktu untuk mengirimkan data yang berkualitas.

Kompresi data terdapat berbagai macam multimedia seperti gambar, audio, video dan teks. Dalam melakukan kompresi data dapat dilakukan 2 metode yaitu kompresi dan dekompresi.

Kompresi adalah metode untuk mencapai tujuan kompresi pada data asli, dibalik setiap algoritma teknik penerapannya dapat mencapai tujuan dengan mempertimbangkan aspek efisiensi dan keefektifan kompresi. Sedangkan untuk dekompresi adalah teknik untuk pengembalian data yang telah di kompresi agar data dapat kembali dalam bentuk asli [8].

Suatu metode kompresi yang optimal bila metode yang digunakan tersebut dapat menghasilkan rasio kompresi yang berukuran minimal setengah dari ukuran data asli.

D. Teks dan Audio

Teks merupakan suatu wacana yang difiksasikan dalam bentuk tulisan dan dapat diartikan sebagai satuan lingual yang dimediasi secara tulisan dengan tata organisasi tertentu sebagai ungkapan makna secara kontekstual. Contoh format atau ekstensi file teks seperti .docx, .txt dan .pdf. Kompresi data teks termasuk dalam *lossless*[9].

Sedangkan audio merupakan suara yang dihasilkan oleh getaran suatu benda agar bisa didengar telinga manusia, getaran tersebut harus cukup kuat minimal 20/detik. Banyaknya getaran suatu benda dapat diukur dengan satuan *Cycles Per Second* (CPS) atau biasa disebut dengan *Hertz* (Hz). Adapun beberapa format audio yang terkenal seperti .mp3, .wav, .wma, .aac dan lainnya.

Adapun penjelasan tentang MP3 dan WAV. MPEG-1 layer (MP3) termasuk dalam *lossy*, file tersebut tidak 100% berisi informasi audio asli. MP3 menggunakan pengkodean perseptual yang dimana dapat menghapus informasi yang tidak dapat diperhatikan oleh telinga manusia sehingga membuat file lebih kecil dan bahkan memungkinkan pengguna untuk mendownload file MP3 dengan kecepatan yang wajar. Sedangkan untuk *Waveform Audio* (WAV) merupakan format standar audio digital dari Windows. Menggunakan ekstensi file .WAV, sampel 8 atau 16-bit dapat diambil pada kecepatan 11.025 Hz, 22.050 Hz dan 44.100 Hz [10].

E. Run Length Encoding

Algoritma *Run Length Encoding* merupakan salah satu bentuk skema metode terbaik kompresi data *lossless* yang paling sederhana dan mudah dipahami. Metode ini digunakan untuk kompresi data yang berkarakter berulang dan bernilai sama lalu disimpan sebagai perhitungan data tunggal tetapi bukan sebagai proses asli. Metode ini sering digunakan untuk kompresi data teks dan audio.

Skema algoritma ini pertama kali digunakan dalam transmisi sinyal televisi analog pada tahun 1967. Di tahun 1983 *Run Length Encoding* dipatenkan oleh Hitachi. Algoritma ini cocok untuk menandai batas sebanyak jumlah karakter tersebut [11].

Sistem kerja metode *Run Length Encoding* adalah bekerja dengan mengurangi ukuran dengan pengulangan string dari deretan karakter, string pengulangan ini dinamakan RUN dan dikodekan dalam 2 bit. Bit pertama merupakan jumlah pengulangan dan bit kedua merupakan karakter yang diulang [12]. *Repetition Suppresion* melakukan perubahan angka atau huruf yang berulang diwakilkan dengan satu huruf dengan jumlahnya.

Contoh : 95900000 = 959A6

Pattern Substitution melakukan substitusi kata-kata menjadi huruf atau Simbol Kompresi.

Contoh : ABCCCCDEFFFFFG = 15 karakter

RLE tipe 1 minimal 4 huruf sama : ABC!5DEF!5G = 11 karakter

RLE tipe 2 1 : AB5CDE5FG = 9 karakter

Best case untuk RLE tipe 2 adalah ketika terdapat 127 karakter yang sama hingga dikompresi menjadi 2 byte saja. *Worst case* untuk tipe 2 ketika terdapat 127 karakter yang berbeda semua maka akan terdapat 1 byte tambahan yang tanda jumlah karakter yang tidak sama tersebut [12].

F. Rasio Kompresi

Persentase atau rasio kompresi merupakan perbandingan antara ukuran data asli yang sebelum melalui proses kompresi dengan ukuran data yang setelah melalui proses kompresi. Angka rasio kompresi menunjukkan perbandingan ukuran yang dicapai dalam satu proses kompresi.

Rumus rasio kompresi:

$$\text{Rasio kompresi} = \frac{\text{ukuran data output}}{\text{ukuran data input}} \quad (1)$$

Kompresi yang digunakan pada algoritma ini yakni termasuk pada lossless compression, dimana saat mengkompresi data hanya untuk diterima ditujukan dalam kondisi asli dan data yang telah dikompresi ukurannya sama atau lebih besar.

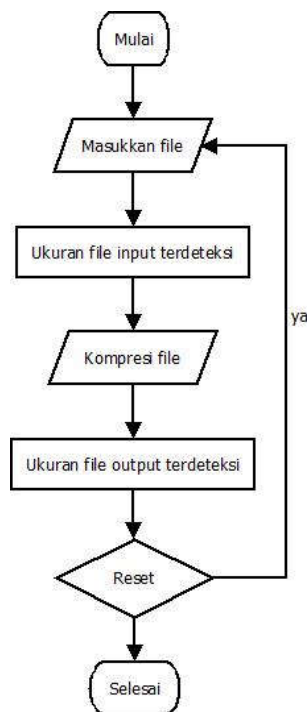
II. METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan tentang langkah-langkah metode penelitian analisa perancangan.

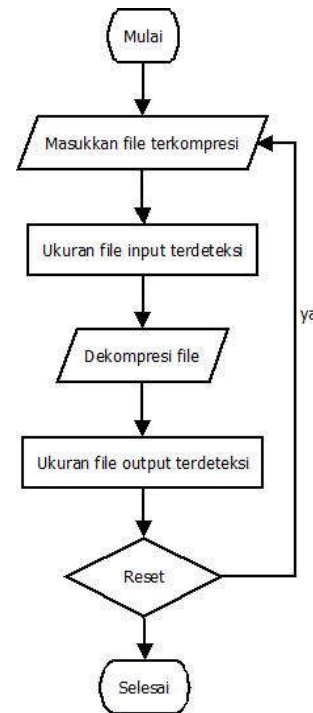
A. Flowchart Kompresi Dekompresi Data

Flowchart merupakan tipe dari diagram untuk menunjukkan proses alur kerja. Dapat diartikan juga dengan representasi diagram dari suatu algoritma dalam langkah untuk menyelesaikan tugas dari aplikasi yang akan dibuat.

Alur proses kompresi dan dekomposisi data akan ditunjukkan pada flowchart di bawah ini:



Alur sistem proses kompresi dari rancangan Gbr. 1 di atas menunjukkan ketika memasukan file yang dikompresi akan muncul ukuran awal suatu file dalam bentuk ukuran *byte*. Kemudian setelah proses perhitungan kompresi selesai maka akan muncul ukuran akhir hasil kompresi dalam bentuk ukuran *byte*, persentase kompresi dalam bentuk % dan waktu proses dalam bentuk ukuran *millisecond* (ms). Kemudian file hasil kompresi telah disimpan secara otomatis dengan nama *compressed*.



Gbr. 2 Flowchart Dekompresi

Sedangkan untuk alur sistem proses dekomposisi dari rancangan Gbr. 2 di atas setelah melakukan proses kompresi data maka selanjutnya adalah proses dekomposisi data yaitu dengan cara memasukan file terkompresi (*compressed*) yang telah disimpan kemudian akan muncul ukuran awal suatu file dalam bentuk ukuran *byte*. Kemudian setelah proses perhitungan kompresi selesai maka akan muncul ukuran akhir hasil kompresi dalam bentuk ukuran *byte*, persentase kompresi dalam bentuk % dan waktu proses dalam bentuk ukuran *millisecond* (ms). Kemudian file hasil dekomposisi juga telah disimpan secara otomatis dengan nama *decompressed*.

III. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dan pembahasan dari analisis penelitian yakni mengukur kinerja rasio dan kecepatan kompresi dekomposisi data pada format .docx, .txt, .pdf .mp3, dan .wav.

Berikut merupakan hasil pengujian kompresi dan dekomposisi kategori teks yang dapat dilihat pada Tabel I,

Tabel II, Tabel III, Tabel IV, Tabel V, dan Tabel VI di bawah ini.

1). *Pengujian Kompresi Data Teks Format *.docx*

Tabel I berikut ini merupakan hasil pengujian kompresi data teks format *.docx.

TABEL I
HASIL PENGUJIAN KOMPRESI DATA TEKS .DOCX

No	Nama File	Ukuran (byte)		Rasio (%)	Waktu (ms)
		Awal	Akhir		
1.	Sample1	19.756	17.874	-9%	5
2.	Sample2	22.528	9.051	-59%	5
3.	Sample3	39.844	35.335	-11%	4
4.	Sample4	40.601	38.701	-4%	4
5.	Sample5	85.188	80.894	-5%	8

Dapat dilihat pada tabel I di atas ini bahwa hasil pengujian kompresi data teks format *.docx dengan algoritma *Run Length Encoding* membuktikan bekerja dengan sangat baik.

Pada *Sample1* dengan ukuran awal 19,2 KB setelah dikompresi menjadi 17,4 KB menghasilkan rasio -9% dengan durasi 5 ms.

Pada *Sample2* dengan ukuran awal 22 KB setelah dikompresi menjadi 8,83 KB menghasilkan rasio -59% dengan durasi 5 ms. Menghasilkan rasio paling terbanyak dikarenakan terdapat banyak pengulangan karakter di dalamnya.

Pada *Sample3* dengan ukuran awal 38,9 KB setelah dikompresi menjadi 34,5 KB menghasilkan rasio -11% dengan durasi 4 ms.

Pada *Sample4* dengan ukuran awal 39,6 KB setelah dikompresi menjadi 37,7 KB menghasilkan rasio -4% dengan durasi 4 ms.

Dan pada *Sample5* dengan ukuran awal 83,1 KB setelah dikompresi menjadi 78,9 KB menghasilkan rasio -5% dengan durasi 8 ms.

2) *Pengujian Dekompresi Data Teks Format *.docx*

Tabel II berikut ini merupakan hasil pengujian dekompresi data teks format *.docx.

TABEL II
HASIL PENGUJIAN DEKOMPRESI DATA TEKS .DOCX

No	Nama File	Ukuran (byte)		Waktu (ms)	Status
		Awal	Akhir		
1.	Sample1_compressed	17.874	19.756	5	Berhasil
2.	Sample2_compressed	9.051	22.528	6	Berhasil
3.	Sample3_compressed	35.335	39.844	8	Berhasil
4.	Sample4_compressed	38.701	40.601	8	Berhasil
5.	Sample5_compressed	80.894	85.188	9	Berhasil

Dapat dilihat pada tabel II di atas ini bahwa hasil pengujian dekompresi data teks format *.docx dengan algoritma *Run Length Encoding* membuktikan bekerja dengan baik.

Pada *Sample1_compressed* dengan ukuran awal 17,4 KB setelah didekompresi kembali menjadi 19,2 KB dengan durasi 5 ms.

Pada *Sample2_compressed* dengan ukuran awal 8,83 KB setelah didekompresi kembali menjadi 22 KB dengan durasi 6 ms.

Pada *Sample3_compressed* dengan ukuran awal 34,5 KB setelah didekompresi kembali menjadi 38,9 KB dengan durasi 8 ms.

Pada *Sample4_compressed* dengan ukuran awal 37,7 KB setelah didekompresi kembali menjadi 39,6 KB dengan durasi 4 ms.

Dan pada *Sample5_compressed* dengan ukuran awal 78,9 KB setelah didekompresi kembali menjadi 83,2 KB dengan durasi 8 ms.

3). *Pengujian Kompresi Data Teks Format *.txt*

Tabel III berikut ini merupakan hasil pengujian kompresi data teks format *.txt

TABEL III
HASIL PENGUJIAN KOMPRESI DATA TEKS .TXT

No	Nama File	Ukuran (byte)		Rasio (%)	Waktu (ms)
		Awal	Akhir		
1.	Sample1	23.166	22.275	-3%	2
2.	Sample2	33.768	17.808	-47%	4
3.	Sample3	41.664	32.832	-21%	3
4.	Sample4	62.524	50.890	-18%	7
5.	Sample5	63.840	12.320	-80%	2

Dapat dilihat pada tabel III di atas ini bahwa hasil pengujian kompresi data teks format *.txt dengan algoritma *Run Length Encoding* bekerja sangat baik.

Pada *Sample1* dengan ukuran awal 22,6 KB setelah dikompresi menjadi 21,7 KB menghasilkan rasio -3% dengan durasi 2 ms.

Pada *Sample2* dengan ukuran awal 32,9 KB setelah dikompresi menjadi 17,3 KB menghasilkan rasio -47% dengan durasi 4 ms.

Pada *Sample3* dengan ukuran awal 40,6 KB setelah dikompresi menjadi 32 KB menghasilkan rasio -21% dengan durasi 3 ms.

Pada *Sample4* dengan ukuran awal 61 KB setelah dikompresi menjadi 49,6 KB menghasilkan rasio -18% dengan durasi 4 ms.

Dan pada *Sample5* dengan ukuran awal 62,3 KB setelah dikompresi menjadi 12 KB menghasilkan rasio -80% dengan durasi 2 ms. Menghasilkan rasio paling terbanyak dikarenakan terdapat banyak pengulangan karakter di dalamnya.

4). Pengujian Dekompresi Data Teks Format *.txt

Tabel IV berikut ini merupakan hasil pengujian dekomposisi data teks format *.txt

TABEL IV
HASIL PENGUJIAN DEKOMPRESI DATA TEKS .TXT

No	Nama File	Ukuran (byte)		Waktu (ms)	Status
		Awal	Akhir		
1.	Sample1_compressed	22.275	23.166	2	Berhasil
2.	Sample2_compressed	17.808	33.768	2	Berhasil
3.	Sample3_compressed	32.832	41.664	8	Berhasil
4.	Sample4_compressed	50.890	62.524	4	Berhasil
5.	Sample5_compressed	12.320	63.840	7	Berhasil

Dapat dilihat pada tabel IV di atas bahwa hasil pengujian dekomposisi data teks format *.txt dengan algoritma *Run Length Encoding* juga dapat bekerja dengan sangat baik.

Pada *Sample1_compressed* dengan ukuran awal 21,7 KB setelah didekompresi kembali menjadi 22,6 KB dengan durasi 2 ms.

Pada *Sample2_compressed* dengan ukuran awal 17,3 KB setelah didekompresi kembali menjadi 32,9 KB dengan durasi 2 ms.

Pada *Sample3_compressed* dengan ukuran awal 32 KB setelah didekompresi kembali menjadi 40,6 KB dengan durasi 8 ms.

Pada *Sample4_compressed* dengan ukuran awal 49,6 KB setelah didekompresi kembali menjadi 61 KB dengan durasi 4 ms.

Dan pada *Sample5_compressed* dengan ukuran awal 12 KB setelah didekompresi kembali menjadi 62,3 KB dengan durasi 7 ms.

5). Pengujian Kompresi Data Teks Format *.pdf

Tabel V berikut ini merupakan hasil pengujian kompresi data teks format *.pdf

TABEL V
HASIL PENGUJIAN KOMPRESI DATA TEKS .PDF

No	Nama File	Ukuran (byte)		Rasio (%)	Waktu (ms)
		Awal	Akhir		
1.	Sample1	178.385	178.309	0%	7
2.	Sample2	184.650	184.540	0%	33
3.	Sample3	186.489	186.388	0%	38
4.	Sample4	198.301	198.210	0%	42
5.	Sample5	206.730	206.606	0%	67

Dapat dilihat pada tabel V di atas ini bahwa hasil pengujian kompresi data teks format *.pdf dengan algoritma *Run Length Encoding* tidak bekerja dengan baik.

Pada *Sample1* dengan ukuran awal 174 KB setelah dikompresi tetap menjadi 174 KB menghasilkan rasio 0% dengan durasi 7 ms.

Pada *Sample2* dengan ukuran awal 180 KB setelah dikompresi tetap menjadi 180 KB menghasilkan rasio 0% dengan durasi 33 ms.

Pada *Sample3* dengan ukuran awal 182 KB setelah dikompresi tetap menjadi 182 KB menghasilkan rasio 0% dengan durasi 38 ms.

Pada *Sample4* dengan ukuran awal 193 KB setelah dikompresi tetap menjadi 193 KB menghasilkan rasio 0% dengan durasi 42 ms.

Dan pada *Sample5* dengan ukuran awal 201 KB setelah dikompresi tetap menjadi 201 KB menghasilkan rasio 0% dengan durasi 67 ms.

6). Pengujian Dekompresi Data Teks Format *.pdf

Tabel IV berikut ini merupakan hasil pengujian dekomposisi data teks format *.pdf

TABEL VI
HASIL PENGUJIAN DEKOMPRESI DATA TEKS .PDF

No	Nama File	Ukuran (byte)		Waktu (ms)	Status
		Awal	Akhir		
1.	Sample1_compressed	178.309	43138031	509	Tidak Berhasil
2.	Sample2_compressed	184.540	43797469	550	Tidak Berhasil
3.	Sample3_compressed	186.388	46086820	581	Tidak Berhasil
4.	Sample4_compressed	198.210	44304473	645	Tidak Berhasil
5.	Sample5_compressed	206.606	129.414.293	2823	Tidak Berhasil

Khusus pada tabel VI di atas bahwa hasil pengujian dekompresi data teks format *.pdf dengan algoritma *Run Length Encoding* tidak dapat bekerja dengan baik, maka dari sinilah menjadi batasan masalah pada penelitian ini dikarenakan hasil pengembalian ukuran akhir tidak seperti yang diharapkan.

7). Pengujian Kompresi Data Audio Format *.mp3

Tabel VII berikut ini merupakan hasil pengujian kompresi data audio format *.mp3

TABEL VII
HASIL PENGUJIAN KOMPRESI DATA AUDIO .MP3

No	Nama File	Ukuran (byte)		Rasio (%)	Waktu (ms)
		Awal	Akhir		
1.	Sample1	52.079	51.214	-1%	25
2.	Sample2	336.890	174.940	-48%	50
3.	Sample3	2.403.008	127.777	-94%	60
4.	Sample4	8.414.449	8.313.886	-1%	229
5.	Sample5	13.200.345	12.786.396	-3%	350

Dapat dilihat pada tabel VII di atas ini bahwa hasil pengujian kompresi data audio format *.mp3 dengan algoritma *Run Length Encoding* membuktikan bekerja dengan sangat baik.

Pada *Sample1* dengan ukuran awal 50,8 KB setelah dikompresi menjadi 50 KB menghasilkan rasio -1% dengan durasi 3 ms.

Pada *Sample2* dengan ukuran awal 328 KB setelah dikompresi menjadi 170 KB menghasilkan rasio -48% dengan durasi 8 ms.

Pada *Sample3* dengan ukuran awal 2,29 MB setelah dikompresi menjadi 124 KB menghasilkan rasio -94% dengan durasi 60 ms.

Pada *Sample4* dengan ukuran awal 8,02 MB setelah dikompresi tetap menjadi 7,92 MB menghasilkan rasio -1% dengan durasi 229 ms.

Dan pada *Sample5* dengan ukuran awal 12,5 MB setelah dikompresi tetap menjadi 12,1 MB menghasilkan rasio 3% dengan durasi 350 ms.

8). Pengujian Dompresi Data Audio Format *.mp3

Tabel VIII berikut ini merupakan hasil pengujian dekompresi data audio format *.mp3

TABEL VIII
HASIL PENGUJIAN DEKOMPRESI DATA AUDIO .MP3

No	Nama File	Ukuran (byte)		Waktu (ms)	Status
		Awal	Akhir		
1.	Sample1_compressed	51.214	52.093	12	Berhasil
2.	Sample2_compressed	174.940	336.890	20	Berhasil
3.	Sample3_compressed	127.777	2.403.008	58	Berhasil
4.	Sample4_compressed	8.313.886	8.455.416	197	Tidak Berhasil
5.	Sample5_compressed	12.786.396	13.217.528	304	Tidak Berhasil

Dapat dilihat pada tabel VIII di atas bahwa hasil pengujian dekompresi data audio format *.mp3 dengan algoritma *Run Length Encoding* ada yang dapat bekerja dengan baik dan tidak.

Pada *Sample1_compressed* dengan ukuran awal 50 KB setelah didekompresi kembali menjadi 50,8 KB dengan durasi 12 ms.

Pada *Sample2_compressed* dengan ukuran awal 170 KB setelah didekompresi kembali menjadi 328 KB dengan durasi 20 ms.

Pada *Sample3_compressed* dengan ukuran awal 124 KB setelah didekompresi kembali menjadi 2,29 MB dengan durasi 58 ms.

Pada *Sample4_compressed* dengan ukuran awal 7,92 MB setelah didekompresi menjadi 8,06 MB dengan durasi 197 ms tidak dapat berkerja dengan dikarenakan ukuran byte bertambah sedikit tidak dapat kembali seperti ukuran awal file.

Dan pada *Sample5_compressed* dengan ukuran awal 12,1 MB setelah didekompresi menjadi 12,6 MB dengan durasi 304 ms tidak dapat berkerja dengan dikarenakan ukuran byte bertambah sedikit tidak dapat kembali seperti ukuran awal file.

9). Pengujian Kompresi Data Audio Format *.wav

Tabel IX berikut ini merupakan hasil pengujian kompresi data audio format *.wav

TABEL IX
 HASIL PENGUJIAN KOMPRESI DATA AUDIO .WAV

No	Nama File	Ukuran (byte)		Rasio (%)	Waktu (ms)
		Awal	Akhir		
1.	Sample1	188.082	188.079	0%	21
2.	Sample2	1.584.080	1.507.949	-4%	45
3.	Sample3	9.629.626	9.494.616	-1%	60
4.	Sample4	10.406.738	10.406.733	0%	71
5.	Sample5	9.796.666	9.632.436	-1	63

Dapat dilihat pada tabel IX di atas bahwa hasil pengujian kompresi data audio format *.wav dengan algoritma *Run Length Encoding* dapat bekerja dengan baik.

Pada *Sample1* dengan ukuran awal 183 KB setelah dikompresi tetap menjadi 183 KB menghasilkan rasio 0% dengan durasi 21 ms.

Pada *Sample2* dengan ukuran awal 1,50 MB setelah dikompresi menjadi 1,43 MB menghasilkan rasio -4% dengan durasi 45 ms.

Pada *Sample3* dengan ukuran awal 9,18 MB setelah dikompresi menjadi 9 MB menghasilkan rasio -1% dengan durasi 60 ms.

Pada *Sample4* dengan ukuran awal 9,92 MB setelah dikompresi menjadi 9,92 MB menghasilkan rasio 0% dengan durasi 71 ms.

Dan pada *Sample5* dengan ukuran awal 9,34 MB setelah dikompresi menjadi 9,18 MB menghasilkan rasio -1% dengan durasi 63 ms.

10). Pengujian Dekompresi Data Audio Format *.wav

Tabel X berikut ini merupakan hasil pengujian dekomposisi data audio format *.wav

TABEL X
 HASIL PENGUJIAN DEKOMPRESI DATA AUDIO .WAV

No	Nama File	Ukuran (byte)		Rasio (%)	Waktu (ms)
		Awal	Akhir		
1.	Sample1_compressed	188.079	188.082	6	Berhasil
2.	Sample2_compressed	1.507.949	1.584.080	30	Berhasil
3.	Sample3_compressed	9.494.616	9.655.975	42	Tidak Berhasil
4.	Sample4_compressed	10.406.733	10.411.889	64	Tidak Berhasil
5.	Sample5_compressed	9.632.436	9.796.666	67	Berhasil

Dapat dilihat pada tabel X di atas bahwa hasil pengujian dekomposisi data audio format *.wav dengan algoritma *Run Length Encoding* ada yang dapat bekerja dengan baik dan ada yang tidak.

Pada *Sample1_compressed* dengan ukuran awal 183 KB setelah didekomposisi kembali tetap menjadi 183 KB dengan durasi 6 ms.

Pada *Sample2_compressed* dengan ukuran awal 1,43 MB setelah didekomposisi kembali menjadi 1,51 MB dengan durasi 30 ms.

Pada *Sample3_compressed* dengan ukuran awal 9 MB setelah didekomposisi kembali menjadi 9,20 MB dengan durasi 42 ms tidak dapat berkerja dengan dikarenakan ukuran byte bertambah sedikit tidak dapat kembali seperti ukuran awal file.

Pada *Sample4_compressed* dengan ukuran awal 9,92 MB setelah didekomposisi kembali tetap menjadi 9,92 MB dengan durasi 64 ms tidak dapat berkerja dengan dikarenakan ukuran byte bertambah sedikit tidak dapat kembali seperti ukuran awal file.

Dan pada *Sample5_compressed* dengan ukuran awal 9,18 MB setelah didekomposisi kembali menjadi 9,41 MB dengan durasi 67 ms.

IV. KESIMPULAN

Setelah melakukan penelitian dari analisis kompresi dan dekomposisi data teks dan audio dengan menggunakan

algoritma *run length encoding* dapat disimpulkan bahwa jika durasi waktu dalam proses kompresi dan dekompresi yang didapatkan menunjukkan semakin kecil maka semakin efisien metode yang digunakan.

Hasil rasio kompresi-dekompresi dari pengujian ini memperoleh banyak varian, jika kompresi menunjukkan nilai minus (-) maka file itu telah terkompresi dengan baik, begitu pula sebaliknya dengan proses dekompresi ukuran dapat kembali seperti semula, tetapi apabila menunjukkan rasio 0% atau ukuran byte bertambah sedikit maka file itu tidak terdekompresi dengan sempurna.

Pada dasarnya metode *Run Length Encoding* digunakan untuk membaca *file text* di setiap karakter *ASCII* pada file tersebut dengan pengulangan *string* dari deretan karakter. Hasil dari pengujian aplikasi ini membuktikan pada proses kompresi dapat bekerja dengan baik tetapi untuk proses dekompresi beberapa file tidak bisa bekerja dengan sempurna dan mengalami kendala khususnya di format *.pdf sehingga menjadi batasan masalah pada penelitian ini.

UCAPAN TERIMA KASIH

Dengan selesainya penelitian ini penulis panjatkan Puji syukur terimakasih Alhamdulillah kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian dan menulis artikel ilmiah dengan baik. Tidak lupa juga penulis mengucapkan terimakasih kepada kedua orang tua, dosen pembimbing, teman dan semua pihak yang telah mendukung, membantu, serta membimbing penulis dalam mengerjakan dan menyelesaikan penelitian ini.

REFERENSI

- [1] Putra, Darma. *Pengolahan Citra Digital*. Yogyakarta: Andi, 2010.
- [2] Salomon, David. *Data Compression: The Complete Reference*. IEEE Signal Processing Magazine, 2007.
- [3] Rahandi, Aditya., Dian Rachmawati., dan Sajadin Sembiring. *Analisis dan Implementasi Kompresi File Audio Dengan Menggunakan Algoritma Run Length Encoding (RLE)*. Jurnal FASILKOM-TI USU, 2013.
- [4] Willfrid, Darno., Garuda Ginting., Yasir Hasan. *Implementasi Algoritma Run Length Encoding Pada Kompresi File MP3*. Jurnal Riset Komputer (JURIKOM) Vol. 3 No. 4, pp 5-9, 2016.
- [5] Dhudhagara, Chetan dan Hasamukh B. Patel. *Performance Analysis of Data Compression using Lossless Run Length Encoding*. Jurnal Oriental Journal of Computer Science and Technology Vol. 10, No. (3) pp. 703-707, 2017.
- [6] Mitra, Sanjoy dan Debaprasad Das. *A Critical Study on The Applications of Run Length Encoding Techniques in Combined Encoding Schemes*. International Journal of Advanced Research in Computer Science. Volume 8, No. 5, 2017.
- [7] Sinaga, Helbert., Poltak Sihombing., Handrizal. *Perbandingan Algoritma Huffmandan Run Length Encoding Untuk Kompresi File Audio*. Jurnal TALENTA Conference Series: Science & Technology. Volume 1 Issue 1, pp 10-15, 2018.
- [8] Pu, I.M. *Fundamental Data Compression*. Butterworth-Heinemann, 2005.
- [9] Sobur, Alex. *Analisis Teks Media: Suatu Pengantar untuk Analisis Wacana, Analisis Semiotik, dan Analisis Framing*. Bandung: PT. Remaja Rosdakarya, 2006.
- [10] Fisher, Darren. *Developing Multimedia Instructional Products*. CSUN EDU, 2003.
- [11] Wikipedia, "Run-Length Encoding", 2020. [Online]. Available: https://en.wikipedia.org/wiki/Run-length_encoding [Diakses 26 Juni 2020]
- [12] Hendri, Nofri. *Komputer Multimedia*. Univesitas Negeri Padang, 2015.