

Wave File Compression Analysis and Implementation using The Huffman Method

Edward Robinson Siagian

Universitas Budidarma Medan, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 28 Juni 2022
Revisi Akhir: 30 Juni 2022
Diterbitkan *Online*: 30 Juni 2022

KATA KUNCI

Compression; Wave; Huffman

KORESPONDENSI

Phone: -

E-mail: Edwardrobin129@gmail.com

A B S T R A K

One of the sound file formats that are widely used in the Windows operating system is the Wave format (*. WAV). This format is widely used for gaming and multimedia purposes. Wave is actually a raw format where the voice signal is directly recorded and quantized into digital data. The basic format of this file by default does not support compression and is known as PCM (Pulse Code Modulation). If you record a CD Audio quality song using a sampling rate of 44.1 kHz, 16 bits per sample, 2 channels (stereo), then the total media required to store this audio data per second is 176,400 bytes, so for a duration of 1 minute it takes 10,584 MB. If the average duration of one song is 5 minutes, it takes more than 50 MB of space to store the audio data of the song. This is of course very wasteful of storage media such as hard disks even though there are currently large hard disk capacities available.

PENDAHULUAN

Suara yang kita dengar sehari-hari adalah merupakan gelombang *analog*. Gelombang ini berasal dari tekanan udara yang ada di sekeliling kita, yang dapat kita dengar dengan bantuan gendang telinga. Gendang telinga ini bergetar, dan getaran ini dikirim dan diterjemahkan menjadi informasi suara yang dikirimkan ke otak, sehingga kita dapat mendengarkan suara. Suara yang kita hasilkan sewaktu berbicara berbentuk tekanan suara yang dihasilkan oleh pita suara. Pita suara ini akan bergetar, dan getaran ini menyebabkan perubahan tekanan udara, sehingga kita dapat mengeluarkan suara. Komputer hanya mampu mengenal sinyal dalam bentuk *digital*. Bentuk *digital* yang dimaksud adalah tegangan yang diterjemahkan dalam angka "0" dan "1", yang juga disebut dengan istilah "*bit*". Tegangan ini berkisar 5 volt bagi angka "1" dan mendekati 0 volt bagi angka "0". Dengan kecepatan perhitungan yang dimiliki komputer, komputer mampu melihat angka "0" dan "1" ini menjadi kumpulan *bit-bit* dan menerjemahkan kumpulan *bit-bit* tersebut menjadi sebuah informasi yang bernilai.

Bagaimana caranya memasukkan suara *analog* ini sehingga dapat dimanipulasi oleh peralatan elektronik yang ada? Alat yang diperlukan untuk melakukan ini adalah *transducer*. Dalam hal ini, *transducer* adalah istilah untuk menyebut sebuah peralatan yang dapat mengubah tekanan udara (yang kita dengar sebagai suara) ke dalam tegangan elektrik yang dapat dimengerti oleh perangkat elektronik, serta sebaliknya. Contoh *transducer* adalah mikrofon dan *speaker*. Mikrofon dapat mengubah tekanan udara menjadi tegangan elektrik, sementara *speaker* melakukan pekerjaan sebaliknya.

Tegangan elektrik diproses menjadi sinyal *digital* oleh *sound card*. Ketika Anda merekam suara atau musik ke dalam komputer, *sound card* akan mengubah gelombang suara (bisa dari mikrofon atau *stereo set*) menjadi data *digital*, dan ketika suara itu dimainkan kembali, *sound card* akan mengubah data *digital* menjadi suara yang kita dengar (melalui

speaker), dalam hal ini gelombang *analog*. Proses perubahan gelombang suara menjadi data digital ini dinamakan *Analog-to-Digital Conversion* (ADC), dan kebalikannya, perubahan data *digital* menjadi gelombang suara dinamakan *Digital-to-Analog Conversion* (DAC).

Dalam masalah transfer data, ukuran *file* yang kecil akan mempercepat waktu transmisi. Dalam beberapa kasus seperti seseorang ingin memberikan datanya kepada temannya tetapi ukuran *file* data misalkan ukurannya sebesar 1,6 MB. Jika ia ingin menyimpannya dalam sebuah disket baru diberikan kepada temannya, maka *file* tersebut tidak akan muat. Untuk itu sebelum di-*copy*-kan *file* tersebut dapat dikompresi dulu sehingga ukurannya lebih kecil dari ukuran semula dan dapat muat ke dalam disket tersebut. *File* merupakan data digital yang berupa representasi atas bit '0' dan '1'. Seringkali dalam sebuah *file* terjadi perulangan data atau *redundancy* [1].

Semua metode kompresi melakukan pemadatan terhadap data berulang tersebut. Seperti diketahui jenis algoritma kompresi terbagi atas *lossless compression* dan *lossy compression*. Pada *lossy compression* ada data yang hilang tetapi tidak banyak setelah data dikompresi. Contoh standar yang menggunakan jenis *lossy compression* adalah JPEG (*Joint Picture Experts Group*) sebagai standar *image* gambar atau *still image*, MPEG (*Motion Picture Experts Group*) untuk *audio video* seperti Video CD, MP3 (*MPEG-1 Layer 3*) untuk *audio*. Data hasil kompresi dengan *lossy compression* jika dikembalikan maka hasilnya tidak akan sama persis lagi dengan data orisinal. Berbeda dengan *lossy compression*, pada *lossless compression* tidak ada data yang hilang setelah proses kompresi dan data dapat dikembalikan seperti data semula. Contoh standar yang menggunakan jenis ini adalah Gzip, Unix Compress, WinZip, GIF (*Graphic Interchange Format*) untuk *still image*, dan Morse Code [2].

Algoritma kompresi Huffman atau disebut dengan *encoding* Huffman adalah algoritma yang dipakai untuk mengkompresi *file*. Teknik kompresi ini dengan menggantikan code yang lebih kecil pada karakter yang sering dipakai dan *code* yang lebih panjang untuk karakter yang tidak begitu sering dipakai. *Code* dalam hal ini adalah urutan bit berupa nilai '0' dan '1' yang secara unik merepresentasikan sebuah karakter. Ide dasar dari *encoding* Huffman adalah mencocokkan *code word* yang pendek pada blok input dengan kemungkinan yang terbesar dan *code word* yang panjang dengan kemungkinan terkecil. Konsep ini mirip dengan *Morse Code*. Suatu *file* merupakan kumpulan dari karakter-karakter. Dalam suatu *file* tertentu suatu karakter dipakai lebih banyak daripada yang lain. Jumlah bit yang diperlukan untuk merepresentasikan tiap karakter bergantung pada jumlah karakter yang harus direpresentasikan. Dengan menggunakan satu bit maka dapat merepresentasikan dua buah karakter. Sebagai contoh 0 merepresentasikan karakter pertama dan 1 merepresentasikan karakter kedua. Dengan menggunakan dua bit maka dapat merepresentasikan 2^2 atau 4 buah karakter [3].

METODOLOGI

Pada tahap ini dilakukan dengan mempelajari teori dasar yang mendukung penelitian, pencarian dan pengumpulan data-data yang dibutuhkan. Untuk mengumpulkan data yang dibutuhkan, maka penulis memakai teknik :

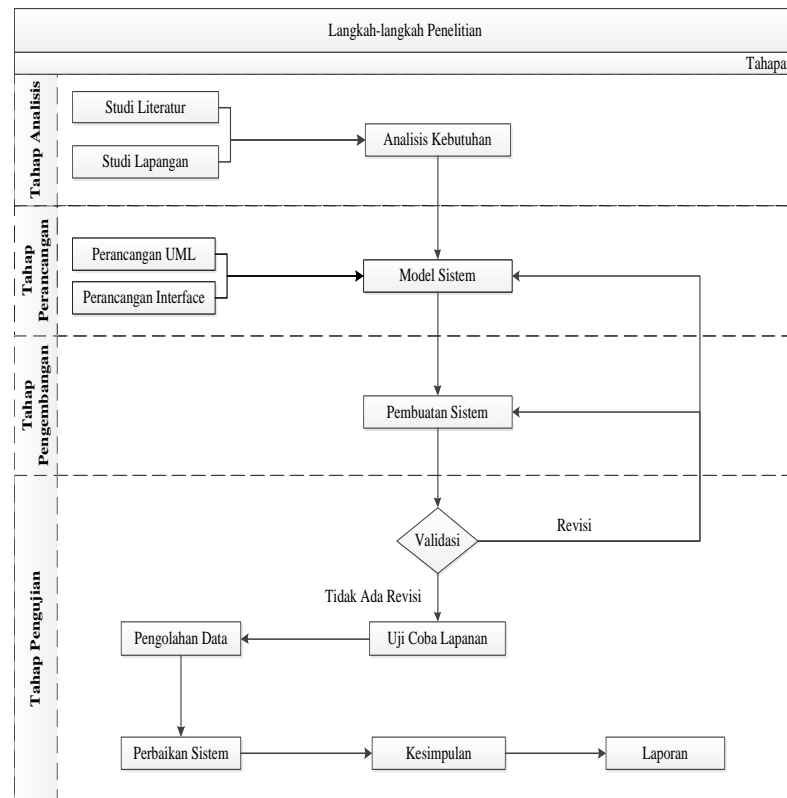
Penelitian Kelapangan

1. Pengamatan Langsung (*Observation*)
2. Penelitian perpustakaan (*Library Research*)

Pada metode ini penulis mengutip dari beberapa bacaan yang berkaitan dengan pelaksanaan penelitian yang dikutip dapat berupa teori yaitu jurnal. Metodologi penelitian yang digunakan dalam melakukan kompresi *file wave* dengan menggunakan metode *Huffman*, antara lain:

Bagan Alur Penelitian

Bagan alur dari penelitian ini dapat dilihat seperti gambar 1 berikut ini :



Gambar 1. Bagan alur penelitian

Tahap Analisis

Pada tahap ini dilakukan proses analisis kebutuhan terhadap sistem yang akan dirancang dan dibangun agar dapat dipahami oleh perangkat lunak seperti apa yang dibutuhkan oleh user berdasarkan studi literature dan studi lapangan. Pada tahap ini juga di analisa kebutuhan perangkat lunak dan perangkat keras dari sistem yang akan dirancang. Spesifikasi kebutuhan perangkat lunak dan perangkat keras pada tahap ini perlu untuk didokumentasikan. Suatu *Software Requirements Spefication* harus mencantumkan tentang deskripsi dengan lingkungannya. Adapun spesifikasi kebutuhan dari sistem yang akan dibangun adalah sebagai berikut :

Spesifikasi perangkat keras yang dibutuhkan adalah:

1. Laptop/PC
2. Ram 2 GB
3. Hardisk 200 GB
4. Dual Core

Spesifikasi perangkat Lunak yang dibutuhkan adalah

1. Sistem operasi Windows 8.1
2. *Visual Basic 2008*
3. Prosesor Core I3

Tahap Perancangan

Tahap perancangan merupakan tahap permodelan dari sistem yang akan dibangun. Permodelan Sistem merupakan suatu bentuk penyederhanaan dari sebuah elemen dan komponen yang sangat kompleks untuk memudahkan pemahaman dari informasi yang dibutuhkan. Tahap perancangan ini terdiri atas :

Perancangan UML

Pada tahap ini dilakukan perancangan dari modelan sistem yang akan dibangun menggunakan model UML. Setelah dirancang model sistem yang akan dibangun selanjutnya dilakukan perancangan *interface* dari sistem yang akan dibangun.

Perancangan *Interface*

Pada tahap ini dilakukan proses perancangan *interface* yakni dengan merancang bentuk tampilan dari sistem yang akan

dibangun baik itu tampilan halaman *input*, halaman proses dan halaman *output* sehingga dapat memberikan kemudahan.

Tahap Pengujian (Validasi)

Validasi merupakan proses untuk menunjukkan seberapa besar nilai keakuratan program terhadap kondisi-kondisi saat pemakaian sebenarnya. Proses ini menjalankan skenario berdasarkan data dan lingkungan yang merepresentasikan dunia nyata dengan menggunakan mesin percobaan. Pada tahap ini akan terjadi beberapa perbaikan, baik itu dari segi perancangan sistem maupun pengembangan sistem sampai didapat sistem yang sesuai dan laporan yang dihasilkan sesuai dengan kebutuhan atau perlu dilakukan perbaikan.

HASIL DAN PEMBAHASAN

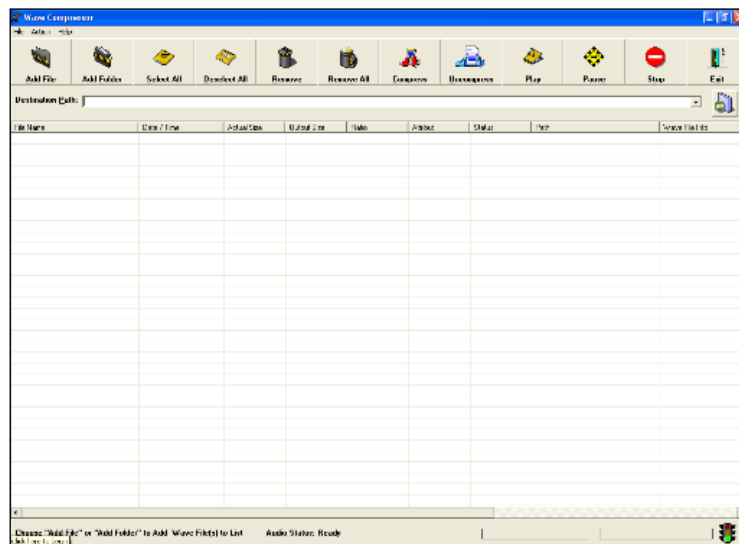
Untuk mengetahui hasil pengujian program kompresi dan dekompresi *file Wave* ini dengan algoritma kompresi Huffman yang telah diimplementasikan maka dilakukan pengujian pada beberapa jenis *file Wave*. *File Wave* yang diuji mempunyai ukuran besar yang bervariasi dan pengujian dilakukan terhadap *file Wave* yang disertakan pada sistem operasi Windows di bagian *folder* “\Windows\Media”.

Dari hasil pengujian proses kompresi didapat bahwa rasio mempunyai *range* antara 67,52% untuk nilai terendah dan tertinggi 91,21%. Jika dicari hasil rasio kompresi tersebut secara rata-rata adalah sebesar 82,11%. Ini berarti ukuran *file* hasil adalah 0,8211 kali ukuran *file* semula dan pengurangan ukuran *file* sebesar $(100\% - 82,11\%) = 17,89\%$. Nilai ini cukup terutama dalam mengkompresi *file Wave* berukuran besar misalnya berukuran di atas 10 MB. Dari hasil tersebut juga menunjukkan bahwa persentase kompresi atau dekompresi *file Wave* tidak bergantung pada ukuran *file* melainkan bergantung pada isi data pada *file Wave*. Semakin banyak perulangan data yang terdapat pada bagian *chunk* data *file Wave* maka rasio kompresi akan semakin rendah. Kecepatan kompresi memang tidak dilakukan pengujian tetapi dari beberapa pengujian yang dilakukan tingkat kecepatan baik untuk proses kompresi dan dekompresi berbanding lurus dengan ukuran *file Wave*, artinya semakin besar ukuran *file Wave* yang diproses maka semakin lama proses berlangsung.

Tampilan-tampilan dalam melakukan kompresi *file wave* dengan metode Huffman, antara lain:

Menu Utama

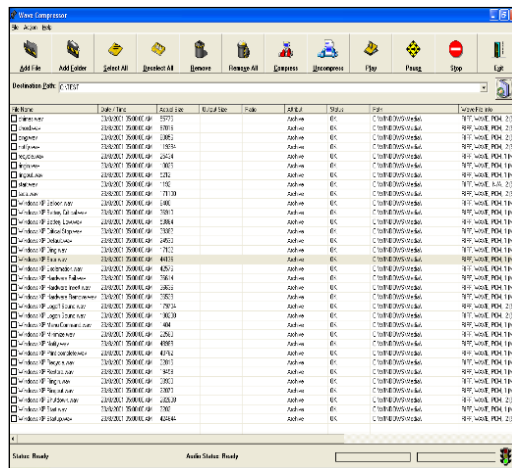
Berikut ini merupakan tampilan menu utama dalam melakukan kompresi *file wave*.



Gambar 1. Menu Utama

File Wave

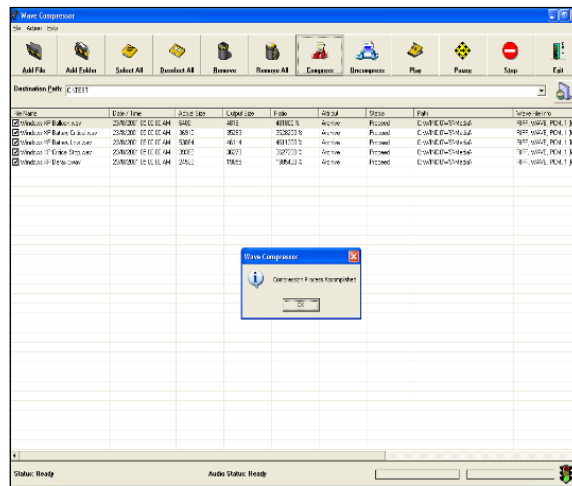
Berikut ini merupakan daftar jenis *file wave* yang dijadikan sebagai sampel untuk dilakukan kompresi.



Gambar 2. List File Wave

Tampilan Proses Kompresi

Untuk proses komnpresi maka dapat ditampilkan seperti gambar 3 dibawah ini.



Gambar 3. Proses Kompresi

KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan selama membuat aplikasi ini, maka dapat ditarik beberapa kesimpulan sebagai berikut:

Reduksi ukuran *file* yang diperoleh dengan algoritma Huffman ini berkisar dari *range* 20% hingga 40%. Jadi dapat dikatakan dengan rasio kompresi ini algoritma Huffman sudah dikatakan baik dalam hal mengkompresi *file* khususnya *file Wave*. Tingkat kompresi dipengaruhi oleh banyaknya nada yang sama dalam *file Wave*. Kecepatan proses tidak bergantung pada data yang diproses tetapi berbanding lurus dengan ukuran *file Wave*, artinya semakin besar ukuran *file Wave* yang diproses maka semakin lama waktu prosesnya. Proses dekomposisi lebih cepat dilakukan dibandingkan dengan proses kompresi karena pada proses dekomposisi tidak dilakukan lagi proses pembentukan pohon Huffman dari data melainkan hanya langsung membaca dari tabel *code* pohon Huffman yang disimpan pada *file* sewaktu proses kompresi. *File Wave* yang telah dikompresi bila dilakukan proses kompresi sekali lagi maka ukuran *file* akan bertambah besar sedikit karena algoritma Huffman merupakan *optimal compression* jadi *file* yang dilakukan kompresi sebanyak dua kali maka proses terakhir tidak akan mereduksi ukuran *file* lagi. Terjadi penambahan *byte* pada proses kompresi kedua kalinya karena program menyimpan struktur pohon Huffman dari hasil kompresi pertama. *File Wave* yang telah dikompresi tersebut hanya dapat dimainkan dari program ini.

DAFTAR PUSTAKA

[1] Basalamah, Affah, Teknologi Multimedia MP3, PT. Elex Media Komputindo, Jakarta, 2001
 [2] Hadi R, Pemrograman Windows API dengan Microsoft Visual Basic, PT. Elex Media Komputindo, Jakarta, 2001.

- [3] Halvorson M, Microsoft Visual Basic 6.0 Professional, Step by Step, PT. Elex Media Komputindo, Jakarta, 2000.
- [4] Microsoft Developer Network (MSDN) Library Visual Studio 6.0, Microsoft Corporation, 1998.
- [5] Shannon, C. E., A Mathematical Theory of Communication, The Bell System Technical Journal, Vol. 27, pp. 379 – 423, 623 – 656, July, October, 1948.
- [6] http://www.replaygain.hydrogenaudio.org/file_format_wav.html, 2022