

# Kompresi Citra Digital Menggunakan Kode Huffman

Fatkhan Masruri 13518053<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13518053@std.stei.itb.ac.id

**Abstract**—Masyarakat pada saat ini cenderung sering menggunakan *smartphonenya* untuk melakukan suwa foto secara tidak langsung hal itu meningkatkan kebutuhan penyimpanan. Oleh karena itu, dibutuhkan sebuah metode untuk menangani hal tersebut, salah satu caranya yaitu dengan melakukan kompresi citra digital. Kompresi citra digital merupakan salah satu metode dalam pengolahan citra yang berfungsi untuk mengurangi ukuran data citra masukan dengan cara mengurangi informasi atau mempertahankannya. Salah satu metode yang dapat digunakan adalah kode huffman.

**Keywords**—Citra Digital, Kode Huffman, Kompresi, Dekompresi.

## 1. PENDAHULUAN

Pada era digital banyak orang yang menggunakan media sosial, penggunaan media sosial secara langsung meningkatkan pertukaran data antar perangkat. Sehingga pertukaran data pada saat ini merupakan salah satu kebutuhan yang sangat esensial. Pada tahun 2020 diperkirakan *global ip traffic* akan meningkat mencapai 230.000 *petabytes*.

Peningkatan trafik digital berdampak pada aliran data multimedia, salah satunya adalah citra digital. Citra digital merupakan gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan atau diskrit nilai digital yang disebut *pixel*. Ukuran data citra digital sangat sensitif terhadap kualitas datanya. Semakin tinggi kualitas datanya maka semakin tinggi juga ukuran datanya. Oleh karena itu perlu dilakukan kompresi citra digital.

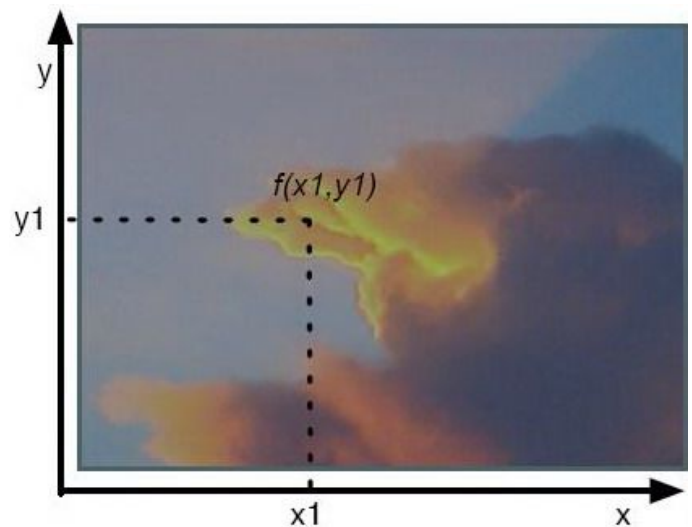
Kompresi citra digital bertujuan untuk mereduksi data tanpa menghilangkan informasi yang terkandung. Dengan dilakukan kompresi maka ukuran data menjadi lebih kecil sehingga semakin kecil juga *bandwidth* yang dibutuhkan ketika ditransmisikan di dalam jaringan. Semakin kecil *bandwidth* semakin cepat juga di dalam proses *upload*.

Kompresi digital secara umum dibagi menjadi dua metode yaitu *lossy compression* dan *lossless compression*. *Lossless compression* mampu melakukan reduksi ukuran data tanpa menghilangkan informasi yang terkandung dalam data asli. Metode ini memungkinkan pengguna mendapatkan data sama persis dengan aslinya.

Pada makalah ini akan melakukan kompresi menggunakan metode kode huffman. Kode huffman termasuk teknik

kompresi losses sehingga kemungkinan data yang dilakukan kompresi akan menampilkan informasi yang sama. Cara kerja dari metode kode huffman adalah dengan melakukan pengkodean dalam bentuk bit untuk mewakili data karakter.

## 2. CITRA DIGITAL



Gambar 1.0 Citra Digital  
(sumber: elektronika-dasar.web.id)

### 2.1. Pengertian Citra Digital dan Pengolahannya

Citra merupakan sebagai sebuah keluaran dari sistem perekaman data yang dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Citra digital merupakan suatu larik dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar. Indek baris dan kolom pada matriks menyatakan suatu titik pada citra tersebut dan elemen matriksnya yang disebut elemen gambar/piksel/ pels. Sehingga informasi-informasi yang tersimpan bersifat diskrit. Matriks yang dinyatakan citra digital yaitu matriks berukuran  $N \times M$  seperti gambar berikut.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Citra digital tidak selalu dihasilkan dari sebuah data rekaman suatu sistem, tetapi juga bisa dihasilkan dari rekaman data bersifat kontinu. Dengan demikian untuk mendapatkan suatu citra digital diperlukan suatu proses konversi, sehingga citra tersebut dapat diproses oleh komputer.

Pengolahan pada citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Pada proses ini secara umum didefinisikan dengan pemrosesan citra dua dimensi menggunakan komputer. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili bit-bit tertentu.

## 2.2. Noise Citra Digital

Kualitas suatu citra sangat dipengaruhi oleh tingkat deraunya. Kualitas yang didapatkan pada suatu citra sangat dipengaruhi oleh alat penginderaan. Kualitas yang kurang baik dari suatu citra dapat dikarenakan oleh derau sensor, kamera yang kurang fokus, dan hal hal lain.

## 2.3. Proses Filter Pada Citra Digital

Seringkali ketika mendapatkan citra mengandung derau sehingga sebelum melakukan analisis harus dihaluskan terlebih dahulu menggunakan tapis citra. Hal ini dapat dilakukan dengan melakukan manipulasi pada pixel tetangga. Desain tapis seperti mampu membuat citra lebih halus dan bentuk sudut dalam pada citra tetap terjaga. Ada beberapa model tapis yang dapat digunakan untuk memperhalus citra. Derau aditif atau Gaussian biasanya dilakukan tapis menggunakan tapis Wiener. Derau Multipikatif atau Speckle biasanya dihilangkan dengan tapis Homomorfik. Sedangkan pada derau salt-and-pepper biasanya digunakan tapis median atau LPF.

## 2.4. Segmentasi Dan Pengembangan Citra Digital

Banyak cara yang dapat dilakukan untuk melakukan pemilihan pada citra, salah satu cara yang dapat digunakan adalah dengan segmentasi. Segmentasi yaitu membagi citra menjadi bagian-bagian yang diharapkan termasuk objek-objek yang dianalisis. Segmentasi sering dideskripsikan sebagai proses analogi terhadap proses pemisah latar depan dengan latar belakang.

Pemilihan bentuk citra sangat berguna dalam melakukan pengukuran atau pemahaman citra. Secara sederhana, pengembangan dapat didefinisikan sebagai proses pendefinisian jangkauan nilai-nilai gelap-terang pada citra yang sebenarnya. Memilih piksel-piksel dalam jangkauan ini dijadikan sebagai latar depan dan sisanya dijadikan sebagai

latar belakang. Dengan demikian citra terbagi menjadi dua bagian yaitu bagian hitam atau putih atau warna-warna yang membatasi setiap wilayah.

## 3. KOMPRESI CITRA

### 3.1 Pengertian Kompresi Citra

Kompresi citra merupakan aplikasi kompresi data yang dilakukan terhadap citra digital dengan tujuan untuk mengurangi redundansi dari data-data yang terdapat di dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien. Pada citra hitam-putih hanya terdapat dua warna sehingga pada pikselnya hanya memuat satu bit yaitu nol atau satu. Sedangkan pada citra grayscale atau warna dengan gradasi warna abu-abu sebanyak 256 warna, pada tiap piksel citra ini memuat informasi sebanyak 8 bit. Pada citra berwarna yang berupa gradasi warna mulai dari 256 warna sampai 16 juta warna. Tiap piksel pada citra ini bisa menyimpan informasi warna mulai dari 8-bit sampai dengan 24-bit.

### 3.2 Teknik Kompresi

Teknik untuk melakukan kompresi citra dengan kompresi lainnya sama yaitu menggunakan teknik *Lossy Compression*. Teknik ini akan melakukan kompres pada citra dengan menghilangkan beberapa informasi dalam citra asli. *Lossy Compression* akan mengubah detail dan warna pada file citra menjadi lebih sederhana tanpa terlihat perbedaan yang mencolok dalam pandangan manusia, sehingga ukurannya menjadi lebih kecil. Teknik dapat dilakukan dengan berbagai cara yaitu salah satunya *color reduction* dan *chroma subsampling*. *Color reduction* akan menyimpan warna - warna mayoritas menjadi *color palette*. Sedangkan *chroma subsampling* adalah teknik yang memanfaatkan fakta bahwa mata manusia merasa *brightness* lebih berpengaruh daripada warnan (chrominance) itu sendiri, maka perlu dilakukan pengurangan resolusi warna dengan sampling ulang. Hal ini biasanya digunakan pada sinyal YUV (Y: *luminance*, U: *CBlue*, V: *CRed*).

### 3.3. Hal Penting Dalam Kompresi Citra

*Scalability* adalah kualitas dari proses pengkompresian citra karena manipulasi bitstream tanpa adanya dekompresi atau rekompresi. Pada kasus ini biasanya terjadi ketika *preview* gambar sementara diunduh. Ketika semakin baik *scalability*nya semakin bagus juga *preview* gambarnya. Tipe dari *scalability* ini ada banyak yaitu:

1. *Quality progressive* : Tipe ini, gambar akan dikompres secara perlahan-lahan dengan penurunan kualitasnya.
2. *Resolution progressive* : Tipe ini, gambar akan dikompresi dengan mengkode resolusi gambar yang rendah terlebih dahulu baru kemudian ke resolusi yang lebih tinggi.

3. *Component progressive* : Gambar akan dikompresi berdasarkan komponennya, pertama mengkode komponen abu-abu baru kemudian dilakukan yang berwarna.

Selain *scalability* hal penting yang perlu diperhatikan dalam melakukan kompresi citra adalah *Region of interest coding* dan informasi meta. *Region of interest coding* adalah daerah-daerah tertentu denkode dengan kualitas yang lebih tinggi daripada yang lain. Informasi meta adalah gambar yang dikompres juga dapat memiliki informasi meta seperti warna, tekstur, *small preview image*, dan *author* atau *copyright information*.

## 4. KODE HUFFMAN

### 4.1 Pengertian Kode Huffman

Kode Huffman merupakan algoritma kompresi citra yang menggunakan pendekatan statistik dengan cara melakukan pengkodean dalam bentuk bit untuk mewakili karakter. Dalam pemrosesannya kode huffman menggunakan struktur pohon.

Algoritma huffman

### 4.2 Sejarah Kode Huffman

Kode Huffman dikembangkan oleh David A. Huffman. Kode huffman ditemukan ketika penemunya sedang menempuh pendidikan doktoral di bidang sains *Massachusetts Institute of Technology*. Beliau mendapatkan metode ini ketika mendapat tugas membuat *papper* untuk menemukan kode biner yang paling efisien. Kemudian ia mendapat ide menggunakan *frequency-sorted binary tree*. Dengan metode tersebut ia mampu menemukan metode yang paling efisien.

Pada tahun 1952, Ia menerbitkan sebuah jurnal yang berjudul "A Method for the Construction of Minimum-Redundancy". Pada buku ini, Ia menjelaskan tentang metode yang telah ditemukannya.

### 4.3 Teknik Dasar Kode Huffman

Cara Kerja metode kode huffman ini adalah sebagai berikut:

1. Menghitung banyaknya jenis karakter dan jumlah dari masing-masing karakter.
2. Menyusun setiap jenis karakter dengan urutan jenis karakter yang jumlahnya paling sedikit ke jumlah yang paling banyak.
3. Membuat pohon biner berdasarkan urutan karakter dari jumlahnya terkecil ke yang terbesar dan memberi kode untuk tiap karakter.
4. Mengganti data yang ada dengan kode bit berdasarkan pohon biner.


Dengan mengikuti langkah-langkah diatas maka kita dapat melakukan kompresi menggunakan metode kode huffman. Kita dapat menyelesaikan kasus kompresi seperti ini : "ABBABABACAACDDD".

Untuk menyelesaikan kasus ini pertama adalah melakukan encoding. Melakukan encoding dapat dilakukan dengan

mengikuti langkah langkah diatas.

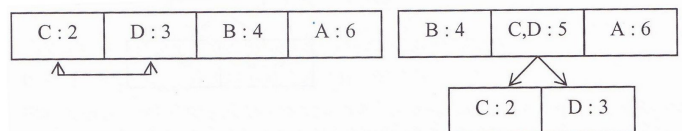
1. Membuat daftar frekuensi tiap-tiap kemunculan karakter dan urutkan dari yang terbesar.

Karakter	Frekuensi
A	6
B	4
C	3
D	2



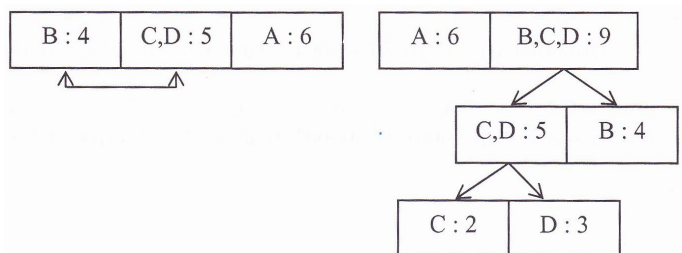
Gambar 2 Teknik Metode Huffman 1  
(sumber: firlizaa.blogspot.com)

2. Gabung dua pohon yang mempunyai frekuensi kemunculan terkecil kemudian diurutkan kembali



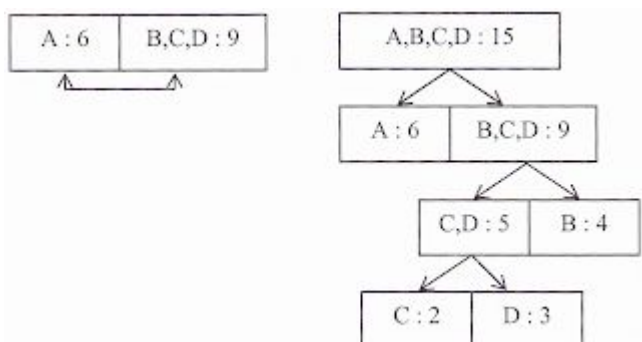
Gambar 3 Teknik Metode Huffman 2  
(sumber: firlizaa.blogspot.com)

3. Gabungkan dua pohon yang mempunyai frekuensi kemunculan terkecil kemudian diurutkan kembali.



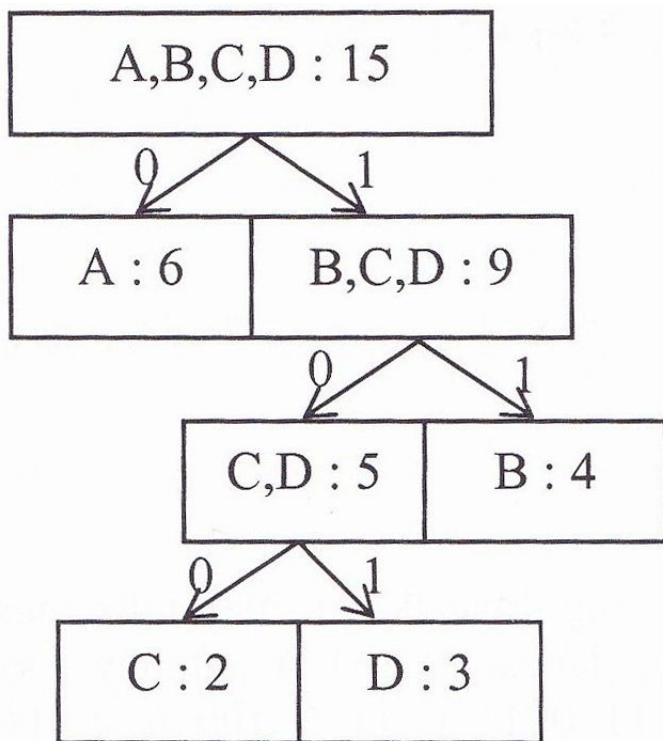
Gambar 4 Teknik Metode Huffman 3  
(sumber: firlizaa.blogspot.com)

4. Gabungkan kembali dua pohon yang mempunyai frekuensi kemunculan paling kecil



Gambar 5 Teknik Metode Huffman 4  
(sumber: firlizaa.blogspot.com)

5. Kemudian di akhir beri label dari akar ke daun, sebelah kiri = 0 dan sebelah kanan = 1



Gambar 6 Teknik Metode Huffman 5  
(sumber: firlizaa.blogspot.com)

Setelah mengikuti langkah - langkah tersebut maka kita akan mendapatkan kode huffman seperti pada tabel dibawah ini.

A = 0	B = 11	C = 100	D = 101
<b>Karakter</b>	<b>Frekuensi</b>	<b>Kode Huffman</b>	
A	6	0 = 1 bit	
B	4	11 = 2 bit	
C	2	100 = 3 bit	
D	3	101 = 3 bit	

Gambar 7 Tabel Kode Huffman  
(sumber: firlizaa.blogspot.com)

Dari kasus di atas yaitu melakukan kompresi kata "ABBABABACAACDDD" dengan ukuran 120 bit setelah dilakukan kompresi menggunakan metode huffman menjadi 29 bit. Artinya kita dapat hitung rasio pemampatannya dengan rumus =  $(100\% - 29/120 \times 100\%) = 75.8\%$ . Hal itu berarti dengan menggunakan metode kode huffman kita dapat memampatkan 75.8% bit dari string tersebut.

Setelah melakukan proses encoding, maka tahap terakhir yang perlu dilakukan adalah proses decoding. Proses decoding ini akan menjadi proses yang lebih mudah karena kode yang dihasilkan dari kode huffman unik. Contohnya saat membaca kode bit pertama dalam rangkain bit "0 11 11 0 11 0 11 0 100 0

0 100 101 101 101", yaitu bit "0" dapat disimpulkan bahwa bit "0" mempresentasikan karakter "A". Kemudian bit "1" tidak ditemukan pada kode huffman tersebut karena kode tersebut maka secara otomatis akan membaca kode selanjutnya yaitu "11". Bit tersebut akan mempresentasikan karakter "B" dan berikut juga seterusnya. Pada proses decoding tidak dapat dilakukan ketika belum melakukan proses encoding. Hal itu dikarenakan pada proses memerlukan *keywords* yang dihasilkan dari proses encoding.

## 5. PEMBAHASAAN KOMPRESI CITRA DIGITAL MENGGUNAKAN METODE KODE HUFFMAN

### 5.1 Algoritma Kompresi

Kompresi citra digital dilakukan untuk meminimalkan kebutuhan memori dengan cara mengurangi duplikasi data di dalam citra sehingga dapat mempresentasikan citra yang mempunyai memori lebih kecil daripada yang semula. Tujuan dari melakukan kompresi citra antara lain adalah menghemat penyimpanan dan juga mempercepat waktu dalam melakukan *upload* citra ke jaringan. Salah satu teknik yang dapat digunakan untuk melakukan kompresi adalah menggunakan metode kode huffman. Kode huffman termasuk ke dalam *lossless compression*. Algoritma pada kompresi menggunakan metode kode huffman adalah berikut:

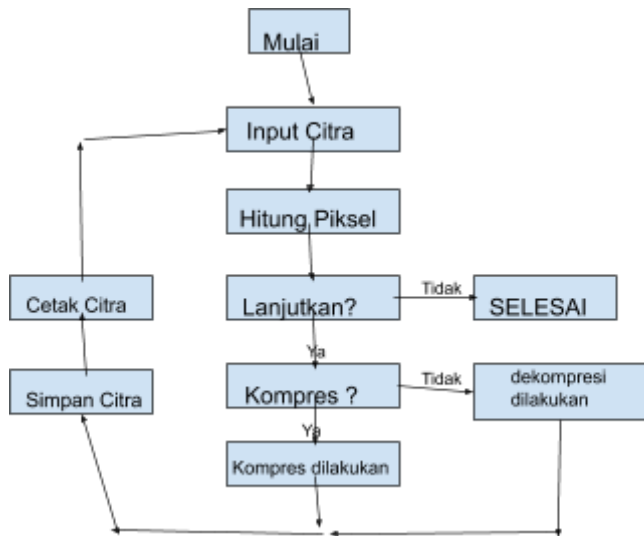
1. Masukan Citra  
Masukan citra merupakan langkah awal dari proses kompresi. Untuk melakukan kompresi kita harus memilih citra yang hendak dikompresi.
2. Menghitung nilai piksel  
Setelah memilih citra yang akan dilakukan kompresi, langkah selanjutnya adalah menghitung piksel pada citra tersebut. Hal ini bertujuan agar kita mengetahui bit-bit file pada citra yang akan kita lakukan kompres.
3. Kompresi  
Pada langkah ini citra akan mulai dilakukan kompresi yang bertujuan untuk mengurangi kelebihan dari data-data yang termuat dalam citra sehingga dapat ditransmisikan secara efisien. Setelah masuk proses ini maka akan didapatkan citra dengan ukuran yang lebih kecil daripada yang sebelumnya.
4. Simpan file hasil kompresi.
5. Lakukan langkah 1 sampai dengan 2  
Pada langkah ini untuk melakukan dekomposisi, maka perlu melakukan input citra yang akan didekomposisi. Kemudian lakukan perhitungan piksel pada citra yang dipilih.
6. Dekompresi  
Setelah menginput citra yang akan dilakukan dekomposisi langkah selanjutnya adalah proses dekomposisi. Langkah ini bertujuan agar informasi pada file citra tidak hilang, sehingga setelah dilakukan



kompresi file citra masih sama dengan file yang sebelumnya. Akan tetapi bit-bitnya telah berubah.

- Lakukan penyimpanan pada file hasil kompresi dan dekompresi.

Untuk lebih memahami algoritma tersebut dapat dilihat pada *flowchart* berikut.



Gambar 8. Flowchart Algoritma

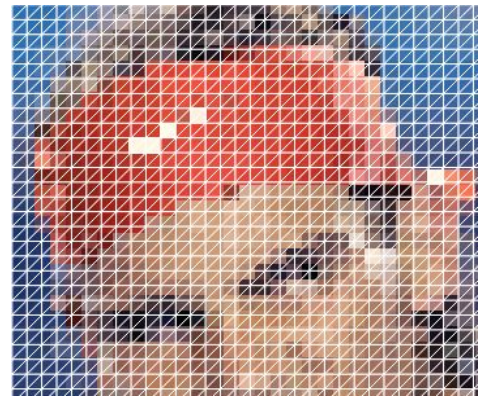
### 5.2 Melakukan Uji Kompresi

Langkah pertama dalam melakukan uji adalah memilih gambar yang akan dilakukan citra. Misal gambar yang akan dilakukan citra adalah gambar berikut ini.



Gambar 9. Gambar Uji  
(sumber: belajarcoreldraw.co)

Gambar uji tersebut kemudian diubah kedalam bentuk piksel



Gambar 10. Hasil Pikel  
(sumber: belajarcoreldraw.co)

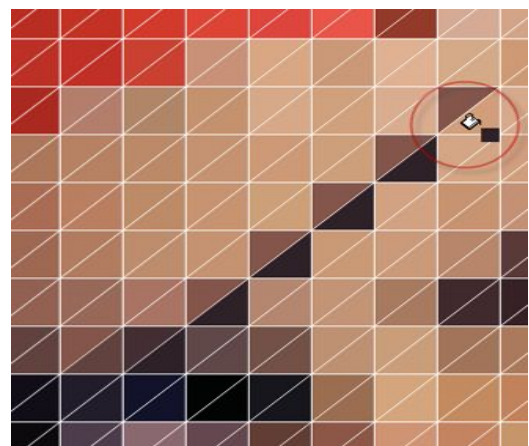
Pada data uji terdapat citra dengan representasi matriks. Tiap piksel pada gambar akan didekodekan dengan 8 bit warna sebagai berikut.

$$\begin{bmatrix} 200 & 100 & 150 & 200 \\ 100 & 100 & 100 & 100 \\ 150 & 250 & 150 & 200 \\ 100 & 200 & 200 & 250 \end{bmatrix}$$

representasi warna:

Dari data tersebut akan dilakukan kompresi. Proses dari melakukan kompresi sebagai berikut:

- Data dari hasil menghitung piksel diubah kedalam bentuk vektor. Bentuk vektor dari citra tersebut adalah [100 100 100 100 100 100 150 150 150 200 200 200 200 200 250 250] jumlah piksel yang didapat adalah enam belas.
- Setelah itu kumpulan piksel pada gambar berdasarkan frekuensi kemunculan warna pada gambar.



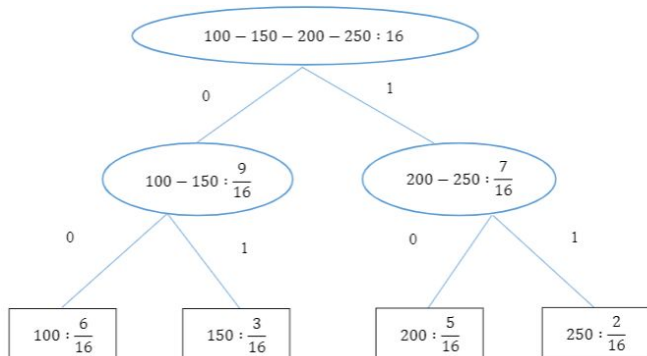
Gambar 11. Gambar Hasil Probabilitas  
(sumber: belajarcoreldraw.co)  
hasil dari perhitungan probabilitasnya sebagai berikut

$$100 = \frac{6}{16}, 150 = \frac{3}{16}, 200 = \frac{5}{16}, 250 = \frac{2}{16}$$

3. Langkah selanjutnya adalah menentukan simpul dari pohon biner. Setelah melakukan perhitungan pada langkah sebelumnya, maka kita mendapatkan simpul dari pohon biner.

$$100 : \frac{6}{16} \quad 150 : \frac{3}{16} \quad 200 : \frac{5}{16} \quad 250 : \frac{2}{16}$$

4. Setelah menentukan simpul maka langkah terakhir adalah membuat pohon binernya. Pohon biner yang didapatkan adalah sebagai berikut.

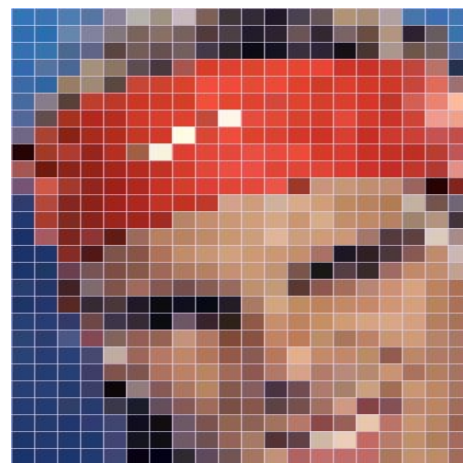


Pada proses ini kita sudah bisa menentukan bit baru yang bisa dijadikan sebagai representasi dari piksel-piksel. Bit yang dihasilkan dari pohon biner akan lebih kecil daripada bit yang sebelumnya sehingga ukuran gambar yang dilakukan zip akan memampat. Untuk lebih mudah melihat hasil bit yang baru dapat dilihat pada tabel dibawah ini

Warna	Peluang kemunculan	Kode Huffman
100	$\frac{6}{16}$	00
150	$\frac{3}{16}$	01
200	$\frac{5}{16}$	10
250	$\frac{2}{16}$	11

Tabel 1. Hasil Kompresi

Pada akhir proses kompresi bit-bit tersebut akan disusun kembali seperti representasi sebelumnya. Sehingga didapatkan gambar yang sama dengan gambar masukan. Hasil dari kompres gambar uji sebagai berikut.



Gambar 12. Gambar Hasil Uji (sumber: belajarcoreldraw.co)

## 6. KESIMPULAN

Kode Huffman merupakan sebuah algoritma kompresi yang praktis dan mudah dipelajari. Akan tetapi karena kepraktisan tersebut hasil yang didapatkan dari kompresi kurang maksimal. Akan tetapi, untuk kalangan pemula sangatlah cocok karena algoritma kode Huffman membangun pola pikir dan gambaran umum proses kompresi file. Ketika sudah terbangun pola pikir maka mahasiswa akan lebih mudah dalam melakukan pengembangan yang dapat menghasilkan hasil kompres yang lebih baik.

## 7. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah memberi kelancaran dalam penulisan makalah ini. Tidak lupa juga kepada Ibu Dra. Harlili M.Sc. selaku pengajar matematika diskrit kelas 02. Tidak lupa juga kepada Bapak Rinaldi Munir yang telah memberi ruang bahan bacaan yang bisa dijadikan acuan dalam mencari sumber-sumber lain yang berkaitan pada makalah ini.

## 8. REFRENSI

- [1] Munir,Rinaldi.(2006).Pohon.  
[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) diakses pada tanggal 5 Desember 2019
- [2] <https://elektronika-dasar.web.id/definisi-dan-pengolahan-citra-digital/> diakses pada tanggal 5 Desember 2019
- [3] <https://docplayer.info/45491066-Teknik-kompresi-citra-menggunakan-metode-huffman.html> diakses pada tanggal 5 Desember 2019
- [4] <https://openlibrary.telkomuniversity.ac.id/pustaka/95350/kompresi-citra-lossless-menggunakan-improved-huffman-coding.html>. diakses pada tanggal 5 Desember 2019
- [5] <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/> diakases pada tanggal 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, 6 Desember 2019



Fatkhan Masruri 13518053