Penerapan Algoritma Huffman dalam Kompresi Gambar Digital

David Theosaksomo
13515131
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*davidtsaksomo@students.itb.ac.id*

***Abstract*—Makalah ini berisi tentang gambaran umum mengeai kompresi data, penjelasan mengenai algoritma huffman, pengenalan bagaimana representasi gambar digital pada komputer, dan penggunaan algoritma huffman dalam melakukan kompresi terhadap gambar digital.**

***Keywords*—gambar, huffman, kompresi.**

I. PENDAHULUAN

Kompresi adalah metode untuk mengurangi jumlah data yang dibutuhkan untuk merepresentasi suatu konten berupa gambar, video, suara, ataupun teks tanpa mengurangi kualitasnya secara signifikan. Dalam dunia digital kompresi berarti mengurangi jumlah bit yang dibutuhkan untuk menyimpan suatu konten. Secara umum kompresi data dibagi menjadi dua yaitu lossless compression dan lossy compression. Lossless compression adalah metode kompresi yang menjaga keutuhan data sama seperti sebelum dikompres, tanda ada ada data yang dibuang. Sebaliknya, lossy compression adalah kompresi data yang mengakibatkan adanya data yang hilang setelah dikompresi. Data yang dibuang biasanya adalah data yang tidak berarti atau tidak signifikan. Contoh dari data ini misalnya adalah frekuensi yang sangat rendah pada suatu file musik yang tidak dapat didengar oleh manusia. Metode yang termasuk dalam lossless compression diantaranya adalah DCT (Discreet Cosine Transform), Vector Quantisation, Entropy coding. Sementara yang termasuk dalam lossy compression adalah RLE (Run Length Encoding), string-table compression, LZW (Lempel Ziff Welch).

Seiring dengan perkembangan sistem informasi penggunaan data multimedia terus bertambah. Termasuk di dalamnya adalah penggunaan gambar sebagai sumber informasi. Ukuran gambar cenderung jauh lebih besar bila dibandingkan dengan ukuran teks. Sehingga penyimpanannya membutuhkan kapasitas penyimpanan yang besar dan dibutuhkkan bandwidth yang banyak untuk mengirimkannya. Untuk mengatasi masalah tersebut digunakanlah sistem kompresi standar, semisal JPEG, GIF, dan PNG. JPEG adalah format yang terkenal dan banyak digunakan dimana-mana. Salah satu algoritma yang digunakan dalam kompresi gambar adalah algoritma huffman.

Algoritma Huffman adalah metode kompresi yang termasuk dalam lossless compression. Ini berarti tidak ada data yang hilang selama proses kompresi. Algoritma huffman dilakukan berdasarkan frekuensi kemunculan dari satuan data. Misal dari satuan data adalah karakter pada data text, atau pixel pada data gambar. Dibandingkan dengan menggunakan jumlah bits yang konstan untuk menyimpan data, Algoritma Huffman menggunakan jumlah bit yang lebih sedikit untuk data yang sering muncul, dansebaliknya menggunnakan jumlah bit yang lebih banyak untuk data yang jarang muncul. Hal ini mengakibatkan pengurangan dari jumlah bit rata-rata untuk merepresentasikan data. Algoritma huffman banyak digunakan dalam kompresi gambar digital, terutama dalam format JPEG.

II. KOMPRESI DATA

Terdapat dua tipe utama dalam kompresi data di berbagai jenis data, yaitu lossy data compression yang biasa digunakan untuk gambar dan lossless data compression yang biasa digunakan untuk teks dan file binary.

1. Lossy Data Compression

   Lossy Data Compression adalah metode kompresi dimana terdapat pembuangan data pada file yang telah dikompresi, sehingga tidak sama persis dengan file awalnya, tetapi tetap berfungsi sama. Data yang dikompresi tidak akan pernah bisa kembali persis seperti semula dimana data belum dikompresi. Bila data di-dekompresi maka hasilnya tidak akan sama persis dengan data awal. Akibatnya, lossy data compression tidak cocok untuk melakukan kompresi terhadap data yang bersifat kritikal, semisal data teks. Jenis kompresi ini banyak digunakan pada Digitally Sampled Analog Data (DSAD). DSAD terdiri dari file audio, video, dan gambar.

   Contoh dari pembuangan data, pada file suara misalnya, data yang menyimpan suara dengan frekuensi sangat rendah atau sangat tinggi yang tidak dapat didengar oleh manusia, dapat dibuang. Contoh sehari-hari dari lossy compression misalnya daam streaming media dari internet. Contoh dari format lossy

compression adalah JPEG, MPEG, MP3.

2. Lossless Compression
   Kebalikan dari lossy compression, lossless data compression adalah metode kompresi dimana terdapat tidak ada pembuangan data pada file yang telah dikompresi, sehingga apabila data didekompresi akan menghasilkan data yang persis dengan data awal. Kompresi data dapat dilakukan berulang-ulang dengan keutuhan data tetap terjaga. Lossless data digunakan apabila penting untuk data hasil kompresi sama persis dengan data awal. Format ZIP yang terkenal menggunakan metode loseless compression dalam kompresi datanya [6].

III. ALGORITMA HUFFMAN

Algoritma huffman adalah algoritma enkoding entropi yang digunakan di komputer sains dan teori informasi. [1]. Algoritma huffman didasarkan pada frekuensi kemunculan dari data. Prinsipnya adalah dengan menggunakan jumlah bit yang sedikit untuk menyimpan data yang sering muncul. Panjang rata-rata dari kode huffan bergantung pada statistik frekuensi kemunculan data[2].

Algoritma Huffman dipakai secara luas dimana-mana. Banyak format file yang populer yang menggunakan algoritma Huffman, semisal GZIP, PKZIP, BZIP2, dan pada gambar JPEG dan PNG. Algoritma Huffman sangat efektif dalam melakukan kompresi terhadap data yang banyak memiliki pengulangan, dimana data tersebut banyak dijumpai, semisal teks dalam bahasa inggris, dan gambar. Algoritma Huffman banyak menjadi back-end dari metode kompresi lainnya. Misalnya Brotli Compression, salah satu algoritma kompresi yang dikeluarkan oleh google, menggunakan algoritma Huffman. Huffman banyak dipakai karena soal masalah paten, algoritma Huffman dapat bebas dipakai.

Algoritma Huffman menggunakan sebuah pohon biner dalam proses kompresi yang dinamakan pohon Huffman, yaitu pohon yang menyimpan enkoding dari item-item data dalam bits.
Pohon huffman ini dibuat pada saat data dikompresi dan digunakan untuk mendekompresi data.

Dalam algoritma huffman pertama-tama hitung frekuensi kemunculan dari setiap data. Lalu berdasarkan data itu kita buat pohon Huffman. Algoritma untuk membentuk pohon huffman adalah sebagai berikut:
1. Pilih dua data dengan frekuensi kemunculan paling sedikit. Kemudian dua data tersebut dijadikan sebagai anak dari simpul baru, yaitu gabungan dari kedua data dengan frekuensi kemunculannya merupakan akumulasi dari frekuensi kemunculan kedua data yang kita pilih.
2. Pilih dua data berikutnya, termasuk simpul yang baru kita buat, yang memiliki peluang terkecil.
3. Selanjutnya kita ulangi langkah 2 hingga seluruh data sudah terpilih. Pohon yang terbentuk adalah pohon Huffman kita.

Dari pohon Huffman kita mendapatkan kode huffman untuk simbol kita. Caranya adalah dengan memberi nomor pada sisi pohon dengan angka 0 dan 1. Pemberian nomor ini dibebaskan, namun untuk memudahkan pemberian nomor orang biasa untuk menggunakan standar tertentu. Semisal dari standar ini adalah sisi pohon kiri diberi nomor 0, sedangkan sisi pohon kanan diberi nomor 1. Untuk mendapatkan kode untuk simbol kita, kita perlu menelusuri jalan dari bagian paling atas pohon kita hingga ke daun dimana simbol kita berada. Nomor yang kita lalui sepanjang jalan adalah kode Hufman untuk simbol kita.

Kita dapat menghitung rasio kompresi yang didapatkan dengan formula:

$$Rasio\ Kompresi = \frac{Jumlah\ bit\ setelah\ kompresi}{Jumlah\ bit\ sebelum\ kompresi} \times 100\%$$

Menghitung jumlah bit setelah kompresi dilakukan dengan cara mengalikan jumlah bit yang dipakai pada simbol dengan jumlah kemunculan simbol untuk setiap simbol, dan menjumlahkan seluruhnya.

$$Jumlah\ bit\ setelah\ kompresi = \sum_{i=1}^{m} n_i \times f_i$$

Dimana: m adalah jumlah simbol unik, n_i adalah jumlah bit untuk kode simbol ke i, dan f_i adalah jumlah kemunculan simbol ke i.

Sedangkan untuk menghitung jumlah bit sebelum kompresi dapat berbeda-beda tergantung format datanya. Misalkan untuk text ASCII yang menggunakan 8 bit untuk merepresentasikan satu karakter, maka jumlah bit yang digunakan adalah 8 x Jumlah karakter. Untuk gambar bitmap maka jumlah bitnya adalah resolusi gambar tersebut dikalikan dengan kedalaman bit (bit depth), yaitu jumlah bit yang diperlukan untuk merepresentasi satu pixel pada gambar.

Hal yang perlu diketahui adalah apabila semua jenis karakter muncul pada sebuah teks, dan setiap karakter tersebut memiliki frekuensi yang sama, maka algoritma Huffman akan menghasilkan rasio kompresi 100%, atau tidak dikompresi sama sekali. Ini menunjukkan bahwa algoritma Huffman sangat bergantung pada statistik kemunculan data. Namun di dunia nyata sangat jarang suatu data memiliki frekuensi kemunculan yang sama untuk setiap simbol. Misalnya pada teks bahasa inggris, suatu kata tertentu akan lebih sering muncul dibandingkan dengan yang lain, dan mungkin huruf ‘a’ lebih sering muncul dibandingkan huruf ‘z’.

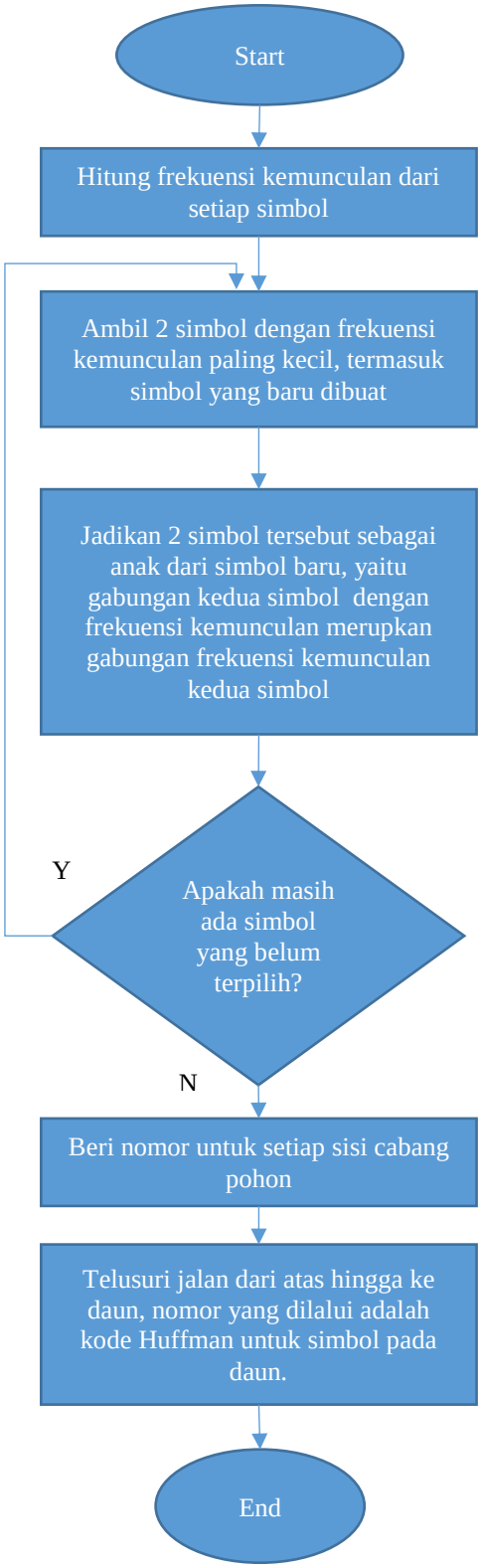


Figure 1: Flowchart penentuan kode Huffman

Pada data gambar simbol yang muncul cenderung lebih beragam, tapi tetap ada kecenderungan pada suatu simbol untuk muncul lebih sering daripada simbol lainnya, tergantung dari apa yang direpresentasikan oleh gambar itu sendiri.

Untuk lebih memahami, mari kita lakukan algoritma Huffman. Misalkan kita memiliki data teks “makan sayur sehat” (termasuk karakter spasi) dan kita ingin mengkompresi teks tersebut dengan algoritma Huffman. Pertama-tama yang kita lakukan adalah menghitung frekuensi kemunculan setiap karakter. Hasil perhitungan dapat dilihat pada tabel 1.

Karakter	Frekuensi Kemunculan
m	1
a	4
k	1
n	1
spasi	2
s	2
y	1
u	1
r	1
e	1
h	1
t	1

Table 1. Frekuensi kemunculan karakter dari teks 'makan sayur sehat'

Dengan menjalani algoritma pembentukan pohon Huffman, kita mendapatkan pohon Huffman seperti pada gambar 2.
Dan dengan memberi nomor pada tiap cabang dengan 0 untuk cabang kiri dan 1 untuk cabang kanan dan menelusuri semua simbol, kita dapatkan kode huffman untuk karakter kita sesuai dengan tabel 2.

Karakter	Kode Huffman	Jumlah Bit
m	11010	5
a	10	2
k	11011	5
n	1111	4
spasi	000	3
s	011	3
y	0011	4
u	0100	4
r	0101	4
e	1110	4
h	1111	4
t	1100	4

Kita memperoleh rata-rata bit yang digunakan untuk satu karakter adalah 3,833 bit. Dan jika kita menghitung rasio kompresi maka rasio kompresi yang didapatkan adalah 47.9125 %.

kombinasi warna atau 16.777.216 warna yang berbeda. Komputer sistem sekarang juga dapat mensupport 32 bit warna, atau 4 byte, dimana 3 byte dialokasikan untuk 3

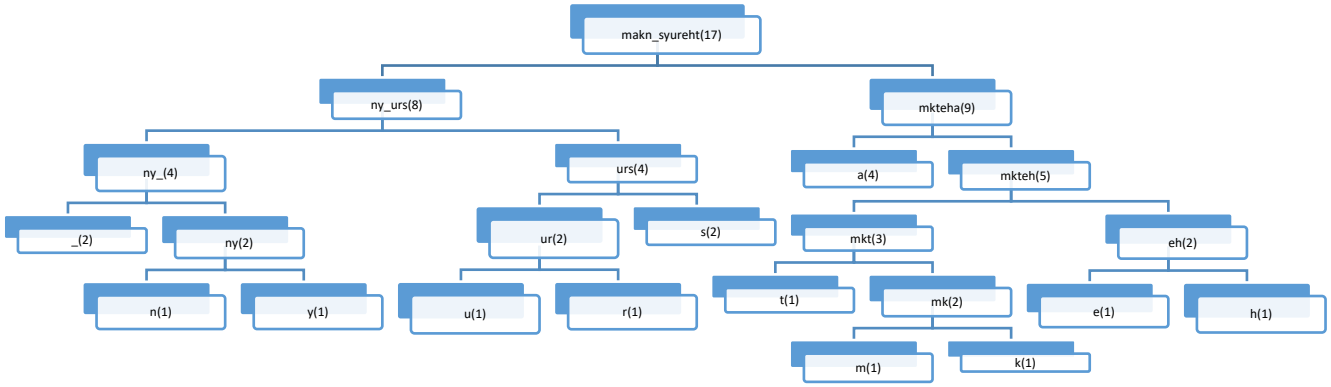


Figure 2 Pohon Huffman untuk teks 'makan sayur sehat'

IV. GAMBAR DIGITAL

Dalam dunia komputer semua data direpresentasikan dalam biner, termasuk gambar. Data gambar ditampilkan pada layar komputer atau alat digital lainnya sebagai bitmap. Bitmap terdiri dari deretan pixel yang membentuk gambar. Setiap gambar mmempunyai resolusi yang diukur dengan jumlah pixel per baris, atau lebar gambar, dan jumlah pixel per kolom, atau tinggi gambar. misalnya gambar dengan resolusi 1024 x 800 pixel berarti dia memiliki 1024 pixel di setiap baris dan 800 piksel di setiap kolom. Seberapa besar gambar muncul di layar, yang diukur dalam satuan panjang semisal cm, bergantung pada dpi gambar. Dpi adalah singkatan dari dots per inch yang dimiliki layar, yaitu jumlah piksel dalam satu inchi layar. Pixel berukuran sangat kecil sehingga sulit dilihat oleh mata telanjang [3].

Jumlah dari bits yang digunakan untuk menyimpan informasi satu pixel sudah bertambah dikarenakan sistem software komputer yang semakin canggih. Penyimpanan 24 bit sudah umum sekarang, dibandingkan 10 tahun lalu dimana kebanyakan masih sebatas 16 bit. 16 bit berarti ada 2^{16} warna yang bisa ditampilkan, atau 65,536 warna [3].

Penambahan ukuran RAM dan kecepatan CPU juga memiliki dampak positif dalam penampilan gambar. Gambar dengan ukuran besar dapat ditampilkan dengan cepat. Serta Penambahan storage dari komputer memungkinkan untuk menyimpan gambar beresolusi besar tanpa harus me-resize nya menjadi ukura yang lebih kecil [3].

Sistem warna yang banyak digunakan dalam gambar digital adalah RGB system. RGB adalah singkatan dari Red, green, dan blue. Warna dari pixel adalah campuran dari ketiga warna dasar tersebut. 8 bit dapat merepresentasi kekuatan yang berbeda dari setiap warna tersebut, yaitu 256 warna, dengan tingkat kekuatan dari 0-255. Untuk ketiga warna maka kita mempunyai 256x256x256 warna, dan 1 byte untuk transparasi (alpha) sehingga dimungkinkan untuk menampilkan gambar transparan [3].

Jumlah bit per pixel yang digunakan untuk menyimpan data gambar bisa bervariasi. Angka ini disebut bit depth atau kedalaman bit. Semakin besar bit depth maka semakin bagus kualitas gambar karena lebih banyak warna yang bisa ditampilkan. Jumlah warna yang bisa ditampilkan adalah 2^n dimana n adalah bit depth. Misalkan apabila bit depth suatu gambar adalah 1 maka gambar hanya bisa menampilkan 2 warna saja.

Ukuran dari data gambar diukur dalam satuan kilobytes atau megabytes. Untuk menghitung ukuran gambar kita dapat meggunakan formula:

$$Image\ Size = \frac{image\ width \times image\ height \times bitdepth}{8}\ bytes$$

Misalkan suatu gambar memiliki resolusi 1024 x 800 pixel dan memiliki bit depth 24 bit. Maka ukuran gambar adalah (1024 x 800 x 24)/ 8 = 19.660.800 bytes atau 18,75 megabytes.

Gambar dapat diresize, baik itu menambah jumlah pixel atau mengurangi jumlah pixel. Tujuannya adalah menghasilkan gambar yang sama, namun dengan ukuran yang lebih besar atau lebih kecil [3].

Beberapa format yang populer dalam penyimpanan data antara lain:

Bitmap

Bitmap memiliki ekstensi bmp. Gambar data disimpan dalam format yang tidak terkompresi.

JPEG

JPEG adalah singkatan dari Joint Photographic Expert group. JPEG adalah format 24 bit yang khusus untuk menyimpan gambar. Dengan 24 bit format maka JPEG dapat menampilkan jutaan warna. JPEG termasuk ke dalam lossy compression format, dimana ada data yang hilang saat dilakukan kompresi.

GIF

Gif adalah format dengan lossless data compression yang menggunakan 256 palet warna. Sehingga ini cocok untuk gambar yang tidak memiliki gradasi warna rumit seperti fotografi.

PNG

PNG atau portable network graphics adalah format yang digunakan pada Adobe Fireworks program. PNG memiliki kesamaan dengan GIF format karena mendukung transparansi dan adalah lossless data compression. Namun PNG dapat menyimpan lebih banyak warna, dengan versi 24 bitnya dapat menyimpan 16 juta warna [3].

V. HUFFMAN CODING DALAM KOMPRESI GAMBAR DIGITAL

Algoritma Huffman dapat digunakan untuk melakukan kompresi pada hampir semua jenis data. Gambar digital adalah salah satu informasi yang dapat dikompresi dengan Algoritma Huffman. Format JPEG misalnya yang populer dan dipakai dimana-mana menggunakan algoritma huffman sebagai salah satu algoritma kompresinya. Kompresi gambar dengan algoritma huffman mirip dengan kompresi teks, hanya saja kita mengganti karakter pada teks dengan warna piksel pada gambar.

Misalnya kita memiliki gambar digital sebagai berikut.

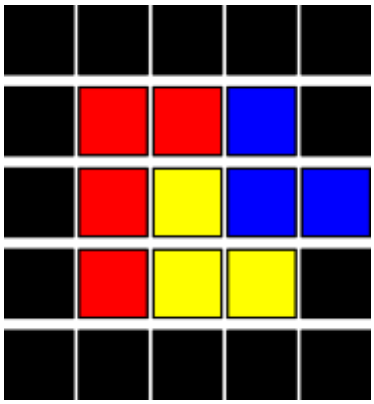
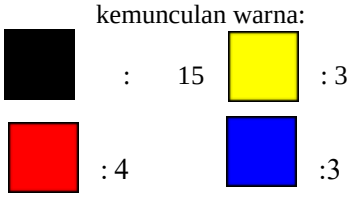


Figure 3 Gambar 5x5

Gambar memilki ukuran 5x5. Misalkan gambar tersebut memiliki kedalaman bit 8 bit. Sehingga ukuran file adalah 5 x 5 x8 /8 = 25 bytes.

Mari kita lakukan algoritma Huffman untuk mengkompresi gambar tersebut. Pertama kita buat tabel kemunculan warna:



Lalu kita buat pohon Huffman nya, yaitu gambar 4.

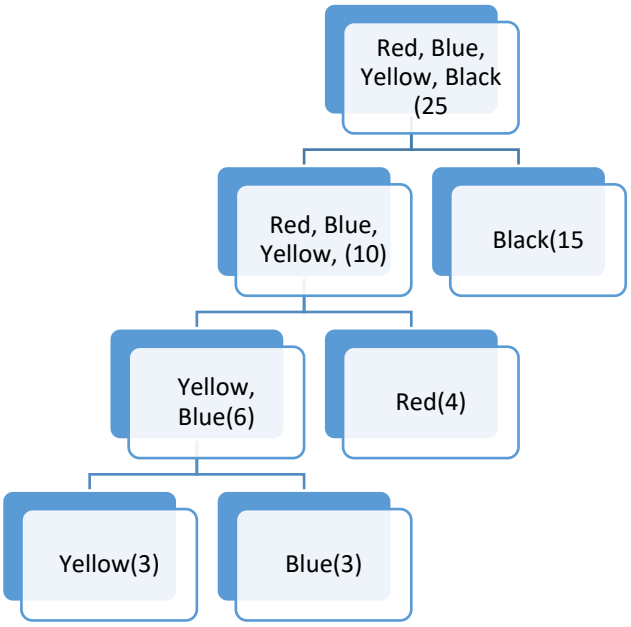


Figure 4 Pohon Huffman untuk gambar 3

Dan dengan memberi nomor pada tiap cabang dengan 0 untuk cabang kiri dan 1 untuk cabang kanan dan menelusuri semua warna, kita dapatkan kode huffman untuk warna pada pixel sesuai dengan tabel 3.

Warna	Kode Huffman	Jumlah bit
Merah	01	2
Hitam	1	1
Kuning	010	3
Biru	011	3

Kita memperoleh rata-rata bit yang digunakan untuk satu warna adalah 2.25 bit. Mari kita hitung jumlah bit yang dibutuhkan untuk menyimpan informasi gambar 3.
Jumlah bit = 2 x 4 + 1 x 15 + 3 x 3 + 3 x 3 = 41 bit.
Sebelum dikompresi, jumlah bit yang dibutuhkan adalah 25 x 8 = 200 bit sehingga rasio kompresi = 41/200 = 20.5%.

V. KESIMPULAN

Algoritma Huffman menggunakan aplikasi pohon prefix biner dalam menentukan kode Huffman. Algoritma Huffman banyak digunakan dalam kompresi data, karena tingkat kompresi data yang tinggi, bergantung pada data yang dikompresi. Algoritma Huffman juga tidak memiliki paten khusus sehingga dapat digunakan secara luas. Terdapat juga algoritma kompresi terkini, semisal Brotli Compression yang menggunakan algoritma Huffman sebagai salah satu algoritma kompresinya. Algoritma Huffman juga dapat digunakan sebagai kompresi gambar digital, dan digunakan sebagai salah satu algoritma kompresi pada format JPEG.

VI. ACKNOWLEDGMENT

Terima kasih kepada Pak Rinaldi Munir, yang karena beliau makalah ini dapat dibuat.

REFERENSI

[1] Mamta Sharma, "Compression Using Huffman Coding". 2010. http://paper.ijcsns.org/07_book/201005/20100520.pdf, 9 Desember 2016.

[2] Asadollah Shahbahrami, Ramin Bahrampour, MobinS abbaghi Rostami, Mostafa Ayoubi Mobarhan, "Evaluation of Huffman and Arithmetic Algorithms for Multimedia Compression Standards" http://airccse.org/journal/ijcsea/papers/0811ijcsea04.pdf, 9 Desember 2016.

[3] petert@nayland.school.dot.nz. 2012. Image Representation – Computer Science For Digital Technologies. https://sites.google.com/site/dtcsinformation/data-representation/image-representation, 9 Desember 2016.

[4] Mitsuharu Shibata, "Huffman Coding – Base of JPEG Image Compression," http://www.print-driver.com/stories/huffman-coding-jpeg, 9 Desember 2016.

[5] Guy E. Blelloch, "Introduction to Data Compression." 2013, *https://www.cs.cmu.edu/~guyb/realworld/compression.pdf*

[6] ijarcsse, "A Review on data compression techniques," https://www.ijarcsse.com/docs/papers/Volume_5/1_January2015/V5I1-0377.pdf, 9 Desember 2016.

[7] Rinaldi Munir, "Diktat Kulih IF2153 Matematika Diskrit," Informatika bandung: Bandug, 2007.

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, 8 Desember 2016

David Theosaksomo 13515131