

## HUFFMAN CODING PADA IMAGE COMPRESSION

Hendarman Lubis<sup>1</sup>, Dwi Budi Srisulistiowati<sup>2</sup>, Adhitya Ilham Ramdhani<sup>3</sup>

<sup>1</sup>Hendarman Lubis, Informatika, Universitas Bhayangkara, [hendarman.lubis@dsn.ubharajaya.ac.id](mailto:hendarman.lubis@dsn.ubharajaya.ac.id)

<sup>2</sup>Dwi Budi Srisulistiowati, Informatika, Universitas Bhayangkara, [dwibudi@dsn.ubharajaya.ac.id](mailto:dwibudi@dsn.ubharajaya.ac.id)

<sup>3</sup>Adhitya Ilham Ramdhani, Manajemen Informatika, STMIK Bani Saleh, [adhityair@gmail.com](mailto:adhityair@gmail.com)

**Abstract**—Dengan perkembangan teknologi informasi, citra telah menjadi arus utama informasi penularan. Dibandingkan dengan karakter, gambar mengandung lebih banyak informasi, tetapi karena gambar membutuhkan lebih banyak kapasitas penyimpanan, itu akan menempati lebih banyak bandwidth dalam transmisi jaringan. Karena kebutuhan kapasitas yang terus meningkat seiring perkembangan teknologi, dibutuhkan metode untuk mengkompresi citra/gambar, peningkatan kecepatan kompresi, dan peningkatan efisiensi transmisi. Metode kompresi citra dalam makalah ini adalah menggunakan Huffman Coding dimana citra disaring oleh wavelet transform untuk menghilangkan informasi yang berlebihan pada gambar, dan kemudian metode Huffman digunakan untuk menyandikan gambar yang menghasilkan ukuran gambar dapat diperkecil dengan efek gambar yang sama, lebih kompleks lagi melalui metode Canonical Huffman yang digunakan untuk kompresi fraktal yang lebih baik daripada pengkodean aritmatika untuk meningkatkan kecepatan kompresi, dan distorsi mikro yang menciptakan sedikit distorsi dibandingkan kompresi distorsi terbatas tradisional untuk meningkatkan efisiensi transmisi sesuai dengan fitur pengkodean Huffman dan untuk mencapai efisiensi pengkodean maksimum teoritis.

**Kata Kunci**—*Compression, Coding, Huffman. Image, Rasio*

### PENDAHULUAN

Sebuah gambar atau citra biasanya terdiri dari sejumlah besar data dan membutuhkan banyak ruang dalam memori. Jika lebih banyak jumlah data yang dibutuhkan untuk transmisi maka dibutuhkan banyak waktu untuk mengirimkan data tersebut ke penerima. Jadi dengan menggunakan teknik kompresi gambar, konsumsi waktu dapat sangat dikurangi. Dalam hal ini, penghapusan data yang berlebihan dalam sebuah citra dapat dilakukan. Gambar yang dikompresi menempati lebih sedikit ruang dalam memori dan membutuhkan lebih sedikit waktu untuk mengirimkan informasi dari pemancar ke penerima. Kompresi berarti memperkecil ukuran file dengan mengatur ulang data dalam file. Mengkompresi citra berbeda dengan mengompresi file. Fungsi utama dari kompresi gambar adalah untuk mengatur ulang data dan dapat menurunkannya untuk mencapai tingkat kompresi yang diinginkan, bergantung pada rasio kompresi. Jika ada rasio kompresi yang lebih baik, semakin kecil ukuran file di sini, semakin banyak data yang dikemas ke dalam ruang yang lebih kecil, tetapi menurunkan kualitas produk yang dikompresi.

Terdapat berbagai metode yang digunakan untuk kompresi multimedia (teks, suara, citra/gambar, animasi, audio, dan video). Metode umum yang biasa digunakan dibagi menjadi metode kompresi lossless dan kompresi lossy metode. Kompresi lossless memampatkan bagian yang berlebihan dari data asli. Menggunakan kompresi lossless, data asli bisa dipulihkan sepenuhnya tanpa kesalahan atau distorsi, yaitu, setelah kompresi dan dekompresi, salinan data asli adalah dihasilkan. Rasio kompresinya umumnya 2:1 - 5:1. Seperti tipikal Teknologi *Double Space*, rasio kompresi berbagai jenis data dan file pada hard disk sekitar 2:1. Karena batasan rasio kompresi, penggunaan kompresi lossless saja tidak bisa menangani masalah penyimpanan dan transmisi suara digital dan gambar video secara real time. Kompresi lossy menjadi beban informasi tertentu, sehingga rasio kompresi dapat lebih tinggi tercapai. Metode kompresi lossy banyak digunakan untuk gambar dengan piksel, video, atau file kualitas suara yang lebih tinggi. Untuk ini jenis kompresi data, rasio kompresi dapat ditingkatkan puluhan atau ratusan kali. Kebanyakan metode kompresi gambar dapat menggunakan pendekatan ini, terutama JPEG, MPEG, dan jenis file lainnya. Metode pengkodean umum yang digunakan

dalam kompresi lossy bersifat prediktif pengkodean dan transformasi pengkodean, yang memungkinkan informasi untuk hilang dalam proses kompresi. Meskipun semua data tidak bisa pulih sepenuhnya setelah dekompresi, tetapi bagian gambar yang hilang, apakah gambar asli atau suara, memiliki sedikit pengaruh pada pemahaman tentang keseluruhan file, itu bisa mendapatkan kompresi yang baik rasio.

Kompresi gambar fraktal adalah salah satu yang tidak dapat diabaikan kompresi mendekati karena memiliki tingkat yang lebih baik kompresi dari JPEG. Kompresi Gambar Fraktal, pendekatan kompresi lossy, yang dapat diterapkan untuk gambar digital dan biasanya didasarkan pada fraktal. Pendekatan ini adalah salah satu pendekatan akuntabel terbaik tekstur dan gambar nyata (gambar alam) yang bergantung pada fakta yang biasanya dimiliki oleh komponen piksel dari suatu gambar kemiripan yang mencolok dengan komponen lain dari gambar itu. Pendekatan fraktal menerjemahkan komponen-komponen ini menjadi data matematika yang disebut sebagai "kode fraktal" dan mereka dapat digunakan untuk membuat ulang gambar yang dikodekan. Algoritma genetik merupakan metode pencarian yang optimal bertindak seperti proses genetik alami yang dapat diterapkan untuk gambar alami karena memiliki banyak fitur.

Pada paper ini berfokus untuk mempelajari bagaimana melakukan Image Compression dengan Huffman Coding. Sesi 2 dari paper ini akan menjelaskan beberapa teori pendukung terkait berbagai metode Image Processing, termasuk Huffman Codingnya itu sendiri. Sesi 3 akan menjelaskan eksperimen dan analisis dari percobaan Image Compression dengan Huffman Coding. Sesi 4 akan berisikan sebuah kesimpulan akhir dari semua penjelasan dan eksperimen yang ada.

**TEORI PENDUKUNG**

Pada sesi ini akan berfokus membahas beberapa teknik atau metode yang berkaitan dengan Huffman image compression.

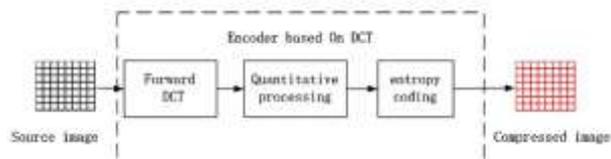
**A. Lossy Compression JPEG berbasis transformasi DCT menggunakan Huffman coding**

JPEG merupakan metode yang biasa digunakan dari *lossy compression* untuk citra/gambar digital. Ada beberapa mode JPEG, yang paling umum yang merupakan mode sekuensial berdasarkan transformasi

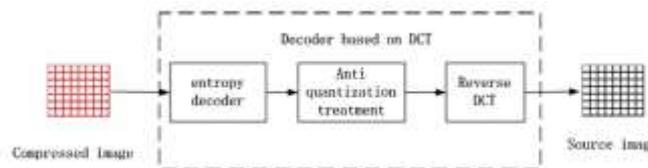
DCT. JPGE adalah algoritma transformasi kosinus diskrit dua dimensi menggunakan sub-blok 8x8.

Algoritme pertama kali membagi yang asli gambar secara berurutan menjadi serangkaian 8 x 8 sub-blok. Dalam 8 x 8 gambar blok, nilai piksel umumnya berubah lebih lembut, jadi citra memiliki frekuensi spasial yang lebih rendah. 8 x 8 blok gambar setelah transformasi DCT, komponen frekuensi rendah terkonsentrasi di sudut kiri atas, frekuensi tinggi komponen di pojok kanan bawah. Kuantifikasi adalah membuang informasi yang memiliki pengaruh kecil pada efek visual di bawah premis untuk mempertahankan kualitas tertentu. Pengukur seragam linier digunakan dalam standar JPGE.

Proses kuantifikasi adalah pembagian 64 koefisien DCT dengan ukuran langkah kuantifikasi dan pembulatan. Komponen frekuensi dipertahankan dan komponen frekuensi tinggi ditekan oleh pemrosesan kuantifikasi. Karena koefisien DC dari dua blok 8x8 yang berdekatan sangat kecil, pengkodean diferensial DCPM (Differential predictive coding modulation) digunakan untuk meningkatkan rasio kompresi yang terdapat pada format tengah. Setelah mendapatkan format tengah, jumlah tanda kurung yang termasuk contoh di atas dikodekan oleh Huffman. Gambar. 1 dan 2 menggambarkan mengenai inti dari langkah-langkah pemrosesan encoder dan decoder berdasarkan DCT dalam JPEG.



**Gambar 1. Encoding DCT encoder**



**Gambar 2. Decoding DCT decoder**

Dalam proses encoding, data gambar sumber dibagi menjadi 8 x 8 blok. DCT maju mengubah setiap blok menjadi 64 DCT koefisien. Amplitudo dari transformasi frekuensi spasial Dalam proses encoding, data gambar sumber dibagi menjadi 8 x 8 blok. DCT maju mengubah setiap blok menjadi 64 DCT

koefisien. Amplitudo koefisien transformasi frekuensi spasial sebagian besar adalah nol atau cenderung nol. Dengan demikian dimungkinkan untuk kompres data. Rumus DCT maju adalah:

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[ \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

$$\text{where } \begin{cases} C(u), C(v) = 1/\sqrt{2}, (u, v = 0) \\ C(u), C(v) = 1, (\text{other}) \end{cases}$$

Setelah mengeluarkan dari DCT maju, pembilang mengkuantisasi nilai koefisien menurut tabel kuantifikasi. Tujuannya adalah menentukan ukuran langkah  $Q(u, v)$  dari pembilang yang sesuai dengan kualitas gambar yang diinginkan, dan untuk mewakili koefisien DCT  $F(u, v)$  dengan presisi yang sesuai untuk mencapai lebih jauh kompresi. Rumus kuantifikasinya adalah:

$$F^Q(u, v) = \text{round} \left( \frac{F(u, v)}{Q(u, v)} \right)$$

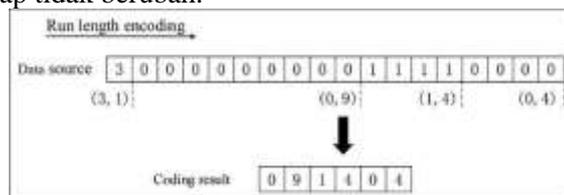
Setelah kuantifikasi, koefisien DC berbeda-beda dikodekan, dan koefisien AC diatur dalam bentuk “Z” dan kemudian pengkodean entropi dilakukan. Langkahnya adalah melakukan lossless pengkodean kompak sesuai dengan karakteristik statistik terkuantisasi koefisien, dan metode pengkodean entropi yang tersedia adalah Pengkodean Huffman dan pengkodean aritmatika. Di antara mereka, Huffman encoding membutuhkan satu atau lebih pernyataan tabel Huffman itu ditentukan oleh aplikasi. Setiap 8 x 8 blok data sampel, setelahnya langkah-langkah pemrosesan di atas, akhirnya mengeluarkan data gambar yang dikompresi untuk penyimpanan atau transmisi.

Proses decoding adalah kebalikan dari proses encoding. Itu decoder entropi melakukan decoding Huffman atau aritmatika. Itu proses kuantifikasi terbalik menggunakan nilai perkiraan diperoleh dari data yang diterjemahkan sebagai input dari DCT terbalik. DCT terbalik mengubah 64 koefisien dengan transformasi terbalik- untuk merekonstruksi gambar output 64 titik.

### B. Run-length encoding

Merupakan bentuk kompresi lossless yang mengambil string atau rangkaian data yang berlebih dan menyimpannya sebagai satu unit. Misal ada sebuah gambar bergaris merah dan putih, terdiri dari

12 piksel putih dan 12 piksel merah. Biasanya, data tersebut akan ditulis sebagai WWWWWWWWWWRRRRRRRRRRRR, dengan W mewakili piksel putih dan R piksel merah. Run length akan menempatkan data tersebut sebagai 12W dan 12R. Sehingga akan jauh lebih kecil dan sederhana dari bentuk awalnya, serta tetap menjaga data tetap tidak berubah.



**Gambar 3.** Cara kerja Run-length encoding

### C. Canonical Huffman Coding

Pendekatan fraktal adalah salah satu pendekatan akuntabel terbaik untuk tekstur dan gambar nyata (natural image) yang bergantung pada klausul yang biasanya dimiliki komponen piksel dari suatu gambar kemiripan yang mencolok dengan komponen lain dari gambar itu. Pendekatan fraktal menerjemahkan komponen-komponen ini menjadi data matematika yang disebut sebagai “kode fraktal” dimana bayangan fraktal dalam bentuk dua dimensi dan dihasilkan dengan menggunakan berbagai jenis transformasi dalam file gambar asli seperti DCT, DWT, FFT dll. Discrete Cosine Transform (DCT) membantu memisahkan gambar menjadi komponen atau spektral subband yaitu fraktal dengan signifikansi yang dibedakan dengan menghormati kualitas visual dan penampilan gambar.

Pendekatan DCT dan Discrete Fourier Transform (DFT) mirip satu sama lain yang mengubah gambar dari domain spasial ke domain frekuensi. Frekuensi komponen domain diterapkan untuk membuat ulang yang dikodekan gambar. Persamaan satu dimensi umum untuk satu dimensi (N item data) didefinisikan oleh:

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} A(i) \cdot \cos \left[ \frac{\pi \cdot u}{2N} (2i + 1) \right] f(i)$$

dengan

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \delta = 0 \\ 1 & \text{else} \end{cases}$$

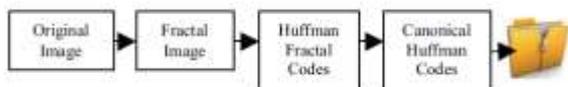
Persamaan dua dimensi umum untuk dua dimensi (gambar N \* M) DCT didefinisikan oleh

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A(i, j) \cdot A$$

dengan

$$A = \cos \left[ \frac{\pi \cdot u}{2N} (2i + 1) \right] \cos \left[ \frac{\pi \cdot v}{2M} (2j + 1) \right] f(i, j)$$

Canonical Huffman Coding adalah kode Huffman yang memungkinkan sangat kompak pada cara menyimpan informasi yang diperlukan untuk memecahkan kodenya karena kode-kode tersebut secara leksikal diurutkan berdasarkan panjangnya. Dengan begitu, ditentukan bahwa panjang kode adalah konstan seperti dengan Kode Huffman sejak kode Canonical Huffman adalah dihasilkan dari mereka dan akibatnya tidak ada kerugian dalam kompresi saat menggunakan kode ini. Misalkan kode Huffman dari 4 bit menjadi "0001" dalam istilah kode kanonik nilainya hanya 4 yang sama dengan totalnya jumlah bit yang ada dalam kode Huffman-nya.



**Gambar 4.** Cara kerja Canonical Huffman encoding

**D. Transform**

Merupakan kompresi lossy yang biasa digunakan untuk JPEG. Ada jutaan corak warna, dan pengkodean transformasi mengambil warna yang memiliki corak serupa dan menjadikannya satu nilai tunggal. Bergantung pada nilai kompresi yang ditentukan (yaitu jumlah bayangan warna yang ingin dikelompokkan bersama), sehingga bisa saja terlihat perbedaan kualitas gambarnya atau tidak sama sekali.

**E. Huffman Coding**

Huffman coding adalah metode kompresi data yang tidak bergantung pada tipe datanya, yaitu data dapat berupa gambar, audio, atau spreadsheet. Skema kompresi ini digunakan dalam JPEG dan MPEG-2. Pengkodean Huffman bekerja dengan melihat aliran

data yang membentuk file yang akan dikompresi. Byte data yang paling sering muncul diberi kode kecil untuk mewakilinya (tentunya lebih kecil dari byte data yang diwakili). Byte data yang paling sering muncul berikutnya memiliki kode yang sedikit lebih besar untuk mewakilinya. Ini berlanjut hingga semua bagian data unik diberi kata kode unik. Untuk distribusi karakter tertentu, dengan menetapkan kode pendek ke karakter yang sering muncul dan kode yang lebih panjang ke karakter yang jarang muncul, pengkodean redundansi minimum Huffman meminimalkan jumlah rata-rata byte yang diperlukan untuk mewakili karakter dalam teks. Pengkodean Static Huffman menggunakan sekumpulan kode tetap, berdasarkan sampel data yang representatif, untuk memproses teks. Meskipun pengkodean dicapai dalam sekali jalan, data yang menjadi dasar kompresi mungkin memiliki sedikit kemiripan dengan teks sebenarnya yang sedang dikompresi. Pengkodean Dynamic Huffman, di sisi lain, membaca setiap teks dua kali; sekali untuk menentukan distribusi frekuensi karakter dalam teks dan sekali untuk menyandikan data. Kode yang digunakan untuk kompresi dihitung berdasarkan statistik yang dikumpulkan selama lintasan pertama dengan teks terkompresi yang diawali dengan salinan tabel pengkodean Huffman untuk digunakan dengan proses decoding. Dengan menggunakan teknik single-pass, di mana setiap karakter dikodekan berdasarkan karakter sebelumnya dalam teks, encoding Huffman adaptif Gallager menghindari banyak masalah yang terkait dengan metode statis atau dinamis.

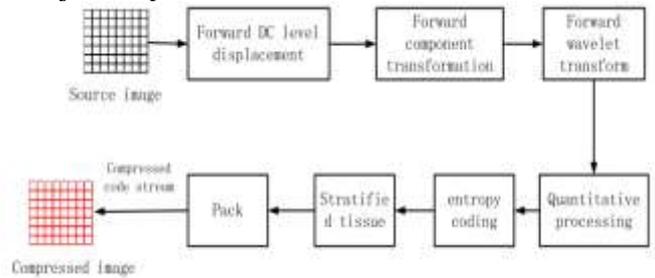
Huffman Coding juga termasuk kedalam algoritma kompresi data lossless. Dalam algoritma ini, kode panjang variabel ditugaskan untuk memasukkan karakter yang berbeda. Panjang kode terkait dengan seberapa sering karakter digunakan. Karakter yang paling sering memiliki kode terkecil dan kode yang lebih panjang untuk karakter yang paling jarang.

Terutama ada dua bagian. Pertama untuk membuat pohon Huffman, dan satu lagi untuk melintasi pohon untuk menemukan kode. Sebagai contoh, perhatikan beberapa string "YYYZZXXYYX", frekuensi karakter Y lebih besar dari X dan karakter Z memiliki frekuensi paling sedikit. Jadi panjang kode untuk Y lebih kecil dari X, dan kode untuk X lebih kecil dari Z. Kompleksitas untuk menetapkan kode untuk setiap karakter menurut frekuensinya adalah  $O(n \log n)$ .

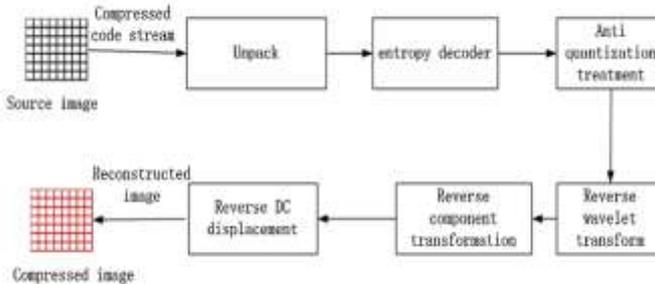
**EKSPERIMEN DAN ANALISIS**

*A. Research on image compression technology based on Huffman coding*

Transformasi wavelet yang digunakan pada JPEG2000 yang diadopsi sebagai multiresolusi metode pengkodean untuk menguraikan gambar menjadi sub-grafik frekuensi rendah, sub-grafik frekuensi tinggi dalam arah horizontal, frekuensi tinggi sub-grafik dalam arah vertikal dan frekuensi tinggi sub-grafik dalam arah garis diagonal dengan menggunakan algoritma Mallat. Setelah wavelet bertransformasi, setiap tingkat dekomposisi wavelet posisi data gambar selalu membagi frekuensi rendah tingkat atas data ke dalam pita frekuensi yang lebih halus. Metode ini tidak hanya dapat memperoleh efek kompresi yang lebih baik, tetapi juga dapat mengatasi file Efek " persegi " yang dihasilkan oleh transformasi DCT di JPGE. Itu JPEG2000 gambar encoding dan decoding diagram aliran diusulkan dalam makalah ini ditunjukkan pada Gambar. 5 dan 6.

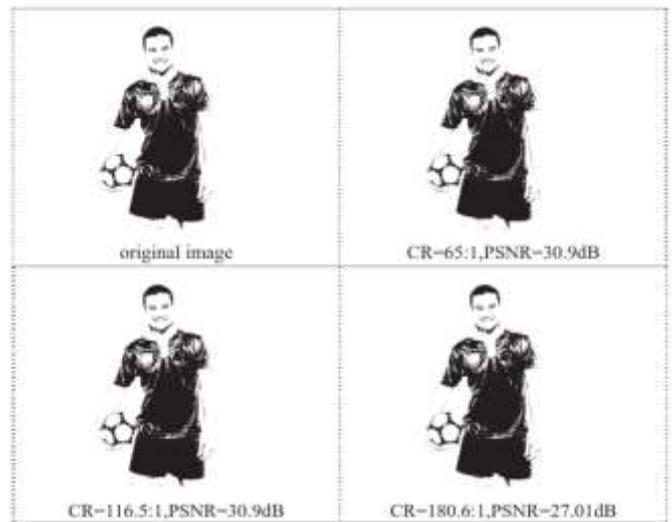


**Gambar 5.** Proses encoding transformasi wavelet

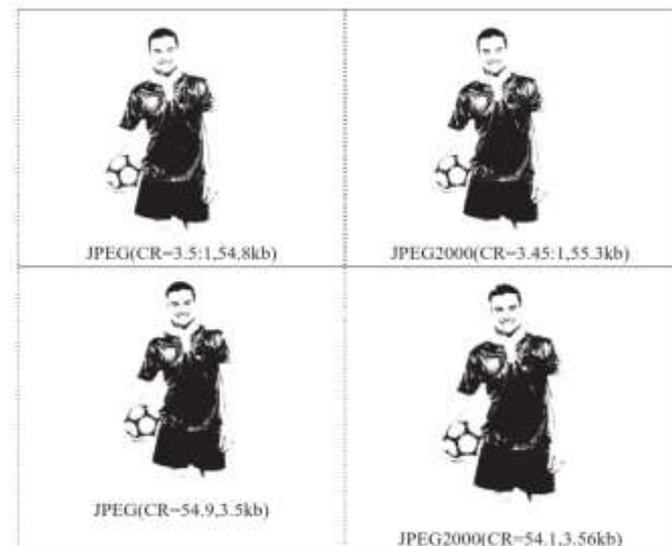


**Gambar 6.** Proses decoding transformasi wavelet

Dari proses tersebut menghasilkan kompresi data pada gambar berikut ini.



**Gambar 7.** Efek dari perbedaan rasio kompresi JPEG2000



**Gambar 8.** Efek dari perbedaan JPEG dan JPEG2000 dengan perbedaan rasio kompresi

Dalam percobaan ini telah dipilih file gambar berformat bitmap picture1.bmp. Setelah metode kompresi JPEG2000 yang telah diusulkan dalam makalah ini digunakan untuk mengompresi file dengan kompresi yang berbeda rasio, efek perbandingan gambar ditunjukkan pada Gambar 7.

Pada Gambar 7 menunjukkan efek gambar yang sesuai setelah kompresi dan SNR (rasio signal-to-noise) gambar pada rasio kompresi yang berbeda untuk gambar JPEG2000. Dimana CR adalah rasio kompresi dan PSNR adalah SNR puncak. Dari efek

gambar, gambar JPEG2000 mempertahankan SNR yang baik pada rasio kompresi yang berbeda.

Dengan rasio kompresi yang lebih tinggi, gambar kompresi JPEG2000 adalah halus dan jernih. Selain itu, gambar JPGE picture2.jpg dikompresi dengan cara yang berbeda rasio kompresi menurut kompresi JPEG2000 metode dan metode kompresi JPEG diperkenalkan dalam makalah ini. Efek kontras gambar ditunjukkan pada Gambar 8. Seperti dapat dilihat dari Gambar 8, dalam kasus rasio kompresi rendah, tidak ada perbedaan yang signifikan antara kedua kompresi tersebut metode, tetapi sebagai rasio kompresi meningkat, perbedaannya menjadi jelas. Efek " persegi " dari gambar JPEG terlihat jelas, dengan bercak putih besar. JPEG2000 belum signifikan terdistorsi.

*B. Canonical Huffman Coding for Image Compression*

Dalam makalah tersebut hasil yang diperhitungkan dalam penelitian ini adalah diekspresikan yang terdiri dari warna dan skala abu-abu gambar untuk aplikasi kompresi. Beberapa dari gambar yang digunakan untuk aplikasi kompresi Algoritma Anime, Onion dan Redeyes dalam gambar berwarna, dan Pout dan Tire dalam gambar skala abu-abu.



**Gambar 9.** Anime



**Gambar 10.** Redeyes



**Gambar 11.** Onion



**Gambar 12.** Pout

Citra skala abu-abu merupakan citra dan warna dua dimensi gambar adalah gambar tiga dimensi jadi untuk melakukan DCT dan untuk menyandikan gambar itu akan diubah menjadi dua gambar dimensi dengan membagi menjadi tiga bidang yaitu menjadi Red Plane, Green Plane dan Blue Plane atau menjadi Luma Plane Komponen, Blue Difference plane dan Red Difference plane.

Karena komponen warna terdiri dari tiga bidang di dalamnya contoh RGB jadi lagi-lagi dipecah menjadi 3 bidang untuk melakukan pengkodean Huffman yang membuat total enam gambar dekomposisi tingkat tunggal DWT yang meningkatkan kompleksitas algoritma. Jadi, bukan file DWT, DCT digunakan. DWT seperti gambar dibawah:

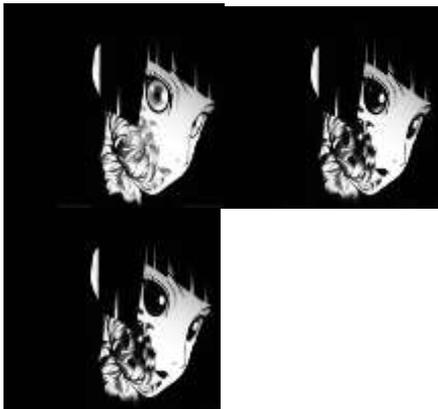


**Gambar 13.** DWT gambar

Sebelum membuat kode Huffman dan Canonical Kode Huffman, gambar berwarna akan dipecah menjadi tiga gambar dua dimensi yaitu menjadi 3 bidang. Jadi ketika gambar berwarna Mata merah rusak, kemudian dapat diekspresikan sebagai berikut:



**Gambar 14.** Plane Pertama, Kedua, dan Ketiga Anime



Gambar 15. Plane Pertama, Kedua, dan Ketiga Redeyes



Gambar 16. Plane Pertama, Kedua, dan Ketiga Onion

Gambar akan diubah menjadi gambar fraktal kompres. Kemudian DCT diterapkan pada gambar dan fraktal gambar dihasilkan di mana komponen frekuensi rendah adalah bagian vital yang berisi informasi.



Gambar 17. DCT di Bidang Pertama, Bidang Kedua, dan Bidang Ketiga Redeyes



Gambar 18. DCT pada Bidang Pertama, Bidang Kedua, dan Bidang Ketiga Onion



Gambar 19. DCT Pout

Setelah generasi gambar fraktal dengan menerapkan DCT pada setiap bidang gambar berwarna atau bidang tunggal gambar skala abu-abu, kode Huffman dihasilkan dan dari kode Huffman standar, kode Canonical Huffman adalah dihasilkan. Rasio jumlah bit dalam real atau original gambar dengan jumlah bit pada gambar setelah kompresi disebut sebagai rasio kompresi. Rasio kompresi bisa dicapai dengan menghitung jumlah koefisien dalam tempat bit sebagai jumlah bit konstan yang digunakan mewakili koefisien DCT. Itu diwakili secara matematis seperti yang ditunjukkan:

$$\text{Compression Ratio} = \left( \frac{\text{size of original image}}{\text{size of compressed image}} \right)$$

Nama File	Ukuran File	Ukuran Setelah di Kompres		Rasio Kompresi
		Symbo l	H. Codes	Huffman
Redeyes.jpg	90.1KB	74.8KB	12.5KB	0.9689
Anime.jpg	383KB	312KB	69KB	0.9947
Onion.png	43.5KB	29KB	11KB	0.9195
Pout.tif	67.3KB	56KB	9.8KB	0.9777
Tire.tif	46.5KB	34KB	9.4KB	0.9333

**Tabel 1.** Rasio kompresi Kompresi (Menggunakan Coding Huffman Standar)

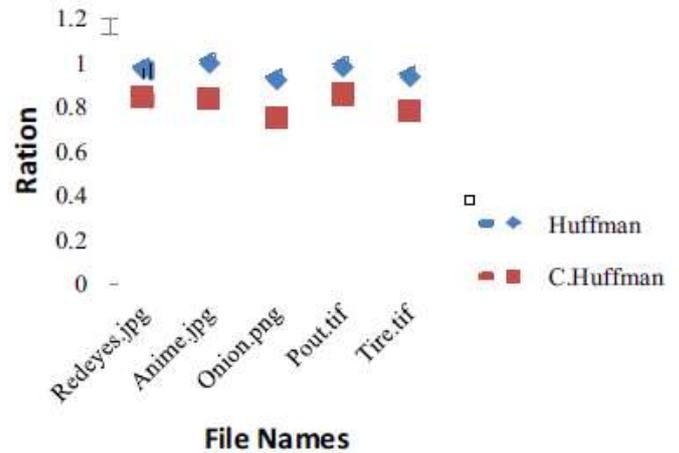
Nama File	Ukuran File	Ukuran Setelah di Kompres		Rasio Kompresi
		Symbol	H. Codes	
Redeyes.jpg	90.1KB	74.8KB	1.1KB	0.8423
Anime.jpg	383KB	312KB	7.1KB	0.8331
Onion.png	43.5KB	29KB	3.5KB	0.7471
Pout.tif	67.3KB	56KB	1.3KB	0.8514
Tire.tif	46.5KB	34KB	2.1KB	0.7763

**Tabel 2.** Rasio kompresi Kompresi (Menggunakan Canonical Huffman Coding)

Dari tabel di atas dinyatakan bahwa kompresi rasio Canonical Huffman Coding lebih baik dari itu dari Standard Huffman Coding tetapi juga menentukan bahwa orisinalitas gambar sebagian besar disimpan dalam kompresi metode pengkodean Huffman Standar. Sehingga rekonstruksi gambar dari file terkompresi di sana pasti akan lebih banyak kehilangan data dalam kasus Canonical Huffman Coding tetapi karena penelitian ini didasarkan pada kompresi sehingga belum lebih fokus dalam hal rekonstruksi.

Dari tabel di atas, kita juga dapat melihat ukuran simbol digunakan dalam kode Huffman dan kode Canonical Huffman sama tetapi hanya ada perbedaan pada kode-kode yang ada dicatat. Rasio kompresi rata-rata Standar Pengkodean Huffman kira-kira setara dengan 97% yaitu ukuran gambar berkurang 3%, sedangkan kompresi rasio dari Canonical Huffman Coding kira-kira setara dengan 82% yaitu ukuran

gambar berkurang 18%. Dari data diatas, tampaknya rasio kompresi Canonical Huffman Pengodean lebih baik.



**Gambar 20.** Perbandingan Grafis pada Rasio Kompresi

Dari grafik di atas, kita dapat mengatakan bahwa rasio kompresi pengkodean Standard Huffman hampir sama dengan Unity. Tetapi dalam kasus Canonical Huffman yang mengkodekan nilainya adalah di bawah nilai perkiraan Standard Huffman Coding. Jadi, Huffman Coding disebut sebagai Teknik Lossless Image Compression. Untuk mengubah Canonical Huffman Codes menjadi Kode Huffman diperhitungkan pendekatan sederhana, menggunakan Pendekatan Gray Code..

### C. Micro Distortion Image Compression Process

Berikut ini proses Image Compression dengan metode micro distortion.

Langkah 1: Ubah nilai RGB asli menjadi nilai diferensial. Dengan urutan serpentine, gunakan nilai RGB piksel saat ini dikurangi nilai RGB piksel sebelumnya.

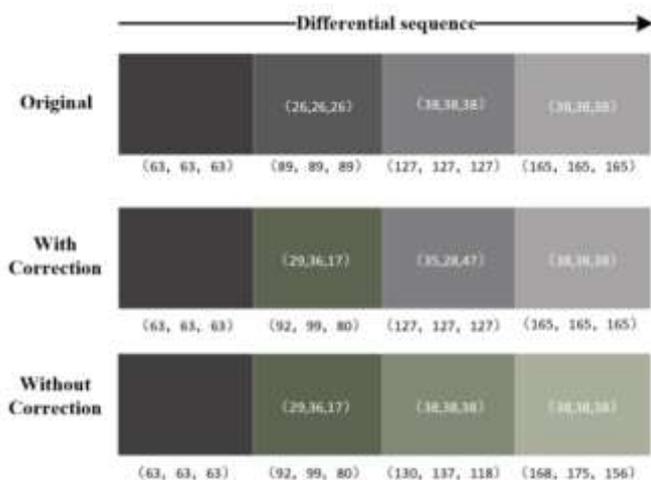
Langkah 2: Dengan urutan serpentine, implementasikan setiap piksel dengan metode pergantian piksel sebelumnya sesuai dengan diagram alir pada gambar 11. Lakukan penerapan ini pada setiap piksel satu per satu.

Langkah 3: Langkah selanjutnya adalah menghapus kemungkinan target dari setiap karakter. Meskipun fitur piksel ('karakter penarikan' atau 'karakter setoran')

ditentukan sebelum traversal, modelnya berlainan sehingga probabilitas target yang tepat tidak dapat dicapai. Menurut teori yang disebutkan dalam solusi optimal untuk transformasi karakter, probabilitas target semua piksel harus paling dekat tetapi lebih besar dari probabilitas optimal. Ketika satu karakter telah mencapai probabilitas targetnya selama transformasi piksel, karakter ini harus dihapus dari kumpulan 'karakter penarikan' atau 'karakter deposit'. Setelah itu, semua piksel dalam karakter ini tidak akan pernah tersentuh.

Langkah 4: Saat semua transformasi piksel selesai, lakukan pengkodean Huffman ke semua karakter. Saat mendekode, kembalikan nilai diferensial ke nilai RGB asli.

Langkah 5: Bandingkan hasil semua hasil traversal, pilih yang optimal sebagai hasil akhir.



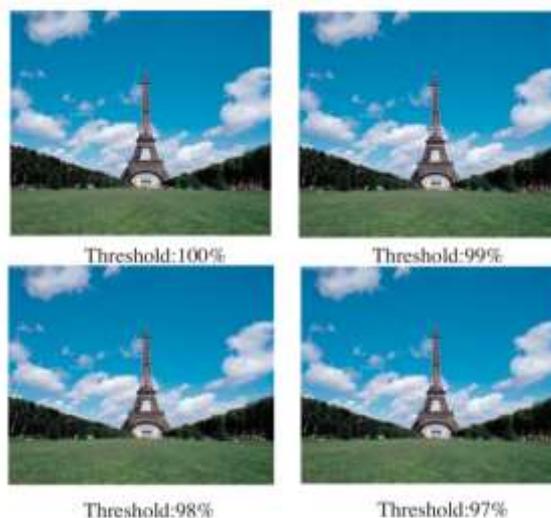
**Gambar 21.** Diagram skematis koreksi piksel tetangga

Sekarang kita menyesuaikan ambang yang berbeda, yaitu mengubah jumlah piksel untuk mengubah efisiensi kompresi. Sehingga didapatkan sebuah hasil yang dapat diamati pada tabel dibawah ini.

Set Threshold	Compression Efficiency	Difference from Original
100 %	0.99581317970 8197	0
99 %	0.99581668879	3.50909026913

	8466	8149e-06
98 %	0.99581554414 7509	2.36443931145 3026e-06
97 %	0.99581324357 4474	6.38662770446 8850e-08

**Tabel 3.** Perbandingan efisiensi kompresi



**Gambar 22.** Hasil Kompresi dengan

Dapat dilihat pada tabel dan gambar diatas, bahwa ketika Thresholdnya 100%, gambar mempunyai resolusi yang sama dengan gambar aslinya, tetapi ketika diturunkan menjadi 97%, gambar menjadi berbeda dengan gambar aslinya.

### I.KESIMPULAN

Seiring zaman berkembang, berbagai teknologi multimedia juga berkembang pesat. Teknologi kompresi data gambar saat ini tidak bisa lagi memenuhi kebutuhan berbagai macam aplikasi multimedia jaringan. Perbandingan teknik kompresi gambar juga sering kali dilakukan, sebagai contoh di atas untuk kompresi menggunakan Standard Huffman Coding dan Canonical Huffman Coding dimana Standard Huffman Coding diperhitungkan untuk verifikasi dan validasi Canonical Huffman Coding, dari contoh tersebut dapat disimpulkan bahwa pengkodean Canonical Huffman tampaknya lebih baik untuk rasio kompresi dan untuk mengkompres gambar

yang kecerahannya sama didistribusikan. Oleh karena itu, semakin banyak perhatian telah diberikan pada penelitian tentang teknologi kompresi multimedia, khususnya file kompresi data citra, peningkatan kecepatan kompresi, dan peningkatan efisiensi transmisi merupakan fokus penelitian. Dalam cakupan penelitian tentang teknologi tersebut banyak metode yang telah digunakan untuk mencari yang terbaik untuk digunakan dan dalam ruang lingkup ini, sebagai Huffman Coding dapat mencakup dalam berbagai hal tersebut, misalnya dalam *lossy compression* JPEG dengan transformasi DCT, Canonical Huffman Coding dalam *image compression*, dan dalam *micro distortion image compression*.

#### REFERENSI

- [1] Yuan, S., & Hu, J. (2019). Research on image compression technology based on Huffman coding. *Journal of Visual Communication and Image Representation*, 59, 33-38.
- [2] S. R. Khaitu and S. P. Panday, "Canonical Huffman Coding for Image Compression," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, 2018, pp. 184-190, doi: 10.1109/ICCCS.2018.8586816.
- [3] K. Bao, Q. Qian and Y. Yang, "Micro Distortion Image Compression Based on Huffman Encoding," 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 2020, pp. 319-323, doi: 10.1109/ICCCS49078.2020.9118558.