

ALGORITMA HUFFMAN CODING UNTUK OPTIMALISASI DAN MENINGKATKAN RASIO KOMPRESI CITRA

Sapta Aji Sri Margiutomo
Program Studi Teknik Informatika
STMIK PPKIA Pradnya Paramita Malang
Jl. L.A Sucipto No. 249-A Malang
e-mail: smargiutomo@yahoo.com

ABSTRACT

The development of storage media on computer technology today is very fast, this technology has proven to be very effective and facilitate all activities related to the exchange of data or information. Information in the form of imagery generally contains a large amount of data, in terms of the number of pixels and the size of the number of bits used for coding the value of gray-level or color values of each pixel encoding. The larger the image, the greater the amount of data it contains, thus requiring storage media with large memory sizes. To solve the above problems, one solution is to compress or compress the image data so that it has a smaller amount of data than the original amount without reducing the content or quality of the image itself.

Keywords: *Algoritma Huffman, Image, Image Compression*

Pendahuluan

Suatu objek atau gambar yang terekam dalam kamera baik itu dalam bentuk foto atau video dalam bentuk digital dalam meyimpan pada besar media berbanding lurus dengan kualitas dari foto (gambar) atau video tersebut. Perkembangan teknologi kamera yang semakin baik dan semakin mini hal ini juga berdampak kepada media penyimpanan yang semakin kecil namun dituntut tetap dapat menyipkan foto atau video dalam jumlah yang cukup banyak. Hal ini berdampak semakin rumitnya metode penyimpanan.

Penyimpanan digital menggunakan metode Algoritma Huffman Coding, Aritmatic Coding dan JPEG. Algoritma Huffman Coding dalam kompresi gambar atau image dapat di kembangkan guna menghasilkan kompresi yang

maksimal tanpa mengurangi kualitas dari gambar tersebut

KAJIAN TEORI

Pengertian Citra Digital

Citra digital merupakan citra yang diperoleh sebagai hasil proses digitalisasi dari citra analog. Teknologi kamera foto dan video digital saat ini telah memungkinkan untuk mendapatkan citra atau video digital secara langsung. Citra adalah suatu representasi informasi visual pada suatu media antara lain lukisan, foto, film dan lain-lain. Informasi citra yang dapat ditangkap dan dianalisis oleh sistem visual mata manusia adalah berupa :

- informasi dasar (warna, bentuk, dan texture)
- informasi abstrak (senang, sedih, susah)
- informasi suasana (ulang tahun, pernikahan)

- Informasi ini direpresentasikan oleh nilai intensitas warna dari sekumpulan piksel yang membentuk setiap citra dan tersusun dalam bentuk larik dua dimensi.

Secara matematis, citra digital didefinisikan sebagai fungsi dua dimensi (pada umumnya), $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ menunjukkan nilai intensitas warna citra pada koordinat tersebut. Sebagai contoh pada gambar 1, pada gambar tersebut $f(x_i, y_j)$ merupakan nilai warna piksel pada posisi x_i dan y_j , dan nilai-nilai piksel dalam suatu citra umumnya dinyatakan dalam bentuk matriks, seperti terlihat pada persamaan 1 dibawah.

$$F = \{f(i, j)\} = \begin{bmatrix} f(0,0) & f(0,1) & \hat{\wedge} & f(0,N-1) \\ f(1,0) & f(1,1) & \hat{\wedge} & f(1,N-1) \\ \hat{\epsilon} & \hat{\epsilon} & \hat{\epsilon} & \hat{\epsilon} \\ f(M-1,0) & f(M-1,1) & \hat{\wedge} & f(M-1,N-1) \end{bmatrix}$$

Gambar 1 Matriks Citra



Gambar 2. Citra Digital

Di mana M dan N menyatakan jumlah baris dan kolom dari citra digital atau $M \times N$ merupakan dimensi citra. Setiap elemen $f(i,j)$

dalam matriks F dinyatakan sebagai picture element (pel) atau lebih dikenal dengan piksel.

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya : *scanner*, kamera digital, dan *handycam*.

Perkembangan teknologi elektronika saat ini telah meningkatkan kecepatan sampling camera digital hingga mampu menghasilkan citra dengan resolusi tinggi dalam kisaran 16 Mpixel. Sensor pada setiap kamera digital atau alat digitizer lainnya menggunakan tiga warna dasar yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*). Ini berarti bahwa setiap pixel untuk citra berwarna membutuhkan 24 bit (8 bit per warna). Untuk citra gray-level (hitam putih) umumnya hanya dibutuhkan 8 bit per pixel. Sebagai contoh, misalkan sebuah foto berwarna berukuran 3 inci x 4 inci diubah ke bentuk digital dengan tingkat resolusi sebesar 500 *dot per inch* (dpi), maka diperlukan $3 \times 4 \times 500 \times 500 = 3.000.000$ dot (pixel). Setiap pixel terdiri dari 3 *byte* dimana masing-masing *byte* merepresentasikan warna merah, hijau, dan biru. sehingga citra digital tersebut memerlukan *volume* penyimpanan sebesar : $3.000.000 \times 3 \text{ byte} + 1080 = 9.001.080 \text{ byte}$ setelah ditambahkan jumlah *byte* yang diperlukan untuk menyimpan *format (header)* citra. Citra tersebut tidak bisa disimpan ke dalam disket yang berukuran 1.4 MB. Selain itu, pengiriman citra berukuran 9 MB memerlukan waktu yang relatif lama (tergantung pada

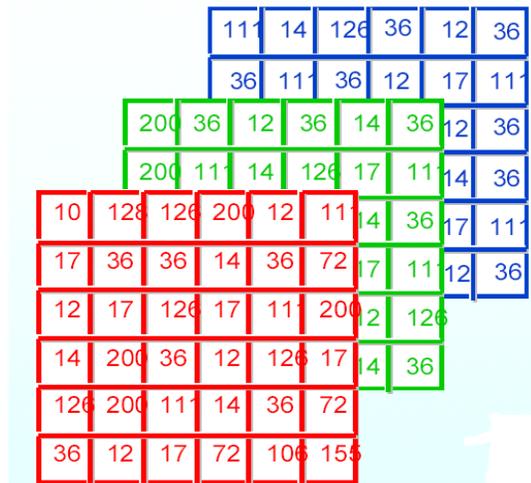
bandwidth media komunikasi yang dipakai). Untuk koneksi internet *dial-up* (56 kbps), pengiriman citra berukuran 9 MB memerlukan waktu 21 menit. Keterbatasan diatas disebabkan oleh beberapa hal diantaranya : pertama adalah *bandwidth* yang terbatas dan relatif mahal, kedua adalah kapasitas data multimedia yang mencapai puluhan ribu kali lipat dari kapasitas *bandwidth* jaringan komunikasi yang sering kita gunakan saat ini.

Solusi yang dapat dilakukan untuk mempercepat waktu komunikasi tanpa memperbesar *bandwidth* dan sekaligus meminimalkan penggunaan memori adalah pengembangan algoritma dan metode yang mampu mengkompres data citra sekecil mungkin dan proses kompresinyapun cepat dengan tetap menjaga kualitas informasi. Citra yang belum dikompres disebut citra mentah (*raw image*). Sementara citra hasil kompresi disebut citra terkompresi (*compressed image*).

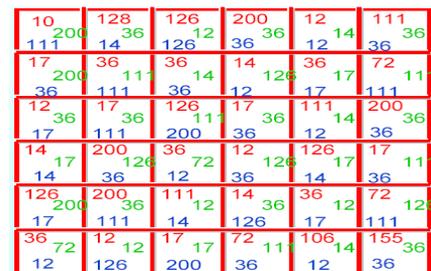
Colour Space

Colour space atau ruang warna merupakan representasi matematik dari sekumpulan warna. Pendefinisian warna dibuat berdasarkan standar dari Commision Internationale de l'Eclairage (CIE). Pada tahun 1931, lembaga ini secara khusus melakukan standarisasi terhadap seluruh warna yang dapat dilihat oleh mata manusia sebagai hasil kombinasi cahaya dengan panjang gelombang berbeda. Dari sekian banyak ruang warna yang ada, umumnya yang paling sering kali digunakan adalah yang memiliki keterkaitan dengan monitor

komputer yang memancarkan sinar dengan warna-warna dasar dengan panjang gelombang 700 nm untuk merah (Red), 546,1 nm untuk Hijau (Green) dan 435,8 nm untuk biru (Blue). Ketiga warna dasar ini lebih dikenal dengan RGB [referensi]. Melalui penggabungan masing-masing kanal dari intensitas warna dasar (R, G, B), tingkatan warna yang dihasilkan akan berbeda-beda hingga mencapai $2^{24} = 16777216$ jenis warna. Contoh representasi nilai warna dasar RGB sebuah citra terlihat pada gambar 3 dan gambar 4 dibawah.



Gambar 3 Tiga Komponen Warna RGB untuk masing-masing Piksel^[FR98]



Gambar 4 Gabungan Tiga Komponen Warna RGB^[FR98]

Untuk kebutuhan analisis warna citra, telah dikembangkan sejumlah ruang warna yang dapat digunakan diantaranya : XYZ, HSV, HSL, Lab, Luv, YCbCr LCH, HCL dll [referenesi]. Khusus untuk kompresi citra, ruang warna yang sering digunakan dan dianggap paling optimal untuk menghasilkan rasio kompresi tinggi dengan kualitas citra yang baik adalah YCbCr dan Luv [referenesi]. Ruang warna yang digunakan pada kompresi JPEG dan JPEG2000 adalah YCbCr atau Yuv, namun yang lebih banyak digunakan adalah ruang warna YCbCr. Kedua ruang warna ini dipilih karena mampu memisahkan antara komponen *luminance* (Y) dan dua komponen *chrominance* (Cb dan Cr atau u dan v). Keduanya dipilih dengan alasan bahwa komponen luminance dapat menyimpan semua informasi visual citra, dimana mata manusia sangat peka terhadap perubahan *luminance*. Sedang kedua *chrominance* menyimpan informasi warna, yang mana mata manusia tidak begitu peka terhadap perubahan warna. Berdasarkan pada kenyataan ini JPEG mengusulkan untuk melakukan pemampatan data yang lebih tinggi pada kedua komponen *chrominance* dari pada komponen *luminance*. Demikian pula dalam penelitian ini, untuk proses kompresi, penulis menggunakan transformasi warna dari RGB ke YCbCr dengan menggunakan rumus berikut ^[FR98]

$$Y = 0.257 R' + 0.504 G' + 0.098 B' + 16$$

$$C_b = -0.148 R' - 0.291 G' + 0.439 B' + 128$$

$$C_r = 0.439R' - 0.368G' - 0.071B' + 128$$

Sedang untuk proses rekonstruksi citra, penulis akan menggunakan transformasi balik seperti pada rumus berikut ^[FR98]

$$R' = 1.164(Y-16) + 1.596(Cr-128)$$

$$G' = 1.164(Y-16) - 0.813(Cr-128) - 0.391(Cb-128)$$

$$B' = 1.164(Y-16) + 2.018(Cr-128)$$

Kompresi Data Citra

Citra merupakan salah satu informasi multimedia yang memiliki jumlah data yang cukup besar. Kualitas sebuah citra selalu dikaitkan dengan resolusi dan kedalaman/tingkat intensitas warna. Resolusi citra menyatakan ukuran panjang kali lebar dari sebuah citra yang dinyatakan dalam satuan piksel. Kedalaman intensitas warna menyatakan jumlah bit yang digunakan untuk pengkodean setiap warna atau dinyatakan dalam satuan *bit/pixel*. Semakin tinggi resolusi sebuah citra berarti semakin banyak jumlah piksel dan semakin tinggi kedalaman intensitas berarti semakin banyak jumlah bit/pekselnya, hal ini mengakibatkan semakin baik kualitas citra tersebut. Namun, tingginya resolusi dan kedalaman intensitas menyebabkan semakin banyaknya jumlah bit yang diperlukan untuk menyimpan dan mentransmisikan data citra tersebut. Sebaliknya, kebanyakan aplikasi menginginkan representasi citra dengan kualitas citra yang baik dan penggunaan memori sekecil mungkin. Untuk meminimalkan jumlah data yang terkandung dalam citra, telah dikembangkan sejumlah algoritma kompresi citra.

Salah satu metode kompresi adalah pengurangan kedalaman bit ini. Di mana prosesnya adalah dengan cara mengurangi jumlah bit yang digunakan untuk merepresentasikan suatu piksel. Misalnya dengan mengurangi kedalaman bit dari 24 bit/piksel menjadi 16 atau 8 bit/piksel. Namun metode ini dapat mengurangi kualitas visual warna citra dan sangaj jarang digunakan.

Metode kompresi lain yang telah dikembangkan adalah pencodean sejumlah data rangkap (redundant) dalam citra dengan menggunakan jumlah bit yang lebih kecil. Hal ini dapat dilakukan karena umumnya dalam sebuah citra terdapat redundansi data yang relatif tinggi. Redundansi merupakan suatu keadaan dimana sejumlah piksel memiliki nilai yang sama atau memiliki kesamaan secara visual. Keadaan ini memungkinkan keseluruhan data yang sama dapat direpresentasikan secara lebih kompak. Metode ini sering digunakan untuk kompresi data spasial citra.

Selain secara spasial, kompresi data citra dapat dilakukan dalam domain frekuensi. Untuk hal ini dibutuhkan suatu proses transformasi (*image transformation*) dari domain spasial ke domain frekuensi. Umumnya, metode transformasi yang digunakan mampu memisahkan antara informasi global (ada pada frekuensi rendah atau lebih dikenal nilai rata-rata sejumlah piksel) dan informasi detail (ada pada frekuensi tinggi atau lebih dikenal nilai diferensial sejumlah piksel) dari setiap citra. Dengan demikian, cara ini akan lebih memudahkan untuk proses kompresi lebih lanjut. Transformasi yang populer digunakan antara lain

Discrete Cosine Transform (DCT) yang diadopsi dalam standar kompresi JPEG dan *Discrete Wavelet Transform* (DWT) yang digunakan dalam kompresi JPEG 2000 ^[WAT94].

Berdasarkan pada perkembangan teori, teknik kompresi citra dapat dibagi menjadi dua yaitu *lossy* dan *lossless* ^[G06]. *Lossless compression* (kompresi tanpa perubahan data) membuat kapasitas *file* sebuah gambar menjadi kecil dengan cara mengoptimalkan teknik pengkodean data redundan dari sebuah gambar yang asli. Metode ini cocok untuk kompresi citra dimana yang mengandung informasi penting dan tidak boleh rusak akibat kompresi tersebut, misalnya untuk citra medis. Teknik kompresi jenis ini adalah dengan mengaplikasikan langsung metode Huffman melalui pendekatan statistik di mana nilai warna atau keabuan yang sering muncul di dalam citra akan dikodekan dengan jumlah bit yang lebih sedikit, sedangkan nilai warna /keabuan yang frekuensi kemunculannya sedikit dikodekan dengan jumlah bit yang lebih panjang.

Lossy compression (kompresi dengan adanya perubahan data) adalah teknik kompresi dengan meminimalkan jumlah bit (mereduksi nilai) pada informasi detail dari *luminance* dan *chrominance* (warna) citra. Hal ini dapat lebih memperkecil jumlah bit yang dibutuhkan untuk pengkodean citra. *Lossy compression* umumnya dilakukan dalam domain frekuensi seperti DCT dan DWT. Kualitas citra kompresi seperti ini sangat tergantung pada seberapa besar pengurangan nilai data/informasi detail. Semakin

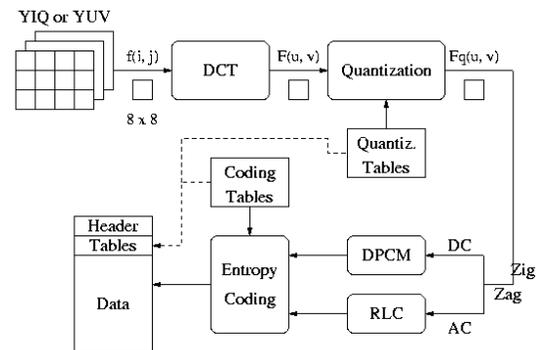
banyak nilai data detail yang hilang semakin rendah kualitas citra.

Metode Kompresi JPEG

JPEG (Joint Photographic Expert Group) secara umum telah digunakan sebagai standar international untuk kompresi citra berwarna digital sejak tahun 1992 (ISO/IEC JTC1 dan CCITT Rec 81). Seperti telah diketahui bahwa citra berwarna digital merupakan kumpulan dari piksel-piksel, dimana masing-masing piksel merupakan elemen-elemen vektor warna tiga dimensi (3-D) seperti RGB, YCbCr dsb. Citra dalam representasi warna YCbCr atau YUV adalah yang digunakan dalam kompresi JPEG.

Teknik kompresi JPEG dilakukan melalui beberapa tahapan yaitu: (1). Transformasi warna dan *down sampling* adalah proses pengkonversian data piksel dari ruang warna RGB ke ruang warna YCbCr dan kemudian dilakukan *down sampling*. Biasanya sampling dilakukan per blok 8x8 piksel. (2). DCT (Discrete Cosine Transform) yaitu proses transformasi data citra dari domain spasial ke domain frekuensi. Masukan proses DCT adalah menghitung dan memisahkan antara informasi global citra (informasi frakuensi rendah) dan informasi detil citra (informasi frekuensi tinggi). (3). Kuantisasi : proses untuk meminimalkan (menol-kan) nilai koefisien frekuensi tinggi hasil DCT. Hal ini dilakukan dengan asumsi bahwa perubahan nilai tersebut tidak terlalu mempengaruhi system visual mata manusia dalam menangkap informasi global pada citra tersebut. Sehingga penkodeannya dapat menggunakan jumlah

bit yang jauh lebih sedikit. Hal ini menyebabkan JPEG bersifat Lossy. (4). Entropy Coding : proses penggunaan algoritma entropy, misalnya Huffman atau Aritmatik untuk mengkodekan koefisien hasil proses DCT dan kuantisasi secara ziz-zag. Gambar (5) menunjukkan proses secara umum dari kompresi JPEG.



Gambar 5 Diagram proses kompresi JPEG ^[K03]

Kuantisasi

Tahap berikutnya adalah proses kuantisasi yaitu suatu proses untuk memperkecil nilai dari setiap elemen matrik AC. Hal ini dilakukan untuk dapat mereduksi jumlah bit dalam pengkodean binar elemen-elemen matriks tersebut. Proses kuantisasi dilakukan dengan cara setiap nilai dari elemen $F(i,j)$ di bagi dengan suatu nilai dari matriks kuantisasi $q(i,j)$. Gambar 6 mengilustrasikan contoh tabel kuantisasi untuk komponen *luminance* Y dan chrominnace Cb dan Cr. Adapun persamaan dari proses kuantisasi ini adalah pada persamaan berikut.

$$F '[i, j] = \text{round} \left(\frac{F [i, j]}{q[i, j]} \right)$$

Di mana, $F'[i,j]$ = Nilai piksel terkuantisasi pada posisi i, j , $F[i,j]$ = Nilai piksel sebelum kuantisasi (hasil DCT) pada posisi i, j .

$q[i,j]$ = Nilai matriks kuantisasi pada posisi i,j .

• The Luminance Quantization Table $q(u, v)$ The Chrominance Quantization Table $q(u, v)$

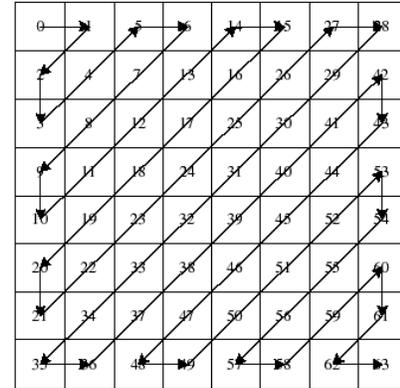
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

Gambar 6 Kuantisasi Luminance dan Chrominance^[H03]

Zig-Zag Coding

Untuk mempermudah pengkodean JPEG mengadopsi model pengkodean zig-zag scan untuk mengelompokkan komponen dari koefisien – koefisien terkuantisasi mulai dari frekuensi rendah hingga ke frekuensi tinggi. Semua komponen DC dikelompokkan kedalam satu vector, sedangkan komponen AC dikelompokkan kedalam satu vector yang lain. Pengelompokan ini dilakukan untuk dapat memisahkan komponen – komponen yang homogen atau yang mempunyai derajat keabuan sama yang tidak lain adalah informasi dengan rendah. Kelompok vektor yang mempunyai komponen frekuensi rendah ini disebut dengan komponen DC. Sedangkan kelompok yang mempunyai frekuensi tinggi disebut dengan komponen AC. Pemisahan kelompok ini selanjutnya digunakan untuk proses entropy coding pada proses berikutnya. Dalam standar JPEG terdapat dua algoritma yang

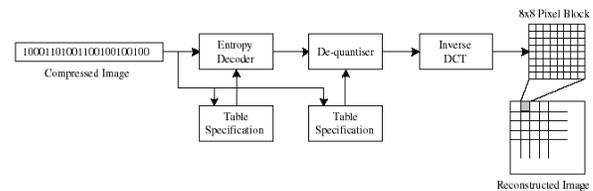
mungkin yang dapat digunakan untuk entropy coding ini, yaitu Huffman coding dan arithmetic coding. Adapun proses dari zig-zag scan ini dapat dilihat pada gambar 7 berikut. Pada gambar ini, scan dimulai dari posisi 0 ke 1, 2, 3 dan seterusnya sampai posisi 63.



Gambar 7 Proses zig-zag scan pada citra 8x8^[W91]

Dekompresi JPEG

Untuk melakukan proses dekompresi dari JPEG, maka prosesnya adalah kebalikan dari proses kompresi yang telah dijelaskan sebelumnya. Blok diagram dari proses dekompresi adalah seperti terlihat pada gambar di bawah ini.



Gambar 8 Blok Diagram proses Dekompresi JPEG^[W91]

Pada gambar 8 ini diawali dengan pengkonversian bitstream terkompresi kedalam proses rekonstruksi formasi zig-zag dari koefisien

DCT dan mendistribusikan kembali koefisien-koefisien tersebut ke dalam masing-masing blok. Blok-blok ini dibentuk oleh entropy decoder. Jika Huffman coding yang digunakan, kode-kode table harus identik dengan yang digunakan pada saat entropy encoded dari citra. Setelah itu proses dekuantisasi dilakukan dengan melakukan perkalian antara nilai dari koefisien DCT terhadap nilai element matriks kuantisasi $Q(i,j)$:

$$F_{Q'}(i, j) = F_Q(i, j) \cdot Q(i, j)$$

Kemudian dilanjutkan dengan proses transformasi invers DCT.

Rasio dan Kualitas Citra

Ada beberapa kriteria yang dapat digunakan untuk mengukur performance metode kompresi citra diantaranya adalah kompleksitas kompresi dan dekompresi, rasio kompresi dan kualitas citra rekonstruksi (decompression fidelity). Kompleksitas kompresi dilihat dari berapa jumlah operasi aritmatika yang dilakukan atau dibutuhkan untuk melakukan proses kompresi ataupun proses rekonstruksi. Nilai rasio kompresi adalah perbandingan antara citra asli dengan resolusi $M \times N$ piksel dan B bit/piksel dan citra yang dimampatkan dengan jumlah bit $\|c\|$ adalah,

$$\text{Rasio Kompresi} = \frac{M \cdot N \cdot B}{\|c\|}$$

Sedang untuk mengukur kualitas citra terkompresi yang bersifat *lossy compression* lebih sering digunakan *peak signal-to-noise ratio* atau disingkat PSNR dan selalu dinyatakan dalam skala logarithmic decibel. Signal yang dimaksud adalah

data citra asli dan noise adalah error yang terjadi akibat kompresi. Secara matematis penghitungan PSNR diuraikan oleh persamaan berikut:

$$PSNR = 20 \cdot \text{Log} \left(\frac{b}{MSE} \right)$$

Dengan MSE (mean squared error) :

$$MSE = \sqrt{\frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M (f_{ij} - f'^i_{ij})^2}$$

dimana f dan f' masing-masing menyatakan nilai piksel citra asli dan nilai piksel citra setelah direkonstruksi, dan b adalah nilai piksel maksimum dari citra rekonstruksi (decompression). Jika citra dengan kedalaman piksel 8 bit, maka nilai piksel maksimumnya adalah 255. Pada persamaan (5), terlihat bahwa nilai PSNR berbanding terbalik dengan nilai MSE. Semakin besar nilai MSE, maka semakin kecil nilai PSNR akibatnya kualitas citra semakin jauh dari kualitas citra aslinya. Sebaliknya, semakin kecil nilai MSE maka semakin besar nilai PSNR dan kualitas citra semakin baik dan bila $MSE = 0$ maka $PSNR = \infty$ dan kualitas citra kompresi sama dengan kualitas citra aslinya.

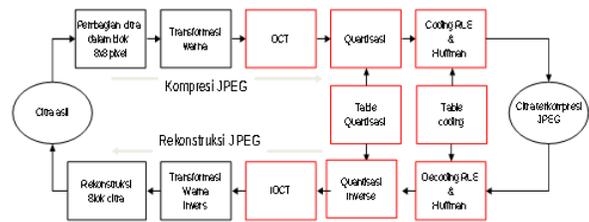
Umumnya nilai PSNR untuk kompresi citra dan video yang bersifat lossy terletak antara 30 dan 50 dB, sedang kualitas citra yang masih dianggap baik untuk kebutuhan transmisi wireless ada pada kisaran 20 dan 25 dB ^{[NV06], [LC07]}.

Optimasi Algoritma Kompresi Citra JPEG

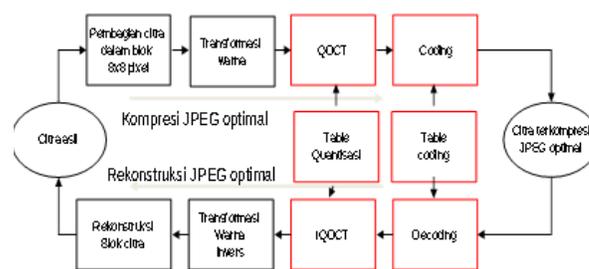
Berdasarkan pada hasil analisis referensi yang telah dilakukan menunjukkan bahwa

algoritma kompresi JPEG yang telah dikembangkan sudah berfungsi dengan baik dan terpakai pada semua teknologi perangkat lunak dan perangkat keras pendukung teknologi informasi termasuk pada kamera foto dan video digital. Namun setelah Penulis mempelajari secara seksama algoritma ini, ternyata masih ada peluang untuk lebih mengoptimalkan kinerjanya, baik dari segi waktu kompresi, rasio kompresi ataupun kualitas hasil kompresi. Bagian dari algoritma JPEG yang masih dapat dioptimalkan adalah pada proses transformasi, proses kuantisasi dan proses koding. Namun dalam penelitian ini Penulis hanya akan mengoptimalkan pada proses transformasi DCT (*Discrete Cosine Transform*) dan kuantisasinya.

Gambar 9 memperlihatkan skema umum proses kompresi JPEG. Pada gambar ini, suatu citra RGB dengan resolusi $M \times N$ piksel akan dibagi menjadi beberapa blok citra dengan ukuran 8×8 piksel dan ditransformasikan kedalam color plane YCbCr. Kemudian masing-masing blok citra dalam domain spasial ini ditransformasikan kedalam domain frekuensi melalui proses *Discrete Cosine Transform* (DCT). Hasil matriks DCT selanjutnya dilakukan proses *Quantization* dengan menggunakan matriks kuantisasi Q . Proses koding merupakan proses pengkodean nilai-nilai hasil kuantisasi kedalam bit stream sesuai dengan format data kompresi JPEG. Terlihat pada gambar ini proses DCT dan kuantisasi yang dilakukan secara terpisah.



Gambar 9 Bagan proses kompresi JPEG Standar



Gambar10 Bagan optimalisasi kompresi JPEG

Gambar 10 menunjukkan skema yang diusulkan untuk mengoptimalkan hasil kompresi JPEG. Dapat dilihat bahwa proses DCT dan kuantisasi digabung menjadi satu proses yang kemudian disebut QDCT. Hal ini dimaksudkan untuk mempercepat proses kompresi dan juga diharapkan dapat meningkatkan kualitas dan rasio kompresi. Uraian tentang proses DCT dan kuantisasi pada JPEG serta QDCT yang diusulkan, diuraikan dalam sub-bab berikut ini.

Proses DCT pada JPEG

DCT adalah proses transformasi citra dari domain spasial ke domain frekuensi. Proses perhitungan DCT pada model kompresi JPEG standar dipresentasikan pada persamaan berikut.

$$F(u, v) = DCT(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

Implementasi 1D :

$$DCT(u, y)_x = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

$$DCT(u, v)_y = \sqrt{\frac{2}{N}} C(v) \sum_{y=0}^{N-1} DCT(u, y)_x \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

$$[DQ]^{-1} = \begin{bmatrix} 0.3536 & 0.4904 & 0.4619 & 0.4157 & 0.3536 & 0.2778 & 0.1913 & 0.0975 \\ 0.3536 & 0.4157 & 0.1913 & -0.0975 & -0.3536 & -0.4904 & -0.4619 & -0.2778 \\ 0.3536 & 0.2778 & -0.1913 & -0.4904 & -0.3536 & 0.0975 & 0.4619 & 0.4157 \\ 0.3536 & 0.0975 & -0.4619 & -0.2778 & 0.3536 & 0.4157 & -0.1913 & -0.4904 \\ 0.3536 & -0.0975 & -0.4619 & 0.2778 & 0.3536 & -0.4157 & -0.1913 & 0.4904 \\ 0.3536 & -0.2778 & -0.1913 & 0.4904 & -0.3536 & -0.0975 & 0.4619 & -0.4157 \\ 0.3536 & -0.4157 & 0.1913 & 0.0975 & -0.3536 & 0.4904 & -0.4619 & 0.2778 \\ 0.3536 & -0.4904 & 0.4619 & -0.4157 & 0.3536 & -0.2778 & 0.1913 & -0.0975 \end{bmatrix}$$

Dari kedua rumus ini dapat dilihat bahwa proses ini tidak lain adalah merupakan perkalian antara matriks citra dengan matriks persamaan cosinus. Sehingga persamaan dapat dituliskan dalam bentuk perkalian matriks sebagai berikut :

$$[DCT(u,y)_x] = [DC] * [f(x,y)]$$

$$[DCT(u,v)] = [DCT(u,y)_x] * [DC]^T$$

Sehingga dalam prosesnya JPEG melakukan perhitungan matriks DC (Discrete cosine) diawal program. Untuk kompresi citra, ukuran matriks ini telah dibuat standar yaitu dengan ukuran 8x8 piksel (N=8). Dengan demikian, matriks DC dapat dihitung berdasarkan pada persamaan diperoleh matriks DC berikut :

$$[DC]_{8x8} = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4903 & -0.2778 & 0.2778 & 0.4903 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4903 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4903 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4903 & 0.4903 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

Karena matriks $[DC]_{8x8}$ ini akan digunakan untuk proses kompresi berarti matriks ini harus mempunyai matriks invers yang nantinya akan digunakan untuk proses dekompresi. Invers dari matriks $[DC]_{8x8}$ adalah :

Agar data citra asli setelah kompresi dapat diperoleh kembali maka perlu pembuktian bahwa perkalian antara matriks $[DC]_{8x8}$ dengan matriks inversnya $[DC]_{8x8}^{-1}$ menghasilkan matriks identitas. Atau secara matematis harus memenuhi syarat berikut :

$$[DC]_{8x8} * [DC]_{8x8}^{-1} = [I] \text{ dan } [DC]_{8x8}^{-1} * ([DC]_{8x8} * f(x,y)) = f(x,y)$$

Untuk membuktikan hal ini maka bila matrik $[DC]$ dan $[DC]^{-1}$ diatas diperkalikan maka diperoleh nilai matrik sebagai berikut :

$$[DC]_{8x8} * [DC]_{8x8}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Sedang bila matriks $[DC]_{8x8}$ dikalikan dengan matriks citra berikut :

$$f = \begin{bmatrix} 164 & 63 & 75 & 95 & 157 & 99 & 91 & 51 \\ 120 & 135 & 55 & 75 & 116 & 67 & 84 & 49 \\ 99 & 132 & 60 & 54 & 100 & 75 & 72 & 58 \\ 64 & 150 & 113 & 50 & 81 & 138 & 70 & 53 \\ 110 & 130 & 162 & 60 & 76 & 109 & 86 & 65 \\ 97 & 82 & 179 & 81 & 74 & 113 & 64 & 56 \\ 61 & 70 & 149 & 159 & 62 & 138 & 67 & 68 \\ 54 & 63 & 115 & 187 & 95 & 141 & 51 & 72 \end{bmatrix}$$

maka diperoleh hasil sebagai berikut :

$$[DC]_{8 \times 8} * f(x, y) = \begin{bmatrix} 271.8826 & 291.6815 & 321.0265 & 269.0541 & 269.0541 & 311.1270 & 206.8287 & 166.8772 \\ 74.5400 & 42.8629 & -96.5309 & -88.5135 & 80.5642 & -57.8407 & 27.3448 & -18.8122 \\ 17.4552 & -72.8608 & -45.9618 & 98.3965 & 44.6496 & 0.0192 & -3.5970 & 2.8837 \\ 51.7730 & -36.4158 & 64.5080 & -14.0354 & 6.3690 & 0.0440 & 15.4925 & -4.5244 \\ 5.3033 & -4.5962 & 7.7782 & 8.1317 & 20.1525 & 33.2340 & 3.8891 & 3.5355 \\ -17.3055 & -18.6836 & 3.0066 & 8.8457 & -4.6437 & 31.5005 & -3.0967 & -1.3098 \\ 15.3481 & -25.3092 & -0.0962 & -12.8213 & 16.3297 & -9.1924 & -9.6079 & -0.4291 \\ 17.7302 & -7.0771 & -3.2332 & 8.0389 & -0.5955 & -14.3935 & 10.3516 & 9.9457 \end{bmatrix}$$

Selanjutnya bila hasil matriks ini dikalikan dengan matriks invers $[DC]_{8 \times 8}^{-1}$ maka diperoleh hasil sebagai berikut :

$$[DC]_{8 \times 8}^{-1} * ([DC]_{8 \times 8} * f(x, y)) = \begin{bmatrix} 164 & 63 & 75 & 95 & 157 & 99 & 91 & 51 \\ 120 & 135 & 55 & 75 & 116 & 67 & 84 & 49 \\ 99 & 132 & 60 & 54 & 100 & 75 & 72 & 58 \\ 64 & 150 & 113 & 50 & 81 & 138 & 70 & 53 \\ 110 & 130 & 162 & 60 & 76 & 109 & 86 & 65 \\ 97 & 82 & 179 & 81 & 74 & 113 & 64 & 56 \\ 61 & 70 & 149 & 159 & 62 & 138 & 67 & 68 \\ 54 & 63 & 115 & 187 & 95 & 141 & 51 & 72 \end{bmatrix} = f(x, y)$$

Hasil pembuktian diatas menunjukkan bahwa matriks $[DC]$ dapat digunakan untuk proses transformasi dari domain spasial ke domain frekuensi dan matriks $[DC]^{-1}$ dapat digunakan untuk proses transformasi dari domain frekuensi ke domain spasial. Matriks $[DC]$ dan $[DC]^{-1}$ inilah yang digunakan dalam algoritma kompresidan dekompresi JPEG.

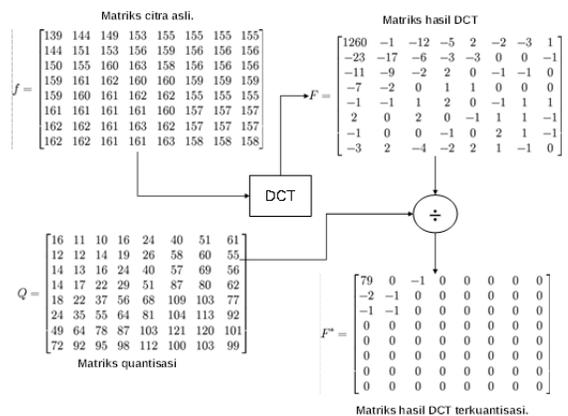
Proses Kuantisasi pada JPEG

Proses kuantisasi dilakukan terhadap hasil keluaran dari proses DCT. Sasaran dari proses ini adalah untuk meminimalkan besarnya nilai-nilai pada matriks hasil DCT, sehingga matriks ini dapat dikodekan dengan jumlah bit yang lebih kecil. Andaikan matriks pada bagian kanan-atas adalah merupakan hasil DCT citra pada matriks di kiri-atas. Kuantisasi dilakukan dengan membagi nilai setiap elemen matriks hasil DCT dengan setiap elemen matriks kuantisasi pada posisi yang sama dan kemudian dilakukan pembulatan. Secara

matematis proses kuantisasi dinyatakan dalam persamaan berikut :

$$F' [i, j] = \text{round} \left(\frac{F[i, j]}{Q[i, j]} \right)$$

Dimana F adalah matriks hasil DCT dua dimensi dan Q adalah matriks kuantisasi. Matriks hasil kuantisasi ditunjukkan oleh matriks F' pada bagian kanan-bawah.



Gambar 11 Proses DCT dan Kuantisasi Kompresi JPEG Standar

Operasi penghitungan matriks $[DCT(u,v)]_{8 \times 8} = F$ ini membutuhkan $2 \times (8 \times 8 \times 8) = 1024$ atau $2 \times N^3$ operasi perkalian dan $2 \times (8 \times 8 \times 7) = 896$ atau $2 \times N^2 \times (N-1)$ operasi penjumlahan. Sedang untuk proses kuantisasi membutuhkan $8 \times 8 = 64$ atau N^2 operasi pembagian. Dengan demikian bila sebuah citra berukuran $K \times L$ piksel (K adalah lebar dan L adalah tinggi citra), maka proses DCT dan kuantisasi untuk citra ini membutuhkan :

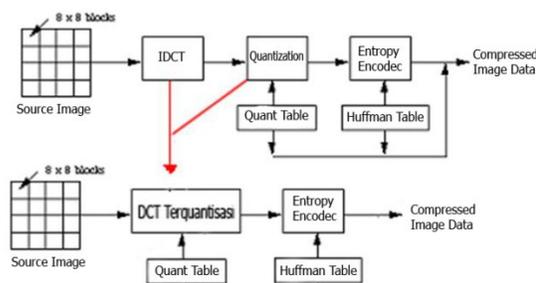
$K \times L \times (2N + 1)$ operasi perkalian/pembagian dan $K \times L \times 2(N-1)$ operasi penjumlahan/ pengurangan. Jumlah kedua operasi ini sangatlah besar,

sehingga muncul suatu pertanyaan dan sekaligus yang menjadi permasalahan dalam penelitian ini adalah mungkinkah jumlah operasi perkalian dan penjumlahan dikurangi sebanyak mungkin tanpa mempengaruhi kualitas citra hasil kompresi ?

Usulan Matriks Transformasi QDCT

Seperti telah diuraikan di atas bahwa model yang diusulkan dalam penelitian ini adalah penggabungan proses DCT dan kuantisasi yang ada pada algoritma kompresi maupun algoritma rekonstruksi JPEG standar. Adapun langkah penelitian yang akan dilakukan sbb :

1. Membuat algoritma DCT-terkuantisasi yang dapat melakukan secara bersamaan proses DCT dan kuantisasi, seperti terlihat dalam diagram dibawah ini :



Gambar 12 Diagram DCT-Terkuantisasi

Untuk dapat menggabungkan proses DCT dan proses kuantisasi menjadi satu proses DCT terkuantisasi, maka penulis mengusulkan setiap elemen [DC] dibagi dengan elemen pada posisi yang sama pada matriks kuantisasi, yang selanjutnya disebut [QDC]. Usulan transformasi QDCT (DCT-Terkuantisasi) :

Implementasi I D :

$$QDCT(u, y)_x = \sqrt{\frac{2}{N}} C(u) Q(u, y) \sum_{x=0}^{N-1} f(x, y) \cos\left(\frac{(2x+1)i\pi}{2N}\right)$$

$$F(u, v) = QDCT(u, v) = \sqrt{\frac{2}{N}} C(v) Q(u, v) \sum_{y=0}^{N-1} QDCT(i, y)_x \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

Dimana $Q(u,y) = Q(u,v)$ adalah konstanta, sehingga kedua persamaan di atas dapat disederhanakan menjadi perkalian matriks sebagai berikut

$$[QDCT(u,y)_x] = [QDC] \times [f(x,y)]$$

$$[F(u,v)] = [QDCT(u,v)] = [F(u,y)_x] \times [QDC]^T$$

Sebagai contoh untuk matriks [QDC] dibawah adalah merupakan hasil kuantisasi dari matrik [DC] oleh matriks kuantisasi save 10 dari perangkat lunak Photoshop CS3, yaitu dengan cara setiap elemen matriks [DC] dibagi dengan setiap elemen matriks kuantisasi yang letaknya bersesuaian.

Dapat ditunjukkan bahwa [QDC] dan $[QDC]^T$ masing - masing mempunyai invers, dan $[QDC] \times [QDC]^{-1} = I$ (matriks identitas)

$$[QDC]^T \times [[QDC]^T]^{-1} = I \text{ (matriks identitas)}$$

[QDC]=

0.2500	0.2500	0.2041	0.1768	0.1581	0.1443	0.1250	0.1066
0.3468	0.2940	0.1964	0.0488	-0.0436	-0.1050	-0.1386	-0.1479
0.2667	0.1353	-0.1105	-0.2066	-0.1746	-0.0638	0.0577	0.1334
0.2079	-0.0488	-0.2193	-0.1050	0.0926	0.1479	0.0282	-0.1200
0.1581	-0.1581	-0.1336	0.1179	0.1066	-0.1021	-0.1021	0.1021
0.1134	-0.1854	0.0325	0.1253	-0.1200	-0.0282	0.1416	-0.0802
0.0676	-0.1540	0.1393	-0.0552	-0.0552	0.1334	-0.1334	0.0552
0.0294	-0.0838	0.1200	-0.1416	0.1416	-0.1200	0.0802	-0.0282

[QDC]=

0.4830	0.6962	0.7540	0.8127	0.8494	0.7616	0.5662	0.3004
0.4897	0.6125	0.3398	-0.1758	-0.8031	-1.2965	-1.3338	-0.8390
0.5527	0.4683	-0.4049	-1.1570	-0.8623	0.3871	1.4908	1.3437
0.6719	0.1921	-1.0922	-0.6723	1.1050	1.1664	-0.8608	-1.7777
0.8325	-0.2355	-1.1751	0.8961	1.0112	-1.4999	-0.6032	1.7577
0.9688	-0.7620	-0.4545	1.4716	-1.2916	-0.2566	1.6679	-1.4341
1.0692	-1.2194	0.6984	0.1149	-1.1627	1.7811	-1.5147	1.0108
1.1267	-1.4736	1.5606	-1.6195	1.3227	-0.9372	0.7857	-0.2124

$[QDC]^T =$

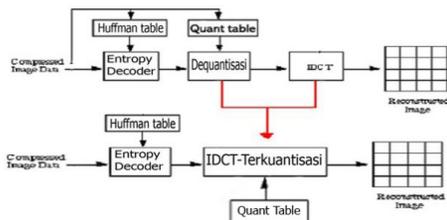
0.2500	0.3468	0.2667	0.2079	0.1581	0.1134	0.0676	0.0294
0.2500	0.2940	0.1353	-0.0488	-0.1581	-0.1854	-0.1540	-0.0838
0.2041	0.1964	-0.1105	-0.2193	-0.1336	0.0325	0.1393	0.1200
0.1768	0.0488	-0.2066	-0.1050	0.1179	0.1253	-0.0552	-0.1416
0.1581	-0.0436	-0.1746	0.0926	0.1066	-0.1200	-0.0552	0.1416
0.1443	-0.1050	-0.0638	0.1479	-0.1021	-0.0282	0.1334	-0.1200
0.1250	-0.1386	0.0577	0.0282	-0.1021	0.1416	-0.1334	0.0802
0.1066	-0.1479	0.1334	-0.1200	0.1021	-0.0802	0.0552	-0.0282

$[QDC]^T)^{-1} =$

0.4830	0.4897	0.5527	0.6719	0.8325	0.9688	1.0692	1.1267
0.6962	0.6125	0.4683	0.1921	-0.2355	-0.7620	-1.2194	-1.4736
0.7540	0.3398	-0.4049	-1.0922	-1.1751	-0.4545	0.6984	1.5606
0.8127	-0.1758	-1.1570	-0.6723	0.8961	1.4716	0.1149	-1.6195
0.8494	-0.8031	-0.8623	1.1050	1.0112	-1.2916	-1.1627	1.3227
0.7616	-1.2965	0.3871	1.1664	-1.4999	-0.2566	1.7811	-0.9372
0.5662	-1.3338	1.4908	-0.8608	-0.6032	1.6679	-1.5147	0.7857
0.3004	-0.8390	1.3437	-1.7777	1.7577	-1.4341	1.0108	-0.2124

Jadi yang diusulkan dalam penelitian adalah mencari satu formula DCT terkuantisasi (selanjutnya disebut QDCT) . Formula ini akan melakukan proses transformasi DCT sekaligus kuantisasi. Dengan demikian proses QDCT akan mempercepat proses kompresi karena mampu mengurangi proses pembagian.

2. Mencari satu formulasi invers DCT-terkuantisasi yang dapat melakukan secara bersamaan proses dekuantisasi dan DCT invers.



Gambar 13 Diagram QIDCT (IDCT-Terkuantisasi)

Sebelum mencari formulasi tersebut, akan dijelaskan proses perhitungan IDCT dan dekuantisasi pada model kompresi JPEG standar. Untuk proses de-kuantisasi menggunakan

persamaan. Sedangkan proses transformasi invers DCT. Menggunakan formula :
IDCT (invers DCT)

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(v)C(u) F(u, v) \cos\left(\frac{(2x+1)j\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right)$$

Implementasi 1D :

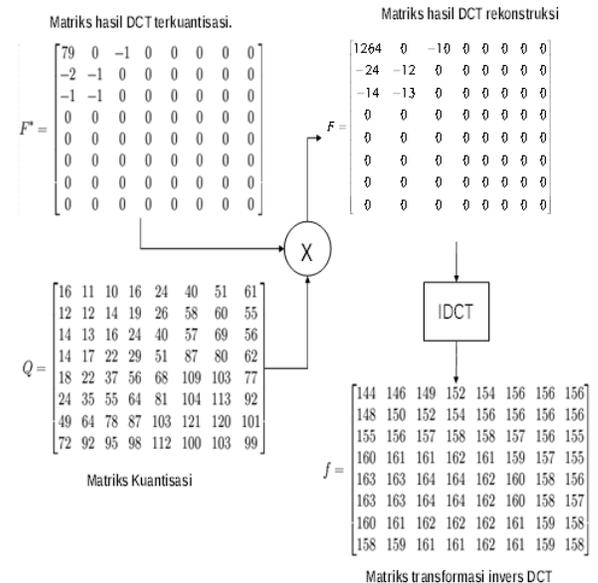
$$F^{-1}(u, y)_x = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} F(u, v) \cos\left(\frac{(2x+1)j\pi}{2N}\right)$$

$$f(x, y) = \sqrt{\frac{2}{N}} C(j) \sum_{y=0}^{N-1} F^{-1}(u, y)_x \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

Dimana,

$$C(z) = \begin{cases} 1/\sqrt{2} & \text{untuk } z = 0 \\ 1 & \text{untuk } z \text{ lainnya} \end{cases}$$

Adapun proses perhitungan IDCT dan Dekuantisasi dapat dijelaskan dalam diagram berikut :



Gambar 14 Diagram Proses IDCT dan Dekuantisasi Kompresi JPEG Standar

Untuk dapat menggabungkan proses dekuantisasi dan invers DCT menjadi satu proses invers DCT terkuantisasi, maka penulis mengusulkan setiap elemen [DC] dikali dengan elemen pada posisi yang sama pada matriks kuantisasi, yang selanjutnya disebut invers DCT Terkuantisasi atau [IQDC].

Usulan transformasi QIDCT (IDCT-Terkuantisasi) Implementasi 1 D :

$$F^{-1}(u, y)_x = \sqrt{\frac{2}{N}} C(u) Q(u, v)^{-1} \sum_{x=0}^{M-1} F(u, v) \cos\left(\frac{(2x+1)i\pi}{2N}\right)$$

$$f(x, y) = \sqrt{\frac{2}{N}} C(v) Q(u, y)^{-1} \sum_{y=0}^{M-1} F^{-1}(u, y)_x \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

Dimana $Q(u, y)^{-1} = Q(u, v)^{-1} = \text{konstanta}$

Dari usulan formula DCT yang diuraikan sebelumnya dalam bentuk matriks, maka diperoleh QIDCT (invers DCT-terkuantisasi) :

$$[\text{QDCT}^{-1}(u, y)_x] = [[\text{QDC}]^T]^{-1} \times [F(u, v)]$$

$$[f(x, y)] = [\text{QDC}]^{-1} \times [\text{QDCT}^{-1}(u, y)]$$

3. Perancangan dan implementasi algoritma DCT dan Kuantisasi dan DCT terkuantisasi

Dari uraian sebelumnya, maka proses kompresi JPEG standar atau DCT dan Kuantisasi dalam bentuk algoritma, yaitu:

Algoritma: Kompresi JPEG Standar

Langkah 1: //Transformasikan Image warna RGB Ix ke YCbCr//
Langkah 2: //Pisahkan masing-masing matriks R, G, B dari matriks I//

Langkah 3: //Bangkitkan dua matriks kuantisasi//
MQ1=

$$= \begin{bmatrix} 2 & 2 & 3 & 4 & 5 & 6 & 8 & 11 \\ 2 & 2 & 2 & 4 & 5 & 7 & 8 & 11 \\ 3 & 2 & 3 & 5 & 7 & 9 & 11 & 12 \\ 4 & 4 & 5 & 7 & 9 & 11 & 12 & 12 \\ 5 & 5 & 7 & 9 & 11 & 12 & 12 & 12 \\ 6 & 7 & 9 & 11 & 12 & 12 & 12 & 12 \\ 8 & 9 & 11 & 12 & 12 & 12 & 12 & 12 \\ 11 & 11 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

MQ2=

$$= \begin{bmatrix} 3 & 3 & 7 & 13 & 15 & 15 & 15 & 15 \\ 3 & 4 & 7 & 13 & 14 & 12 & 12 & 12 \\ 7 & 7 & 13 & 14 & 12 & 12 & 12 & 12 \\ 13 & 13 & 14 & 12 & 12 & 12 & 12 & 12 \\ 15 & 14 & 12 & 12 & 12 & 12 & 12 & 12 \\ 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

Langkah 4: //Mengambil matriks 8x8 dari citra red//

Langkah 5: //Lakukan proses DCT//

Langkah 6: //Lakukan Proses DCT 8x8 & proses kuantisasi//

Langkah 7: //Lakukan /Proses DCT 8x8 dan kuantisasi//

Langkah 8: //Lakukan Proses Rekonstruksi //

Langkah 9: //Lakukan Proses Rekonstruksi

Langkah 10: // Hitung kualitas kompresi (fidelity) //

Algoritma: Kompresi JPEG-DCT-Terkuantisasi

Langkah 1: //Transformasikan Image warna RGB Ix ke YCbCr//

Langkah 2: //Pisahkan masing-masing matriks R, G, B dari matriks I//

Langkah 3: //Bangkitkan dua matriks kuantisasi//

MQ1=

$$= \begin{bmatrix} 2 & 2 & 3 & 4 & 5 & 6 & 8 & 11 \\ 2 & 2 & 2 & 4 & 5 & 7 & 8 & 11 \\ 3 & 2 & 3 & 5 & 7 & 9 & 11 & 12 \\ 4 & 4 & 5 & 7 & 9 & 11 & 12 & 12 \\ 5 & 5 & 7 & 9 & 11 & 12 & 12 & 12 \\ 6 & 7 & 9 & 11 & 12 & 12 & 12 & 12 \\ 8 & 9 & 11 & 12 & 12 & 12 & 12 & 12 \\ 11 & 11 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

MQ2=

$$= \begin{bmatrix} 3 & 3 & 7 & 13 & 15 & 15 & 15 & 15 \\ 3 & 4 & 7 & 13 & 14 & 12 & 12 & 12 \\ 7 & 7 & 13 & 14 & 12 & 12 & 12 & 12 \\ 13 & 13 & 14 & 12 & 12 & 12 & 12 & 12 \\ 15 & 14 & 12 & 12 & 12 & 12 & 12 & 12 \\ 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

Langkah 4: //Mengambil matriks 8x8 dari citra red//
 Langkah 5: //Lakukan proses Transformasi DCT terkuantisasi//
 Langkah 6: //Lakukan Proses DCT 8x8
 Langkah 7: //Lakukan Proses DCT 8x8 baris kedua dan ketiga dan proses kuantisasi//
 Langkah 8:
 //Lakukan Proses $iDCTX \circ transpose(iDCTY)$;
 end
 end

Langkah 9: //Lakukan Proses Rekonstruksi //
 Langkah 10: // Hitung kualitas kompresi (fidelity)

Eksperimen dan Hasil Eksperimen

Untuk mengetahui apakah metode yang diusulkan dapat bekerja dengan baik, maka perlu dilakukan evaluasi terhadap implementasi usulan metode ini agar hasil kompresi yang diinginkan adalah citra hasil kompresi yang mempunyai ukuran file sekecil mungkin atau dengan rasio kompresi besar, namun tetap mempunyai kualitas citra yang baik. Kualitas citra dapat diukur melalui nilai PSNR menggunakan persamaan (2.10). Semakin besar nilai PSNRnya maka kualitas citranya semakin baik.

Lingkungan Eksperimen

Seperti telah diuraikan diatas, untuk mendapatkan data dilakukan pengujian terhadap beberapa citra, diantaranya 3 citra yang sering digunakan sebagai referensi untuk uji coba setiap algoritma kompresi citra yaitu : Baboon, Lena, dan Pappers. Ketiga citra tersebut sering digunakan karena mempunyai karakteristik yang berbeda-beda. Citra baboon memiliki banyak variasi warna, citra lena memiliki sedikit variasi warna namun mengandung region yang memiliki tingkatan

homogenitas yang berbeda-beda., dan citra pappers memiliki sedikit variasi warna serta tingkat homogenitas tinggi, namun memiliki tepi objek yang sangat tajam. Ukuran dari ketiga citra tersebut sama : 512x512 piksel (769 KB) dalam format bmp seperti terlihat di tabel dibawah ini

Tabel 1 Jenis, besar piksel, ukuran dan format file

JENIS CITRA	UKURAN CITRA	UKURAN FILE	FORMAT
Image1	286 x 380	330 KB	BMP
Image2	857 x 553	1390 KB	BMP
Image3	599 x 403	709 KB	BMP
Image4	1200 x 1600	5626 KB	BMP
Image5 (baboon)	512 x 512	769 KB	BMP
Image6	640 x 426	799 KB	BMP
Image7	366 x 332	357 KB	BMP
Image8	458 x 640	1146 KB	BMP
Image9	512 x 384	577 KB	BMP
Image10	1280 x 720	2701 KB	BMP
Image11	1153 x 621	2099 KB	BMP
Image12 (lena)	512 x 512	769 KB	BMP

Image13	709 x 473	984 KB	BMP
Image14	591 x 480	833 KB	BMP
Image15 (pappers)	512 x 512	769 KB	BMP

Cara Pengambilan Data

Dalam melakukan eksperimen ini setiap citra dengan format bmp tersebut dimampatkan dengan menggunakan perangkat lunak Adobe Photoshop CS3, sehingga akan didapatkan citra hasil kompresi yang tersimpan dengan format jpg sesuai dengan kualitas yang diinginkan. Kualitas yang dipilih dalam memampatkan citra tersebut menentukan jenis matriks kuantisasi yang akan digunakan dalam eksperimen, jenis matriks kuantisasi tersebut dinamakan save. Jadi matriks kuantisasi yang digunakan dalam eksperimen baik menggunakan dct standar ataupun dct terkuantisasi adalah matriks kuantisasi yang digunakan adobe photoshop untuk memampatkan citra tersebut. Pengambilan data dilakukan untuk setiap citra dengan menggunakan rasio yang sama untuk mendapatkan psnr dan waktu kompresi baik dct standar JPEG ataupun dct terkuantisasi. Hal itu dilakukan secara berulang-ulang untuk rasio yang berbeda-beda sehingga diperoleh data dibutuhkan.

KESIMPULAN DAN SARAN

Kesimpulan

Setelah tahap eksperimen selesai dilakukan dengan menggunakan program dct standar dan Program DCT-Terkuantisasi, selanjutnya adalah

melakukan perbandingan hasil kompresi dari kedua program tersebut. Dalam melakukan eksperimen ini setiap citra dengan format bmp dimampatkan dengan menggunakan Adobe Photoshop CS3 menjadi format jpg dengan tujuan untuk menghitung rasio kompresi agar matriks kuantisasi yang akan digunakan dapat diketahui. Parameter yang akan dianalisis adalah Waktu Eksekusi, Rasio Kompresi, Kualitas citra hasil kompresi dan kompleksitas waktu algoritma.

Saran

Untuk dapat mencari hasil yang optimal bagi peneliti selanjutnya, Metode Algoritma Huffman coding dapat di bandingkan dengan Arimatic coding sehingga dapat diperoleh perbandingan yang lebih baik lagi.

DAFTAR PUSTAKA

- Blelloch, G.E., 2001. *Introduction to Data Compression*. Carnegie Mellon University, Computer Science Department.
- HM, Jogyanto. 1999. *Analisis Dan Desain Sistem Informasi*. Yogyakarta : Andi Offset.
- Imadewira., 2009. *Definisi dan Pengenalan Algoritma*. (<http://kuliah.imadewira.com/definisi-dan-pengenalan-algoritma/> tanggal 17 April 2011 jam 8:39).
- Mengyi, Ida Pu., 2006. *Fundamental Data Compression*. London, Licensing Agency Ltd.
- Nelson, M., Gailly, J.L., 1996. *The Data Compression Book, Second Edition*. New York, M&T Books.

- R.C Gonzales., 1987. *“Digital Image Processing”*
Second Edition. Pp. 255-330, Addison
Wesley Publishing Company.
- Sarifuddin Madenda, 1994. *“Kompresi Citra
Gray-Level dengan Metode Block
Coding”*. Proceeding Workshop on ECI.
HLM 8-100. ITB. March 3-4.
- Salomon, David., 2004. *Data Compression The
Complete Reference Third Edition*.
California, Springer-Verlag.
- Widhiarti, Putu. 2000. *Pengantar Kompresi Data*.
([http://repository.upi.edu/operator/uploa
d/s_d545_060855_chapter2.pdf](http://repository.upi.edu/operator/upload/s_d545_060855_chapter2.pdf) tanggal
31 maret 2011 jam 08:15).