

Penerapan Matematika Diskrit dalam Pemadatan File dengan Format WAV Menggunakan Algoritma Huffman

Syarifuddin Fakhri Al Husaini - 13518095
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518095@std.stei.itb.ac.id

Abstract—Semakin berkembangnya zaman, ukuran data yang kita simpan akan semakin besar. Oleh sebab itu teknologi kompresi sangat membantu dan menjadi aspek penting dalam perkembangan zaman ini. Dengan memanfaatkan kompresi maka file-file yang ada tidak terlalu besar sehingga dapat menghemat ruang dalam media penyimpanan kita, mempercepat dalam proses pengiriman atau penerimaan sebuah file, bahkan dapat dijadikan sebagai cara untuk mengamankan data. Terdapat dua teknik dalam kompresi yaitu *lossy data-compression* dan *lossless data-compression*. Salah satu metode yang sering digunakan dalam kompresi data adalah dengan algoritma Huffman. Teknik kompresi yang digunakan dalam algoritma Huffman adalah *lossless data-compression*. Penerapan dari kompresi ini adalah untuk memadatkan file suara dengan format WAV. File format WAV ini merupakan file audio standar yang digunakan pada komputer dengan sistem operasi Windows yang umumnya digunakan dalam *game* dan multimedia. Format WAV ini umumnya berukuran besar karena merupakan format kasar dalam bentuk data digital. Oleh sebab itu, kompresi data file dengan format WAV sangat diperlukan untuk menghemat ruang dalam media penyimpanan.

Keyword—kompresi, wafe, algoritma Huffman

I. PENDAHULUAN

1.1. Latar Belakang

Dalam kehidupan sehari-hari kita selalu berurusan dengan data baik itu secara langsung ataupun tidak langsung dan sadar ataupun tidak sadar. Salah satu data yang dapat terlihat adalah data dalam bentuk file yang terdapat di komputer. Bentuk file pun beragam dari file suara, file dokumen, file gambar, file video, dan lain-lain. File-file ini masih dapat dikategorikan menjadi beberapa format. Contohnya untuk file suara terdapat format MP3, AAC, dan WAV; untuk file dokumen terdapat format DOC, PDF, PPT, dan XLS; untuk file gambar terdapat format JPG, GIF, dan PNG; serta untuk file video terdapat format MP4, 3GP, dan AVI.

File yang cukup sering digunakan adalah file suara. Salah satu format file suara yang sering dipakai dalam sistem operasi Windows adalah WAV (waveform). Kegunaan format WAV adalah untuk keperluan *game* (terutama pada pembuatannya) dan keperluan multimedia (seperti dalam pengolahan atau mengedit file suara). Format WAV sebenarnya merupakan *raw format* atau sering disebut dengan format yang paling kasar sehingga ukuran filenya cukup besar.

Ukuran yang besar ini dapat dilihat dengan ukuran 1 menit file suara umumnya besarnya adalah 10 MB. Ukuran ini dengan perhitungan perekaman suatu suara atau audio dengan kualitas CD Audio menggunakan sampel dasar 44,1 kHz, 16 bit per sampel, dan 2 kanal (stereo). Dengan perhitungan tersebut didapat ukuran file suara per detik adalah 176.400 *byte*. Misalkan kita menyimpan sebuah lagu berdurasi 4 menit dengan format WAV, maka kita harus menyiapkan media penyimpanan sebesar 40 MB. Untuk sebuah file suara, ukuran tersebut sangat memboroskan media penyimpanan. Misal kita memiliki *flashdisk* ukuran 4 GB, kita hanya dapat menyimpan file suara dengan durasi 4 menit sebanyak 100 file.

Dengan ukuran sebesar itu, file suara dengan format WAV tidak disukai untuk menyimpan lagu yang diunduh dari internet dan didengarkan secara portabel. Permasalahan ini dapat diatasi dengan kompresi file suara. Dengan kompresi, format file WAV dapat ditekan dan diminimalkan ukurannya sehingga kita dapat menghemat ruang dalam media penyimpanan.

1.2. Tujuan dan Manfaat

Tujuan dari pembuatan makalah ini adalah sebagai berikut:

- Memenuhi tugas perkuliahan mata kuliah matematika diskrit teknik informatika semester 3 sebagai salah satu indikator dalam penilaian;
- Mengetahui cara kerja algoritma Huffman yang akan digunakan untuk kompresi dan dekompresi file suara dengan format WAV.

Manfaat dari pembuatan makalah ini adalah sebagai berikut:

- Meningkatkan kesadaran mahasiswa pentingnya membaca;
- Meningkatkan kemampuan mahasiswa teknik informatika semester 3 dalam hal menganalisis sebuah permasalahan;
- Meningkatkan kemampuan mahasiswa teknik informatika semester 3 dalam memecahkan permasalahan dengan menggunakan pengetahuan matematika diskrit yang telah diajarkan dalam perkuliahan;
- Mengasah kemampuan mahasiswa teknik informatika semester 3 dalam membuat sebuah karya ilmiah ataupun makalah dengan waktu yang telah ditentukan;
- Sebagai media penyebaran informasi hasil pemikiran dan penelitian. Makalah yang dibuat oleh mahasiswa teknik informatika semester 3 nantinya akan dimuat dalam *website* dosen sehingga siapa pun dapat membaca karya ilmiah yang dibuat oleh mahasiswa tersebut.

II. LANDASAN TEORI

2.1. Format File WAV

Dalam sebuah multimedia memerlukan berbagai format yang dapat diakses. Misalkan dalam satu aplikasi multimedia harus terdapat manajemen file video, file suara, *bitmap*, dan kontrol dari aplikasi itu sendiri. Dalam hal ini RIFF dapat berperan di dalamnya. RIFF atau *The Resource Interchange File Format*, sebuah label struktur file, adalah sebuah spesifikasi umum pada berbagai format yang dapat didefinisikan dengan keuntungan ekstensibilitasnya[1].

```

<WAVE-form> ->
RIFF( 'WAVE'
    <fmt-ck>           //Format
    [<fact-ck>]       //Fact chunk
    [<cue-ck>]        //Cue Point
    [<playlist-ck>]   //Playlist
    [<assoc-data-list>] //Associated data list
    <wave-data>      ) //Wave data
  
```

Gambar (2.1) WAVE Form

Dalam standar RIFF seperti di gambar (2.1) , format file WAV mengelompokkan segala isi file yang terdiri dari sampel suara digital, sampel format, dan lain sebagainya menjadi *chunk* yang terpisah. Setiap *chunk* memiliki *header* dan data *Byte*. *Header chunk* digunakan untuk menentukan jenisnya dan ukuran dari data *Byte* dari *chunk* tersebut. Sedangkan data *Byte* berisi informasinya. Dengan metode seperti ini, program hanya akan mengenali *chunk* sesuai dengan jenisnya sehingga dapat lebih efisien. Dalam suatu jenis *chunk*, data *Byte*-nya dapat diisi dengan sub-*chunk*. Seperti *chunk* “RIFF” pada gambar (2.2) mempunyai 2 sub-*chunk* yaitu *chunk* “fmt” dan *chunk* “data”.

Ukuran file RIFF pada *chunk* mempunyai ukuran kelipatan dari 2 *byte* (seperti 2, 4, 6, 8, 10, dan seterusnya) dikarenakan *chunk* merepresentasikan suatu string yang harus diatur tiap katanya[2]. Apabila tiap *chunk* memiliki jumlah *byte* ganjil maka perlu penambahan *byte* dengan cara menambahkan nilai nol pada *byte* terakhirnya. Maka dari itu dalam kalkulasi offset dari *chunk* berikutnya sebuah program harus selalu

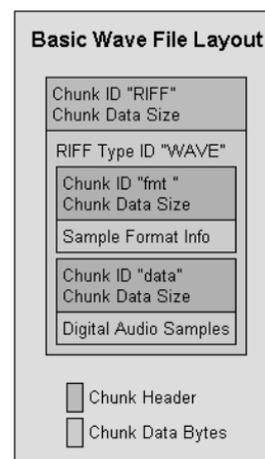
Tugas-tugas yang dapat dilakukan oleh RIFF seperti penukaran data multimedia lintas platform ataupun perekaman data multimedia[1]. File yang dapat dibaca RIFF beraneka ragam seperti:

- Bundle File Format
- Device-Independent Bitmap (DIB) and RIFF DIB file format
- Musical Instrument Digital Interface (MIDI) and RIFF MIDI file formats
- Palette File Format
- Rich Text File Format
- Waveform Audio File Format

Dalam pengimplementasian RIFF, struktur format file WAV sekarang banyak yang dimodifikasi oleh perusahaan lain sesuai keperluan mereka. Karena kesederhanaan dari file WAV itu sendiri, mereka mengembangkan spesifikasi yang dimiliki oleh format file WAV sesuai kebutuhan ataupun keinginan mereka. Yang benar-benar mengimplementasikan RIFF seutuhnya hanyalah format file AVI.

Format file WAV dirancang oleh Microsoft dan The International Business Machine Cooperation (IBM) untuk penyimpanan file suara bitstream di komputer. Pengaplikasiannya menggunakan RIFF yang jurnalnya rilis tahun 1991 dengan metode penyimpanan *chunks*. Format ini mulai populer seiring populernya sistem operasi Windows dan banyak dipakai dalam berbagai aplikasi. Penyimpanan format file WAV pun dalam urutan *little-endian* dan *big-endian* karena dalam perancangannya di Windows, umumnya menggunakan prosesor dari Intel[2].

melakukan pengaturan kata untuk menentukan ukuran nilai dari *header* sebuah *chunk*.



Gambar (2.2) Susunan File Wave

2.2. Kompresi Data

Kompresi data atau pemadatan data adalah suatu tindakan untuk mengurangi ukuran data atau file. Dengan pemadatan ini data atau file yang kita miliki akan semakin kecil. Akibatnya apabila data itu dikirim akan mempercepat transmisi data dan apabila file tersebut disimpan maka akan lebih menghemat ruang pada media penyimpanan kita.

Teknik untuk kompresi data dibedakan menjadi 2 yaitu *lossy data-compression* dan *lossless data-compression*[4]. *Lossy data-compression* adalah sebuah teknik dalam kompresi dan dekompresi dengan memperbolehkan terjadinya kesalahan selama tidak mengubah pola pokok atau inti dari data yang dikompres. Umumnya teknik *lossy data-compression* digunakan untuk gambar atau suara. Sedangkan *lossless data-compression* adalah sebuah teknik dalam kompresi dan dekompresi dengan persyaratan tidak boleh terjadi kesalahan dalam kompresi ataupun dekompresi. Teknik *lossless data-compression* digunakan untuk sebuah data atau file yang memerlukan keakuratan tinggi dalam proses pemadatan seperti file dokumen, file rekaman *database*, dan lain sebagainya.

Kompresi data awalnya berasal dari sebuah cabang dari ilmu matematika yaitu pada teori matematika dari komunikasi. Teori tersebut ditemukan pada tahun 1948 oleh C. E. Shannon di laboratorium Bell. Teori ini muncul karena dorongan bagaimana cara penyimpanan dan pengiriman sebuah informasi. Pada teori ini dibuktikan bahwa terdapat sebuah batas dasar pada sebuah kompresi dengan teknik yang dinamakan dengan *entropy*[3]. Pembuktian ini menggunakan teknik *lossless*. Nama *entropy* diambil dari dari sebuah istilah kimia yang berarti ketidakteraturan. *Entropy* sendiri disimbolkan dengan huruf H. Nilai dari *entropy* dapat dihasilkan menggunakan (1) dengan nilai P adalah probabilitas dari simbol. Pada (1) dapat diketahui semakin tinggi banyak simbol yang muncul pada sebuah informasi atau probabilitas kemunculan simbol tinggi akan menyebabkan nilai dari *entropy*-nya akan semakin besar juga. Untuk menghitung jumlah bit dari simbol dapat menggunakan (2). Sedangkan nilai *entropy* keseluruhannya dapat menggunakan (3) dengan P(i) adalah probabilitas dari simbol ke i dari 1 hingga n total simbol.

$$H = -\log P \quad (1)$$

$$N_{bit} = -\log_2 P \quad (2)$$

$$H_{tot} = \sum_{i=1}^n -\log P(i) \quad (3)$$

Dalam kompresi data ini, menggunakan bilangan yang tidak absolut yaitu tergantung cara memandang kita dalam memperoleh probabilitas sebuah simbol. Kita akan mengabaikan simbol yang telah muncul sebelumnya pada orde-0 sehingga kemunculan tiap simbol akan sama. Hal

ini digunakan untuk menghitung nilai *entropy zero-memory sources*. Pada perhitungan nilai *entropy zero-memory sources* akan menghasilkan S buah simbol dengan perhitungan pada persamaan (4) dengan P(Si) adalah probabilitas setiap simbol.

$$H(S) = \sum_S P(Si) \log_2 \frac{1}{P(Si)} \quad (4)$$

Pengembangan teori oleh shannon tidak hanya sampai di teknik *lossless*. Pada teknik *lossy* ia memperkenalkannya dengan *rate-distortion theory* yaitu fungsi R(D) dengan nilai D adalah jumlah distorsi data yang dapat ditoleransi. Karena D bersifat toleransi terhadap distorsi maka R(D) adalah kemungkinan terbaik dari sebuah kompresi. Nilai D yang mendekati 0 dengan nilai D selalu lebih besar sama dengan 0, maka semakin baik laju kompresinya. Dengan ini dapat dikatakan ketika menggunakan teknik *lossless*, maka mempunyai nilai D=0 dan R(0)=H. Teori *rate-distortion* adalah sebuah penyamarataan dari teknik kompresi *lossless* dengan nilai D=0 hingga D>0.

Terdapat banyak jenis algoritma untuk mengompres sebuah data. Beberapa algoritma yang sering digunakan dengan teknik *lossless data-compression* yaitu RLE, LZ77, LZ88, dan LZW. Sedangkan algoritma yang sering digunakan dengan teknik *lossy data-compression* yaitu Differential Modulation dan Adaptive Coding and Discrete Cosine Transform (DCT).

Contoh kompresi paling sederhana dan sering digunakan dalam kehidupan sehari-hari adalah sebuah singkatan[5]. Misal kata "Tuan" dapat disingkat menjadi "Tn" dan kalimat "Yang Terhormat" disingkat menjadi "Yth.". Dengan penyingkatan ini akan lebih cepat dalam pengetikan dan mengurangi jumlah memori yang digunakan sehingga lebih efisien dalam mengirim pesan.

2.3. Algoritma Huffman

Algoritma Huffman adalah sebuah algoritma yang digunakan untuk mengoptimalkan suatu tipe kode. Algoritma Huffman ditemukan oleh seorang mahasiswa MIT (Massachusetts Institute of Technology) yang bernama David A. Huffman pada tahun 1951. Pada waktu itu Huffman diberi tugas oleh profesornya untuk membuat makalah dengan topik pencarian kode biner paling efisien. Hasil karyanya ini diberi judul "A Method for Construction of Minimum Redundancy Codes".

Algoritma Huffman menggunakan algoritma pohon biner dalam prosesnya. Algoritma ini jika diimplementasikan dalam kompresi termasuk menggunakan teknik *lossless data-compression*. Oleh karena itu algoritma ini cukup terkenal dan sering digunakan dalam kompresi. Algoritma Huffman sendiri cocok untuk mengompres data yang berbentuk dokumen

teks atau file program. Hal ini dikarenakan algoritma Huffman termasuk algoritma keluarga dengan variabel *codeword length*. Cara kerjanya adalah mengganti bit yang panjang menjadi bit urutan dari nilai terkecil (yaitu 0 atau 1) hingga terbesar sesuai jumlah simbol dengan bit terkecil adalah simbol yang sering muncul dan bit terbesar adalah nilai yang jarang muncul.

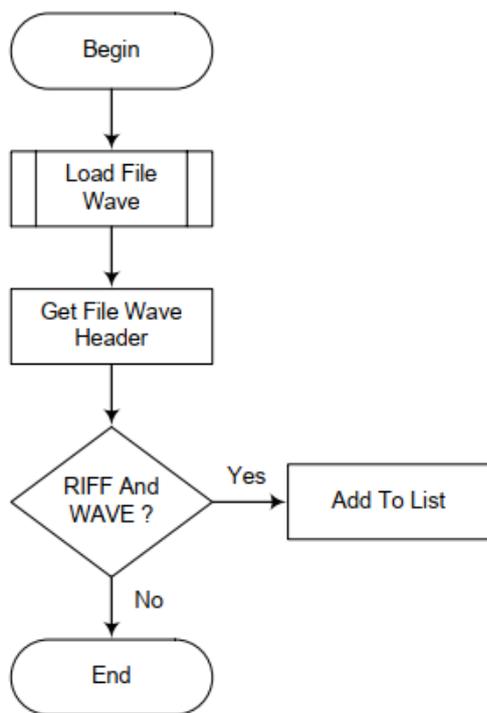
Walaupun algoritma Huffman ideal untuk mengompres data dokumen teks dan file program, algoritma ini jarang dipakai untuk mengompres data dalam bentuk lainnya. Misalkan file gambar atau video umumnya tidak menggunakan algoritma ini kecuali dalam format JPEG dan MPEG. Algoritma Huffman dipakai dalam program kompresi konvensional seperti PKZip, ZOO, dan ARJ.

III. PEMBAHASAN

3.1 Pembacaan Format File WAV

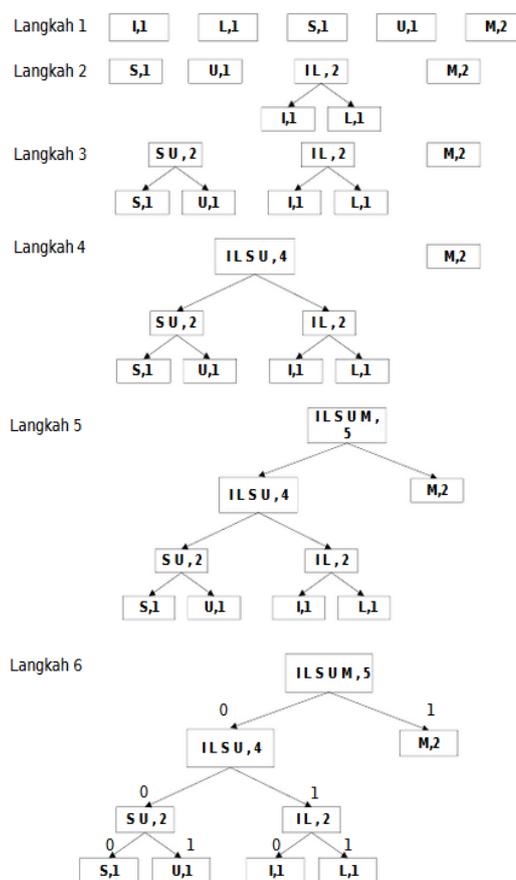
Pada program kali ini hanya dibuat kompresi dan dekompresi khusus hanya untuk audio berjenis file Wave dan mempunyai format audio PCM (*Pulse Code Modulation*) dengan maksimum kanal 2 buah (stereo dan mono). Untuk selain itu tidak dapat dilakukan proses kompresi ataupun dekompresi. File yang belum dikompres akan mempunyai audio format 1 (PCM) dan apabila sudah dikompres mempunyai audio format 88. Nilai ini berada di sub-*chunk* audio format.

Pembacaan file WAV pertama dengan memuat file ke program. Kemudian file WAV dibaca *header*-nya. Apabila *header* mempunyai format RIFF dengan tipe WAV, maka akan dimasukkan ke dalam list, apabila tidak, akan berakhir prosesnya. Untuk lebih jelasnya dapat dilihat pada gambar (3.1)



Gambar (3.1) Diagram Pembacaan File Wave

3.2 Kompresi File



Gambar (3.2) Pembentukan *Encoding* Huffman (Diambil dari:

https://www.researchgate.net/publication/317671197_IMPLEMENTASI_ALGORITMA_HUFFMAN_DAN_LZ78_UNTUK_KOMPRESI_DATA, diakses pada 5 Desember 2019)

Pembentukan algoritma Huffman menggunakan algoritma pohon biner adalah sebagai berikut (dimisalkan dalam bentuk teks):

- Membaca seluruh karakter di dalam teks
- Hitung frekuensi atau probabilitas kemunculan tiap simbol atau karakter, anggap setiap simbol adalah pohon dengan simpul tunggal
- Gabungkan 2 pohon yang mempunyai frekuensi atau probabilitas terkecil pada sebuah akar
- Berikan nilai frekuensi atau probabilitas dari akar tersebut dengan menjumlahkan 2 frekuensi atau probabilitas dari hasil penggabungan 2 pohon tadi
- Ulangi langkah c dan d hingga tersisa 1 buah pohon Huffman
- Berikan nilai 0 pada sisi cabang kiri dan nilai 1 pada sisi cabang kanan

Sebagai ilustrasi, diberikan contoh kata "MUSLIM". Jika diubah menjadi biner beserta frekuensi kemunculan tiap simbolnya akan menjadi sebagai berikut:

M= 01001101 (muncul 2 kali)

U = 01010101 (muncul 1 kali)
 S = 01010011 (muncul 1 kali)
 L = 01001100 (muncul 1 kali)
 I = 01001001 (muncul 1 kali)

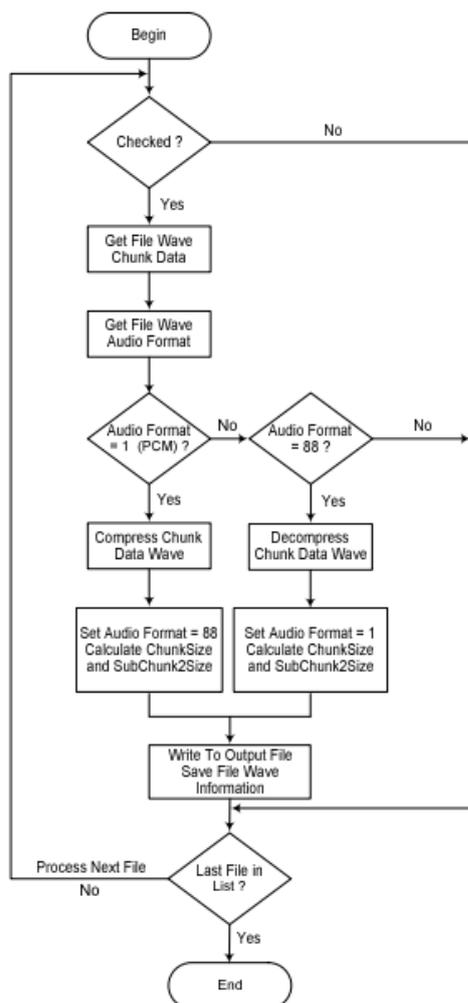
Kemudian lakukan *encoding* untuk Huffman pada teks tersebut. Ilustrasi terdapat pada gambar (3.2). Sehingga hasilnya adalah sebagai berikut

M = 1
 U = 001
 S = 000
 L = 011
 I = 010

3.3 Dekompresi File

Langkah dekomposisi file (dimisalkan dalam bentuk teks) adalah sebagai berikut:

- Mengeset node saat ini menjadi root node
- Membaca bit ke-i dari karakter ke-j
- Jika bit ke-i = 1 maka pindah ke node sebelah kanan. Apabila bit ke-i = 0 maka pindah ke node sebelah kiri
- Apabila bit ke-i adalah yang terakhir dari karakter j, maka cetak karakter yang terdapat pada node tersebut. Jika tidak $i = i + 1$ dan ulangi langkah b
- Lakukan hingga semua karakter telah di dekomposisi



Gambar (3.3) Diagram Proses Pembacaan File hingga berakhir

3.4 Hasil Kompresi

Hasil kompresi akan mempunyai ekstensi .cmp dengan struktur sederhana. *Header* info-info penting untuk dekomposisi sedangkan bagian data berisi data hasil kompresi file Wafe. Proses pembacaan file hingga kompresi dapat dilihat pada gambar (3.3).

Hasil dari kompresi file WAV direduksi berkisar antara range 20% hingga 40%. Jadi dapat dikatakan algoritma Huffman cukup baik dalam memadatkan file WAV. Waktu kompresi berbanding lurus dengan ukuran file yang dikompres. Untuk proses dekomposisi lebih cepat daripada kompresi karena tidak ada pembuatan pohon biner.

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Pada dasarnya, pengaplikasian matematika diskrit masih banyak lagi. Salah satunya adalah pengompresan file menggunakan algoritma Huffman. File WAV cukup baik dikompres menggunakan algoritma Huffman. Hasil reduksinya file WAV diantara 20% hingga 40%. Ini menandakan bahwa dengan teori sederhana dapat mengompres file dengan cukup baik.

4.2 Saran

Pembuatan makalah ini tidak terlepas dari kesalahan pembuatan. Apabila ada yang melanjutkan riset tentang optimasi kompresi data sangat kami dukung. Hal ini dikarenakan semakin majunya zaman, data yang akan diproses akan lebih besar ukurannya.

V. UCAPAN TERIMA KASIH

Puji syukur kepada Allah SWT. karena penulis dapat menyelesaikan tugas makalah matematika diskrit ini. Terima kasih kepada seluruh dosen pengampu matematika diskrit atas ilmu yang telah diberikan. Terima kasih juga kepada keluarga dan teman-teman yang selalu mensupport penulis dalam menjalani perkuliahan pada semester ini.

VI. REFERENSI

- [1] IBM Corporation and Microsoft Corporation, "Multimedia Programming Interface and Data Specifications 1.0", Agustus 1991
- [2] H Purwanto, "Penerapan Algoritma Huffman pada Kompresi File Wave", vol. 2, no. 1, 2015.
- [3] C. E. Shannon, "A Mathematical Theory of Communication", 1948
- [4] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=2ahUK-EwjM9dbqh5_mAhVFY8AKHXq1ArUQFjADegQI

[ARAC&url=http%3A%2F%2Fstaffnew.uny.ac.id%2Fupload%2F132048522%2Fpendidikan%2FKOMPRESI%2BDATA_Eko.pdf&usg=AOvVaw3m-lkqPBRQ3lu4kPzo1CAQ](http://www.staffnew.uny.ac.id/2Fupload/2F132048522%2Fpendidikan%2FKOMPRESI%2BDATA_Eko.pdf&usg=AOvVaw3m-lkqPBRQ3lu4kPzo1CAQ), diakses pada 5 Desember 2019

- [5] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=2ahUKEwjM9dbqh5_mAhVFY8AKHXq1ArUOFjAEegQIBxAC&url=http%3A%2F%2Feprints.undip.ac.id%2F18380%2F1%2FKompresi_Data_pert6.pdf&usg=AOvVaw1aV0XVSvmgzoX7on7ygGZa, diakses pada 5 Desember 2019

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2017



Syarifuddin Fakhri Al Husaini
13518095