

KOMPRESI DALAM SOURCE CODING DITINJAU DARI BENTUK TRANSFORMASINYA

Muhammad Kusban¹

¹Jurusan Teknik Elektro Fakultas Teknik Universitas Muhammadiyah Surakarta
Jl. Ahmad Yani, Tromol Pos I Pabelan Kartasura, Surakarta
E-mail: muhammadkusban@gmail.com

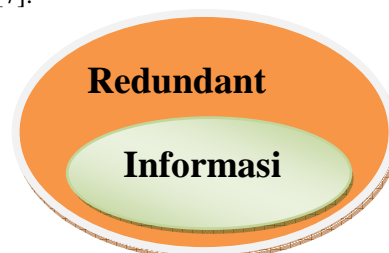
Abstrak

Image merupakan presentasi array nilai pixel, pada umumnya tidak efisien dalam hal memori. Lebih lanjut, adanya keterbatasan bandwidth jaringan menjadikan image tersebut perlu di kompresi. Banyak metode kompresi yang telah ditawarkan dalam rangka mendapatkan ukuran optimal suatu image baru yang masih dapat diterima oleh indera penglihatan dan indera pendengaran. Dengan operasi dalam kawasan transformasi, DCT dan FFT, telah banyak digunakan dalam bidang pemrosesan sinyal, termasuk di dalamnya image processing. Bila hanya untuk mendapatkan kecepatan dan rasio dalam proses kompresi, penggunaan DCT lebih baik yaitu 15.01% dibandingkan 13.28% untuk blok 16x16. Sedangkan FFT akan lebih unggul dibandingkan DCT bila proses diteruskan ke bentuk image enhancement yang memberikan nilai SNR 24.65dB dibandingkan dengan nilai 19.73dB.

Kata kunci : *DCT-based; FFT-based; pixels; SNR; compression; quality decoding*

Pendahuluan

Perkembangan teknologi saat ini memerlukan penyimpanan data secara masal dan cepat dalam koneksinya. Adanya korelasi erat antara besarnya tempat penyimpanan dengan lambatnya koneksi jaringan, mengakibatkan terus menerus berkembang algoritma kompresi data. Tujuan dari kompresi data seperti image adalah untuk menghasilkan image baru, dengan jumlah nilai bit menjadi seminimal mungkin serta dengan metode seefisien mungkin di dalam hal pembuatannya (*recording*) [1]. Bentuk terkompresi, diperlukan terutama dalam bidang: *television conferencing, remote sensing, document medical imaging, facsimile transmission, dan control of remotely piloted* yang digunakan dalam bidang militer dan ruang angkasa [2]. Pada hakekatnya, bentuk informasi yang penting senantiasa dilingkupi oleh informasi yang tidak penting lainnya (*redundant*). Proses kompresi dirancang untuk meminimalkan informasi *redundant* yang dikategorikan menjadi tiga bentuk [3]: *coding redundancy* yang berurusan dengan pixel individual, *interpixel redundancy* yang berurusan dengan hubungan antar pixel, dan *psycho-visual redundancy* yang berkaitan dengan kompresi *lossy*. DCT (*discrete cosines transform*) dan FFT (*fast Fourier transform*) telah menjadi standard baku dalam kompresi image [4,5], contohnya kompresi JPEG [6]. Bahkan Nikolai dan kawan-kawannya telah membuktikan, bahwa penggunaan DCT atau FFT lebih ampuh dibandingkan dengan metode yang lebih baru seperti DWT (*discrete wavelet transforms*) [7].



Gambar 1. Data = *Redundant* + Informasi

Dalam *coding redundancy*, memerlukan perangkat analisa berupa graylevel histogram, dengan rata-rata jumlah bit yang mewakili pixel tersebut. Dalam *interpixel redundancy*, nilai yang dicari adalah koefisien autokorelasi. Semakin besar autokorelasi, maka semakin tinggi *interpixel redundancy*. Sedangkan untuk *psychovisual redundancy*, mengasumsikan bahwa beberapa bagian kecil suatu data memiliki informasi yang lebih penting dibandingkan nilai lainnya dalam data tersebut. Eliminasi dari *psychovisually redundancy* menghasilkan informasi yang bersifat *lossy*.

Dalam mengukur besaran kompresi, terdapat pengertian kompresi *lossless* dan *lossy*. Kompresi *lossless* terjadi bilamana tidak ada informasi yang hilang sehingga image dapat dikembalikan ke bentuk aslinya. Sedangkan dalam kompresi *lossy*, terdapat beberapa informasi yang dihilangkan dalam batas yang wajar sehingga bila

kembalikan ke bentuk awal, image bentukan tidak akan persis sama dengan bentuk awalnya. Hal yang mendasari dari kompresi *lossless* adalah ikatan dalam *spatial redundancy* diantara pixel yang berdekatan (*neighboring pixels*) yang tidak berdiri sendiri, akan tetapi saling berkorelasi satu sama lainnya. Metode yang digunakan untuk menjaga kualitas image hasil kompresi pada umumnya masih dari sudut pandang manusia. Dalam pengamatan, seseorang secara sekilas tidak dapat membedakan antara image asli dengan image terkompresi. Tetapi bila pengamatan diteruskan maka, baru akan muncul perbedaan antara kedua image tersebut.

Faktor yang sangat berpengaruh atas kompresi adalah warna. Dua komponen dasar warna yaitu *luminance* dan *chrominance*. *Luminance* dideskripsikan dengan sejumlah sinar yang lewat atau memancar dalam area tertentu dengan sudut jatuh yang kompak. Lain halnya dengan *brightness*, yaitu penerimaan cahaya yang proporsional terhadap total energi di dalam lingkup (*band*) tampak. *Chrominance* (*chroma* atau disingkat C) dideskripsikan sebagai sinyal sistim video yang menghantarkan informasi warna (*color*) suatu image, terpisah dari sinyal *luma* yang menemaninya (disingkat Y). *Chrominance* direpresentasikan dengan dua warna berbeda: $U = B' - Y'$ (*blue - luma*) dan $V = R' - Y'$ (*red - luma*). Dalam sinyal penyusun video, U dan V memodulasi warna sinyal *subcarrier* yang akan menjadi sinyal *chrominance*. Sinyal *chrominance* berkaitan dengan *phase* dan *amplitude* yang nilainya sebanding dengan *hue* dan *saturation* untuk bentuk warna. *Hue* dapat pula diartikan sebagai *color tone*. Sedangkan *saturation* dapat diterjemahkan sebagai warna itu sendiri yang besaran nilainya tergantung atas *bandwidth*. Guna mendapatkan nilai diskrit terpisah antara *luminance* dan *chrominance*, digunakan ruang warna YUV dengan Y adalah komponen *luminance*, Cb dan Cr merupakan komponen *chrominance*. Korelasi antara YUV dengan RGB dinyatakan sebagai berikut.

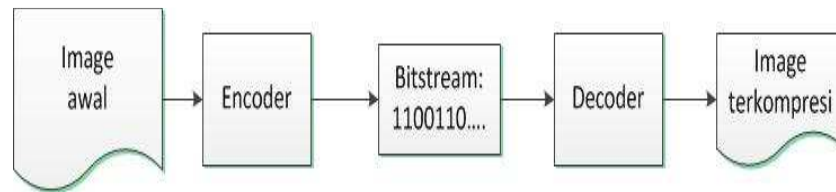
$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

Bentuk image berwarna atau gray yang *smooth*, merupakan rangkaian nilai bit yang berkorelasi satu dengan lainnya. Nilai tersebut, bila dipetakan dalam lingkup 2D akan mengisi ruang di sekitar garis lurus dari posisi kiri bawah menuju titik kanan atas. Hubungan antar bit atau pixel dalam image tersebut sangat erat dan berkaitan satu sama lainnya. Guna kompresi, maka diperlukan algoritma untuk mengurangi korelasi antar pixel tersebut, diantaranya:

- ✓ *Orthogonal transform coding*, yaitu proses memberi nilai suatu konstanta agar berbeda dari asalnya serta upaya mengelompokkan nilai pixel dalam area tertentu. Dalam aljabar, orthogonal matrik Q berupa matrik kotak yang memiliki sifat nilai transpose-nya sama dengan inversinya. $Q^T = Q^{-1}$ dengan $Q^T Q = Q Q^T = I$. Algoritma yang memanfaatkan orthogonal transform coding antara lain KLT (*Karhunen-Loeve transform*) dan DCT (*discrete cosine transform*). Kompresi JPEG menggunakan dasar DCT dalam pengkompresiannya yang bersifat *lossy* sehingga bentukannya tidak dapat dikembalikan sama persis dengan aslinya.
- ✓ *Subband coding*, yaitu proses memberi nilai suatu konstanta dengan cara memecah suatu sinyal menjadi beberapa alokasi frekuensi berbeda, dengan masing-masing alokasi diperlakukan dengan cara berbeda. Untuk kompresi image, misalnya JPEG2000 dengan algoritma DWT (*discrete wavelet transform*). Menggunakan proses subband coding dengan membagi alokasi frekuensi menjadi komponen *lowpass* dan *highpass*.
- ✓ *Predictive coding* yaitu proses memberi nilai suatu konstanta dalam lingkup *envelope* sinyal, dengan menggunakan informasi model *linier predictive*. Untuk kompresi image yang memanfaatkan predictive coding misalnya DPCM (*differential pulse code modulation*) yang bekerja dengan menghilangkan *redundancy* yang sama diantara pixel yang berurutan. Proses transformasi diperlukan untuk menghilangkan keterkaitan nilai awal dalam korelasi antar bit. Dalam statistika, *covariance* merupakan tolok ukur berapa besar nilai yang diperlukan untuk merubah dua variable (*variance* merupakan bentuk khusus *covariance* yang berarti dua variable tersebut identik).

Metodologi

Proses kompresi merupakan proses rangkaian konversi nilai biner (*bit stream*) dengan algoritma tertentu untuk dibentuk nilai baru yang efisien. Untuk bentuk *lossless*, integritas nilai biner tersebut tetap terjaga yaitu nilainya akan tetap sama setelah proses *compression* dan *decompression*, karena proses yang digunakan menggunakan algoritma inversi yang persis sama. Berbeda dengan kompresi *lossy*, dimana nilai biner setelah proses *compress/decompress* berbeda, sehingga tidak akan didapat nilai awal yang sama setelah proses kompresi dilakukan. Meskipun nilai berubah, secara inderawi mata maupun telinga tidak dapat merasakan perubahan kecil yang terjadi. Karena nilai senantiasa berubah, maka bentuk *lossy* tidak dapat digunakan untuk file text maupun file program. Akan tetapi, *lossy* banyak digunakan untuk file gambar dan filem.



Gambar 2. Diagram proses kompresi image

Guna memahami lebih lanjut tentang usaha merubah nilai pixel, perlu memahami konsep konversi nilai biner yang ada dalam file gambar. Sebuah file gambar yang terdiri atas beragam warna sama seperti lukisan yang dihasilkan dari palet dan kanvas [9]. Palet merupakan kumpulan warna yang dapat membentuk gambar. Sama halnya saat melukis, palet tersebut berisikan cat air untuk memberi warna pada lukisan. Setiap warna yang berbeda dalam palet diberi nomor ataupun lubang yang berbeda, contohnya untuk penomoran warna *gray* terdasi atas unsur nilai dari 0 hingga 255. Setelah itu, nomor nomor tersebut dapat digunakan dalam gambar atau lukisan di atas kanvas. Sehingga nilai warna di atas kanvas dianalogikan sebuah matrik yang berisikan elemen dari beragam unsur nomor warna. Untuk itu, guna memahami proses kompresi lebih lanjut, sebuah file gambar direpresentasikan atas matrik yang berisi angka-angka. Selanjutnya proses matrik dipilih dalam blok, baris, atau kolom dalam bidang yang sama alih-ragamnya atau transformasinya.

DCT digunakan sebagai algoritma kompresi image dengan berdasarkan sifat yang dimilikinya, yaitu menempatkan nilai energinya di frekuensi rendah (dalam *subband area*, berada di tepi kiri atas) sehingga dapat mengurangi pengaruh *blocking*. Disamping itu DCT memiliki nilai real (sedangkan DFT bernilai kompleks). Blok fungsi DCT 64 (8 x 8) yang dapat diasumsikan sebagai sum *weighted*. Proses DCT 2D juga merupakan proses mencari nilai *weighted*, seperti tertera dalam persamaan (1) untuk proses DCT dan persamaan (2) untuk proses inverse DCT.

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

Untuk $u = 0, \dots, N-1$ dan $v = 0, \dots, N-1$

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

Untuk $u = 0, \dots, N-1$ dan $v = 0, \dots, N-1$

Tujuan *quantization* adalah untuk pengurangi kepresisian nilai bit sehingga dicapai rasio kompresi yang tinggi. Hal demikian dilakukan mengingat mata manusia lebih respon terhadap koefisien DC serta daerah koefisien spatial frekuensi rendah. Bila koefisien *magnitude* frekuensi tertinggi dibawah ambang *threshold*, maka mata tidak dapat mendeteksinya. Mengatur koefisien frekuensi dalam matriks yang digunakan sebagai transformasi, memiliki kisaran *amplitude* yang lebih kecil dari nilai *threshold* hingga nol (koefisien bentuk demikian tidak dapat di-*recovery* saat proses *decoding*). Selama *quantization*, ukuran koefisien DC dan AC dikurangi serta operasi pembagian dilakukan dengan *threshold* yang telah ditentukan sebagai pembagi. Secara garis besar kompresi dapat dilakukan dalam urutan berikut:

- ✓ Memisahkan *luminance* dan *chrominance* karena informasi dapat dihilangkan dalam porsi yang lebih besar dalam lokasi *chrominance*.
- ✓ sub-sampling komponen warna dalam posisi horisontal dan vertikal.
- ✓ Merubah blok elemen dalam nilai integer dengan mengurangkan separuh skala *gray*.
- ✓ blok posisi (0,0) merupakan blok DC dan blok lainnya merupakan blok AC.
- ✓ Memberi nilai kuantisasi dan *threshold*.
- ✓ Koefisien DC di-deferensiasi yang nilainya relatif terhadap koefisien sub-image sebelumnya.
- ✓ Koefisien AC dipecah menjadi nilai berturutan, mulai dari nol hingga bilangan tidak nol. masing-masing blok bervariasi akan panjang code yang digunakan.
- ✓ Kelompokkan masing-masing koefisien dalam kategori yang sama berdasar atas *range*-nya.
- ✓ variasi panjang kode dan kelompok koefisien menentukan *basecode* dan jumlah bit dalam kode tersebut.
- ✓ Menentukan blok bit kode didasarkan proses koefisien *least significant bits*.

Hasil dan Pembahasan

Dengan menggunakan software Matlab Ver. 7.12.0.635 proses simulasi kompresi atas beragam image dilakukan dalam kawasan transformasi DCT dan FFT. Untuk DCT nilai yang dihasilkan, merupakan jumlah perubahan frekuensi dan *magnitude* dengan menggunakan fungsi `dct2` dan inversinya `idct2`. Sedangkan fungsi FFT

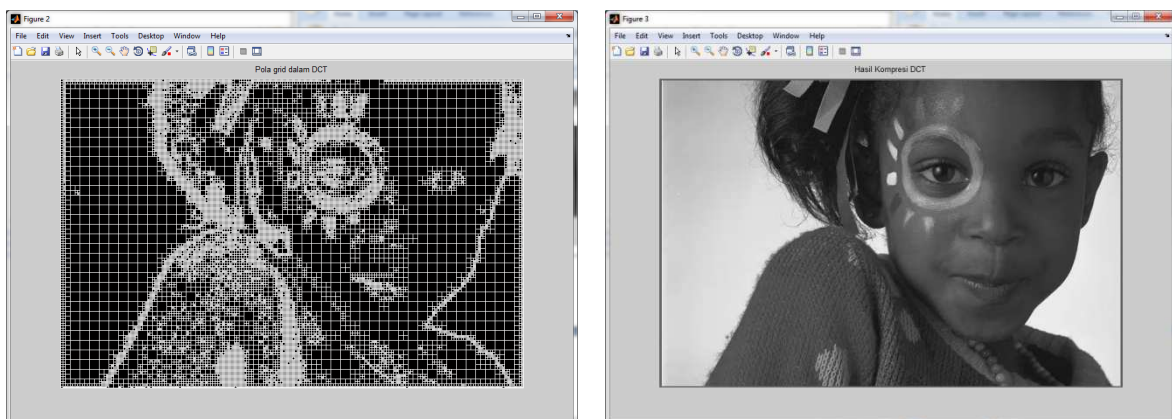
menggunakan perintah *fft2* dan invers-nya menggunakan *ifft2*. Ragam image yang digunakan dalam penelitian didapatkan dalam komunitas Internet.

Tabel 1. Nilai gain DCT untuk blok 4x4, 8x8, 16x16, dan 32x32

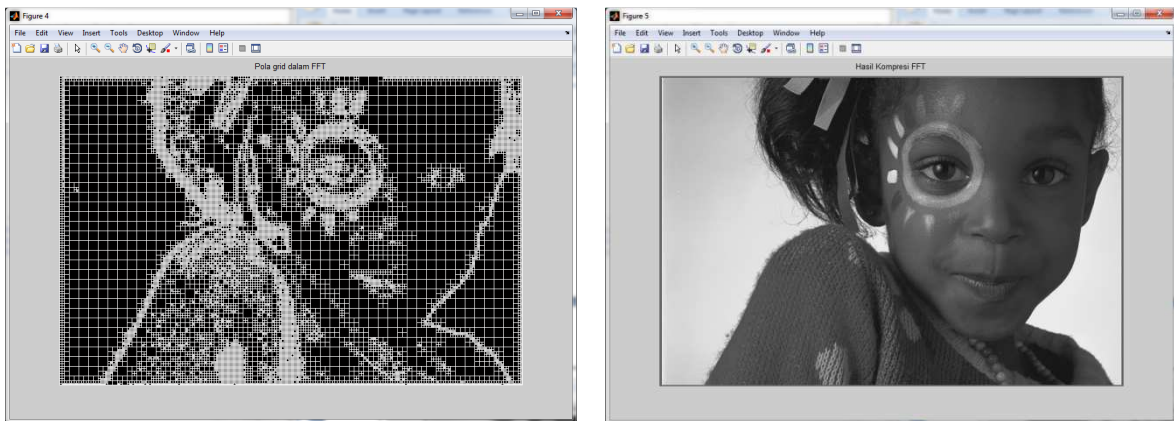
Image	DCT gain			
	4x4	8x8	16x16	32x32
Cameraman.tif	10.97	12.18	12.82	13.35
Baboon.png	5.75	6.51	6.89	7.13
Bigben.png	12.56	14.24	15.10	15.65
Peppers.png	18.42	20.50	21.45	21.96
Railway.png	16.17	18.52	18.20	19.88
Fruits.png	14.44	16.06	16.82	17.27
Lena.png	15.32	16.95	17.71	18.10
Barbara.png	10.94	12.85	14.31	15.47
Boat.png	12.57	14.18	14.97	15.42
Cat.png	15.25	17.58	18.32	19.34
Girl.png	18.20	19.61	20.50	21.06
Goldhill.bmp	0.24	0.33	0.39	0.42
Pool.png	14.24	15.93	16.73	17.40
Sails.png	8.90	10.24	10.94	11.31
Tulips.png	16.68	19.00	20.08	20.62
Σ / μ	190.65 / 12.71	214.68 / 14.31	225.23 / 15.01	234.38 / 15.62

Tabel 2. Nilai gain DCT untuk blok 4x4, 8x8, 16x16, dan 32x32

Image	FFT gain			
	4x4	8x8	16x16	32x32
Cameraman.tif	10.05	11	11.92	12.74
Baboon.png	5.21	5.98	6.50	6.83
Bigben.png	11.29	12.80	13.85	14.65
Peppers.png	15.95	17.24	18.40	19.30
Railway.png	14.73	16.30	16.77	18.80
Fruits.png	12.60	14.06	15.29	16.06
Lena.png	13.19	14.53	15.44	19.30
Barbara.png	9.33	11.00	12.08	13.42
Boat.png	10.80	12.09	13.15	14.03
Cat.png	12.44	14.18	15.37	17.54
Girl.png	17.14	18.25	19.16	19.97
Goldhill.bmp	0.24	0.32	0.37	0.40
Pool.png	11.98	13.18	14.44	15.76
Sails.png	7.64	8.83	9.81	10.55
Tulips.png	13.83	15.33	16.71	18.05
Σ / μ	166.42 / 11.09	185.09 / 12.34	199.26 / 13.28	217.40 / 14.49



Gambar 3. Image yang dihasilkan dengan proses DCT berupa nilai grid dan bentuk *gray* dengan menggunakan blok 8x8



Gambar 4. Image yang dihasilkan dengan proses FFT berupa nilai grid dan bentuk *gray* dengan menggunakan blok 8x8

Tabel 3. Nilai SNR baik DCT dan FFT

Image	2x2 %	4x4 %	8x8 %	16x16 %	SNR DCT (dB)	SNR FFT (dB)
Baboon.png	68.48	17.73	11.45	2.34	14.10	19.56
Bigben.png	10.03	5.56	4.50	74.92	18.88	24.36
Peppers.png	10.97	18.03	22.82	48.18	22.29	26.81
Railway.png	23.08	28.92	12.22	35.78	19.70	24.68
Fruits.png	16.11	21.61	31.52	30.76	20.49	25.12
Lena.png	20.28	21.39	23.56	34.77	19.72	24.55
Cat.png	48.71	23.74	13.26	14.30	20.22	25.27
Girl.png	16.54	17.93	22.75	42.77	24.78	29.83
Pool.png	6.25	8.09	10.62	75.04	20.71	25.26
Sails.png	48.98	40.18	10.32	0.52	15.02	20.05
Tulips.png	27.78	34.32	26.97	10.94	21.18	25.70
\sum / μ	297.21/27.78	237.5/21.59	189.99/17.27	370.32/33.66	217.09/19.73	271.19/24.65

Kesimpulan

Kompresi dalam kawasan transformasi yaitu DCT dan FFT memiliki keuntungan dan kekurangan. Untuk DCT lebih mudah komputasinya karena menghasilkan nilai real, sedangkan FFT lebih berat dalam komputasinya karena menghasilkan nilai kompleks. Nilai energi frekuensi dalam FFT dibawa ke tengah image sehingga mudah dalam penjabaran *enhancement*, yaitu proses selanjutnya setelah kompresi dilakukan. Hal ini berbeda dengan yang menggunakan DCT, yang memusatkan energi frekuensi berada di ujung kiri atas. Dari sudut pandang matrik, proses DCT akan menghasilkan nilai kompresi yang lebih tinggi dibandingkan dari FFT. Dalam simulasi semua nilai kompresi DCT memiliki rasio yang lebih tinggi dibandingkan dengan FFT, misalkan untuk blok 32x32 dalam proses FFT dihasilkan rasio 14.49% sedangkan untuk DCT bernilai 15.62%. Akan tetapi bila menghendaki proses lebih lanjut yaitu adanya proses lagi setelah kompresi seperti menghilangkan noise, membuat kontras suatu image, ataupun meningkatkan kualitas image dengan cara *enhancement* maka penggunaan FFT lebih baik dibandingkan dengan DCT.

DaftarPustaka

[1] <http://ppsc.pme.nthu.edu.tw/prof/pme421600/Chapter%208%20Image%20compression.pdf>
 [2] http://www.dsp.ee.ccu.edu.tw/lie/File/96-IP/Lie-Gonzalez_ch8.pdf
 [3] www.ecse.rpi.edu/courses/S04/ECSE-4540/Compression.ppt
 [4] <http://vision.arc.nasa.gov/publications/mathjournal94.pdf>
 [5] K.S. Thyagarajan, *Still Image and Video Compression with Matlab*, Wiley – 2011.
 [6] Christopoulos, C. Skodras, and A. Ebrahimi T., ‘*The JPEG2000 Still Image Coding System: an overview*’, IEEE Trans. On consumer Electronics, Vol 46, Issue: 4 (2000) 1103-1127.
 [7] NikolayPonomarenko, Vladimir Lukin, Karen Egiazarian, and JaakkoAstola, ‘*DCT Based High Quality Image Compression*’, Tampere International Center for Signal Processing, Tampere-finland.
 [8] Wei Yi Wei, ‘*An Introduction to Image Compression*’, <http://webcache.googleusercontent.com/search?q=cache:KSKd9XPBhpJ:disp.ee.ntu.edu.tw/meeting/%E7%B6>

%AD%E6%AF%85/An%2520Introduction%2520to%2520Image%2520Compression/An%2520Introduction%2520to%2520Image%2520Compression.ppt+an+introduction+to+image+compression+wei+yi+wei&cd=1&hl=id&ct=clnk&gl=id&source=www.google.co.id.

- [9] Aniati Murni dan Dina Chahyati, *Pengolahan Citra Digital: Konsep dasar representasi citra*, Fakultas Ilmu Komputer Universitas Indonesia.