

**KOMPRESI FILE AUDIO WAVE MENGGUNAKAN ALGORITMA
HUFFMAN SHIFT CODING**

JURNAL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer



Disusun oleh:

GALANG BAGUS PRASETYO

0710963022-96

**PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

UNIVERSITAS BRAWIJAYA

MALANG

2013

KOMPRESI FILE AUDIO WAVE MENGGUNAKAN ALGORITMA HUFFMAN SHIFT CODING

Galang Bagus Prasetyo¹; Edy Santoso²; Marji³

Jurusan Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer
Universitas Brawijaya

Jalan MT. Haryono 167, Malang 65145, Indonesia

e-mail: galangbagusp@gmail.com¹, edy144@ub.ac.id², marji@ub.ac.id³

ABSTRAK

File WAV adalah file audio standar yang digunakan oleh Windows. Format WAV banyak digunakan untuk keperluan game dan multimedia. Wave sebenarnya merupakan format kasar dimana signal suara langsung direkam dan dikuantisasi menjadi data digital. Format dasar dari file ini secara default tidak mendukung kompresi dan dikenal dengan nama PCM (Pulse Code Modulation). Algoritma Huffman merupakan algoritma kompresi lossless, yaitu teknik kompresi yang tidak mengubah data aslinya. Hal tersebut yang menyebabkan algoritma ini banyak dipakai dalam proses kompresi. Algoritma Huffman bekerja dengan cara melakukan pengkodean dalam bentuk bit untuk mewakili data karakter. Algoritma ini kurang maksimal jika ada banyak variasi simbol. Untuk mengoptimalkan algoritma huffman ini bisa digunakan algoritma Huffman Shift Coding yang akan membagi simbol awal menjadi beberapa blok. Penggunaan Algoritma Huffman Shift Coding untuk kompresi file audio wave menghasilkan ratio kompresi rata-rata sebesar 14,87% untuk nilai $k=2$ dan 8,72% untuk nilai $k=3$.

Kata Kunci: Huffman Coding, Huffman Shift Coding, File Audio Wave, Kompresi

1. Pendahuluan

File WAV adalah file audio standar yang digunakan oleh Windows. Format WAV banyak digunakan untuk keperluan game dan multimedia. Wave sebenarnya merupakan format kasar dimana signal suara langsung direkam dan dikuantisasi menjadi data digital. Format dasar dari file ini secara default tidak mendukung kompresi dan dikenal dengan nama PCM (Pulse Code Modulation).[1]

Jika direkam suatu lagu sekualitas CD Audio menggunakan *sampling rate* 44,1 kHz, 16 bit per sample, 2 kanal (stereo), maka total media yang diperlukan untuk menyimpan data audio ini per detik adalah 176.400 byte sehingga untuk durasi 1 menit diperlukan 10,584 MB. Jika rata-rata durasi satu lagu selama 5 menit, maka dibutuhkan tempat lebih dari 50 MB untuk menyimpan data audio lagu tersebut. Ini tentunya sangat memboroskan media penyimpanan seperti hard disk meskipun saat ini telah tersedia kapasitas hard disk yang besar. Masalah tersebut dapat diatasi bila file Wave tersebut dikompresi untuk mengurangi ukurannya.

Kompresi data dilakukan untuk mereduksi ukuran data atau file. Dengan melakukan kompresi atau pemadatan data maka ukuran file atau data akan lebih kecil sehingga dapat mengurangi waktu transmisi sewaktu data dikirim dan tidak banyak menghabiskan ruang media penyimpan. Ada banyak algoritma untuk kompresi, salah satunya yaitu algoritma Huffman Coding.[2]

Algoritma Huffman merupakan algoritma kompresi lossless, yaitu teknik kompresi yang tidak mengubah data aslinya. Hal tersebut yang menyebabkan algoritma ini banyak dipakai dalam proses kompresi. Algoritma Huffman bekerja dengan cara melakukan pengkodean dalam bentuk bit untuk mewakili data karakter. Algoritma ini kurang maksimal jika ada banyak variasi simbol. Untuk mengoptimalkan algoritma huffman ini bisa digunakan algoritma Huffman Shift Coding yang akan membagi simbol awal menjadi beberapa blok.

Berdasarkan latar belakang yang telah dikemukakan, maka judul yang diambil dalam skripsi ini adalah "Kompresi

File Audio Wave Menggunakan Algoritma Huffman Shift Coding”.

1.1 Rumusan masalah

Perumusan masalah yang disimpulkan pada pembuatan skripsi ini adalah

- Bagaimana merancang dan mengembangkan aplikasi yang menggunakan *huffman shift coding* untuk kompresi file audio wave.
- Bagaimana tingkat kompresi penggunaan algoritma *huffman shift coding* pada file audio wave.

1.2 Tujuan

Tujuan dari pembuatan skripsi ini adalah

- Menghasilkan aplikasi *huffman shift coding* untuk kompresi file audio wave.
- Mengetahui tingkat kompresi penggunaan algoritma *huffman shift coding* pada file audio wave.

1.3 Batasan masalah

Batasan masalah adalah:

- File* Input hanya berupa *file* format *Wave*.
- Program tidak dapat melakukan perubahan jumlah kanal (*channel*), *bit per sample*, dan *sampling rate* file *Wave*.
- Program dapat melakukan *play* file audio.

2. LANDASAN TEORI

2.1. Pengertian Audio Digital

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. [3]

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.

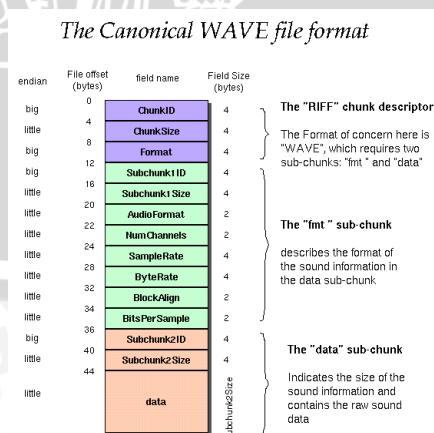
2.2. Kelebihan Audio Digital

Kelebihan *audio digital* adalah kualitas reproduksi yang sempurna. Kualitas reproduksi yang sempurna yang dimaksud adalah kemampuannya untuk menggandakan sinyal audio secara berulang-ulang tanpa mengalami penurunan kualitas suara.

Kelebihan lain dari *audio digital* adalah ketahanan terhadap *noise* (sinyal yang tidak diinginkan). Pada saat transmisi data dan pemrosesan dengan komponen-komponen elektrik, pada sinyal analog sangat mudah sekali terjadi gangguan-gangguan berupa *noise*. Suara desis pada kaset rekaman merupakan salah satu contoh terjadinya *noise* berupa gangguan pada frekuensi tinggi.

2.3. Format Wave PCM

Jenis format Wave ini merupakan jenis *file* Wave yang paling umum dan hampir dikenal oleh setiap program. Format Wave PCM (*Pulse Code Modulation*) adalah *file wave* yang tidak terkompresi, akibatnya ukuran *file* sangat besar jika *file* mempunyai durasi yang panjang. Berikut ini diagram (Gambar 2.1) yang menggambarkan format *file* Wave PCM.



Gambar 2.1 Diagram Format File Wave

2.4. Pengertian Kompresi

Kompresi adalah pengubahan data kedalam bentuk yang memerlukan *bit* yang lebih sedikit, biasanya dilakukan agar data dapat disimpan atau dikirimkan dengan lebih efisien. Kebalikan dari proses kompresi, yaitu dekompresi. Dekompresi merupakan proses untuk mengembalikan data baru yang telah dihasilkan oleh proses kompresi menjadi data awal. Dekompresi yang menghasilkan data sama persis dengan data aslinya sebelum dikompresi, maka kompresi tersebut disebut *lossless compression*. Sebaliknya, jika hasil dekompresi tersebut tidak sama dengan data aslinya sebelum dikompresi, karena ada data yang dihilangkan karena dirasa tidak terlalu penting tetapi tidak mengubah informasi yang dikandungnya, maka disebut *loosy compression*. [2]

2.5. Algoritma Kompresi Huffman

Algoritma kompresi Huffman dinamakan sesuai dengan nama penemunya yaitu David Huffman, seorang profesor di MIT (*Massachusetts Institute of Technology*).

Kompresi Huffman merupakan algoritma kompresi *lossless* dan ideal untuk mengkompresi teks atau *file* program. Ini yang menyebabkan mengapa algoritma ini banyak dipakai dalam program kompresi.

Kompresi Huffman termasuk dalam algoritma keluarga dengan *variable codeword length*. Ini berarti simbol individual (karakter dalam sebuah *file* teks sebagai contoh) digantikan oleh urutan *bit* yang mempunyai suatu panjang yang nyata (*distinct length*). Jadi simbol yang muncul cukup banyak dalam *file* akan memberikan urutan yang pendek sementara simbol yang jarang dipakai akan mempunyai urutan *bit* yang lebih panjang. [4]

Adapun bentuk algoritma dari Huffman dalam membentuk sebuah pohon biner adalah sebagai berikut:

1. Dimulai dengan penyusunan frekuensi simbol sebagai frekuensi dari pohon
2. Jika terdapat lebih dari satu pohon:
 - a. Carilah dua pohon dengan jumlah *weight* yang paling kecil
 - b. Gabungkan dua pohon tersebut menjadi satu dan mempunyai nilai setara dengan jumlah

keduanya, atur salah satunya yang bernilai paling kecil sebagai *child* sisi kiri dan yang lainnya sebagai *child* sisi kanan

3. Lakukan langkah di atas hingga membentuk satu pohon biner tunggal
4. Untuk setiap *child* sisi kiri beri simbol '0' dan beri simbol '1' untuk merepresentasi *child* sisi kanan. [5]

2.6. Huffman Shift Coding

Pada *Huffman Shift Coding*, simbol dibagi menjadi beberapa blok dengan ukuran yang sama. Biasanya ukuran dari blok tersebut adalah 2^k-1 simbol, dimana k adalah bilangan bulat positif. Jika $k=1$, maka *huffman shift coding* sama dengan *standart huffman coding*.

Simbol dari blok pertama akan dikodekan persis menggunakan *huffman coding* standart. Ketika mengkodekan simbol dari blok pertama, ikut pula dikodekan simbol hipotesis yang frekuensi kemunculannya sama dengan jumlah frekuensi kemunculan simbol-simbol dari blok yang lainnya. Yang membedakan antara satu blok dengan blok selanjutnya adalah penambahan satu atau lebih kode *prefix* hasil pengkodean simbol hipotesis untuk menandai tiap blok.

Dengan cara *huffman shift coding*, bisa meningkatkan penggunaan waktu yang lebih sedikit dan rata-rata panjang kode yang lebih efisien. [6]

2.7. Algoritma Kompresi Huffman Shift Coding

1. *Source Symbol* disusun sehingga kemungkinan yang muncul dari yang terbesar sampai yang terkecil.
2. Jumlah total dari *source symbol* dibagi menjadi beberapa blok simbol dengan ukuran yang sama.
3. Simbol dari blok pertama akan dikodekan menggunakan *huffman coding standart*. Ketika mengkodekan simbol dari blok pertama, ikut pula dikodekan simbol hipotesis yang frekuensi kemunculannya sama dengan jumlah frekuensi kemunculan simbol-simbol dari blok yang lainnya. Yang membedakan antara satu blok dengan blok selanjutnya adalah penambahan satu atau lebih kode *prefix*

hasil pengkodean simbol hipotesis untuk menandai tiap blok.

4. Simbol hipotesis dari huffman code kita anggap C.
5. Kode simbol ke-i dari blok k adalah C^{k-1} digabungkan dengan simbol ke-i dari Huffman code blok pertama. [6]

3. ANALISA DAN PERANCANGAN SISTEM

3.1 Arsitektur Sistem

Pada program ini hanya akan dibuat kompresi dan dekompresi khusus hanya pada file audio berjenis Wave dan mempunyai audio format berjenis PCM (Pulse Code Modulation) dan hanya mendukung jumlah kanal maksimum 2 buah kanal (*mono* dan *stereo*). Untuk jenis Wave dengan *Multi Channel* tidak dapat dilakukan proses kompresi.

Bagian yang dikompresi dan didekompresi pada *fileWave* adalah bagian chunk data selain itu file output hasil kompresi akan diberi nilai "88" pada sub chunk audio format untuk membedakan file tersebut dengan file tidak terkompresi yang biasanya bernilai "1" pada bagian audio format-nya.

Langkah pertama sebelum *fileWave* yang dimasukkan ke dalam list, maka terlebih dahulu *fileWave* di-load dan dibuka. Setelah itu lakukan pembacaan pada header *fileWave* untuk 44 byte pertama. Selanjutnya lakukan pengambilan nilai 4 byte pertama lakukan pengecekan apakah merupakan string "RIFF", berikutnya adalah pengambilan dari byte ke-8 hingga byte ke-12 dan lakukan pengecekan apakah merupakan string "WAVE". Jika keduanya benar maka file tersebut merupakan *fileWave* dan langsung ditambahkan di bagian list, jika tidak lakukan loading file berikutnya.

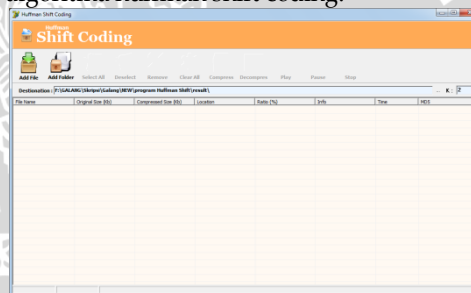
File dibaca untuk mengambil nilai Audio Format, seluruh header file dan chunk data yang merupakan data audio. Jika bernilai 1 berarti file belum dikompresi maka dapat dilakukan proses kompresi. Kompresi dilakukan hanya pada bagian chunk data dengan algoritma Huffman Shift Coding. Hasil kompresi berupa data yang dikompresi berikut pohon Huffman Shift Coding disimpan sekaligus akan ditulis ke file output. Setelah proses kompresi selesai informasi file yang diproses ditulis kembali ke file output berikut dengan pohon

Huffman Shift Coding dan data hasil kompresi.

Untuk proses dekompresi kembali seperti halnya dengan proses kompresi. Chunk data *fileWave* dan informasi header file diambil dan dicek nilai audio format apakah bernilai 88, jika ya maka dilakukan proses dekompresi yang merupakan kebalikan dari proses kompresi. Prosesnya dengan membaca dan membentuk pohon Huffman Shift Coding kembali dimulai dari byte ke-45 file Wave, setelah itu seluruh data hasil kompresi dikembalikan ke nilai semula berdasarkan pohon Huffman Shift Coding tersebut. Selanjutnya informasi pada file Wave ditulis pada file output dan hasil dekompresi ditulis kembali.

3.2 Implementasi Sistem

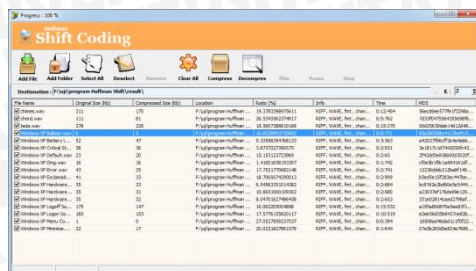
Berikut tampilan awal dari aplikasi kompresi file audio wave menggunakan algoritma huffman shift coding.



Gambar 3.1 Tampilan Awal Aplikasi

Langkah pertama yang dilakukan untuk mengkompresi atau mendekompresi sebuah atau beberapa file Wave adalah me-load file Wave tersebut terlebih dahulu ke dalam list. Ini dapat dilakukan dengan mengklik pada tombol "Add File" untuk memilih sebuah file tunggal dengan memunculkan sebuah kotak dialog untuk memilih satu buah file Wave. Sedangkan untuk memilih semua file Wave yang terdapat pada folder tertentu dapat dilakukan dengan mengklik "Add Folder".

Untuk melakukan kompresi file dapat dilakukan dengan menekan tombol "Compress", dan untuk melakukan proses dekompresi dengan menekan tombol "Decompress", sedangkan untuk memainkan file wave dalam list bisa dilakukan dengan menekan tombol "Play". Antarmuka proses kompresi ditunjukkan pada gambar 3.2.



Gambar 3.2 Tampilan Proses Kompresi

3.3 Uji Coba Sistem

Dalam pengujian, *file* yang digunakan adalah *file wave* pada sistem operasi Windows XP yang mempunyai ukuran *file* yang bervariasi. Penjelasan lebih lengkap mengenai *file audio* yang akan di uji dapat dilihat pada table 3.1.

Tabel 3.1 Daftar file audio wave uji coba

No	Nama File	Ukuran (kb)
1	Chimes.wav	211
2	Chord.wav	111
3	Tada.wav	278
4	Windows XP Balloon.wav	6
5	Windows XP Battery Low.wav	52
6	Windows XP Critical Stop.wav	38
7	Windows XP Default.wav	23
8	Windows XP Ding	16
9	Windows XP Error	43
10	Windows XP Exclamation	41
11	Windows XP Hardware Fail	35
12	Windows XP Hardware Insert	35
13	Windows XP Hardware Remove	35
14	Windows XP Logoff Sound.wav	175
15	Windows XP Logon Sound.wav	185
16	Windows XP Menu Command.wav	1
17	Windows XP Minimize.wav	22

Pada pengujian ini kasus yang diberikan adalah kompresi huffman shift coding dengan nilai $k=2$ dan $k=3$ dan yang diukur rasionya adalah ukuran dari chunk data file wave. Hasil pengujian ditunjukkan pada table 3.2. dan 3.3.

Tabel 3.2 Hasil Pengujian Kompresi dengan nilai $k=2$

Nama File	Ratio
Chimes	19,4%
Chord	26,5%
Tada	18,5%
Windows XP Balloon	16,8%
Windows XP Battery Low	9,3%
Windows XP Critical Stop	3,8%
Windows XP Default	15,1%
Windows XP Ding	1,4%
Windows XP Error	17,7%
Windows XP Exclamation	18,7%
Windows XP Hardware Fail	6,5%
Windows XP Hardware Insert	10,6%
Windows XP Hardware Remove	8%
Windows XP Logoff Sound	16%
Windows XP Logon Sound	17,5%
Windows XP Menu Command	27%
Windows XP Minimize	20%

Tabel 3.2 Hasil Pengujian Kompresi dengan nilai $k=3$

Nama File	Ratio
Chimes	7,7%
Chord	19,2%
Tada	9,1%
Windows XP Balloon	7,6%
Windows XP Battery Low	5,9%

Windows XP Critical Stop	2,6%
Windows XP Default	7,1%
Windows XP Ding	0,13%
Windows XP Error	9,6%
Windows XP Exclamation	8,7%
Windows XP Hardware Fail	4,3%
Windows XP Hardware Insert	2,8%
Windows XP Hardware Remove	6,5%
Windows XP Logoff Sound	7,8%
Windows XP Logon Sound	9,9%
Windows XP Menu Command	23%
Windows XP Minimize	16,3%

4. Kesimpulan

Berdasarkan sistem yang telah dibangun dan uji coba yang telah dilaksanakan dapat diperoleh kesimpulan sebagai berikut :

1. Telah berhasil mengimplementasikan aplikasi kompresi file audio wave menggunakan algoritma huffman shift coding.
2. Tingkat kompresi file audio wave pada pengujian menunjukkan nilai rata-rata 14,87% pada nilai $k=2$ dan 8,72% pada nilai $k=3$. Dengan rentang ratio 1,4% - 27% untuk nilai $k=2$ dan 0,13% - 23% untuk nilai $k=3$. Pada pengujian, file yang mempunyai ratio kompresi terbesar adalah file Windows XP Menu Command, dan yang mempunyai ratio kompresi terkecil adalah Windows XP Ding. Hal ini menunjukkan tingkat ratio kompresi tidak tergantung pada ukuran file, tetapi tergantung pada jumlah variasi simbol dan frekuensi kemunculannya.

5. Saran

Dalam penelitian ini masih banyak hal yang perlu dikembangkan untuk menjadikan sistem ini lebih baik, antara lain:

1. Menyempurnakan beberapa tampilan aplikasi agar lebih memiliki usability yang lebih baik sehingga user nyaman menggunakan aplikasi.
2. Untuk meningkatkan rasio kompresi maka algoritma kompresi Huffman Shift Coding dapat digabungkan dengan rasio kompresi yang lain seperti LZW.
3. Untuk pengembangan selanjutnya agar file Wave yang telah dikompresi dapat dimainkan lebih cepat maka dapat dilakukan dekompresi dengan teknik streaming dimana file tidak perlu didekompresi sampai utuh di memory tetapi bagian file didekompresi tersebut langsung dimainkan.
4. Mengimplementasikan algoritma huffman shift coding untuk file text, image, atau video

DAFTAR PUSTAKA

- [1] Nor Azizah. 2006. *Kompresi Format Audio File Wave Dengan Menggunakan Metode Lossless*. Fakultas Teknik dan Ilmu Komputer Unikom, Bandung.
- [2] Hananto, Indra. 2006. *Kompresi Video Menggunakan Discrete Cosine Transform*. Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, Institut Teknologi Bandung, Bandung.
- [3] Yenny. 2004. *Kompresi File Wave Dengan Algoritma Huffman*. STMIK Mikroskil, Medan.
- [4] Adhitama, Gagarin. 2009. *Perbandingan Algoritma Huffman dengan Algoritma Shannon-Fano*. Institut Teknologi Bandung, Bandung.
- [5] Sakti, Agus. 2010. *Implementasi Algoritma Huffman pada Kompresi dan Dekompresi File Bertipe WAV*. Fakultas MIPA, Universitas Pendidikan Bandung, Bandung.
- [6] Jalal Al Roumy. 2010. *Huffman Shift Coding*. The Islamic University of Gaza, Gaza.