JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Perancangan Aplikasi Kompresi *File* Audio Dengan Menggunakan Algoritma *Sequitur*

Winda Remi Sianturi

Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email : nwinda317@gmail.com

Abstrak− Kompresi audio adalah proses mengubah sebuah aliran data input menjadi aliran audio baru yang memiliki ukuran lebih kecil. Aliran yang dimaksud adalah berupa audio, data, dokumen, dalam memori. Kebutuhan terhadap kompresi audio dipengaruhi oleh dua alasan, yaitu kecenderungan manusia untuk mengumpulkana udio dan kebutuhan terhadap proses transfer audio yang terlalu tidak memenuhi kapasitas. Salah satu solusi untuk melakukan pengiriman audio atau tentu diperlukan sebuah media penyimpanan yang dapat menampung sebuah audio atau informasi tersebut besarnya suatu ukuran yang akan dipindahkan sangat berpengaruh padaruang penyimpanan. Artinya jika ukuran yang di pindahkan tersebut berukuran besar maka dibutuhkan ruang penyimpanan yang cukup besar, juga untuk menampung audio tersebut. Hasil dari penelitian ini maka untuk mengatasi masalah seperti ini dapat dilakukan dengan proses teknik kompresi data. Kompresi data bertujuan untuk memperkecil sebuah ukuran *file* yang dapat mengurangi ukuran bit yang ada pada setiap audio, dan juga dengan berkurangnya ukuran audio tersebut berkurang jugalah ruang penyimpanan yang dibutuhkan

Kata Kunci : *Kompresi, Audio, Algoritma sequitur*

Abstract− *Audio compression is the process of changing an input data stream into a new audio stream that has a smaller size. The stream in question is in the form of audio, data, documents, in memory. The need for audio compression is influenced by two reasons, namely the human tendency to collect audio and the need for an audio transfer process that is too inadequate. One solution for sending audio or of course requires a storage medium that can accommodate audio or information. The size of the size to be transferred greatly influences the storage space. This means that if the size being moved is large, quite large storage space is needed, also to accommodate the audio. The results of this research are that solving problems like this can be done using data compression techniques. Data compression aims to reduce the file size which can reduce the bit size of each audio, and also by reducing the size of the audio the storage space required is also reduced.*

Keywords*: Compression, Audio, Sequitur algorithm*

1. PENDAHULUAN

Kompresi data merupakan proses mengubah sebuah aliran data input sebagai peredaran data baru yang mempunyai ukuran lebih kecil. Kebutuhan terhadap kompresi data dipengaruhi oleh dua alasan, yaitu kecenderungan manusia untuk mengumpulkan data serta kebutuhan terhadap proses transfer data yang cepat[1].

Mp3 merupakan suatu format audio yang tak jarang dipergunakan sebab data yang tersimpan hampir sama dengan data asli dan ukurannya tidak terlalu besar dibandingkan format lainnya. Namun MP3 tetap saja *user* (pengguna) menginginkan data menggunakan kualitas terbaik serta kuantitas (berukuran) yang rendah.

Untuk melakukan pengiriman data atau tentu diperlukan sebuah media penyimpanan yang bisa menampung sebuah data atau informasi tersebut besarnya suatu ukuran yang akan dipindahkan sangat berpengaruh pada ruang penyimpanan. artinya, bila ukuran yang akan dipindahkan tersebut berukuran tinggi hingga membutuhkan ruang penyimpanan yang relatif besar serta buat menampung data tersebut. [2].

Dari hasil latar belakang masalah tersebut, maka untuk mengatasi masalah seperti ini dapat dilakukan dengan proses teknik kompresi data. Kompresi data bertujuan untuk memperkecil sebuah ukuran *file* yang dapat mengurangi ukuran bit yang ada pada setiap audio, serta juga dengan berkurangnya ukuran audio tersebut berkurang jugalah ruang penyimpanan yang dibutuhkan. Algoritma *Sequitur* adalah suatu algortima kompresi yang bisa melakukan kompresi data Algortima ini akan bekerja dengan mencari kata yang sama, kata yang sama akan digantikan menjadi karakter baru yang lebih singkat sehingga menggurangi penggunaan data[3].

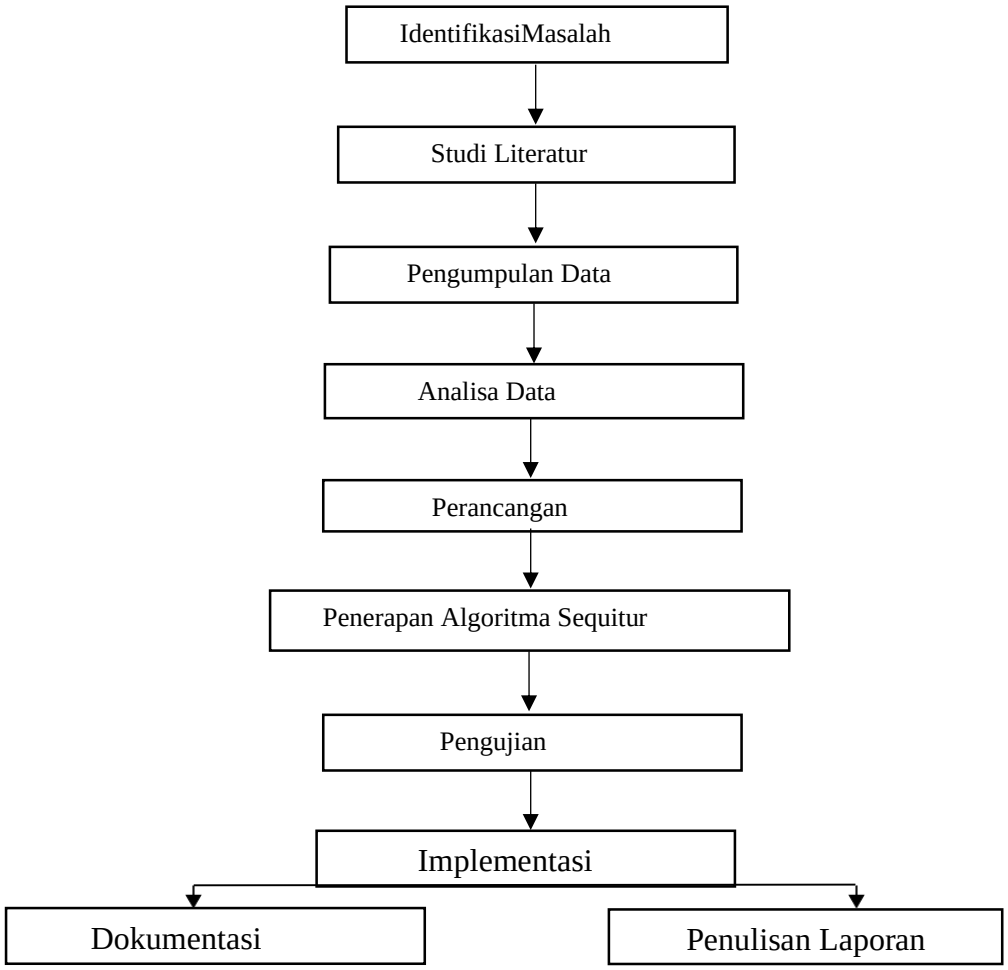
Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Berdasarkan algoritma kompresi yang digunakan pada penelitian terdahulu, penulis menerapkan algoritma *sequitur* untuk mengetahui bagaimana proses kompresi pada audio. Untuk itu penulis mengangkat sebuah judul penelitian yaitu: **"Perancangan Aplikasi Kompresi *File* Audio Dengan Menggunakan Algoritma *Sequitur*".**

2. METODOLOGI PENELITIAN

Kerangka kerja penelitian terdiri dari beberapa tahapan yang saling berkaitan secara sistematis. Berikut merupakan kerangka kerja dalam proses penelitian ini.



Gambar 1 Kerangka kerja penelitian

2.1 Perancangan

Perancangan adalah langkah awal pada *fase* pegembangan rekayasa *system*. Perancangan yaitu proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefenisikan secara rinci prangkat, proses atau system yang memungkinkan realisasi fisiknya, satu proses atau satu system secara detail yang membolehkan dilakukan realisasi fisik. Fase ini adalah inti teknis dari proses rekayasa perangkat lunak. Pada *fase* ini elemen-elemen dari model analisa dikonversikan. Dengan mengunakan satu dari sejumlah metode perancangan data, perancangan antarmuka, perancangan arsitektur dan perancangan prosedur.

2.3 Aplikasi

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Aplikasi yaitu gabungan komponen yang saling berinteraksi dan saling berhubungan untuk aktivitas bersama-sama dalam mencapai tujuan tertentu[4]. Aplikasi pada dasarnya adalah hasil suatu proses pengembangan yang dilakukan dengan teknologi terkeni tersedia perangkat lunak untuk mempermudah proses pengelolaan maupun pemantauan suatu kegiatan. Aplikasi berasal dari kata *Aplication*. Pengertian aplikasi yaitu istilah yang digunakan para pengguna komputer untuk pemecahan masalah.

2.3 Kompresi

Kompresi memiliki tujuan untuk mengecilkan ukuran audio dengan kapasitas besar. Untuk kompresi audio, pengkompresian data dapat dilakukan dengan memanfaatkan dua factor utamanya, yaitu pada data suara, gambar atau audio dan kepemilikan dari manusia, dan dalam cara lain di ilmu komputer dapat memadatkan data sehingga ruang penyimpanan lebih kecil dan lebih efisien dalam penyimpanan atau mempersingkat waktu pertukaran data. Setelah data ke format yang lebih kecil data tersebut dapat tersimpan dengan berukuran kecil. Oleh sebab itu, kompresi yang bertujuan untuk mengkompresi data agar dapat memperkecil tempat penyimpanan data. Salah satunya yaitu kompresi audio agar dapat diperkecil dengan kapasitasyangberukuran besar[5].

2.3 Audio

Audio yaitu suara atau bunyi yang dihasilkan dari getaran suatu benda, agar dapat tertangkap oleh telinga manusia getaran tersebut harus kuat minimal 20 kali /detik. Suara yaitu suatu getaran yang dihasilkan dari gesekan, pantulan dll., antara benda-benda. Sedangkan gelombang yaitu suatu getaran yang terdiri dari Amplitudo dan juga waktu. Pengertian audio lainnya yaitu salah satu elemen yang penting, karena ikut berperan dalam membangun sebuah sistem Komunikasi dalam bentuk suara, ialah suatu sinyal elektrik yang akan membawa unsur-unsur bunyi didalamnya. Audio itu terbentuk melalui beberapa tahap, diantaranya adalah tahap pengambilan atau penangkapan suara, sambungan transmisi yang membawa bunyi, *amplifier*[6].

2.3 AlgoritmaSequitur

Algoritma *sequitur* merupakan algoritma waktu linear yang menyimpulkan tata bahasa bebas konteks ke dalam suatu pemampatan untuk mengurangi masukan yang berulang. Operasi *sequitur* terdiri dari pemastian dua sifat yang berlaku. Ketika menjelaskan algoritma, sifat-sifatnya bertindak sebagai batasan. Algoritma beroperasi menjalankan batasan didalam sebuah tata bahasa yang ketika diagram keunikan (*diagram uniqueness)* dilanggar, sebuah aturan baru dibentuk, dan ketika batasan aturan kegunaan (*ruleutility*) dilanggar, aturan yang sia-sia dihapus. Algoritma *Sequitur* membangun tata bahasanya dengan menggunakan dua prinsip,yaitu :

1. *DiagramUniqueness*
   Tidak ada pasangan simbol yang berdekatan yang muncul lebih dari satu.
2. Semua aturan harus digunakan lebih dari satu kali dan aturan yang hanya digunakan sekali harus diabaikan atau ditiadakan.

Algoritma ini sendiri beroperasi menjalankan batasan didalam sebuah tata bahasa yang ketika *digram uniqueness* dilanggar, sebuah aturan baru dibentuk, dan ketika batasan aturan kegunaan *Ruleutility* dilanggar aturan yang sia-sia dihapus. Sedangkan algoritma *Sequitur* beroperasi dengan menjalankan batasan diagram keunikan *(Diagram uniqueness*) dan aturan kegunaan (*Ruleutility*).

Dibawah ini ialah contoh penerapan algoritma *sequitur* yang akan masukkan karakter pertama s untuk mendapatkan hasil yaitu :

S → abcababcd; Mengulang dan Mencocokkan pada non-terminal yang ada;
S → AcAab → A → ab S → Ac Aa Membuat suatu non-terminal baru
S→AcAAcA→ab→B→ Ac Memindahkan non-terminal
S→AcAAcA→ abS→BAB→B→Ac Memasukkan karakter baru
S→ BABd Sampai tidak ada karakter yang tersisa[7].

3. HASIL DAN PEMBAHASAN

Dalam penelitian dilakukakan analisa kinerja terhadap algoritma *Sequitur* dengan menerapkan algoritma *Sequitur* pada audio berformat MP3. Selanjutnya akan dibuat perancangan perangkat

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

lunak untuk kompresi audio. Algoritma *Sequitur* merupakan satu dari algoritma kompresi yang memperoleh suatu pengurangan dari ukuran data dan mengurangi kapasitas pada media penyimpanan sehingga proses pengiriman data menjadi lebih cepat.

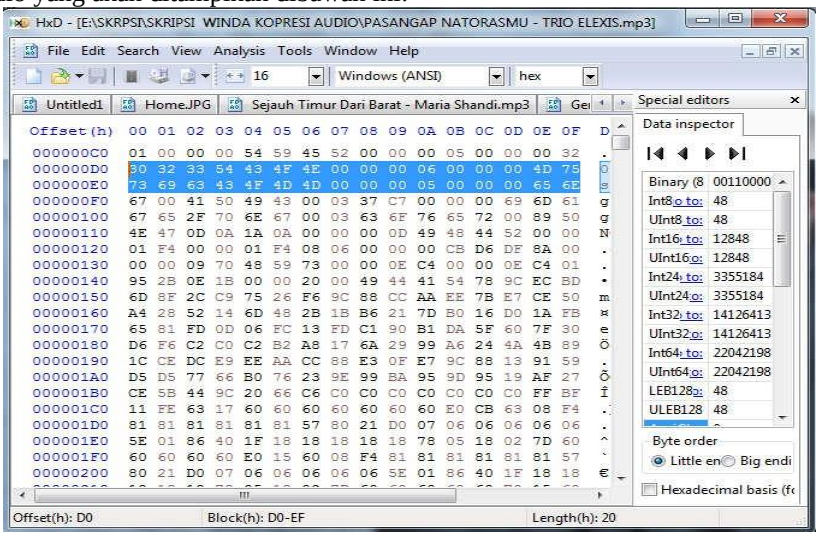
Pada tahap awal yang akan dianalisa penulis ialah dengan mencari audio yang akan dikompresi kemudian diubah menjadi nilai *hexadecimal* melalui *Software HxD*, setelah didapat nilai *hexadecimal* audio maka akan dilakukan proses kompresi dan diperoleh hasil kompresi. Berikut merupakan prosedur alur sederhana proses kompresi dan dekompresi audio.

a. Menginput *File*

Tabel 1. Sampel Audio yang akan dikompresi

NamaAudio	Size	Type
Pasangap Natorasmu- Trio Elexis	6,91 Mb	MP3

Berdasarkan tahapan yang akan diproses kompresi dan dekompresi pada sampel gambar audio, sebelum audio dikompresi terlebih dahulu dikonfersi dalam bentuk nilai hexadesimal. Berikut gambar audio yang akan ditampikan dibawah ini:



Gambar 2. Niai Hexadesimal Sampel Audio

b. Kompresi Audio Menggunakan Algoritma *Sequitur*

Pada sampel nilai *hexadecimal* audio diatas, lalu digunakan 32 sampel nilai *hexadecimal* untuk proses kompresi audio.

Tabel 2. Tampilan Nilai Bilangan Hexadecimal

30	32	33	54	43	4F	4F	00	00	00	06	00	00	00	4D	75
73	69	63	43	4F	4D	4D	00	00	00	05	00	00	00	65	6E

Berdasarkan tabel di atas, lalu terdapat nilai bilangan *hexadecimal* audio MP3 untuk dihitung mulai dari manual. Nilai bilangan *hexadecimal* tersebut terlebih dahulu diubah kedalam bentuk karakter untuk proses perhitungan kompresi dapat dilihat pada tabel di bawah ini.

Tabel 3. Nilai Hexadecimal yang telah diubah dalam bentuk karakter

No.	Nilai Hexadecimal	Decimal	Karakter
1.	30	48	0
2.	32	50	2
3.	33	51	3
4.	63	99	c
5.	43	67	C

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

6.	4F	79	O
7.	4F	77	M
8.	00	0	Null(tidakterlihat)
9.	00	0	Null(tidakterlihat)
10.	00	0	Null(tidakterlihat)
11.	06	6	-
12.	00	0	Null(tidakterlihat)
13.	00	0	Null(tidakterlihat)
14.	00	0	Null(tidakterlihat)
15.	4D	77	M
16.	75	117	u
17.	73	115	s
18.	69	105	i
19.	63	99	c
20.	43	67	C
21.	4F	79	O
22.	4D	77	M
23.	4D	77	M
24.	00	0	Null(tidakterlihat)
25.	00	0	Null(tidakterlihat)
26.	00	0	Null(tidakterlihat)
27.	05	5	?
28.	00	0	Null(tidakterlihat)
29.	00	0	Null(tidakterlihat)
30.	00	0	Null(tidakterlihat)
31.	65	101	e
32.	6E	110	n
Total Karakter			3

Setelah nilai *hexadecimal* diubah kedalam bentuk karakter langkah selanjutnya melakukan perhitungan menggunakan algoritma *Sequitur.*
Karakter: 023cCOM-MusicCOMM?en'

Hasil Pemindaian pertama : 02, 3c, CO, M-, Mu, si, c, CO, MM, ?e, n dan terjadi perulangan pada 'CO' maka diganti dengan simbol non-terminal 'A'➔'CO'menjadi '023cAM-MusicAMM?en.

Hasil pemindaian kedua : 02, 3c, AM, -M, us, ic, AM, M?, en., dan terjadi perulangan pada 'AM' maka diganti dengan simbol non-terminal 'B' ➔'AM'menjadi023cBMusicBM?en. Bentuk karakter tidak ditemukan pengulangan. Sehingga proses algoritma *Sequitur* sudah selesai.

Tabel 4. Hasil pemindaian karakter

Karakte	Simbol Berulang	Rule	Perubahan Karakter	Simbol non Terminal dihapus
023cCOM-MusicCOMM?en	CO	A➔CO	023cAM-MusicAMM?en	-
023cAM-MusicAMM?en	AM	A➔CO B➔AM	023cB-MusicBM?en	A

Tahap selanjutnya untuk menghitung ukuran dari karakter yang akan dikompresi, terlebih dahulu diurutkan berdasarkan frekuensi kemunculan karakter terbanyak hingga terkecil kemudian menghitung *bit* dari setiap karakter.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Tabel 5. Karakter yang belum dikompresi

No	Karakter	Frekuensi	Decimal	Biner	Bit	Bit x Frekuensi
1.	Null(tidak terlihat)	12	0	00000000	8	96
2.	M	4	77	01001101	8	32
3.	C	2	67	01000011	8	16
4.	O	2	79	01001111	8	16
5.	c	2	99	01100011	8	16
6.	0	1	48	00110000	8	8
7.	2	1	50	00110010	8	8
8.	3	1	51	00110011	8	8
9.	u	1	117	01110101	8	8
10.	s	1	115	01110011	8	8
11.	i	1	105	01101001	8	8
12.	e	1	101	01100101	8	8
13.	n	1	110	01101110	8	8
14.	?	1	5	00000101	8	8
15.	-	1	6	00000110	8	8
	Total	32	-	-	-	256 Bit

Berdasarkan perhitungan sebelumnya didapat ukuran awal karakter 023cCOM-MusicCOMM?en adalah 256 *bit*, dan pada tahap sebelumnya juga sudah dilakukan proses kompresi dengan algoritma *Sequitur* dan menghasilkan karakter baru yaitu ‘023cB-MusicBM?en kemudian diurutkan berdasarkan frekuensi terbesar kefrekuensi terkecil dapat dilihat pada tabeldi bawah ini.

Tabel 6. Perhitungan hasil kompresi algoritma *Sequitur*

No	Karakter	Frekuensi	Decimal	Biner	Bit	BitxFrekuensi
1.	B	2	42	01000010	8	16
2.	M	2	77	01001101	8	16
3.	c	2	99	01100011	8	16
4.	0	1	48	00110000	8	8
5.	2	1	50	00110010	8	8
6.	3	1	51	00110011	8	8
7.	-	1	6	00000110	8	8
8.	u	1	117	01110101	8	8
9.	s	1	115	01110011	8	8
10.	i	1	105	01101001	8	8
11.	?	1	5	00000101	8	8
12.	e	1	101	01100101	8	8
13.	n	1	110	01101110	8	8
	Total	16	-	-	-	128 Bit

Setelah dilakukan perhitungan kompresi maka selanjutnya menghitung tingkat kinerja hasil kompresi sesuai dengan parameter yang telah ditentukan. Semakin kecil hasil dari file kompresinya maka semakin berkualitas kinerja kompresinya, begitu sebaliknya. Berikut merupakan parameter untuk mengukur
Kinerja algoritma *Sequitur.*

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

1. *Ratioof Compression* (RC) ialah nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi.

$$RC = \frac{UkuranDataSebelumDikompresi}{UkuranDataSesudahDikompresi}$$

$$RC = \frac{256}{128} = 2$$

2. *Compression Ratio* (CR) ialah presenate perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.

$$CR = \frac{UkuranDataSetelahDikompresi}{UkuranDataSebelumDikompresi} \times 100\%$$

$$CR = \frac{128}{256} \times 100\% = 50\%$$

3. *Space Saving* merupakan selisih antara data yang belum dikompresi dengan besar data yang sudah dikompresi.
   S_s= 100%- C_R
   S_s=100%-*50%* = 50%

Dari perhitungan diatas dapat diketahui berapa ukuran audio setelah dikompresi dengan menggunakan algoritma *sequitur* yaitu:

=Ukuran audio awal x *Compression Ratio*
= 6,91 MB x 50% = 3,45 MB
= Ukuran audio awal – Hasil Perkalian (ukuran audio x *compression ratio*)
=6,91MB– 3,45MB =3,45 MB

c. Dekompresi Algoritma Sequitur

Sebelum dilakukan proses dekompresi pada algoritma *Sequitur* perlu diketahui bahwa pada saat kompresi sebelumnya sudah menyimpan rule maka gunakan rule tersebut kembali untuk proses dekompresi atau mengembalikan karakter ke bentuk semula dapat dilihat pada tabel dibawah ini.

Tabel 7. Hasil dekompresi algoritma *Sequitur*.

Karakte	Rule	Keterangan	Perubahan Karakter
023cCOM-MusicCOMM?en	B➔AM	Simbol Bakanmengembalikan kata‘isom’ke karakter awal	023cB-MusicBM?en

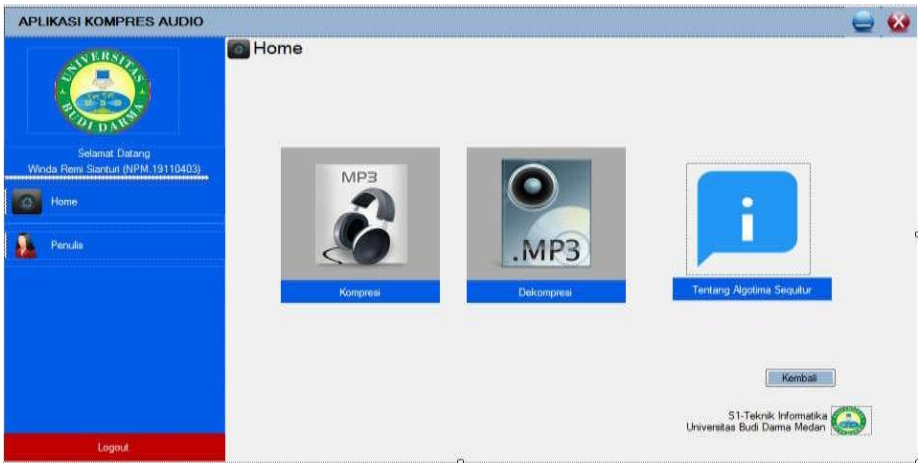
3.2 Implementasi

Implementasi program adalah tahapan uji coba dari sebuah sistem yang telah dirancang. Sebagian kecil membahas spesifikasi perangkat keras *(hardware)* dan perangkat lunak *(software)*, serta tampilan sistem yang sedang berjalan.

1. *Form* Home
   Dalam tampilan ini terdapat beberapa pilihan menu untuk mengakses *form* yang terdapat pada sistem. Berikut merupakan tampilan dari *form* menu utama.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin



Gambar 3. Tampilan Output Enkripi

2. *Form* Kompresi

Form ini menampilkan proses kompresi audio dengan digunakannya algoritma *Sequitur*, yakni *user* meng-*input*-kan terlebih dahulu audio yang akan dikompresi.

Gambar Error! No text of specified style in document.**.** Proses penginputan audio

Setelah memilih audio yang akan dikompresi, maka aplikasi akan menampilkan audio yang akan dikompresi beserta ukuran dan type dari audio.

Gambar 5. Tampilan audio yang dikompresi

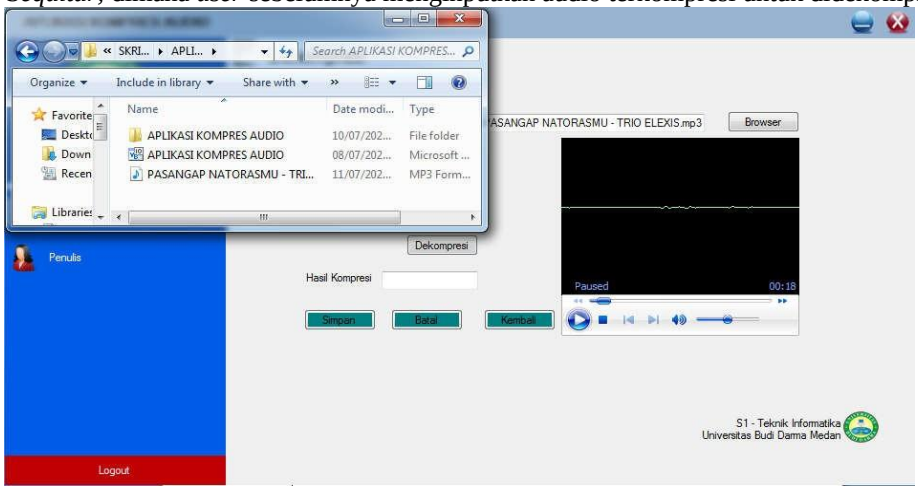
3. *Form* Dekompresi

Form ini akan menampilkan cara dekompresi audio dengan digunakannya algoritma

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

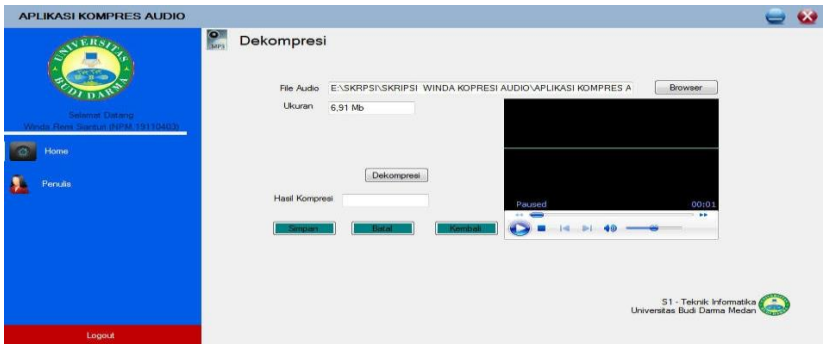
JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Sequitur, dimana *user* sebelumnya menginputkan audio terkompresi untuk didekompresi.



Gambar 7. Proses penginputan audio terkompresi

Setelah memilih audio terkompresi untuk dilakukan proses dekompresi, maka aplikasi akan menampilkan audio terkompresi seperti gambar dibawah ini.



Gambar 8. Tampilan Audio terkompresi

4. *Form* Penulis

Form penulis berisi tentang biodata penulis penelitian ini. Adapun *form* penulis dapat dilihat pada gambar dibawah ini.

Gambar 9. *Form* Penulis

4. KESIMPULAN

Setelah dilakukannya proses kompresi dan dekompresi audio dengan menggunakan algoritma Sequitur terdapat pengurangan penggandaan data pada audio akibatnya memori yang

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

Volume 5, Nomor 1, Desember 2023, Page 1-10
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

diperlukan membuat lebih kecil dari pada audio sebelumnya. Penerapan algortima Sequitur untuk kompresi ukuran audio dilakukan dengan menggunakan tiga kriteria yaitu Ratio of compression (RC), CR, dan Space Saving (SS). Perancangan aplikasi kompresi audio menggunakan algoritma Sequitur berhasil dibangun dengan menerapkan aplikasi visual Studio 2010.

REFERENCES

[1] I.Lestari, "Analisa Perbadingan Algoritma Goldbach Codes Dengan Algoritma Sequitur Pada Kompresi File Text Menggunakan MetodeExponential, "*PelitaInform.Inf. Dan Inform.*, vol.8,no.1, pp.15–18,2019.

[2] S. H. Silitonga and S. D. Nasution, "Implementasi Algoritma Boldi-VignaCodes Untuk Kompresi File Audio Pada Aplikasi Pemutar Audio BerbasisWeb," *KOMIK Konf. Nas. Teknol. Inf. Dan Komput.*, vol. 6, no. 1, pp. 586–595, 2023.

[3] H.Hidayat,T.Pamungkas,andW.Zarman,"Implementasi Algoritma Kompresi Lzw Pada Database Server,"*Komputa J.Ilm.Komput. Dan Inform.*,vol. 2, no. 1, 2013.

[4] D. A. Megawaty and M. E. Putra, "Aplikasi Monitoring Aktivitas Akademik Mahasiswa Program Studi Informatika Universitas Xyz BerbasisAndroid," *J.Inform. Dan Rekayasa Perangkat Lunak*,vol. 1,no. 1,pp. 65–74,2020.

[5] Y.Darnita,K.Khairunnisyah,andH.Mubarak,"Kompresi Data Teks Dengan Menggunakan Algoritma Sequitur," *Sist. J. Sist. Inf.*, vol. 8, no. 1,pp. 104–113, 2019.

[6] M. Mustapa, S. Susanti, and R. Anggreraeni, "Perbaikan Kualitas AudioTempat Sampah Elektronik (E-Trash) Berbasis Arduino Uno," *J. Elektron.Telekomun.Comput.*, vol. 15, no. 1, 2020.

[7] S.Siahaan,"Penerapan Algoritma Sequitur Pada Kompresi Record Database Pada Database," *JURIKOM J. Ris. Komput.*, vol. 6, no. 5, pp.511–516, 2019.

[8] B. R. O. Sinambela, M. Syahrizal, and K. Siregar, "Penerapan Algoritma Start-Step-Stop Code Pada Kompresi File Audio, "*KOMIK Konf.Nas. Teknol.Inf. Dan Komput.*, vol. 4, no. 1,2020.

[9] R. Handayani, "Perancangan Aplikasi Kompresi File Audio Menerapkan Algoritma Universal Codes," *KOMIK Konf. Nas. Teknol. Inf. Dan Komput.*,vol. 5, no. 1, 2021.

[10] R. Syahputra, "Perbandingan Algoritma Sequitur dan RLE Dalam KompresiTeks,"*Bull. Artif. Intell.*, vol. 1, no. 1, pp. 33–35, 2022.

[11] C.A.N.Ginting,"Penerapan Algoritma Sequitur Pada Kompresi FileTeks," *KOMIK Konf. Nas. Teknol. Inf. Dan Komput.*, vol. 6, no. 1, pp. 333–339, 2023.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**