



Penerapan Algoritma Boldy Vigna Untuk Mengkompresi Pada File Audio Aplikasi Kajian Dan Murottal Islami

Lobe Lahmuddin

Program Studi Teknik Informatika, Fakultas Teknik Informatika, Universtias Budi Darma Medan, Indonesia

Jl. Sisingamangaraja No.338, Siti Rejo I, Kec. Medan Kota, Kota Medan, Sumatera Utara, Indonesia

Email: lahmuddin804@gmail.com

Abstrak—Pada saat ini telepon genggam telah berkembang menjadi sarana untuk mempelajari suatu pokok permasalahan dalam ruang lingkup kehidupan yang berbeda dengan cakupan yang lebih luas, salah satunya dapat memudahkan setiap orang untuk mengakses berbagai bidang ilmu tanpa terkecuali, bahkan bidang agama sekalipun, kini dapat diakses dengan sebuah aplikasi Kajian Dan Murottal Islami yang tersebar luas di play store. Aplikasi Kajian Murottal Isalami merupakan sebuah aplikasi dengan teknik penyampaian kajian yang jelas. Akan tetapi pada aplikasi kajian dan murottal islami terdapat 192 audio kajian dengan panjangnya durasi audio tersebut maka juga berpengaruh pada kapasitas penyimpanan yang juga akan semakin membesar. Tak heran jika kebanyakan aplikasi kajian dan murottal islami memiliki kapasitas yang besar sehingga akan membutuhkan banyak ruang untuk menyimpan file aplikasi tersebut. Hal ini dapat dihindari apabila file audio aplikasi tersebut melewati proses kompresi terlebih dahulu. Salah satu metode yang digunakan dalam melakukan kompresi atau pemampatan data yaitu algoritma Boldy Vigna. Algoritma Boldy vigna mengurutkan karakter yang paling banyak muncul ke bit terkecil dan karakter yang langka ke bit terbesar. Dengan begitu, ukuran file dapat diminimalisir dari ukuran aslinya..

Kata Kunci: Kompresi, Audio, Boldy Vigna

Abstract—At this time mobile phones have developed into a means to study a subject matter in different spheres of life with a wider scope, one of which can make it easier for everyone to access various fields of knowledge without exception, even in the field of religion, now it can be accessed with a Islamic Studies and Murottal applications that are widespread in the play store. The Murottal Isalami Study Application is an application with a clear study delivery technique. However, in the Islamic study and murottal application there are 192 audio studies with the length of the audio duration, which also affects the storage capacity which will also increase. No wonder most Islamic studies and murottal applications have a large capacity so it will require a lot of space to store the application files. This can be avoided if the application's audio file goes through the compression process first. One of the methods used in compressing or compressing data is the Boldy Vigna algorithm. The Boldy vigna algorithm sorts the most appearing characters to the smallest bit and the rare characters to the largest bit. That way, the file size can be minimized from the original size..

Keywords: Compression, Audio, Boldy Vigna

1. PENDAHULUAN

Pemampatan data (data compression) merupakan salah satu kajian di dalam ilmu komputer yang bertujuan untuk mengurangi besarnya ukuran file audio sebelum menyimpan atau memindahkan data tersebut ke dalam media penyimpanan (storage device). Jika data yang akan disimpan pada media penyimpanan semakin bertambah dan berukuran besar, maka penyimpanan harddisk dapat full karena melebihi kapasitas penyimpanan. Kebiasaan manusia untuk mengumpulkan file dan tidak menghapusnya menyebabkan ruang penyimpanan semakin berkurang.

Dalam mengumpulkan file terkadang sering terjadi berulangnya data atau kumpulan data yang sama dan dapat menyebabkan pemborosan media penyimpanan. Oleh karena itu dilakukan kompresi untuk memadatkan file audio dengan cara menghilangkan berbagai redundansi dan menemukan sumber redundansi disetiap data sehingga ukuran file audio menjadi lebih kecil. Salah satu yang sering terjadi yaitu pada file audio, dimana sekarang ini file audio sering digunakan untuk informasi yang berguna. Jika terjadi berulangnya file audio yang ada ataupun karena sudah terlalu banyak file audio sehingga ruang penyimpanan menjadi lebih sedikit, itu terjadi karena dalam suatu media penyimpanan tidak hanya file audio saja yang ada tetapi banyak file-file lainnya yang berbagi ruang penyimpanan. Salah satu solusi untuk memecahkan masalah tersebut maka diperlukan suatu algoritma kompresi.

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

Dalam proses pembuatan skripsi ini penulis melakukan penerapan metode penelitian dalam memperoleh data-data yang dibutuhkan sehingga menyusun skripsi ini dapat diselesaikan dengan baik. Adapun metode penelitian yang penulis lakukan adalah:

1. Penelitian Kepustakaan (Library Research)
Penulis mempelajari beberapa buku, karya ilmiah dan sumber lainnya agar mendapat data yang terioritas yang berhubungan dengan penelitian ini sebagai bahan acuan untuk memecahkan permasalahan.
2. Perancangan (Design)
Merancang suatu program untuk dapat melihat hasil metode dalam kompresi File audio.



3. Pengkodean (Coding)
Yaitu untuk memudahkan dalam membedakan antara data yang satu dengan yang lainnya.
4. Pengujian (Testing)
Perangkat lunak akan diuji untuk mengetahui apakah sistem tersebut sudah berjalan sesuai fungsi atau tidak. Jika perangkat lunak sudah berfungsi dengan baik maka akan ditarik suatu kesimpulan
5. Penerapan (Application)
Menerapkan algoritma Boldy Vigna dalam pembuatan aplikasi kompresi pada file teks.
6. Laporan (Reprt)
Merupakan laporan ilmiah untuk memudahkan pembaca dalam memahami langkah – langkah yang telah ditempuh dalam penelitian dan hasilnya.

2.2 Kompresi Data

Kompresi sendiri adalah sebuah proses yang dapat mengubah sebuah aliran data masukan (data asli) ke dalam aliran data yang lain (data yang telah dimampatkan) yang memiliki ukuran yang lebih kecil(David, 2010).

$$\text{rasio kompresi} = 100 - \left(\frac{\text{ukuran file setelah dikompresi}}{\text{ukuran file sebelum dikompresi}} \right) \times 100\% \dots\dots\dots [2.1]$$

2.2.1 Jenis-jenis Kompresi Data

Ada dua jenis teknik dalam melakukan proses kompresi data diantaranya adalah sebagai berikut :(David, 2010)

- a. Metode *Lossless*
Metode lossless merupakan metode kompresi dimana data hasil kompresi dapat dikembalikan ke data semula tanpa menghilangkan informasi pada data.
- b. Metode *Lossy*
Metode lossy merupakan data hasil kompresi menjadi lebih kecil ukurannya dibanding dengan metode lossless tetapi menyebabkan hilangnya beberapa informasi pada file hasil kompresi.

2.2.2 Pengukuran Kinerja Kompresi

Terdapat beberapa faktor atau fariabel yang biasa digunakan untuk mengukur kualitas dari suatu teknik kompresi data, yaitu:

- a. *Compression Ratio* (Cr)
$$C_R = \left(\frac{\text{Ukuran data setelah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \right) \times 100\% \dots\dots\dots [2.3]$$

- b. *Space Saving* (Ss)
$$SS = (1 - C_R) \times 100 \% \dots\dots\dots [2.3]$$

2.3 Audio

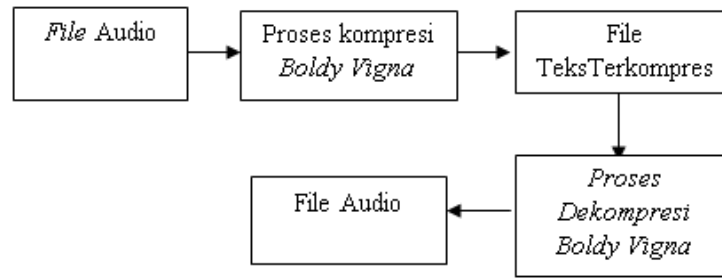
datapat *Audio* merupakan salah satu elemen dalam multimedia, dimana elemen ini dapat dirasakan dengan indera pendengaran. *Audio* terdiri dari kata yang diucap, suara-suara, musik dan bahkan gangguan (*noise*).*Audio* sendiri adalah gelombang yang merambat akibat dari tekanan yang melalui beberapa media (benda padat, cair, atau gas). Suara terdiri dari *frekuensi* dan dapat didengar dalam jangkauan pendengaran tingkatan yang cukup kuat untuk didengar(Arifin et al., 2015).

2.4 Algoritma Boldy Vigna

Zeta (ζ) kode juga dikenal sebagai Boldy Vigna Code, diperkenalkan oleh Paolo Boldi dan Sebastiano Vigna keluarga Variabel-Length Code yang merupakan pilihan terbaik untuk kompresi. Dimulai dengan hokum Zipf, seorang kuasa hukum empiris [Zipf 07] diperkenalkan oleh Linguis George K. Zipf. Menyatakan bahwa frekuensi setiap kata dalam bahasa alami kira-kira berbanding terbalik dengan posisinya dalam table frekuensi.

3. HASIL DAN PEMBAHASAN

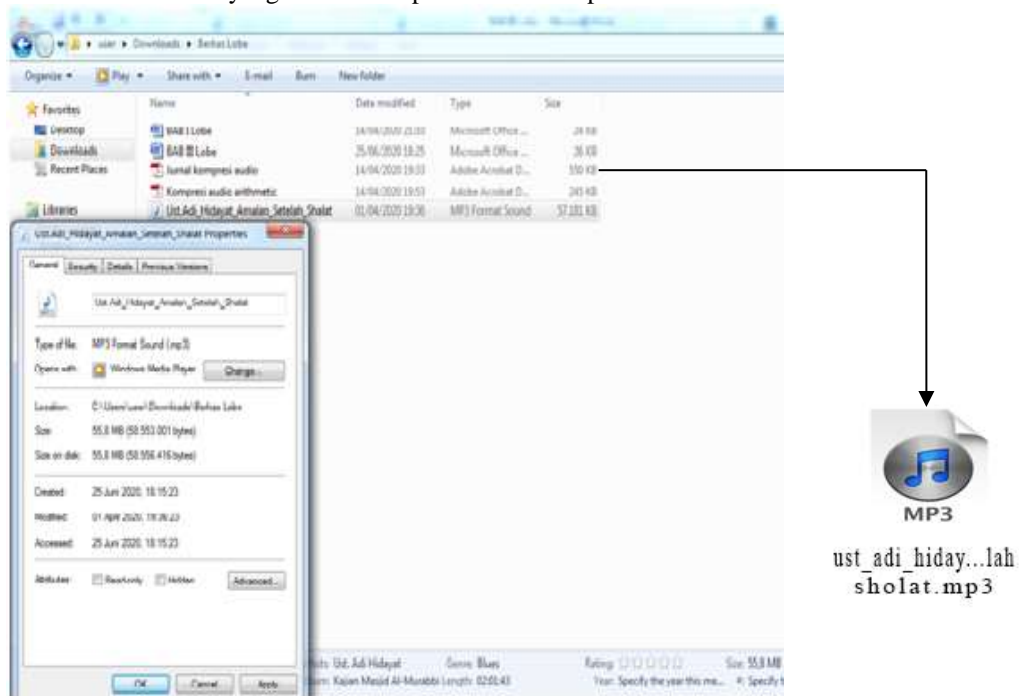
Prosedur untuk mengkompresi file audio dimulai dari memilih file audio yang akan dikompresi, lalu file tersebut akan dikompresi dengan aloritma Boldy Vigna sehingga menghasilkan sebuah file terkompresi yang memiliki ukuran lebih kecil. Sedangkan untuk proses dekompresinya yang menjadi output adalah file terkompresi lalu file tersebut akan didekompresi menunakan algoritma Boldy Vigna sehingga menghasilkan file audio yang memiliki ukuran yang sebelum dikompresi atau balik menjadi file awal. Untuk mengetahui prosedur kompresi dan dekompresi suatu file audio dapat dilihat pada gambar dibawah.



Gambar 1. Diagram Kompresi dan Dekompresi File Audio

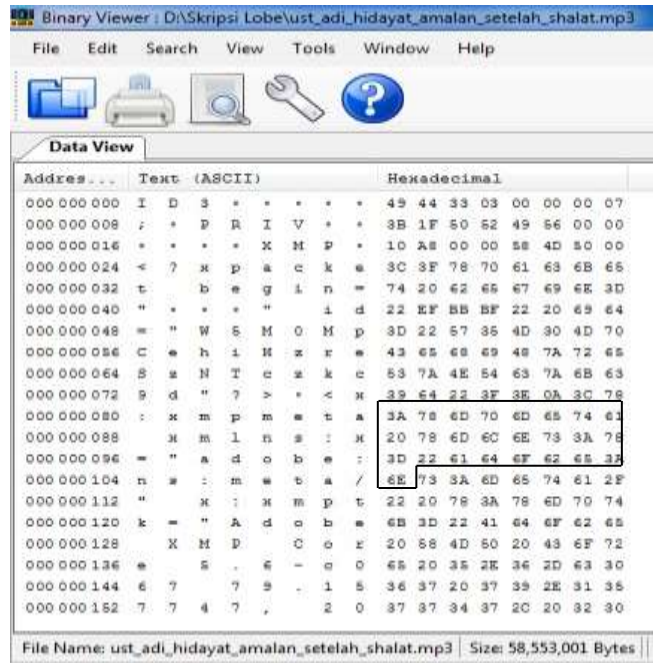
3.1 Penerapan Algoritma Boldy Vigna

File audio yang akan dikompres adalah file berektensi MP3 yang akan dilakukan sebuah proses kompresi. Sebelum file dikompresi, terlebih dahulu dilakukan pembacaan pada file audio untuk mendapatkan file berupa biner. Membaca biner yang terdapat pada file audio menggunakan aplikasi Binary Viewer untuk mencari biner pada aplikasi file MP3. Berikut adalah contoh file audio yang akan dikompresi dan dekompresi.



Gambar 2. File Audio Yang Akan Dikompresi

Memasukkan file Dari sample MP3 maka dapat dilihat nilai hexadecimal menggunakan bantuan software binary viewer seperti Berikut :



Gambar 3. Sampel Nilai *Hexadecimal* File Audio

Berdasarkan contoh diatas didapat nilai binary viewer, adapun nilai hexadecimal yang akan dijadikan sampel untuk penelitian ini pada file audio sampel D:\SkripsiLobe\ust_adi_hidayat_amalan_setelah_shalat.mp3 tersebut adalah :

Tabel 1 Nilai *Hexadecimal* Berdasarkan *Binary Viewer*

3A	78	6D	70	6D	65	74	61
20	78	6D	6C	6E	73	3A	78
3D	22	61	64	6F	62	65	3A
6E							

1. Buat daftar frekuensi kemunculan tiap-tiap karakter dan mengurutkan dari frekuensi terbesar sampai frekuensi terkecil.

Tabel 2. Daftar Frekuensi Kemunculan Karakter

N	Hexadecimal	ASCII Biner	Bit	Frekuensi	Bit x Frekuensi
1	3A	00111010	8	3	24
2	78	01111000	8	3	24
3	6D	01101101	8	3	24
4	65	01100101	8	2	16
5	61	01100000	8	2	16
6	6E	01101110	8	2	16
7	70	01110000	8	1	8
8	74	01110100	8	1	8
9	20	00100000	8	1	8
10	6C	01101100	8	1	8
11	73	01110011	8	1	8
12	3D	00111101	8	1	8
13	22	00100010	8	1	8
14	64	01100100	8	1	8
15	6F	01101111	8	1	8
16	62	01100001	8	1	8
TOTAL BIT					200



2. Mencari codeword dari setiap karakter yang ada berdasarkan data yang ada pada tabel 3.1. Untuk mencari codeword, pertama sekali menentukan nilai $h = 1$, Dengan demikian, $h + 1 = 2$ dan kode unary dari 2 adalah 01. Setelah menentukan nilai h , maka masukan rumus berikut :
 - a. Untuk nilai $k = 3$ dan $h + 1 = 2$ dan kode unary 2 adalah 01.
N interval $[2^3, 2^6 - 1]$
 1. Diberikan $k = 3$ dan $n = 1$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $1 - 2^3 = -7$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -7 akan dikodekan sebagai $x - z - 2^s = -7 - 56 - 2^6 = 15$ dalam 6 bit, sehingga 001111.
 2. Diberikan $k = 3$ dan $n = 2$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $2 - 2^3 = -6$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -6 akan dikodekan sebagai $x - z - 2^s = -6 - 56 - 2^6 = 14$ dalam 6 bit, sehingga 001110.
 3. Diberikan $k = 3$ dan $n = 3$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $3 - 2^3 = -5$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -5 akan dikodekan sebagai $x - z - 2^s = -5 - 56 - 2^6 = 13$ dalam 6 bit, sehingga 001101.
 4. Diberikan $k = 3$ dan $n = 4$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $4 - 2^3 = -4$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -4 akan dikodekan sebagai $x - z - 2^s = -4 - 56 - 2^6 = 12$ dalam 6 bit, sehingga 001100.
 5. Diberikan $k = 3$ dan $n = 5$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $5 - 2^3 = -3$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -3 akan dikodekan sebagai $x - z - 2^s = -3 - 56 - 2^6 = 11$ dalam 6 bit, sehingga 001011.
 6. Diberikan $k = 3$ dan $n = 6$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $6 - 2^3 = -2$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -2 akan dikodekan sebagai $x - z - 2^s = -2 - 56 - 2^6 = 10$ dalam 6 bit, sehingga 001010.
 7. Diberikan $k = 3$ dan $n = 7$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $7 - 2^3 = -1$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -1 akan dikodekan sebagai $x - z - 2^s = -1 - 56 - 2^6 = 9$ dalam 6 bit, sehingga 001001.
 8. Diberikan $k = 3$ dan $n = 8$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $8 - 2^3 = 0$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 0 akan dikodekan sebagai $x - z - 2^s = 0 - 56 - 2^6 = 8$ dalam 6 bit, sehingga 001000.
 9. Diberikan $k = 3$ dan $n = 9$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $9 - 2^3 = 1$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 1 akan dikodekan sebagai $x - z - 2^s = 1 - 56 - 2^6 = 9$ dalam 6 bit, sehingga 001001.
 10. Diberikan $k = 3$ dan $n = 10$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $10 - 2^3 = 2$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 2 akan dikodekan sebagai $x - z - 2^s = 2 - 56 - 2^6 = 10$ dalam 6 bit, sehingga 001010.
 11. Diberikan $k = 3$ dan $n = 11$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $11 - 2^3 = 3$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai -6 akan dikodekan sebagai $x - z - 2^s = 3 - 56 - 2^6 = 11$ dalam 6 bit, sehingga 001011.
 12. Diberikan $k = 3$ dan $n = 12$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $12 - 2^3 = 4$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 4 akan dikodekan sebagai $x - z - 2^s = 4 - 56 - 2^6 = 12$ dalam 6 bit, sehingga 001100.
 13. Diberikan $k = 3$ dan $n = 13$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $13 - 2^3 = 5$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 5 akan dikodekan sebagai $x - z - 2^s = 5 - 56 - 2^6 = 13$ dalam 6 bit, sehingga 001101.
 14. Diberikan $k = 3$ dan $n = 14$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $14 - 2^3 = 6$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 6 akan dikodekan sebagai $x - z - 2^s = 6 - 56 - 2^6 = 14$ dalam 6 bit, sehingga 001110.
 15. Diberikan $k = 3$ dan $n = 15$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $15 - 2^3 = 7$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 7 akan dikodekan sebagai $x - z - 2^s = 7 - 56 - 2^6 = 15$ dalam 6 bit, sehingga 001111.
 16. Diberikan $k = 3$ dan $n = 16$, di interval $[2^3, 2^6 - 1]$, yang sesuai dengan $h = 1$, Dengan demikian biner minimal $16 - 2^3 = 8$ Panjang z dari interval $[2^3, 2^6 - 1]$ adalah 56. Ini berarti $s = \lceil \log_2 56 \rceil = 6$. Nilai 8 akan dikodekan sebagai $x - z - 2^s = 8 - 56 - 2^6 = 16$ dalam 6 bit, sehingga 010000.
 3. Mengganti karakter yang ada dengan codeword yang telah dihasilkan, dan dapat dilihat pada tabel 3.4 :



Tabel 3. Pergantian Karakter Menjadi Codeword

N	Hexadecimal	Codeword	Bit	Frekuensi	Bit x Frekuensi
1	3A	001111	6	3	18
2	78	001110	6	3	18
3	6D	001101	6	3	18
4	65	001100	6	2	12
5	61	001011	6	2	12
6	6E	001010	6	2	12
7	70	001001	6	1	6
8	74	001000	6	1	6
9	20	001001	6	1	6
10	6C	001010	6	1	6
11	73	001011	6	1	6
12	3D	001100	6	1	6
13	22	001101	6	1	6
14	64	001110	6	1	6
15	6F	001111	6	1	6
16	62	010000	6	1	6
TOTAL BIT					150

Berdasarkan pada tabel di atas dapat di bentuk nilai bit baru hasil kompresi dari susunan nilai hexadecimal sampel awal sebelum kompresi yaitu, 3A, 78, 6D, 70, 6D, 65, 74, 61, 20,78, 6D, 6C, 6E, 73, 3A, 78, 3D, 22, 61, 64, 6F, 62, 65, 3A, 6E. (tanpa ada tanda koma dan spasi) menjadi nilai bit biner “00111100111000110100100100110100110000100000101100100100111000110100101000101000101100111100111000110000110100101100111000111101000001100001111001010”.

Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan stringbit itu sendiri yaitu padding bit dan flag bit.

$$7 - n + "1"$$

$$7 - 6 + "1" = 01$$

Bit Akhir 9 - n

$$\text{Bit Akhir} = 9 - 6 = 3 = 00000011$$

“0011110011100011010010010011010011000010000010110010010011100011010010100010100010110011110011100011000110100101100111000111101000001100001111001010010000011” Total panjang bit keseluruhan setelah ada penambahan bit adalah 150+2+8=160. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti berikut.

00111100-11100011-01001001-00110100-11000010-00001011-00100100-11100011-01001010-00101000-10110011-11001110-00110000-11010010-11001110-00111101-00000011-00001111-00101001-00000011

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 3 kelompok nilai biner baru yang sudah terkompresi beserta nilai biner penambahan bit. Setelah pembagian dilakukan, maka nilai yang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari string bit tersebut menggunakan kode ASCII untuk mengetahui nilai yang sudah terkompresi. nilai simbol yang sudah terkompresi dapat dilihat pada tabel di bawah ini:

Tabel 4. Nilai Desimal Sample Terkompresi

Biner	Nilai Desimal Terkompresi	Simbol
00111100	60	<
11100011	227	Ã
01001001	73	I
00110100	52	4
11000010	192	À
00001011	11	
00100100	36	\$
11100011	227	Ã
01001010	74	J
00101000	40	(
10110011	179	³
11001110	206	î
00110000	48	0



11010010	210	Ò
11001110	206	Î
00111101	61	=
00000011	3	
00001111	15	
00101001	41)
00000011	3	

Setelah mengetahui hasil kompresi tersebut, hasil kompresi dapat di ukur atau di hitung sebagai berikut :

$$Cr = 100\% - \frac{ukuranbitdatasetelahdikompresi}{ukuranbitdatasebelumdikompresi} \times 100\%$$

$$Cr = 100\% - \frac{150bit}{200bit} \times 100\%$$

$$Cr = 100\% - 75\%$$

$$Cr = 25\%$$

Space Savings(SS) adalah persentase selisih antara data yang belum dikompresi dengan data yang telah dikompresi dan berikut ini adalah rasio hasil perbandingan kompresi.

$$SS = 1 - C_r \times 100 \%$$

$$SS = 1 - 25 \% \times 100 \%$$

$$SS = 0,75 \%$$

3.2 Hasil Implementasi Program

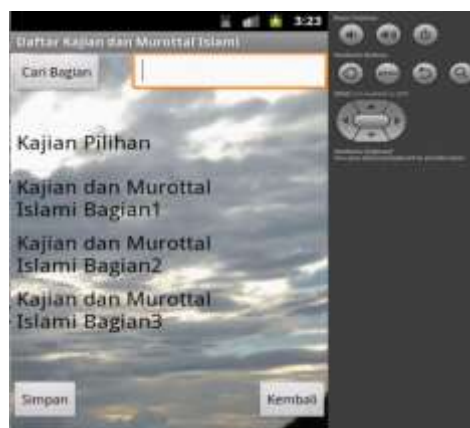
Pengujian program merupakan tahap mengidentifikasi hasil dari penerapan sistem aplikasi. Hasil pengujian menampilkan hasil output dari sebuah input data pada program, dimana proses awal untuk mendapatkan hasil pengujian program.

1. Perancangan Menu Utama



Gambar 4. Form Menu Utama

2. Rancangan Form Menu Daftar Bagian



Gambar 5. *Form Menu Daftar Bagian*

3. *Form* Daftar Kajian Islami

**Gambar 6.** *Form* Daftar Kajian Islami

4. *Form* About

**Gambar 7.** *Form* About

4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. Berdasarkan prosedur kompresi dengan menggunakan Algoritma Boldy Vigna telah berhasil melakukan proses kompresi file audio berformat MP3. Sehingga proses kompresi dapat berjalan sesuai dengan teknik kompresi. Berdasarkan penerapan Algoritma Boldy Vigna telah membuktikan bahwa suatu file audio yang memiliki ukuran besar dapat dikompres menjadi ukuran yang lebih kecil. Berdasarkan dari hasil pengujian terhadap sistem bahwa ukuran file audio lebih kecil setelah dilakukan kompresi. Kompresi file audio dapat dirancang dan dibangun dengan menggunakan aplikasi Eclipse dengan menerapkan algoritma Boldy Vigna sehingga diharapkan memudahkan penulis dalam mengkompresi ukuran file audio.

REFERENCES

- [1] Mutiara Diah, Annisa, Sutardi, Ramadhan, Rahmat, Aplikasi Kompresi File Audio Menggunakan Algoritma Arithmetic Coding. *semanTIK*, pp. 29-38, 2016
- [2] R. Warsita, Rahmat Agus, yoannita, Rancangan Bangunan Aplikasi Kompresi Audio Berbasis Android Menggunakan Algoritma Huffman, *STMIK Global Informasi MDP*, pp. 1-11
- [3] S. David, "Hand Book Of Data Compression," in *Compression*, 5th ed., California: Springer London Dordrecht Heidelberg New York, 2010, p. 2.
- [4] H. Rahmat, "IMPLEMENTASI KOMPRESI DATA PADA JARINGAN KOMPUTER MENGGUNAKAN ALGORITMA ZLIB," *Kompresi*, vol. 9, pp. 2-7.
- [5] D. Salomon, *Data Compression*, 4th ed. California: Springer Verlag London New York, 2007.



- [6] Y. Arifin, S. Kom, M. Y. Ricky, S. Kom, V. Yesmaya, and S. Kom, *DIGITAL MULTIMEDIA*. Jakarta Barat, DKI Jakarta Raya 11480: PT WIDIA INOVASI NUSANTARA, 2015.
- [7] P. Studi, I. Komputasi, U. Telkom, and L. Firmansah, “Kompresi Data Audio Lossless format FLAC menjadi Audio Lossly format MP3 dengan Algoritma Huffman Shift Coding AUDIO LOSSY FORMAT MP3 DENGAN ALGORITMA HUFFMAN,” vol. 2, no. 3, pp. 8066–8073, 2015.
- [8] C. R. C. P. Llc, *THE TRANSFORM AND DATA COMPRESSION*. 2001.
- [9] M. Syahlan, A. Kunda, S. Alam, Aplikasi Manajemen Informasi Jadwal Kajian Islami Berbasis Android. 2019.
- [10] Permata, Ahmad Nora, Metodologi Studi Agama, Ypgyakarta : Pustaka Pelajar, 2000.
- [11] Fl.Sigit Suyantoro, Ed., *ALGORITMA DAN PEMOGRAMAN*, 2nd ed. Yogyakarta: C.V Andi, 2012.
- [12] R. A.S, *REKAYASA PERANGKAT LUNAK TERSTRUKTUR DAN BERORIENTASI OBJEK*. Bandung: Informatika Bnadung, 2016.
- [13] A. Nugroho, *Pemograman JAAVA untuk Aplikasi Basis Data dengan Teknik XP Menggunakan IDE Eclipse*, 1st ed. Yogyakarta, 2007.
- [14] M. S. Alfa Satyaputra, *JAVA for Beginners with eclipse 42 Juno*. Jakarta, 2012.