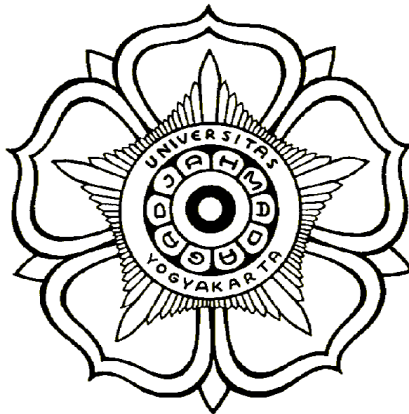


SKRIPSI

**IMPLEMENTASI TEKNIK WATERMARKING
DIGITAL PADA DOMAIN DCT UNTUK
CITRA BERWARNA**

DESI ALEX LESTARI

98 / 121408 / PA / 07478



**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS GADJAH MADA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
YOGYAKARTA**

2003

**IMPLEMENTASI TEKNIK WATERMARKING
DIGITAL PADA DOMAIN DCT UNTUK
CITRA BERWARNA**

SKRIPSI

Skripsi ini disusun untuk memenuhi salah satu syarat guna memperoleh derajat sarjana S1 Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada

DESI ALEX LESTARI

98 / 121408 / PA / 07478

**DEPERTEMEN PENDIDIKAN NASIONAL
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2003

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah S.W.T. atas segala nikmat dan karunia-Nya yang telah diberikan, sehingga penulis dapat menyelesaikan penulisan tugas akhir ini

Penulis menyadari sepenuhnya bahwa dalam penelitian dan penulisan skripsi ini tidak terlepas dari dukungan, dorongan, kerjasama maupun bimbingan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih kepada :

1. Prof. Dr. Subanar, SU., selaku Dekan Fakultas MIPA Universitas Gadjah Mada.
2. Drs. Agus Harjoko, M.Sc., Ph.D., selaku dosen pembimbing skripsi atas bimbingan, kesabaran, dan pengertian yang telah diberikan kepada penulis dari awal penulisan proposal sampai akhir selesainya skripsi ini.
3. Drs. Medi, M.Kom., selaku dosen pembimbing akademik atas segala pengarahan dalam menentukan pengambilan mata kuliah selama penulis belajar di Fakultas MIPA Universitas Gadjah Mada.
4. Segenap staf pengajar di Fakultas MIPA Universitas Gadjah Mada yang telah mengajar, membimbing dan memberikan pemahaman-pemahaman penulis tentang Ilmu Komputer.
5. Segenap karyawan Fakultas MIPA Universitas Gadjah Mada yang telah membantu penulis selama belajar.
6. Ibu, Bapak, dan kakak-kakakku yang telah memberikan semangat dan mendoakan selama belajar di Fakultas MIPA Universitas Gadjah Mada Yogyakarta dan dalam mengerjakan skripsi.
7. *Yeyen, Ian, Arya, Ozy, Trias, Beng2, Eko 'software' dan Yoyok (Alm.)*, serta teman-teman angkatan 1998 lainnya yang tidak dapat penulis sebutkan namanya atas kekompakan, bantuan dan dorongan semangatnya kepada

penulis selama melalui masa kuliah bersama dan dalam mengerjakan skripsi.

8. *Basman, Soniy, Onix*, dan semua warga Trenggalek yang ada di Jogja, yang telah membantu dan *sharing* dengan penulis selama berada di Jogja.
9. Seluruh warga wisma Malvinas tercinta dan sekitarnya, *i-one, fbi, uut, kemsu, anif, onix, pakAdjie, masHarry, hadjar, gogo, suko, aris, ariefa, heksa & hepta, alex* dan semua yang pernah tinggal disana atas segala yang pernah diberikan kepada penulis.
10. Teman-teman dari *mailing-list* Watermarking World yang telah banyak membantu penulis dalam tugas akhir ini.
11. Semua pihak dan teman-teman yang tidak dapat penulis sebutkan satu persatu, yang telah membantu sehingga selesainya pengerjaan skripsi ini dan memberikan arti sendiri dalam hidup penulis.

Tiada hal lain yang dapat menggantikan semua bantuan, dukungan, kerjasama maupun bimbingan dari seluruh pihak yang telah disebutkan di atas, selain doa penulis, kiranya Tuhan yang Maha Pengasih dan Maha Penyayang serta sumber dari segala kekuatan, berkenan membalas semua yang telah diberikan kepada penulis.

Yogyakarta, Juni 2003

Penulis

DAFTAR ISI

SKRIPSI.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSEMBAHAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
INTISARI.....	xiii
ABSTRACT.....	xiv
<u>BAB I</u>	
<u>PENDAHULUAN.....</u>	<u>1</u>
Latar Belakang.....	1
Perumusan Masalah.....	2
Batasan Masalah.....	2
Tujuan Penulisan.....	3
Tinjauan Pustaka.....	3
Metodologi Penelitian.....	4
Sistematika Penulisan.....	4
<u>BAB II</u>	
<u>LANDASAN TEORI.....</u>	<u>6</u>
Definisi Watermark Digital.....	6
Framework Watermarking Digital.....	7
Karakteristik Watermark Digital.....	8
Klasifikasi Teknik Watermarking Digital.....	9
BAB I Spatial Domain Watermarking.....	10
BAB II Transform Domain Watermarking.....	10
BAB III Hybrid Techniques Watermarking.....	11
Discrete Cosine Transform (DCT).....	11
BAB IV Discrete Cosine Transform Dimensi Satu (1-D DCT).....	12
BAB V Discrete Cosine Transform Dimensi Dua (2-D DCT).....	13
Watermarking pada Domain Frekuensi.....	15
Warna dan Ruang Warna	18
BAB VI RGB (Red Green Blue).....	18
BAB VII HSL (Hue Saturation Lightness) dan HSV (Hue Saturation Value).....	19
BAB VIII YCbCr.....	21
Watermarking Digital Pada Citra	23
BAB IX Watermarking pada Citra Gray Scale.....	23

BAB X Watermarking pada Citra Berwarna.....	24
Serangan terhadap Teknik Watermarking Digital Pada Citra	25
BAB XI Kompresi citra JPEG.....	25
BAB XII Perataan histogram (histogram equalization).....	27
BAB XIII Window averaging.....	29
BAB XIV Median filtering	30
BAB XV Penambahan noise.....	31
BAB XVI Resizing	33
<u>BAB III</u>	
<u>ALGORITMA WATERMARKING DIGITAL PADA DOMAIN DCT UNTUK</u>	
<u>CITRA BERWARNA.....</u>	
Struktur Algoritma Watermarking Digital pada Domain DCT untuk Citra Berwarna	35
BAB XVII Ruang Warna.....	35
BAB XVIII Watermark Digital.....	36
BAB XIX Algoritma Penyisipan Watermark (Enkoder).....	36
BAB XX Algoritma Pendeteksian Watermark (Dekoder).....	37
BAB XXI Pengurutan zig-zag.....	38
Analisis Teoritis Algoritma Watermarking Digital pada Domain DCT....	39
BAB XXII Analisis Teoritis.....	39
BAB XXIII Pemilihan Threshold.....	42
<u>BAB IV</u>	
<u>IMPLEMENTASI DAN PEMBAHASAN</u>	
<u>HASIL UJI COBA</u>	
<u>.....</u>	
Implementasi Algoritma Watermarking Digital pada Domain DCT untuk Citra Berwarna	47
BAB XXIV Sarana Implementasi	47
<u>IV.1.1.1 Perangkat Keras.....</u>	<u>47</u>
<u>IV.1.1.2 Perangkat Lunak.....</u>	<u>48</u>
BAB XXV Implementasi Algoritma Watermarking Digital.....	48
<u>IV.1.1.3 Pembuatan Watermark.....</u>	<u>48</u>
<u>IV.1.1.4 Fungsi Scan Zig-zag</u>	<u>50</u>
<u>IV.1.1.5 Fungsi Penyisipan Watermark.....</u>	<u>53</u>
<u>IV.1.1.6 Prosedur Pengujian Keberadaan Watermark.....</u>	<u>56</u>
Uji Coba Hasil Implementasi.....	60
BAB XXVI Hasil Uji Coba tanpa Serangan	60
BAB XXVII Hasil Uji Coba dengan Serangan.....	63
<u>IV.1.1.7 Kompresi JPEG.....</u>	<u>63</u>
<u>IV.1.1.8 Perataan Histogram (Histogram Equalization).....</u>	<u>65</u>
<u>IV.1.1.9 Window averaging</u>	<u>67</u>
<u>IV.1.1.10 Median filtering.....</u>	<u>70</u>
<u>IV.1.1.11 Penambahan noise.....</u>	<u>72</u>

IV.1.1.12 Resizing.....	75
<u>BAB V</u>	
<u>PENUTUP.....</u>	<u>78</u>
Kesimpulan.....	78
Saran.....	79
<u>DAFTAR PUSTAKA.....</u>	<u>80</u>
<u>LAMPIRAN.....</u>	<u>84</u>

DAFTAR GAMBAR

Gambar II-1 : Delapan basis vektor untuk DCT dengan $n = 8$	13
Gambar II-2 : 64 basis fungsi dari 2-D DCT matriks 8×8 .	
Sumber : MATLAB Image Processing Toolbox, The MathWorks Inc.....	14
Gambar II-3 : Operasi yang biasa dialami oleh sebuah media digital.	
Sumber : Cox et al (1997).....	16
Gambar II-4 : Ruang warna RGB.	
Sumber : MATLAB Image Processing Toolbox, The MathWorks Inc.....	19
Gambar II-5 : Ruang Warna HSL.	
Sumber : SPOT Imaging Forum.....	20
Gambar II-6 : Ruang Warna HSV	
Sumber : SPOT Imaging Forum.....	21
Gambar II-7 : Dekomposisi citra RGB ke dalam komponen luminance dan chrominance-nya. Searah jarum jam dari kiri atas adalah citra RGB, komponen luminance (Y), komponen chrominance blue (Cb) dan komponen chrominance red (Cr). Original picture courtesy of Lativi.....	
Gambar II-8 : Diagram encoding berbasis DCT.	23
Sumber : CCITT (1993).....	26
Gambar II-9 : Persiapan kuantisasi koefisien DCT bagi prosedur entropy encoding. (i) Pengkodean beda nilai koefisien DC, dan (ii) Pengurutan zig-zag.	
Sumber : CCITT (1993).....	27
Gambar II-10 : Diagram decoding berbasis DCT	
Sumber : CCITT (1993).....	27
Gambar II-11 : Contoh penerapan histogram equalization pada citra gray level Fishing boat. Citra pada bagian atas adalah citra asal dengan histogramnya, dan yang dibawah adalah hasil perataan histogramnya. Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	
Gambar II-12 : 3x3 neighborhood disekitar titik (x,y) sebuah citra.....	30
Gambar II-13 : Contoh hasil dari window averaging dengan ukuran window 5x5 (tengah) dan 9x9 (kanan). Citra fishing boat asal (kiri). Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	
Gambar II-14 : Contoh hasil dari median filtering dengan ukuran window 5x5 (tengah) dan 15x15 (kanan). Citra peppers asal dengan noise 'salt & pepper' dan density=0.05 (kiri). Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	31

Gambar II-15 : Contoh hasil penambahan noise pada citra pentagon. Berturut turut dari kiri adalah citra pentagon asal, citra hasil gaussian noise dengan mean=0, var=0.03, citra hasil 'salt & pepper' noise dengan density=0.07 dan citra hasil speckel noise dengan var=0.06. Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	32
Gambar III-16: Pengurutan secara zig-zag.....	39
Gambar III-17: pdf dari korelasi untuk hipotesis 0 dan 1. Tidak terjadi serangan pada citra.....	44
Gambar III-18: pdf dari korelasi untuk hipotesis 0 dan 1. Terjadi serangan pada citra.....	44
Gambar III-19: Penentuan threshold yang baru, berdasarkan pada batasan maksimal probabilitas false positive.	46
Gambar IV-20 : Histogram empat buah watermark dengan n = 10000 dan masing-masing state-nya adalah 7,25,46,79.....	50
Gambar IV-21 : Flowchart scan zig-zag.....	52
Gambar IV-22 : Diagram alir prosedur penanaman watermark.....	56
Gambar IV-23 : Diagram alir prosedur pengujian keberadaan watermark....	59
Gambar IV-24 : Citra asli Kid. Original image courtesy of Karel de Gendre	62
Gambar IV-25 : Versi terwatermark dari citra Kid.....	62
Gambar IV-26 : Respon detektor terhadap 1000 watermark penguji pada citra Kid terwatermark dan kinerja ketiga threshold.....	62
Gambar IV-27 : Citra Baboon terwatermark. Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	64
Gambar IV-28 : Hasil kompresi JPEG dengan kualitas 0% dari citra Baboon terwatermark.....	64
Gambar IV-29 : Respon detektor terhadap 1000 watermark penguji pada citra Baboon terwatermark dengan kompresi JPEG 0%.....	65
Gambar IV-30 : Citra Bear terwatermark. Original image courtesy of Robert E. Barber, Barber Nature Photography....	66
Gambar IV-31 : Hasil perataan histogram pada citra Bear terwatermark.....	66
Gambar IV-32 : Respon detektor terhadap 1000 watermark penguji pada citra Bear terwatermark dengan perlakuan perataan histogram.....	67
Gambar IV-33 : Citra Lena terwatermark. Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	69
Gambar IV-34 : Hasil filter window averaging (5x5) pada citra Lena terwatermark.....	69
Gambar IV-35 : Respon detektor terhadap 1000 watermark penguji pada citra F16 terwatermark dengan perlakuan filter window averaging (5x5).....	69

Gambar IV-36 : Citra Skyline Arch terwatermark. Original image courtesy of Robert E. Barber, Barber Nature Photography.....	71
Gambar IV-37 : Hasil median filtering dengan window 5x5 pada citra Skyline Arch terwatermark.....	71
Gambar IV-38 : Respon detektor terhadap 1000 watermark penguji pada citra Skyline Arch terwatermark dengan median filtering (window 5x5).	71
Gambar IV-39 : Citra New York terwatermark. Original image courtesy of Patrick Loo, University of Cambridge.....	74
Gambar IV-40 : Hasil penambahan Gaussian noise dengan mean = 0 dan variance = 0,79 dari citra New York terwatermark.....	74
Gambar IV-41 : Respon detektor terhadap 1000 watermark penguji pada citra New York terwatermark dengan penambahan gaussian noise (mean=0,var=0.79).....	74
Gambar IV-42 : Citra F16 terwatermark. Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.....	76
Gambar IV-43 : Hasil bilinear resizing dari citra F16 terwatermark. Ukuran asli citra 512x512 dan hasil resizing 256x256.....	76
Gambar IV-44 : Respon detektor terhadap 1000 watermark penguji resizing pada citra F16 terwatermark dengan teknik bilinear dari 512x512 menjadi 256x256.....	77

DAFTAR TABEL

Tabel IV-1 : Spesifikasi Perangkat Keras.....	47
Tabel IV-2 : Syntax fungsi randn.....	49
Tabel IV-3 : Hasil pengujian pada citra terwatermark tanpa serangan.....	61
Tabel IV-4 : Hasil pengujian pada citra terwatermark dengan beragam kualitas kompresi JPEG.....	64
Tabel IV-5 : Hasil pengujian pada citra terwatermark dengan Perataan Histogram	66
Tabel IV-6 : Hasil pengujian window averaging pada citra terwatermark. ..	68
Tabel IV-7 : Hasil pengujian median filtering pada citra terwatermark.....	70
Tabel IV-8 : Hasil pengujian penambahan noise pada citra terwatermark ...	73
Tabel IV-9 : Hasil pengujian pada citra terwatermark dengan resizing	76

INTISARI

IMPLEMENTASI TEKNIK WATERMARKING DIGITAL PADA DOMAIN DCT UNTUK CITRA BERWARNA

Oleh

Desi Alex Lestari
98 / 121408 / PA / 07478

Kemudahan distribusi media digital, khususnya melalui internet ternyata memberikan dampak negatif bagi usaha-usaha perlindungan hak cipta atas media digital. Watermarking digital sebagai salah satu solusi yang bisa digunakan untuk mengatasi masalah ini, dihadapkan pada permasalahan ketahanan terhadap distorsi yang mungkin terjadi selama proses distribusi.

Ada beberapa teknik watermarking digital yang pernah diusulkan oleh para peneliti, baik itu yang bekerja pada domain spasialnya maupun yang beroperasi pada domain frekuensinya. Teknik-teknik watermarking pada domain frekuensi ternyata cukup baik dalam mengatasi permasalahan ketahanan terhadap distorsi. DCT (*discrete cosine transform*) sebagai salah satu jenis transformasi yang banyak digunakan dalam dunia citra, kompresi JPEG adalah salah satu contohnya, mentransformasikan nilai intensitas pikselnya ke dalam frekuensi dasarnya. Implementasi watermarking digital dengan menggunakan domain DCT ternyata menunjukkan kinerja ketahanan yang cukup baik, khususnya terhadap kompresi JPEG.

Kata kunci : Watermarking Digital,
DCT (*discrete cosine transform*),
Perlindungan Hak Cipta,

ABSTRACT

THE IMPLEMENTATION OF DIGITAL WATERMARKING TECHNIQUES ON DCT-DOMAIN FOR COLOR IMAGES

Desi Alex Lestari
98 / 121408 / PA / 07478

The ease of distributing digital media, specially using Internet, apparently brings the negative impact for the effort of copyright protection on digital media. Digital watermarking as one of solution to overcome these problems, facing the question of robustness against the distortion that could be happens along the distribution process.

There are several digital watermarking techniques that have been proposed by the scientist, some of them operate directly on its spatial domain and the other one work on frequency domain. The watermarking techniques that operate on frequency domain, apparently can figure the problems out. DCT (*discrete cosine transform*) as one of transformation that used in image environment, i. e. JPEG compression, converting the intensity value of the pixel into its elementary frequency components. The implementation of digital watermarking using DCT domain shows a quite good performance on its robustness, specially again JPEG compression.

Key words : Digital Watermarking,
DCT (*discrete cosine transform*),
Copyright Protection,

BAB I PENDAHULUAN

Latar Belakang

Perkembangan teknologi Internet dalam beberapa tahun terakhir ini, telah membawa perubahan besar bagi distribusi media digital. Media digital yang dapat berupa teks, citra, audio dan video dapat dengan mudah didistribusikan melalui Internet. Kemudahan distribusi media digital melalui Internet disisi lain dapat menimbulkan permasalahan ketika media tersebut terlindungi hak cipta (*copyright*). Sesuai dengan sifatnya, media digital memungkinkan tak terbatasnya salinan yang sulit dibedakan dengan aslinya, dan dengan mudah didistribusikan maupun diperbanyak oleh pihak-pihak yang tidak berhak.

Pemasalahan diatas, membawa perubahan cara pandang peneliti terhadap metode yang digunakan untuk melindungi hak cipta pada media digital. Kriptografi dalam konteks perlindungan terhadap hak cipta media digital ternyata tidak memberikan jawaban yang memuaskan. Teknik kriptografi hanya mengijinkan pemegang kunci yang benar saja yang dapat mengakses media digital terenkripsi, tetapi ketika media ini telah didekripsi tidak ada lagi cara untuk melacak hasil reproduksi. Watermarking digital menawarkan solusi lain yang lebih tepat untuk masalah ini, teknik watermarking melindungi media digital dengan data tertentu yang tertanam secara permanen di dalam media yang bersangkutan.

Discrete Cosine Transform (DCT) banyak sekali digunakan didunia citra khususnya dalam hal kompresi citra. JPEG merupakan salah contohnya, ia memanfaatkan DCT untuk kompresi citra (Wallace, 1991). Faktor rasio kompresi JPEG yang sangat kompetitif menyebabkan popularitasnya meningkat dengan pesat dewasa ini. Watermark sebagai komponen utama dalam algoritma watermarking digital haruslah tahan terhadap kompresi yang populer ini.

Perumusan Masalah

Algoritma-algoritma watermarking yang bekerja pada domain DCT menurut ukuran bloknnya dapat dibagi kedalam 2 kelompok, yaitu algoritma yang memanfaatkan blok-blok DCT dengan ukuran tertentu yang lebih kecil dari ukuran citra aslinya, misalnya blok dengan ukuran 8x8 atau 16x16. Kelompok kedua menerapkan DCT keseluruhan dimensi citra secara langsung.

Masalah yang akan dibahas pada penulisan tugas akhir ini adalah cara kerja beserta implementasi algoritma watermarking digital pada citra berwarna yang bekerja pada domain DCT secara langsung, serta ketahanannya terhadap beberapa serangan, khususnya terhadap kompresi JPEG.

Batasan Masalah

Pembahasan atas domain frekuensi pada citra dan *Discrete Cosine Transform* (DCT) hanya akan diulas secara umum, hal ini dimaksudkan hanya untuk memberikan penjelasan secara global tentang pokok permasalahan yang akan dibahas, sehingga pembahasan tidak dilakukan secara mendalam.

Teknik watermarking yang digunakan mengadopsi proposal Barni et al (1998) dengan beberapa perubahan yang dilakukan penulis. Ujicoba serangan

terhadap teknik ini dibatasi pada kompresi JPEG dan beberapa teknik pemrosesan citra yang umum, seperti *histogram equalization*, *median filtering*, *resizing* dan penambahan *noise*.

Implementasi teknik watermarking pada tugas akhir ini menggunakan program Matlab dari MathWorks Inc.

Tujuan Penulisan

Tujuan yang ingin dicapai pada penulisan skripsi ini adalah untuk memahami teknik watermarking digital pada citra yang memanfaatkan domain DCT secara langsung serta dapat mengimplementasikannya untuk citra berwarna. Selain itu juga untuk mengetahui ketahanan teknik ini terhadap beberapa serangan, seperti kompresi citra maupun pemrosesan citra.

Tinjauan Pustaka

Penelitian mengenai algoritma watermarking digital berbasis DCT, telah banyak diusulkan oleh para peneliti melalui jurnal – jurnal ilmiah, diantaranya adalah jurnal yang berjudul *Secure Spread Spectrum Watermarking for Multimedia*, oleh Cox et al (1997), ia membahas penggunaan konsep *spread spectrum communication* untuk aplikasi watermarking digital. Barni et al (1998) mengusulkan algoritma berbasis DCT pada jurnalnya dengan judul *A DCT-Domain System for Robust Image Watermarking*. Teknik watermarking berbasis DCT juga dibahas oleh Langelaar et al (2000) didalam *Watermarking Digital Image and Video Data : A State-of-the-Art Overview*.

Di Indonesia sendiri ada beberapa ulasan mengenai watermarking digital, diantaranya oleh Supangkat et al (2000) dalam artikel *Watermarking sebagai Teknik*

Penyembunyian Label Hak Cipta pada Data Digital yang diterbitkan oleh jurnal Teknik Elektro, dan Agung (2001) yang mengulas *Digital Watermarking: Teknologi Pelindung HAKI Multimedia*.

Selain referensi diatas, buku buku yang membahas citra digital serta beberapa literatur mengenai teknik-teknik watermarking digital lainnya juga penulis gunakan sebagai bahan pustaka pada penulisan tugas akhir ini.

Metodologi Penelitian

Metode yang digunakan dalam penelitian tugas akhir ini adalah dengan terlebih dahulu melakukan studi literatur mengenai algoritma watermarking digital pada beberapa jurnal maupun paper ilmiah yang diperoleh dari Internet serta beberapa buku yang berhubungan dengan citra digital, kemudian penulis melakukan perancangan dan menerapkan algoritma tersebut dengan menggunakan program Matlab.

Sistematika Penulisan

Di dalam skripsi ini, penulis menyusun pembahasan materi skripsi menjadi beberapa bab. Materi tersebut disusun dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi mengenai latar belakang masalah, batasan permasalahan, tujuan penulisan tugas akhir ini, metode penelitian yang dipakai, serta sistematika penulisan tugas akhir.

BAB II LANDASAN TEORI

Pada bab ini dibahas definisi watermarking digital secara umum, beberapa karakteristik yang perlu dimiliki sebuah algoritma watermarking, pembahasan singkat akan *discrete cosine transform* (DCT), watermarking pada frekuensi domain, dan juga ulasan mengenai ruang warna. Serangan terhadap algoritma watermarking digital pada citra juga diulas pada bab ini.

BAB III ALGORITMA WATERMARKING DIGITAL PADA DOMAIN DCT UNTUK CITRA BERWARNA

Bab ini mengulas algoritma watermarking digital pada domain DCT untuk citra berwarna beserta analisis teoritis terhadap algoritma ini dan juga pemilihan *threshold*.

BAB IV IMPLEMENTASI DAN PEMBAHASAN HASIL UJI COBA

Pada bab ini dibahas implementasi algoritma watermarking digital dari bab III, dan beberapa hasil uji-coba terhadap citra terwatermark, termasuk citra yang dikenai serangan.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran.

BAB II

LANDASAN TEORI

Pada bab ini akan dibahas beberapa konsep dasar yang berhubungan dengan algoritma watermarking digital pada domain DCT untuk citra berwarna, seperti definisi, framework, karakteristik dan klasifikasi teknik watermarking digital. Kemudian dilanjutkan dengan pembahasan singkat mengenai *discrete cosine transform* (DCT), watermarking pada domain frekuensi, ruang warna serta watermarking pada citra *gray scale* dan berwarna.

Definisi Watermark Digital

Ada beberapa pengertian mengenai watermark digital yang dapat diperoleh dari berbagai referensi, baik itu dari paper-paper ilmiah maupun dari sumber-sumber lain yang penulis dapatkan dari Internet.

Watermarking World (2002) mendefinisikan watermark sebagai data tersembunyi yang ditambahkan pada sinyal pelindung (*cover signal*), sedemikian rupa sehingga penambahan tersebut tidak terlihat. Watermark dapat juga merupakan suatu pola yang terbentuk oleh kumpulan bit data tertentu, yang disisipkan kedalam file citra, audio ataupun video yang mengidentifikasi informasi hakcipta file tersebut (Webopedia, 2003). Lebih jauh lagi watermark bisa juga berupa kode yang membawa informasi mengenai pemilik hak cipta, pencipta, pembeli yang sah dan segala sesuatu yang diperlukan untuk menangani hak kepemilikan media digital. Watermark sengaja ditanamkan secara permanen pada

data digital sedemikian hingga pengguna yang berwenang dapat dengan mudah membacanya, disisi lain watermark tersebut haruslah tidak mengubah isi media kecuali sedikit atau perubahan tersebut tidaklah tampak atau kurang begitu tampak bagi indera manusia (Barni et al, 1998).

Dari beberapa definisi diatas ada benang merah yang dapat ditarik untuk memberikan definisi watermark digital, yaitu sebuah watermark merupakan sebuah pola/kode/data tertentu yang membawa informasi tertentu sesuai dengan tujuannya dan sengaja ditanamkan secara permanen kedalam data media induknya.

Framework Watermarking Digital

Jika watermark merupakan sesuatu yang ditanamkan, maka watermarking merupakan proses penanaman watermark tersebut. Secara umum *framework* sebuah algoritma watermarking tersusun atas tiga bagian, yaitu [(Duan and King, 1999) dan (Mohanty, 1999)] :

- (1) Watermark,
- (2) Algoritma penyisipan watermark (enkoder), dan
- (3) Algoritma pendeteksian watermark (dekoder).

Watermark dapat berupa representasi identitas kepemilikan media digital, maupun informasi lain yang dipandang perlu untuk ditanamkan kedalam media yang bersangkutan. Algoritma penyisipan watermark menangani bagaimana sebuah watermark ditanamkan pada media induknya. Algoritma pendeteksian watermark menentukan apakah didalam sebuah media digital terdeteksi watermark yang sesuai atau tidak.

Karakteristik Watermark Digital

Ada beberapa karakteristik atau sifat khusus tertentu yang harus dimiliki oleh sebuah watermark. Sifat-sifat tersebut sangat bergantung kepada aplikasi watermarking yang akan dibuat, atau dengan kata lain tidak ada sekelompok sifat tertentu yang harus dipenuhi oleh semua teknik watermarking. Meskipun demikian ada beberapa sifat yang secara umum dimiliki aplikasi watermarking. Sifat-sifat tersebut diantaranya adalah sebagai berikut [(Swanson et al, 1998) dan (Langelaar et al, 2000)] :

- *Perceptual transparency*

Sebagian besar aplikasi watermarking mengharuskan algoritma watermarking digital menanamkan watermark sedemikian hingga ia tidak mempengaruhi kualitas media yang disisipi watermark. Media yang telah ditanami watermark haruslah sulit dibedakan dengan media aslinya oleh indera manusia. Atau dengan kata lain penanaman watermark pada citra haruslah tidak terdeteksi oleh indera penglihatan manusia dan penanaman watermark pada audio haruslah tidak dikenali oleh indera pendengaran.

- *Robustness*

Untuk watermark yang memang ditujukan untuk membuktikan keotentikan media induk, atau yang disebut dengan *fragile watermark*, tidak disyaratkan memiliki sifat ini. Tetapi jika watermark digunakan untuk aplikasi yang lain, diperlukan watermark yang selalu tertanam didalam media induk, meskipun media induknya mengalami penurunan kualitas akibat serangan.

- *Security*

Dalam banyak aplikasi watermarking, proses penanaman watermark haruslah aman sedemikian hingga pihak yang tidak berhak harus tidak dapat mendeteksi keberadaan data yang ditanamkan, dan mampu menghilangkan data tersebut. Sekali lagi keamanan disini juga sangat bergantung pada aplikasinya. Sebuah prosedur penanaman watermark dikatakan aman jika ia tidak dapat dipecahkan kecuali pihak yang tidak berhak tersebut memiliki akses terhadap kunci yang mengendalikan proses penyisipan data pada media induk.

- Public vs Private Watermarking

Dalam beberapa aplikasi, seperti *copyright protection*, pengekstrakan watermark bisa saja membutuhkan media aslinya. Teknik ini disebut dengan *private* atau *nonoblivious watermarking*. Sedangkan pada aplikasi yang lain, seperti *copy protection* dan *indexing*, proses pendeteksian watermark seharusnya tidak memiliki akses terhadap media asli yang belum terwatermak. Teknik ini sering disebut sebagai *public, blind*, atau *oblivious watermarking*.

Klasifikasi Teknik Watermarking Digital

Dalam satu dasawarsa terakhir ini cukup banyak teknik-teknik watermarking yang sudah diusulkan oleh para peneliti. Teknik-teknik tersebut berdasarkan domain kerjanya, dapat diklasifikasikan menjadi tiga kelompok, yaitu:

1. Teknik watermarking yang bekerja pada domain spasial (*spatial domain watermarking*).

2. Teknik watermarking yang bekerja pada domain *transform* / frekuensi (*transform domain watermarking*)
3. Teknik yang bekerja pada kedua domain diatas (*hybrid techniques watermarking*).

BAB ISpatial Domain Watermarking

Teknik ini bekerja dengan cara menanamkan watermark secara langsung kedalam domain spasial dari suatu citra. Istilah domain spasial sendiri mengacu pada piksel-piksel penyusun sebuah citra. Teknik watermarking jenis ini beroperasi secara langsung pada piksel-piksel tersebut. Beberapa contoh teknik yang bekerja pada domain spasial adalah teknik penyisipan pada *Least Significant Bit (LSB)* (Johnson and Jajodia, 1998), metode *patchwork* yang diperkenalkan oleh Bender et al (1996), Teknik *adaptive spatial-domain watermarking* diusulkan oleh Lee dan Lee (1999).

BAB IITransform Domain Watermarking

Pada *transform domain watermarking* (sering juga disebut dengan *frequency domain watermarking*) ini penanaman watermark dilakukan pada koefisien frekuensi hasil transformasi citra asalnya. Ada beberapa transformasi yang umum digunakan oleh para peneliti, yaitu: *discrete cosine transform (DCT)*, *discrete fourier transform (DFT)*, *discrete wavelet transform (DWT)* maupun *discrete laguerre transform (DLT)*.

Berikut ini beberapa contoh algoritma watermarking digital pada domain frekuensi : Koch dan Zhao (1995) memperkenalkan teknik *randomly sequenced pulse position modulated code (RSPPMC)* yang bekerja pada domain DCT. Kemudian Cox et

al (1997) mengusulkan teknik watermarking digital yang dianalogikan dengan teknik *spread spectrum communication*. Teknik yang hampir serupa dengan proposal Cox et al (1997) diperkenalkan oleh Fotopoulos et al (2000), letak perbedaannya adalah dalam penggunaan blok DCT tempat penanaman watermark. Teknik lain yang memanfaatkan DCT adalah yang diusulkan oleh Barni et al (1998), ia memanfaatkan *pseudo-random number sequence* sebagai watermark yang disisipkan ke dalam vektor koefisien DCT citra yang disusun secara zig-zag seperti dalam algoritma JPEG (Wallace, 1991). Pemanfaatan domain DLT dalam watermarking digital dapat ditemui di (Gilani dan Skodras, 2000). Teknik yang berbasis wavelet ternyata juga tidak kalah populer digunakan dalam watermarking digital, seperti penggunaan wavelet pada watermarking video yang diusulkan oleh Swanson et al (1997). Salah satu alasan pemanfaatan wavelet dalam watermarking adalah kemampuan watermark untuk bertahan dalam berbagai skala resolusi citra (Swanson et al, 1997).

BAB III Hybrid Techniques Watermarking

Teknik watermarking jenis ini bekerja dengan menggabungkan kedua teknik diatas. Pada teknik ini biasanya penanaman watermark dilakukan pada domain frekuensi beberapa bagian citra yang dipilih berdasarkan karakteristik spasial citra tersebut.

Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) biasa digunakan untuk mengubah sebuah sinyal menjadi komponen frekuensi dasarnya. DCT pertama kali diperkenalkan

oleh Ahmed, Natarajan dan Rao pada tahun 1974 dalam makalahnya yang berjudul "On image processing and a discrete cosine transform" (Watson, 1994).

BAB IV Discrete Cosine Transform Dimensi Satu (1-D DCT)

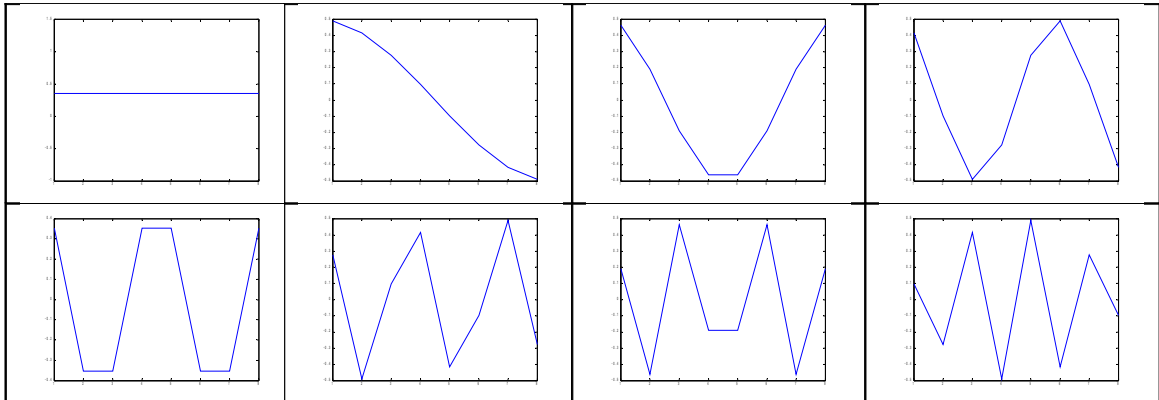
Discrete Cosine Transform dari sederet n bilangan real $s(x)$, $x = 0, \dots, n-1$, dirumuskan sebagai berikut (Watson 1994) :

$$S(u) = \sqrt{2/n} C(u) \sum_{x=0}^{n-1} s(x) \cos \frac{(2x+1)u\pi}{2n},$$

dengan $u = 0, \dots, n-1$ (2.1)

$$\text{dimana } C(u) = \begin{cases} 2^{-1/2}, & \text{untuk } u = 0 \\ 1 & \text{, untuk lainnya} \end{cases}$$

Setiap element dari hasil transformasi $S(u)$ merupakan hasil *dot product* atau *inner product* dari masukan $s(x)$ dan basis vektor. Faktor konstanta dipilih sedemikian rupa sehingga basis vektornya orthogonal dan ternormalisasi. DCT juga dapat diperoleh dari produk vektor (masukan) dan $n \times n$ matriks orthogonal yang setiap barisnya merupakan basis vektor. Delapan basis vektor untuk $n = 8$ dapat dilihat pada gambar II-1. Setiap basis vektor berkorespondensi dengan kurva sinusoid frekuensi tertentu.



Gambar II-1 : Delapan basis vektor untuk DCT dengan $n = 8$.

Barisan $s(x)$ dapat diperoleh lagi dari hasil transformasinya $S(u)$ dengan menggunakan *invers discrete cosine transform* (IDCT), yang dirumuskan sebagai berikut :

$$s(x) = \sqrt{2/n} \sum_{u=0}^{n-1} S(u) C(u) \cos \frac{(2x+1)u\pi}{2n},$$

dengan $x = 0, \dots, n-1$ (2.2)

$$\text{dimana } C(u) = \begin{cases} 2^{-1/2}, & \text{untuk } u = 0 \\ 1, & \text{untuk lainnya} \end{cases}$$

Persamaan diatas menyatakan s sebagai kombinasi linier dari basis vektor. Koefisien adalah elemen transformasi S , yang mencerminkan banyaknya setiap frekuensi yang ada didalam masukan s (Watson, 1994).

BAB V Discrete Cosine Transform Dimensi Dua (2-D DCT)

DCT dimensi satu berguna untuk mengolah sinyal-sinyal dimensi satu seperti bentuk gelombang suara. Sedangkan untuk citra yang merupakan sinyal dua dimensi, diperlukan versi dua dimensi dari DCT (Watson, 1994). Untuk sebuah matriks $n \times m$, 2-D DCT dapat dihitung dengan cara: 1-D DCT diterapkan pada

setiap baris dari s dan kemudian hasilnya dihitung DCT untuk setiap kolomnya.

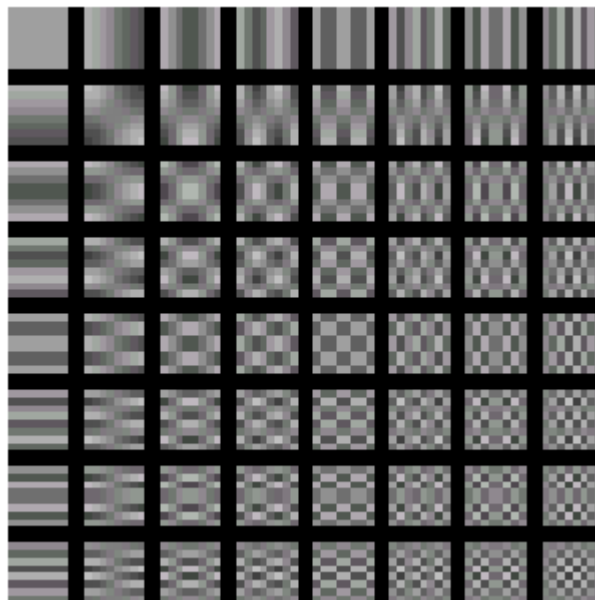
Rumus transformasi 2-D DCT untuk s adalah sebagai berikut :

$$S(u, v) = \frac{2}{\sqrt{nm}} C(u)C(v) \sum_{y=0}^{m-1} \sum_{x=0}^{n-1} s(x, y) \cos \frac{(2x+1)u\pi}{2n} \cos \frac{(2y+1)v\pi}{2m},$$

dengan $u = 0, \dots, n-1$; $v = 0, \dots, m-1$ (2.3)

Rumus 2-D DCT diatas sering juga disebut sebagai *forward discrete cosine transform* (FDCT). 2-D DCT dapat dihitung dengan menerapkan transformasi 1-D secara terpisah pada baris dan kolomnya, sehingga kita dapat mengatakan bahwa 2-D DCT *sparable* dalam dua dimensi.

Seperti pada kasus satu-dimensi, setiap elemen $S(u, v)$ dari transformasi merupakan *inner product* dari masukan dan basis fungsinya, dalam kasus ini, basis fungsinya adalah matriks $n \times m$. Setiap dua-dimensi basis matriks merupakan *outer product* dari dua basis vektor satu-dimensinya.



Gambar II-2 : 64 basis fungsi dari 2-D DCT matriks 8×8 .
Sumber : MATLAB Image Processing Toolbox, The MathWorks Inc.

Setiap basis matriks dikarakterisasikan oleh frekuensi spasial horizontal dan vertikal. Frekuensi horizontal meningkat dari kiri ke kanan, dan dari atas ke bawah secara vertikal. Dalam konteks citra, hal ini menunjukkan tingkat signifikansi secara perseptual, artinya basis fungsi dengan frekuensi rendah memiliki sumbangan yang lebih besar bagi perubahan penampakan citra dibandingkan basis fungsi yang memiliki frekuensi tinggi. Nilai konstanta basis fungsi yang terletak di bagian kiri atas sering disebut sebagai basis fungsi DC, dan DCT koefisien yang bersesuaian dengannya disebut sebagai koefisien DC (*DC coefficient*).

Invers discrete cosine transform dimensi dua (2-D IDCT) dapat diperoleh dengan rumus berikut ini :

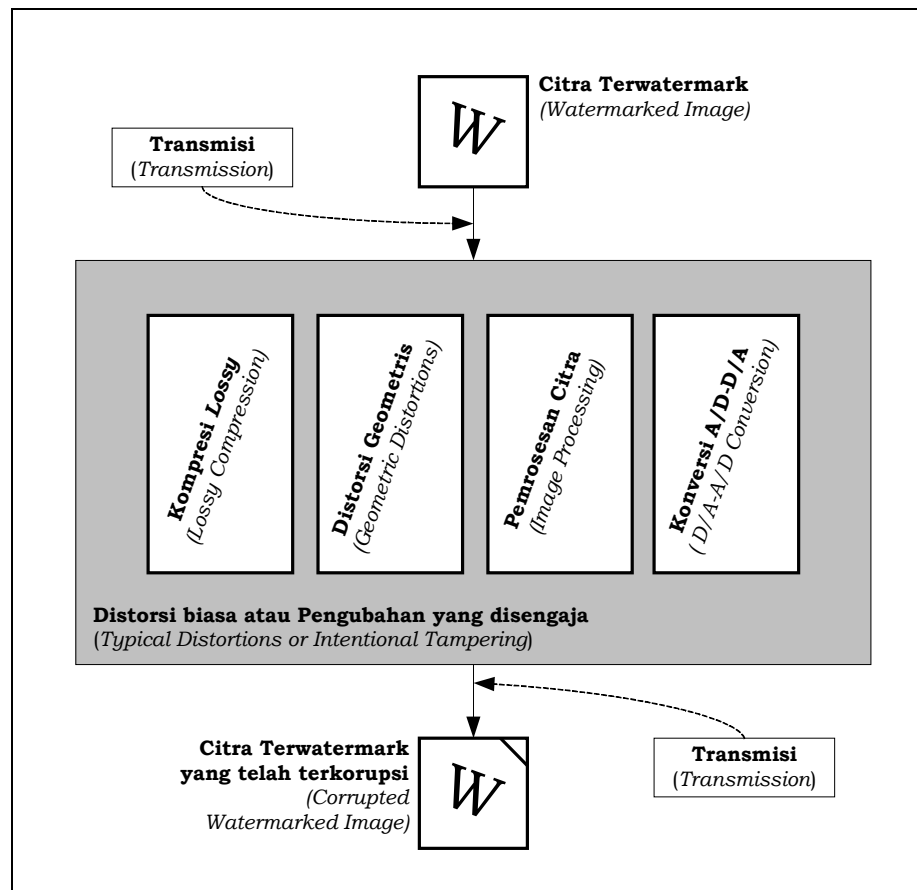
$$s(x, y) = \frac{2}{\sqrt{nm}} \sum_{v=0}^{m-1} \sum_{u=0}^{n-1} S(u, v) C(u) C(v) \cos \frac{(2x+1)u\pi}{2n} \cos \frac{(2y+1)v\pi}{2m},$$

dengan $x = 0, \dots, n-1$; $y = 0, \dots, m-1$ (2.4)

Watermarking pada Domain Frekuensi

Sebuah media digital dalam perjalannya mulai dari si pencipta dan akhirnya sampai ke tangan pengguna (konsumen), tentunya akan sangat mungkin mengalami banyak sekali distorsi ataupun penyimpangan-penyimpangan. Gambar II-3 adalah contoh gambaran proses penyimpangan yang mungkin terjadi pada sebuah media digital khususnya citra digital. Sebuah citra digital seperti dalam gambar II-3 mengalami berbagai proses tertentu yang tentu saja dapat mempengaruhi keutuhan data yang berada didalamnya. Jalur penyebaran (*transmission*) mengacu pada segala aplikasi berbasis kode sumber (*source code*) maupun kode kanal (*channel code*), dan atau teknik-teknik enkripsi standar terhadap

data citra. Meskipun dalam sebagian kasus proses-proses yang terjadi bersifat *lossless*, atau data yang diproses dapat dikembalikan sebagaimana data aslinya, tetapi dalam kasus lain seperti kompresi, ada teknik kompresi (seperti kompresi JPEG) yang bersifat *lossy*, yang secara potensial dapat menurunkan kualitas suatu citra, karena data yang terkompresi tidak dapat dikembalikan seperti data aslinya lagi. Pada kompresi *lossy* dan distorsi geometris (seperti *scaling*) bagian dari citra yang terpengaruh adalah pada komponen frekuensi tinggi.



Gambar II-3 : Operasi yang biasa dialami oleh sebuah media digital.

Sumber : Cox et al (1997).

Uraian diatas memberikan gambaran singkat bahwa sebuah watermark seharusnya tidak ditanamkan pada bagian citra yang secara perseptual tidak

penting, karena banyak pemrosesan citra dan geometris yang mempengaruhi bagian ini. Kemudian yang menjadi permasalahan adalah bagaimana menyisipkan watermark pada spektrum citra yang secara perseptual cukup penting, sedemikian hingga tidak menurunkan kualitas citra. Hal ini dapat dilakukan jika perubahan nilai koefisien cukup kecil dan koefisien yang diubah, secara persentual memang cukup penting.

Untuk mengatasi masalah ini, domain frekuensi dapat dilihat sebagai sebuah kanal komunikasi dan watermark dapat diibaratkan sebagai sinyal yang dikirimkan melewatinya. Serangan dan distorsi sinyal yang tidak disengaja dapat diperlakukan sebagai *noise*, dan watermark harus kebal terhadapnya.

Teknik watermarking pada domain frekuensi sebenarnya dapat dianalogikan seperti komunikasi *spread spectrum (SS)*. Didalam komunikasi SS, seseorang mengirimkan sebuah sinyal cukup lemah diatas kanal dengan yang lebar sedemikian hingga energi sinyal tersebut yang berada pada sebuah frekuensi tidak terdeteksi. Demikian juga dengan watermark, ia disebar ke dalam banyak satuan frekuensi sedemikian hingga energi watermark yang berada pada setiap frekuensi cukup kecil dan secara langsung tidak terdeteksi.

Sebuah watermark yang ditempatkan dengan baik pada domain frekuensi citra, tidak akan tampak oleh mata. Hal ini akan terjadi jika energi watermark yang tersebar cukup kecil di setiap frekuensi. Energi ini dapat ditingkatkan dengan memanfaatkan pengetahuan tentang fenomena *masking* dalam sistem penglihatan manusia (Cox et al, 1997).

Warna dan Ruang Warna

Warna pada dasarnya merupakan hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400nm sampai dengan 700nm (Poynton , 1997).

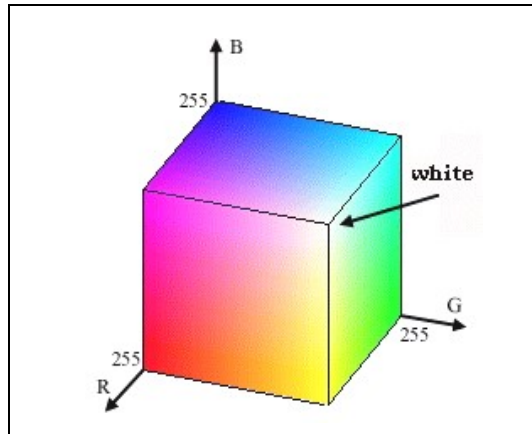
Ruang warna atau yang sering juga disebut sebagai model warna merupakan sebuah cara atau metode untuk mengatur, membuat dan memvisualisasikan warna (Ford and Roberts, 1998). Untuk aplikasi yang berbeda ruang warna yang dipakai bisa juga berbeda, hal ini dikarenakan beberapa peralatan tertentu memang membatasi secara ketat ukuran dan jenis ruang warna yang dapat digunakan. Di dalam penulisan tugas akhir ini, hanya akan dibahas secara singkat beberapa ruang warna yang biasa digunakan untuk aplikasi watermarking. Beberapa ruang warna tersebut antara lain adalah sebagai berikut (Ford and Roberts, 1998) :

1. RGB (*Red Green Blue*)
2. HSL (*Hue Saturation Lightness*) dan HSV (*Hue Saturation Value*)
3. YCbCr (*Luminance – Chrominance*)

BAB VIRGB (*Red Green Blue*)

Citra berwarna yang selama ini biasa kita kenal umumnya memiliki ruang warna RGB. Ruang warna RGB dapat divisualisasikan sebagai sebuah kubus seperti pada gambar II-4, dengan tiga sumbunya yang mewakili komponen warna merah (*red*) R , hijau (*green*) G dan biru (*blue*) B . Salah satu pojok alas kubus ini menyatakan warna hitam ketika $R = G = B = 0$, sedangkan pojok atasnya yang berlawanan menyatakan warna putih ketika $R = G = B = 255$ (untuk sistem warna 8 bit bagi

setiap komponennya). RGB sering digunakan didalam sebagian besar aplikasi komputer karena dengan ruang warna ini, tidak diperlukan transformasi untuk menampilkan informasi di layar monitor. Alasan diatas juga menyebabkan RGB banyak dimanfaatkan sebagai ruang warna dasar bagi sebagian besar aplikasi.



Gambar II-4 : Ruang warna RGB.

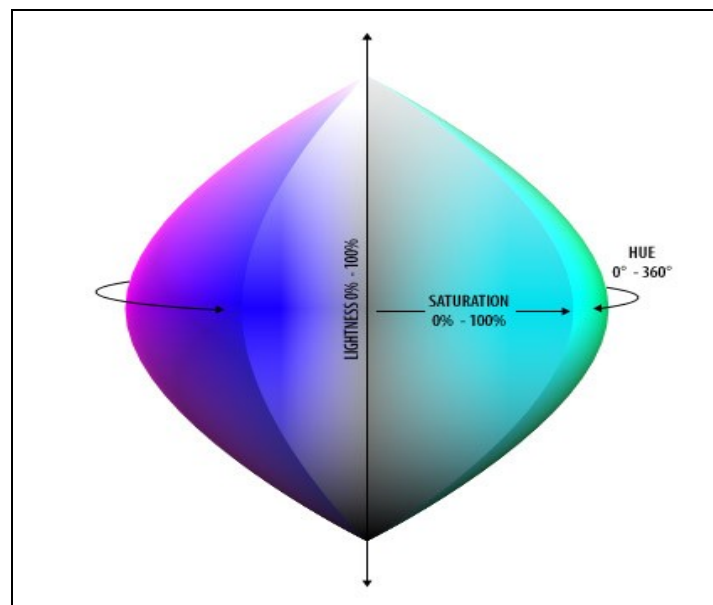
Sumber : MATLAB Image Processing Toolbox, The MathWorks Inc.

BAB VIII HSL (*Hue Saturation Lightness*) dan HSV (*Hue Saturation Value*)

Untuk menyediakan representasi warna bagi antar-muka pengguna (*user interface*), biasa digunakan ruang warna HSL. HSL sendiri merupakan kependekan dari *Hue*, *Saturation*, *Lightness* / *Luminancy*. *Hue* atau corak warna merupakan sensasi penglihatan manusia berdasarkan pada kemiripan suatu daerah tampak seperti daerah yang lain sesuai dengan warna yang diterimanya, merah, kuning, hijau dan biru, atau kombinasi keduanya. *Saturation* adalah kekayaan warna pada suatu daerah sesuai dengan proporsi gelap-terangnya. Kita bisa menemukan warna biru langit sampai dengan biru tua dengan mengubah nilai dari komponen ini. Sedangkan *Luminancy* atau *Lightness* merupakan persepsi suatu daerah warna yang tampak ketika menerima sedikit atau banyak cahaya dengan referensi warna putih.

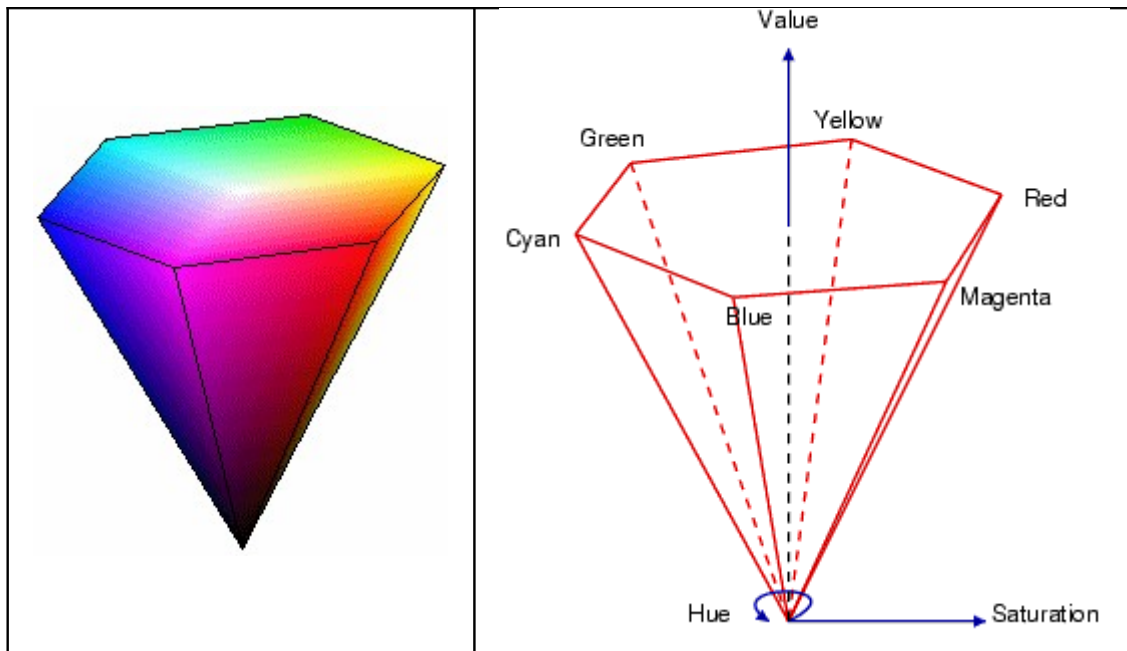
Gambar II-5 melukiskan ruang warna HSL, *Hue* adalah sudut warna tertentu yang melingkar dari suatu titik awal, *Saturation* jarak suatu warna terhadap sumbu *Lightness*, *Lightness* (*Luminancy*) merupakan sumbu tegak yang menghubungkan warna hitam sampai dengan putih.

Alternatif lain dari HSL adalah HSV. Pada ruang warna HSV *Luminancy* digantikan dengan *Value*. HSV dapat divisualisasikan dengan sebuah poligon seperti pada gambar II-6. *Hue* seperti pada HSL merupakan sudut warna yang melingkari poligon, jadi misalnya jika untuk warna merah $hue = 0^\circ$ maka hue untuk warna hijau = 120° dan untuk warna biru nilai hue -nya 240° . *Saturation* sama seperti pada HSL, merupakan jarak terhadap sumbu tegak. Dan *value* merupakan sumbu tegak yang menghubungkan puncak dan dasar poligon.



Gambar II-5 : Ruang Warna HSL.

Sumber : *SPOT Imaging Forum*.



Gambar II-6 : Ruang Warna HSV
 Sumber : *SPOT Imaging Forum*.

BAB VIII YCbCr

YCbCr merupakan standar internasional bagi pengkodean digital gambar televisi yang didefinisikan di CCIR Recommendation 601 (Ford dan Roberts, 1998). Y merupakan komponen *luminance*, Cb dan Cr adalah komponen *chrominance*. Pada monitor monokrom nilai *luminance* digunakan untuk merepresentasikan warna RGB, secara psikologis ia mewakili intensitas sebuah warna RGB yang diterima oleh mata. *Chrominance* merepresentasikan corak warna dan saturasi (*saturation*). Nilai komponen ini juga mengindikasikan banyaknya komponen warna biru dan merah pada warna (Cuturicu, 1999).

Retina mata mempunyai dua macam sel yang berfungsi sebagai analisis visual, yaitu : Sel yang digunakan untuk penglihatan di waktu malam dan sel yang dipakai untuk penglihatan di siang hari. Jenis yang pertama hanya menerima corak keabuan mulai dari warna putih terang sampai dengan hitam pekat. Dan jenis

kedua menerima corak warna. Jika sebuah warna RGB diberikan, sel jenis yang pertama mendeteksi tingkat keabuan (*gray level*) yang serupa dengan nilai *luminance*-nya, sedangkan sel jenis kedua yang bertanggungjawab terhadap penerimaan corak warna, mendeteksi nilai yang sesuai dengan nilai *chrominance*-nya (Cuturicu, 1999).

YCbCr (256 level) dapat diperoleh dari RGB 8-bit dengan menggunakan rumus berikut (Hamilton,1992) :

$$Y = 0.299 R + 0.587 G + 0.114 B \quad (2.5)$$

$$Cb = -0.1687 R - 0.3313 G + 0.5 B + 128 \quad (2.6)$$

$$Cr = 0.5 R - 0.4187 G - 0.0813 B + 128 \quad (2.7)$$

Sedangkan untuk konversi YCbCr ke RGB dapat dilakukan dengan rumus berikut (Hamilton,1992) :

$$R = Y + 1.402 (Cr - 128) \quad (2.8)$$

$$G = Y - 0.34414 (Cb - 128) - 0.71414 (Cr - 128) \quad (2.9)$$

$$B = Y + 1.772 (Cb - 128) \quad (2.10)$$

Gambar II-7 menunjukkan dekomposisi citra RGB ke dalam komponen *luminance* dan *chrominance*-nya.



Gambar II-7 : Dekomposisi citra RGB ke dalam komponen *luminance* dan *chrominance*-nya. Searah jarum jam dari kiri atas adalah citra RGB, komponen *luminance* (Y), komponen *chrominance blue* (Cb) dan komponen *chrominance red* (Cr).
Original picture courtesy of Lativi.

Watermarking Digital Pada Citra

Watermarking pada citra digital secara umum dapat dikategorikan menurut warnanya menjadi 2 kelompok, yaitu :

1. Watermarking pada citra *gray scale*
2. Watermarking pada citra berwarna

BAB IX Watermarking pada Citra *Gray Scale*

Citra *gray scale* merupakan jenis citra yang banyak dimanfaatkan sebagai obyek algoritma watermarking yang pernah diusulkan. Salah satu alasannya adalah bahwa citra ini tidak terlalu kompleks untuk diolah karena ia dapat direpresentasikan sebagai matriks dua dimensi dengan nilai masing-masing

elemennya merupakan representasi tingkat intensitas keabuan dari setiap pikselnya. Informasi tingkat keabuan setiap pikselnya disimpan dalam satuan 8 bit, atau dengan kata lain disini ada 2^8 atau 256 warna keabuan yang berbeda, mulai dari 0 yang merepresentasikan warna hitam sampai dengan 255 yang mewakili warna putih.

Pada dasarnya watermarking pada citra *gray scale* dapat diterapkan secara langsung pada citra ini sesuai dengan jenis teknik watermarking yang digunakan. Sebagai contoh : jika teknik *spatial domain* yang digunakan maka nilai intensitas piksel-piksel dari citra ini yang akan diubah sesuai dengan prosedur yang digunakan untuk menanamkan watermark. Begitupun dengan teknik yang bekerja pada domain frekuensi, transformasi secara langsung dapat dilakukan pada citra, meskipun pada algoritma tertentu kadang-kadang diperlukan pembagian citra menjadi blok-blok yang lebih kecil. Kemudian watermark ditanamkan pada koefisien-koefisien frekuensi yang terpilih.

BAB X Watermarking pada Citra Berwarna

Berbeda dengan teknik watermarking pada citra *gray scale*, dalam kasus citra berwarna, watermark dapat ditanamkan didalam satu atau lebih kanal warna yang ada pada ruang warna citra tersebut. Beberapa teknik watermarking hanya menggunakan kanal warna biru sebab sistem penglihatan manusia (*human visual system* (HVS)) tidak terlalu peka pada komponen warna ini (Swanson, 1998). Cara lain adalah dengan mentransformasikan ruang warna RGB ke ruang warna lain seperti HSL atau YCrCb, kemudian watermark ditanamkan pada komponen *luminance*-nya saja, dan ditransformasikan kembali ke ruang warna RGB.

Serangan terhadap Teknik Watermarking Digital Pada Citra

Dalam subbab ini akan diulas secara singkat beberapa serangan yang akan digunakan didalam pengujian terhadap teknik watermarking yang diimplementasikan dalam tugas akhir ini. Beberapa serangan tersebut antara lain adalah kompresi citra JPEG, *histogram equalization*, *window averaging*, *median filtering*, penambahan *noise* dan *resizing*.

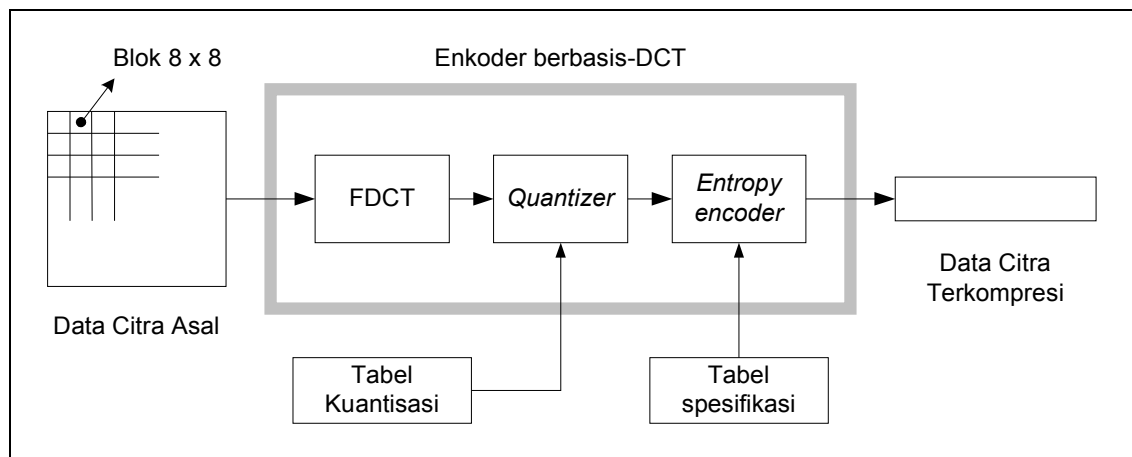
BAB XI Kompresi citra JPEG

Citra terdiri dari beberapa *array* dua dimensi yang biasa disebut dengan komponen (*component*). Dan komponen ini tersusun oleh *array* dua dimensi yang setiap elemennya disebut dengan sampel (*sample*). Sebuah citra yang komponen-komponennya memiliki lebih dari satu bit setiap sample-nya disebut sebagai citra *continuous-tone* (CCITT, 1993). JPEG adalah suatu standar kompresi citra *continuous-tone* yang dikembangkan oleh *Joint Photographic Expert Group*.

Secara umum, proses kompresi citra JPEG dapat dibagi menjadi dua kelompok, yaitu: proses *encoding* dan proses *decoding*. Proses *encoding* disini berarti proses pengkodean dari citra asal (*input*) kedalam citra terkompresi (*output*), sedangkan pada proses *decoding* berlaku sebaliknya. Gambar II-8 menunjukkan proses *encoding* berbasis-DCT.

Didalam proses *encoding*, sampel-sampel dari komponen citra masukan dikelompokkan kedalam blok 8 x 8, dan setiap bloknya ditransformasikan dengan *forward DCT* (FDCT) ke dalam sebuah himpunan 64 nilai yang biasa disebut dengan koefisien DCT. Salah satu anggota dari himpunan ini sering disebut sebagai koefisien DC dan sisanya 63 disebut dengan koefisien AC. Selanjutnya setiap 64

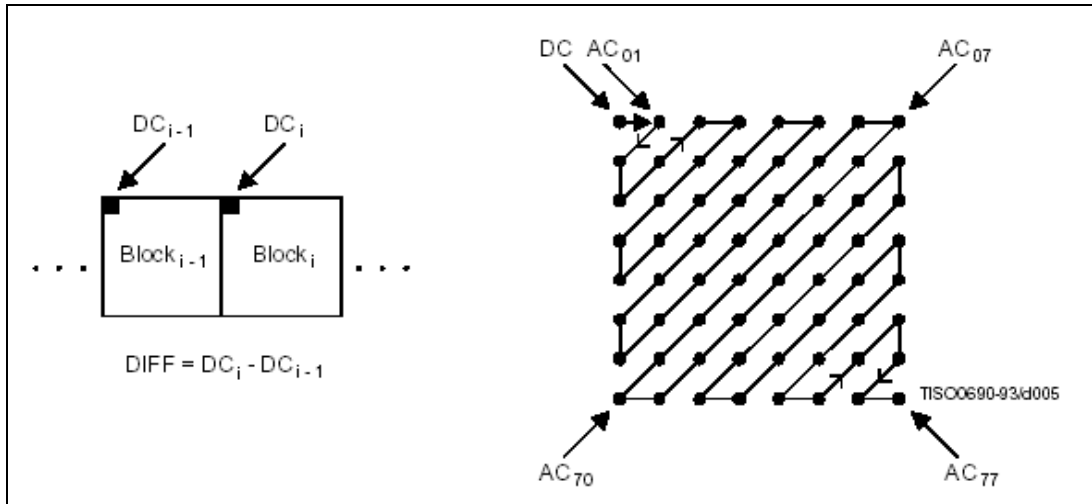
koefisien ini, dilakukan proses kuantisasi menggunakan 64 nilai yang disesuaikan pada tabel kuantisasinya. Setelah proses kuantisasi, koefisien DC dan 63 koefisien AC disiapkan untuk *entropy coding*, seperti terlihat pada gambar II-9. Nilai koefisien DC yang telah dikuantisasi sebelumnya digunakan untuk memprediksi nilai koefisien DC terkuantisasi sekarang dan perbedaan nilainya juga dikodekan. Kemudian 63 koefisien AC ini dikodekan seperti pada koefisien DC tetapi sebelumnya diurutkan secara zig-zag seperti pada gambar II-9. Koefisien yang telah dikuantisasi ini selanjutnya dilewatkan pada prosedur *entropy encoding* yang bertugas mengkompresi data.



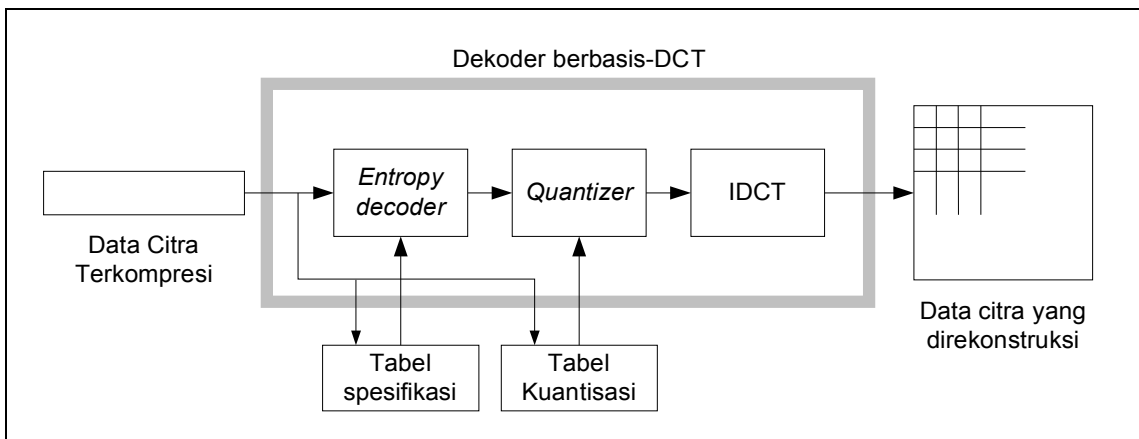
Gambar II-8 : Diagram *encoding* berbasis DCT.
Sumber : CCITT (1993).

Pada gambar II-10 dapat dilihat diagram proses *decoding* berbasis DCT. Langkah-langkah pada proses decoding pada dasarnya merupakan kebalikan dari proses encoding diatas. *Entropy decoder* melakukan *decoding* pada koefisien-koefisien DCT terkuantisasi yang telah terurut secara zig-zag, setelah proses ini DCT koefisien ditransformasikan kedalam 8×8 blok sampelnya menggunakan

invers DCT (IDCT), kemudian citra baru dibentuk dari blok-blok sampelnya tersebut (CCITT, 1993).



Gambar II-9 : Persiapan kuantisasi koefisien DCT bagi prosedur *entropy encoding*. (i) Pengkodean beda nilai koefisien DC, dan (ii) Pengurutan zig-zag.
Sumber : CCITT (1993).



Gambar II-10 : Diagram *decoding* berbasis DCT
Sumber : CCITT (1993).

BAB XII Perataan histogram (*histogram equalization*)

Salah satu teknik yang biasa digunakan dalam proses peningkatan kualitas citra (*image enhancement*) adalah teknik perataan histogram (*histogram equalization*).

Histogram dari suatu citra sebenarnya menyediakan deskripsi umum akan

penampakkannya. Untuk histogram sebuah citra berwarna, ketiga histogram komponen warnanya dapat diperlakukan secara terpisah sebagaimana histogram sebuah citra *gray-level*.

Sebuah citra *gray scale*, dengan r_k menunjukkan tingkat keabuan (*gray level*) dari piksel-pikselya yang tersusun oleh L tingkatan warna keabuan. Kita memiliki persamaan probabilitas berikut (Gonzalez dan Wintz, 1987):

$$p_r(r_k) = \frac{n_k}{n} \quad (2.11)$$

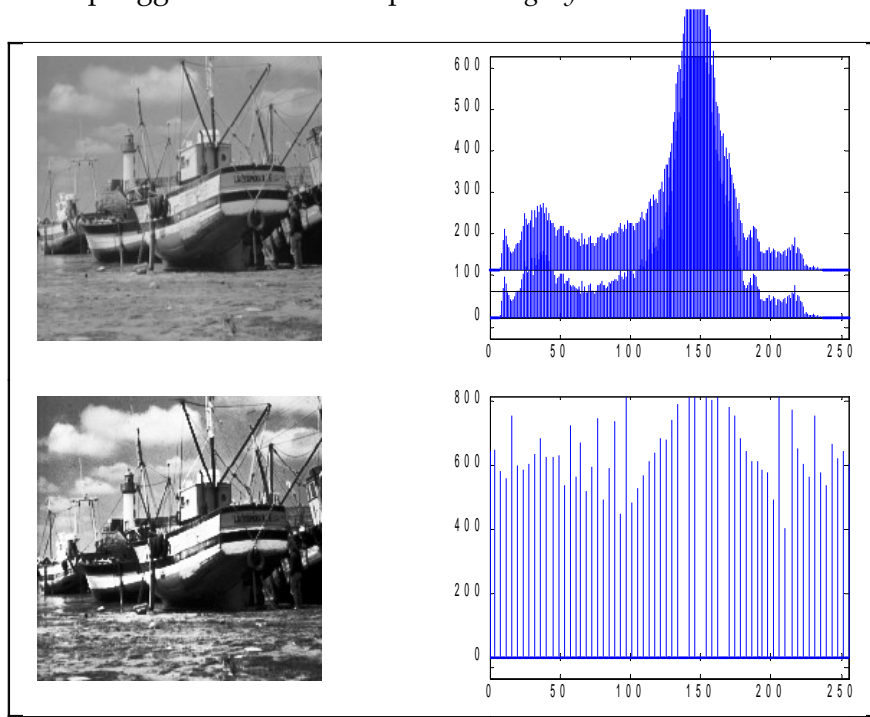
dengan $0 \leq r_k \leq 1$ dan $k = 0, 1, \dots, L-1$. $p_r(r_k)$ adalah probabilitas ke- k tingkat keabuan piksel-piksel penyusun citra, n_k merupakan banyaknya piksel dengan tingkatan warna ke- k yang muncul didalam citra, dan n adalah jumlah total piksel dari citra. Grafik plot antara $p_r(r_k)$ dengan r_k inilah yang biasa disebut dengan histogram, dan teknik yang digunakan untuk mendapatkan histogram yang seragam dikenal dengan nama *histogram equalization* atau perataan histogram.

Cara kerja teknik ini adalah sebagai berikut: sebuah citra *gray level*, sebelumnya tingkat keabuannya r_k perlu dinormalisasikan pada *range* $0 \leq r_k \leq 1$, kemudian dihitung n_k dan $p_r(r_k)$. Selanjutnya dihitung *cumulative distribution function* CDF dari citra dengan rumus berikut (Gonzalez dan Wintz, 1987):

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j) \quad (2.12)$$

dengan $0 \leq r_k \leq 1$ dan $k = 0, 1, \dots, L-1$.

Setiap CDF s_k ini kemudian dipetakan kedalam tingkat keabuan r_k yang terdekat yang terdapat pada citra. Proses ini diakhiri dengan mentransformasikan ulang jumlah piksel yang bersesuaian dengan tingkat keabuan asalnya kedalam tingkat keabuan baru yang merupakan hasil pemetaan CDF sebelumnya. Gambar II-11, adalah contoh penggunaan teknik ini pada citra *gray level*.



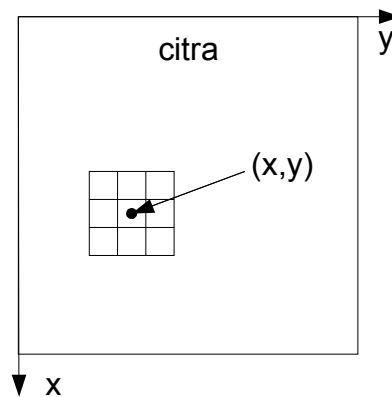
Gambar II-11 : Contoh penerapan *histogram equalization* pada citra *gray level* *Fishing boat*. Citra pada bagian atas adalah citra asal dengan histogramnya, dan yang dibawah adalah hasil perataan histogramnya. *Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.*

BAB XIII Window averaging

Teknik lain yang juga biasa digunakan dalam peningkatan kualitas citra adalah teknik *window averaging* atau *neighborhood averaging*. Teknik yang bekerja secara langsung pada domain spasialnya ini dapat dimanfaatkan untuk penghalusan citra (*image smoothing*). Tetangga (*neighborhood*) atau jendela (*window*) dari sebuah titik (x,y) didalam citra dapat didefinisikan sebagai bagian dari citra

yang berupa bujur sangkar dengan pusat (x,y) . Pada gambar II-12 adalah contoh *window* dengan ukuran 3×3 .

Jika dari sebuah citra *gray scale* $f(x,y)$ ingin diperoleh citra yang lebih halus (*smooth*) $g(x,y)$ dengan menggunakan teknik *window averaging*, maka yang harus dilakukan adalah pada setiap titik (x,y) nilai keabuannya diisi dengan nilai rata-rata (*average*) tingkat keabuan dari tetangganya. Kelemahan utama dari teknik ini adalah ia mengaburkan suatu obyek tepian dan detail tajam lainnya. Gambar II-13 adalah contoh penerapan *window averaging* dengan beragam ukuran *window*.



Gambar II-12 : 3×3 neighborhood disekitar titik (x,y) sebuah citra.

BAB XIV Median filtering

Selain dua teknik diatas, masih ada teknik lain yang juga sering digunakan didalam penghalusan citra (*image smoothing*), teknik ini dinamakan *median filtering*.



Gambar II-13 : Contoh hasil dari *window averaging* dengan ukuran *window* 5x5 (tengah) dan 9x9 (kanan). Citra *fishing boat* asal (kiri). *Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.*

Median filtering sebenarnya merupakan pendekatan alternatif yang digunakan untuk mengatasi kelemahan *window averaging*. Teknik ini bekerja dengan cara mengisi nilai dari setiap piksel dengan nilai median tetangganya. Proses pemilihan median ini diawali dengan terlebih dahulu mengurutkan nilai-nilai piksel tetangga, baru kemudian dipilih nilai tengahnya (median). Gambar II-14 contoh *median filtering* dengan beberapa ukuran *window*.



Gambar II-14 : Contoh hasil dari *median filtering* dengan ukuran *window* 5x5 (tengah) dan 15x15 (kanan). Citra *peppers* asal dengan noise 'salt & pepper' dan *density*=0.05 (kiri). *Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.*

BAB XVPenambahan *noise*

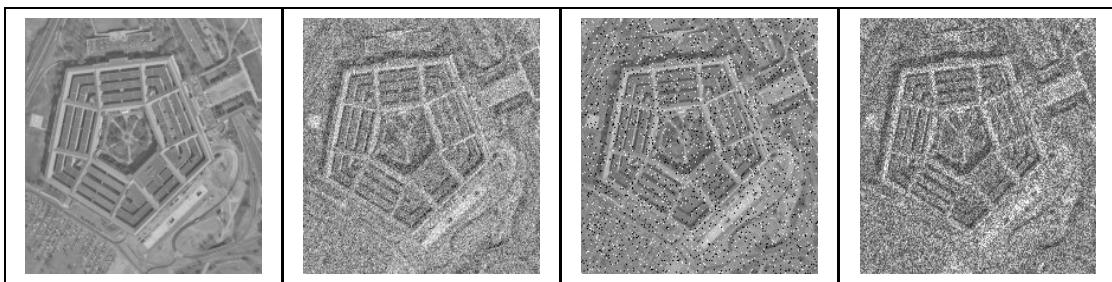
Citra digital sangat rentan mendapatkan serangan berbagai macam jenis *noise*. Ada beberapa cara yang menyebabkan suatu *noise* dapat berada didalam sebuah citra, bergantung bagaimana citra tersebut diciptakan. Sebagai contoh, jika citra merupakan hasil scan foto yang berasal dari negatif film, maka negatif film ini merupakan sumber *noise*. *Noise* juga bisa merupakan akibat dari kerusakan film atau juga bisa berasal dari scanner itu sendiri. Jika citra diperoleh secara langsung dalam

format digitalnya, mekanisme dalam mendapatkan data digital tersebut juga dapat menyebabkan adanya *noise*. Penyebaran data citra secara elektronik bisa juga menghasilkan *noise*.

Didalam program Matlab, dikenal tiga jenis *noise* yang biasa digunakan untuk mensimulasikan penambahan *noise* pada citra (The MathWorks Inc., 2000a). Ketiga jenis *noise* tersebut adalah : *gaussian noise*, '*salt & pepper*' *noise* dan *speckle noise*. Jenis pertama dari *noise* ini bekerja dengan cara menambahkan *Gaussian white noise* dengan mean dan variansi tertentu. Jenis kedua bekerja dengan cara menghidupmatikan nilai intensitas suatu piksel dengan kepadatan tertentu, istilah hidup-mati ini mengacu pada pengaturan nilai dari piksel terpilih dengan nilai tertinggi atau terendah tingkatan warnanya. Jenis terakhir bekerja dengan menambahkan perkalian *noise* dengan citranya berdasarkan persamaan berikut ini :

$$J = I + n * I \quad (2.13)$$

dimana n adalah *noise* acak yang terdistribusi secara seragam (*uniform*), I adalah citra asalnya dan J merupakan citra hasil penambahan *noise*.



Gambar II-15 : Contoh hasil penambahan *noise* pada citra *pentagon*. Berturut turut dari kiri adalah citra *pentagon* asal, citra hasil *gaussian noise* dengan $mean=0$, $var=0.03$, citra hasil '*salt & pepper*' *noise* dengan $density=0.07$ dan citra hasil *speckle noise* dengan $var=0.06$. *Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.*

BAB XVI Resizing

Ketika kita bekerja dengan citra digital, operasi pengubahan ukuran citra (*resizing*) mungkin merupakan operasi yang biasa dilakukan. Baik itu memperbesar maupun memperkecil ukuran citra. Teknik yang umum digunakan untuk operasi ini adalah teknik interpolasi.

Interpolasi secara sederhana dapat dikatakan sebagai sebuah cara untuk memperkirakan nilai tertentu diantara nilai-nilai yang diketahui dalam *range* yang sama (*Pocket Oxford Dictionary*, 1994). Dalam konteks citra digital, interpolasi merupakan suatu proses yang dilakukan untuk memperkirakan nilai citra pada lokasi diantara piksel-piksel citra (*The MathWorks Inc.*, 2000a). Contohnya jika operasi perbesaran pada citra dilakukan maka kita akan mendapatkan lebih banyak piksel dibandingkan citra asalnya. Piksel-piksel baru ini bisa didapat dari interpolasi.

Ada beberapa teknik interpolasi yang biasa dipakai untuk pengubahan ukuran (*resizing*), teknik-teknik tersebut adalah *nearest neighbor*, *bilinear* dan *bicubic*. Pada interpolasi *nearest neighbor*, nilai piksel yang baru diisi dengan nilai piksel terdekat yang sudah ada. Sehingga ketika sebuah citra diperbesar, teknik ini menggandakan piksel, dan ketika citra ukurannya diperkecil maka ia akan menghapus piksel. Cara ini memang lebih cepat dibandingkan dua teknik interpolasi lainnya, meskipun demikian kelemahan utama teknik ini terlihat pada operasi perbesaran, tepian obyek didalam citra terlihat bergerigi. Pada operasi pengecilan, citra yang dihasilkan terlihat kurang halus. Teknik kedua, *bilinear*, bekerja dengan menggunakan nilai dari keempat piksel-piksel disekitarnya, yaitu :

atas, bawah, kiri dan kanan dari titik dimana piksel baru akan dibuat. Nilai dari piksel ini ditentukan dengan menghitung bobot rata-rata dari empat piksel (*array 2x2*) yang menurut jaraknya paling dekat. Hasil dari teknik ini memang lebih halus dibandingkan dengan teknik yang pertama. Pada teknik terakhir, *bicubic*, nilai piksel baru ditentukan dengan menghitung bobot rata-rata 16 piksel (*array 4x4*) yang menurut jaraknya paling dekat. Citra hasil dari teknik ini jauh lebih halus dibandingkan dengan menggunakan teknik yang pertama (Zinneman, 2002).

BAB III

ALGORITMA WATERMARKING DIGITAL PADA DOMAIN DCT UNTUK CITRA BERWARNA

Pada bab ini akan diulas algoritma watermarking digital pada domain DCT untuk citra berwarna, kemudian dilanjutkan dengan analisis terhadap algoritma tersebut termasuk didalamnya pembahasan mengenai pemilihan *threshold* yang digunakan dalam proses deteksi.

Struktur Algoritma Watermarking Digital pada Domain DCT untuk Citra Berwarna

BAB XVIIIRuang Warna

Seperti yang pernah disinggung sebelumnya pada sub bab 2.7.1, citra berwarna yang biasa dikenal pengguna komputer adalah bekerja pada ruang warna RGB. Ruang warna ini membagi komponen warnanya menjadi tiga kanal terpisah, yaitu: kanal merah (*red*) R, hijau (*green*) G dan biru (*blue*) B. Kombinasi ketiga kanal warna inilah yang kemudian membentuk sebuah warna tertentu yang dapat dilihat pada layar monitor. Sebenarnya watermark secara langsung bisa saja ditanamkan pada salah satu kanal warna ini. Tetapi akan lebih baik jika penanaman watermark dilakukan pada kanal *luminance* Y dalam ruang warna YCbCr. Pemilihan model warna ini tidak lain mengacu pada algoritma JPEG yang lebih memprioritaskan kanal *luminance* dibandingkan dengan kedua kanal *chrominance*-nya. Untuk itulah perlu dilakukan transformasi dari citra dari ruang warna RGB ke ruang warna YCbCr-nya.

BAB XVIII Watermark Digital

Watermark $X = \{x_1, x_2, \dots, x_n\}$ berupa deretan bilangan real dengan sebanyak n , terdistribusi normal, memiliki nilai varian (*variance*) σ^2 satu dan nilai *mean* nol (Barni et al, 1998). Deretan bilangan-bilangan real ini dibangkitkan oleh komputer secara acak dengan suatu kunci tertentu, sedemikian hingga apabila kunci yang sama dimasukkan akan didapatkan sederetan bilangan real yang sama juga. Penggunaan watermark yang terdistribusi normal dimotivasi oleh ketahanannya terhadap serangan yang menggunakan beberapa salinan citra terwatermark yang berbeda untuk menghasilkan citra tanpa watermark (Cox et al, 1997).

BAB XIX Algoritma Penyisipan Watermark (Enkoder)

Algoritma penyisipan watermark merupakan bagian yang cukup penting dalam *framework* sebuah algoritma watermarking digital. Dalam penulisan tugas akhir ini, algoritma penyisipan watermark yang digunakan mengadopsi algoritma yang pernah diusulkan oleh Barni et al (1998). Secara singkat, algoritma ini bekerja dengan menyisipkan watermark $X = \{x_1, x_2, \dots, x_n\}$ kedalam sekelompok koefisien DCT citra terpilih $V = \{v_{L+1}, v_{L+2}, \dots, v_{L+n}\}$ yang telah diurutkan secara zig-zag, seperti algoritma kompresi JPEG (Wallace, 1991), mulai koefisien ke $L+1$ sampai dengan koefisien DCT ke $L+n$ sesuai aturan berikut ini (Barni et al, 1998):

$$v'_{L+i} = v_{L+i} + \alpha |v_{L+i}| x_i \quad (3.1)$$

dimana $i = 1, 2, \dots, n$. Kemudian V' disisipkan kembali kedalam urutan zig-zagnya kembali, dan dilakukan inverse DCT untuk mendapatkan citra terwatermark.

Berikut ini urutan algoritma penyisipan watermark tersebut:

Algoritma 3.1 : Algoritma Penyisipan Watermark

1. Ambil komponen *luminance* Y hasil transformasi RGB citra asal kedalam ruang warna YCbCr.
2. Hitung DCT komponen Y dan urutkan secara zigzag.
3. Pilih koefisien DCT V hasil pengurutan zig-zag mulai dari koefisien ke $L+1$ sampai dengan koefisien ke $L+n$. Dan kemudian tanamkan watermark X kedalam koefisien terpilih V_{L+i} .
4. Sisipkan kembali koefisien DCT yang telah ditanami watermark V'_{L+i} sesuai urutan zig-zagnya dan lakukan invers DCT untuk mendapatkan komponen *luminance* Y yang baru.
5. Transformasikan kembali kedalam ruang warna RGB untuk mendapatkan citra terwatermark.

BAB XX Algoritma Pendeteksian Watermark (Dekoder)

Algoritma pendeteksian watermark berfungsi untuk mendeteksi keberadaan watermark didalam suatu citra yang diduga mengandung watermark. Lebih jauh lagi algoritma ini bertugas memberikan verifikasi apakah citra yang diuji tertanami watermark pembanding atau tidak. Algoritma ini bekerja dengan menghitung korelasi r antara koefisien terpilih V' dengan watermark pembanding W . Korelasi r ini dapat dihitung dengan aturan berikut ini (Barni et al, 1998):

$$r(W, V') = \frac{W \cdot V'}{n} = \frac{1}{n} \sum_{i=1}^n w_i v'_i \quad (3.2)$$

Kemudian nilai dari korelasi r ini dibandingkan dengan sebuah nilai *threshold* T_r , untuk menentukan apakah citra yang diuji mengandung watermark W . Jika nilai r lebih besar dari harga T_r maka dapat diputuskan citra yang bersangkutan mengandung watermark W , dan jika tidak maka disimpulkan bahwa citra yang diujikan tidak mengandung watermark pembanding. Pemilihan nilai T_r akan dibahas lebih lanjut pada bagian 3.2.2 Pemilihan *Threshold*.

Algoritma pendeteksian watermark tersebut dapat dirumuskan sebagai berikut:

Algoritma 3.2 : Algoritma Pendeteksian Watermark

1. Ambil komponen Y (luminance) hasil transformasi citra RGB yang terwatermark kedalam ruang warna YCbCr.
2. Hitung DCT komponen Y dan urutkan secara zig-zag.
3. Pilih koefisien DCT hasil pengurutan zig-zag mulai dari koefisien ke $L+1$ sampai dengan koefisien ke $L+n$. Dan kemudian hitung korelasi r antara watermark W dengan koefisien terpilih V^* .
4. Bandingkan korelasi r dengan *threshold* T_r untuk menentukan apakah watermark W terdapat didalam koefisien terpilih atau tidak.

BAB XXIPengurutan zig-zag

Pada bahasan sebelumnya, disebutkan bahwa algoritma penyisipan dan pendeteksian watermark memerlukan pengurutan zig-zag sebagaimana yang dipergunakan dalam algoritma kompresi JPEG (Wallace, 1991). Pengurutan secara

$r(X, V')$ antara watermark penguji X dan koefisien DCT terpilih V' dari citra yang dideteksi dihitung sesuai dengan persamaan (3.2) untuk menentukan apakah citra yang dimaksud mengandung watermark X atau tidak. Pada proses pendeteksian ada tiga kemungkinan yang dapat terjadi, yaitu :

Hp. A : $V' = V$ (citra tidak ditanami watermark);

Hp. B : $V' = V + \alpha W|V|$ (watermark $W \neq X$ ada didalam citra);

Hp. C : $V' = V + \alpha X|V|$ (watermark X ada didalam citra);

Secara umum, dekoder hanya bertugas untuk menentukan apakah sebuah citra tertanami suatu watermark atau tidak. Sehingga ketiga hipotesis diatas dapat disederhanakan menjadi dua hipotesis berikut :

Hp.0 : Hp. A dan Hp. B (citra tidak ditandai dengan X);

Hp.1 : Hp. C (citra ditandai dengan X);

Untuk membedakan antara kedua hipotesis diatas dekoder menghitung $r(X, V')$ dan membandingkannya dengan sebuah *threshold* T_r , jika $r(X, V')$ dibawah T_r , maka dekoder memutuskan bahwa citra tidak ditandai dengan watermark X , sebaliknya demikian juga jika nilai $r(X, V')$ berada diatas T_r , maka dekoder akan menyimpulkan bahwa citra ditandai oleh watermark X . Sebelum memilih nilai T_r yang sesuai, perlu diketahui dahulu karakteristik dari $r(X, V')$ (Piva et al, 1998). Berangkat dari persamaan (3.1) dan persamaan (3.2) kita bisa mendapatkan persamaan korelasi sebagai berikut :

$$r = \frac{1}{n} \sum_{i=1}^n (v_i w_i + \alpha |v_i| x_i w_i) \quad (3.3)$$

dengan $r = r(X, V')$. Jika watermark penguji W sesuai dengan watermark X yang tertanam didalam citra maka r akan menjadi :

$$r = \frac{1}{n} \sum_{i=1}^n (v_i x_i + a |v_i| x_i^2) \quad (3.4)$$

Berdasarkan hipotesis bahwa v_i dan x_i sama-sama memiliki nilai mean nol, dan merupakan variabel random yang *independent* dan terdistribusi secara seimbang, maka mean dan varian dari r dapat dihitung sebagai berikut (Barni et al, 1998) :

$$\mu_\rho = \begin{cases} \alpha \mu_{|v|} & \text{jika } X = W \\ 0 & \text{jika } X \neq W \\ 0 & \text{jika watermark tidak ada} \end{cases} \quad (3.5)$$

$$\sigma_\rho^2 = \begin{cases} (1 + 2\alpha^2) \frac{\sigma_v^2}{n} + \alpha^2 \frac{\sigma_{|v|}^2}{n} & \text{jika } X = W \\ (1 + \alpha^2) \frac{\sigma_v^2}{n} & \text{jika } X \neq W \\ \frac{\sigma_v^2}{n} & \text{jika watermark tidak ada} \end{cases} \quad (3.6)$$

Dari persamaan (3.5) dan (3.6), untuk hipotesis 0, diperoleh (Piva et al, 1998):

$$\mu_{\rho|H_{p,0}} = 0 \quad (3.7)$$

dan

$$\sigma_{\rho|H_{p,0}}^2 = (1 + \alpha^2) \frac{\sigma_v^2}{n} \quad (3.8)$$

dan untuk hipotesis 1, didapat :

$$\mu_{\rho|H_{p,1}} = \alpha \mu_{|v|} \quad (3.9)$$

dan

$$\sigma_{\rho|Hp.1}^2 = (1 + 2\alpha^2) \frac{\sigma_v^2}{n} + \alpha^2 \frac{\sigma_{|v|}^2}{n} \quad (3.10)$$

dimana $\mu_{|v|} = E[|v|]$, $\sigma_v^2 = \text{var}[t]$, dan $\sigma_{|v|}^2 = \text{var}[|v|]$. Dengan menyatakan $\sigma_{|v|}^2 < \sigma_v^2$ dan asumsi bahwa $\alpha \ll 1$, maka :

$$\sigma_{\rho|Hp.0}^2 ; \sigma_{\rho|Hp.1}^2 ; \frac{\sigma_v^2}{n} @ \sigma_{\rho}^2 \quad (3.11)$$

BAB XXIII Pemilihan *Threshold*

Pemilihan nilai *threshold* $\boxed{\tau}$ yang tepat dan benar merupakan suatu hal yang sangat penting bagi sistem pendeteksian watermark. Untuk menentukan nilai $\boxed{\tau}$, probabilitas kesalahan yang dibuat dekoder perlu diperhatikan. Probabilitas kesalahan P_e , yang merupakan probabilitas dekoder memutuskan hipotesis yang salah, dapat ditulis sebagai berikut (Piva et al, 1998):

$$P_e = P(0|1)P(1) + P(1|0)P(0) \quad (3.12)$$

dimana $P(0|1)$ adalah probabilitas dekoder menyangkal eksistensi watermark, meskipun sebenarnya ia ada (*false negative*), dan $P(1|0)$ adalah probabilitas dekoder menyatakan watermark ada, meskipun sebenarnya ia tidak ada (*false positive*), $P(0)$ dan $P(1)$ masing-masing merupakan probabilitas hipotesis 0 dan 1. Berdasarkan asumsi bahwa $H_{p.0}$ dan $H_{p.1}$ mempunyai probabilitas yang seimbang, maka $\boxed{\tau}$ pada persamaan (3.12) dapat ditulis sebagai berikut (Piva et al, 1998):

$$P_e = \frac{1}{2} [P(\rho < T_{\rho} | 1) + P(\rho > T_{\rho} | 0)] \quad (3.13)$$

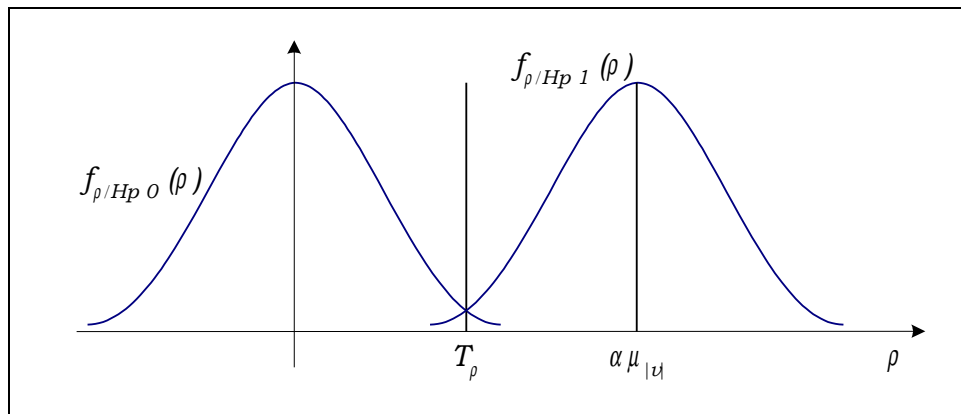
dimana $\rho = \rho(X, V')$.

Pada gambar III-2 dapat dilihat *probability density function* (pdf) dari ρ baik untuk hipotesis 0 maupun hipotesis 1. Nilai *threshold* T_ρ haruslah dipilih sedemikian rupa sehingga $P(0|1) + P(1|0)$ bernilai paling kecil. Karena $\alpha \ll 1$, maka $P(0|1) = P(1|0)$, sehingga harga optimum *threshold* T_ρ berada ditengah-tengah nilai nol dan $\mu_{\rho|Hp.1}$:

$$T_\rho = \frac{\alpha}{2} \mu_{|v|} \quad (3.14)$$

Analisis ini akan bekerja dengan baik untuk citra yang tidak mendapat serangan. Tetapi jika serangan terhadap citra terjadi, maka analisis T_ρ tidak lagi sesuai, karena nilai mean dan varian dari $\rho(X, V')$ sangat mungkin akan berubah akibat serangan. Dalam aplikasi praktisnya, perhitungan nilai T_ρ pada dekoder menggunakan asumsi

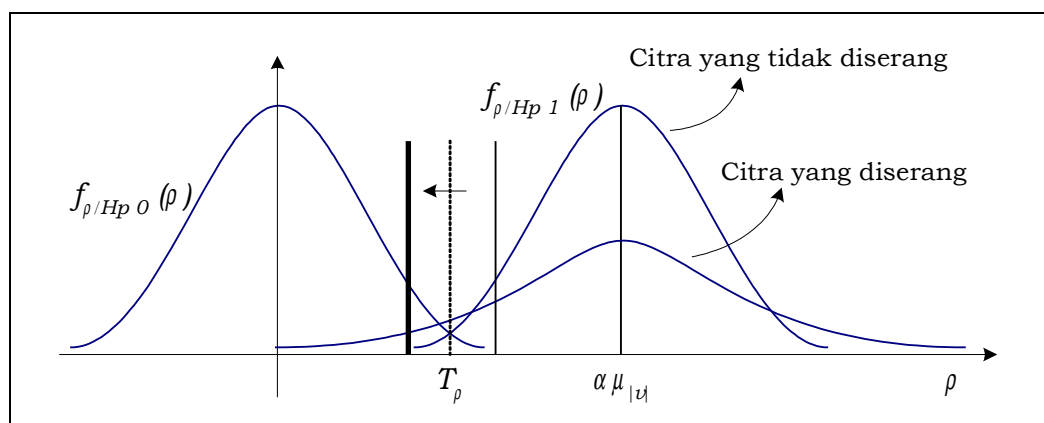
$$\mu_{|v|} ; \frac{1}{n} \sum_{i=1}^n |v'_i| \quad (3.15)$$



Gambar III-17: pdf dari korelasi ρ untuk hipotesis 0 dan 1. Tidak terjadi serangan pada citra.

Secara umum, akibat adanya serangan, distribusi pdf dari ρ untuk Hp.0 dan Hp.1 dapat dikatakan masih tetap ada, tetapi pdf dari Hp.1 yang berpusat pada $\mu_{|v|}$ sekarang memiliki varian yang cukup lebar, seperti terlihat pada gambar II-3. Hal ini mendorong pemilihan harga T_ρ sedekat mungkin dengan nol, meskipun masih tetap terletak diantara nol dan $\mu_{|v|}$, sehingga rumus T_ρ diubah menjadi:

$$T_\rho = \frac{\alpha}{3} \mu_{|v|} \quad (3.16)$$



Gambar III-18: pdf dari korelasi ρ untuk hipotesis 0 dan 1. Terjadi serangan pada citra.

Dari beberapa hasil uji-coba serangan terhadap citra yang ditanami watermark, menunjukkan bahwa harga *threshold* diatas (persamaan (3.16)), ternyata tidak sepenuhnya bekerja sesuai dengan yang diharapkan. Dalam prakteknya, probabilitas pendeteksi yang menyatakan watermark tidak ditandai meksipun sebenarnya ia ada (*false negative*), ternyata lebih besar dibandingkan dengan probabilitas dekoder mendeteksi keberadaan watermark yang sebenarnya tidak ada (*false positive*). Penjelasan akan fenomena ini dapat dilihat dari fakta bahwa akibat serangan biasanya $\mu_{\rho|Hp,1} < \alpha \mu_{|\rho|}$ (lihat pada gambar II-4). Untuk mencari jalan keluar dari permasalahan ini, diperlukan pendekatan yang berbeda untuk menentukan harga T_ρ . Piva et al (1998) menetapkan batasan tetap bagi probabilitas maksimum untuk kasus *false positive* (misalnya 10^{-6}), untuk itulah *threshold* T_ρ digeser lagi kekiri (perhatikan lagi gambar II-4). Sehingga jika diberikan $P_f = P(\rho > T_\rho | 0) = \frac{1}{2} \text{erfc}\left(\frac{T_\rho}{\sqrt{2\sigma_\rho^2}}\right) = 10^{-6}$, maka didapat hubungan berikut

(Piva et al, 1998):

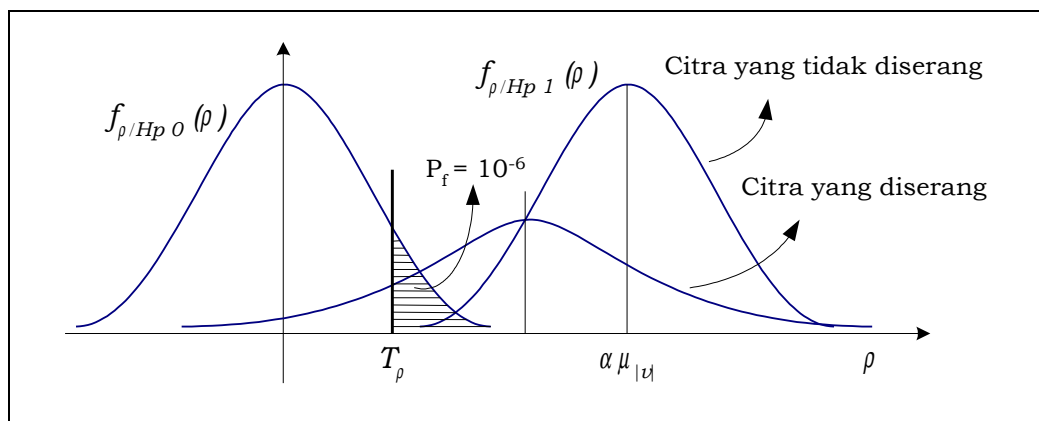
$$\frac{T_\rho}{\sqrt{2\sigma_\rho^2}} \geq 3.3 \quad (3.17)$$

dan akhirnya didapat juga harga *threshold* yang baru (Piva et al, 1998):

$$T_\rho = 3.3\sqrt{2\sigma_\rho^2} = 3.3\sqrt{\frac{2(1+\alpha^2)\sigma_v^2}{n}} \quad (3.18)$$

Threshold ini dapat dihitung secara langsung pada citra yang ditanami watermark dan mungkin juga telah terkorupsi : harga $(1 + \alpha^2)\sigma_v^2$ dapat diganti dengan σ_v^2 , sehingga didapat (Piva et al, 1998):

$$T_\rho = 3.3 \sqrt{\frac{2\sigma_v^2}{n}} \quad (3.19)$$



Gambar III-19: Penentuan *threshold* yang baru, berdasarkan pada batasan maksimal probabilitas *false positive*.

BAB IV IMPLEMENTASI DAN PEMBAHASAN HASIL UJI COBA

Pada bab ini dibahas implementasi algoritma watermarking digital pada domain DCT untuk citra berwarna dan dilanjutkan dengan pembahasan hasil-hasil percobaan pada citra yang telah ditanami watermark.

Implementasi Algoritma Watermarking Digital pada Domain DCT untuk Citra Berwarna

BAB XXIV Sarana Implementasi

Untuk mengimplementasikan algoritma ini penulis menggunakan perangkat keras dan perangkat lunak berikut ini :

IV.1.1.1 Perangkat Keras

Perangkat keras yang penulis gunakan berupa seperangkat komputer personal (PC) yang memiliki spesifikasi sebagai berikut :

Tabel IV-1 : Spesifikasi Perangkat Keras

Jenis Perangkat	Spesifikasi
Prosesor	Intel Pentium Celeron 466 MHz
Memori	Visipro SDRAM 128 MB
Hardisk	Quantum 8,4 GB
Kartu VGA / AGP	AGP Riva TNT 8 MB

IV.1.1.2 Perangkat Lunak

Sedangkan perangkat lunak yang digunakan oleh penulis didalam implementasi algoritma ini adalah sebagai berikut :

1. Sistem Operasi Windows 98 dari Microsoft Inc.
2. Matlab Release 12 dari MathWorks Inc.
3. ACD FotoCanvas Lite versi 2.0 dari ACD System Ltd.
4. ACDDSee versi 5.0 dari ACD System Ltd.

BAB XXV Implementasi Algoritma Watermarking Digital

Penulis dalam skripsi ini hanya mengimplementasikan prosedur penyisipan watermark dan prosedur pengujian keberadaan watermark. Prosedur pertama mengimplementasikan **Algoritma 3.1 : Algoritma Penyisipan Watermark** yang dibahas pada bagian 3.1.3 bab sebelumnya, sedangkan prosedur kedua mengimplementasikan pengujian watermark yang didasarkan pada **Algoritma 3.2 : Algoritma Pendeteksian Watermark**, prosedur ini hanya menunjukkan respon pendeteksi atas citra terwatermark yang merupakan masukan dari prosedur ini.

IV.1.1.3 Pembuatan Watermark

Watermark sebagaimana yang telah dibahas pada bab sebelumnya merupakan sebuah vektor bilangan real yang terdistribusi normal dan memiliki nilai mean nol dan variansi satu. Vektor watermark dengan panjang n dapat dibangkitkan secara acak oleh komputer dengan hanya memberikan sebuah kunci tertentu, atau dengan kata lain dengan kunci yang sama juga akan didapat vektor watermark yang serupa. Untuk mengimplementasikan

pembuatan watermark ini, penulis memanfaatkan perintah-perintah yang telah disediakan oleh Matlab. Dalam hal ini Matlab menyediakan fungsi `randn`. Fungsi ini bermanfaat untuk membangkitkan senarai bilangan acak yang elemennya terdistribusi dengan mean 0, variansi $\sigma^2 = 1$ dan standar deviasi $\sigma = 1$ (The Mathworks Inc., 2000b). Ada beberapa *syntax* fungsi `rand` yang bisa digunakan, diantaranya adalah seperti dalam tabel dibawah ini:

Tabel IV-2 : *Syntax* fungsi `randn`

Syntax	Keterangan
<code>R = randn(m,n)</code> atau <code>R = randn([m n])</code>	untuk membangkitkan bilangan acak dalam bentuk matriks $m \times n$, dan hasilnya disimpan didalam variabel R.
<code>randn('state', j)</code>	untuk me- <i>reset</i> status keadaan (<i>state</i>) dari generator pada nilai j, tetapi jika j adalah integer maka generator akan di- <i>reset</i> pada status keadaan (<i>state</i>) ke-j.

Untuk membangkitkan watermark, yang perlu dilakukan pertama adalah dengan mengisi status keadaan (*state*) di dalam generator dengan suatu nilai, pada contoh dibawah ini penulis mengisi `state` dengan nilai 46.

```
randn('state', 46);
```

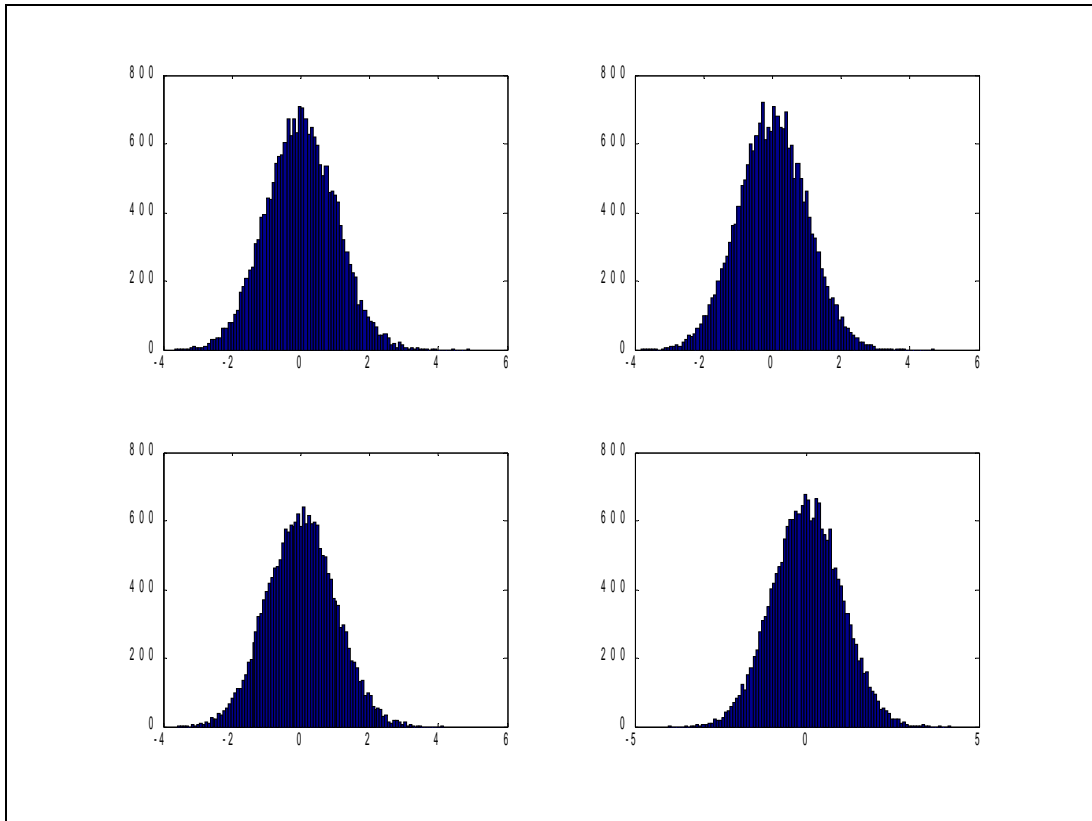
Kemudian vektor watermark dengan panjang n dapat dibangkitkan dengan memberikan perintah berikut:

```
Wm = randn(n, 1);
```

Perintah diatas menghasilkan sebuah vektor watermark dengan panjang n yang disimpan didalam variabel `Wm`.

Beberapa contoh watermark ($n = 20000$ dan masing masing *state*-nya adalah 7, 25, 46 79) dapat dilihat dari gambar grafik histogram watermark tersebut pada

gambar IV-1. Dari gambar tersebut terlihat bahwa histogram watermark-watermark tersebut terdistribusi secara normal disekitar nol.



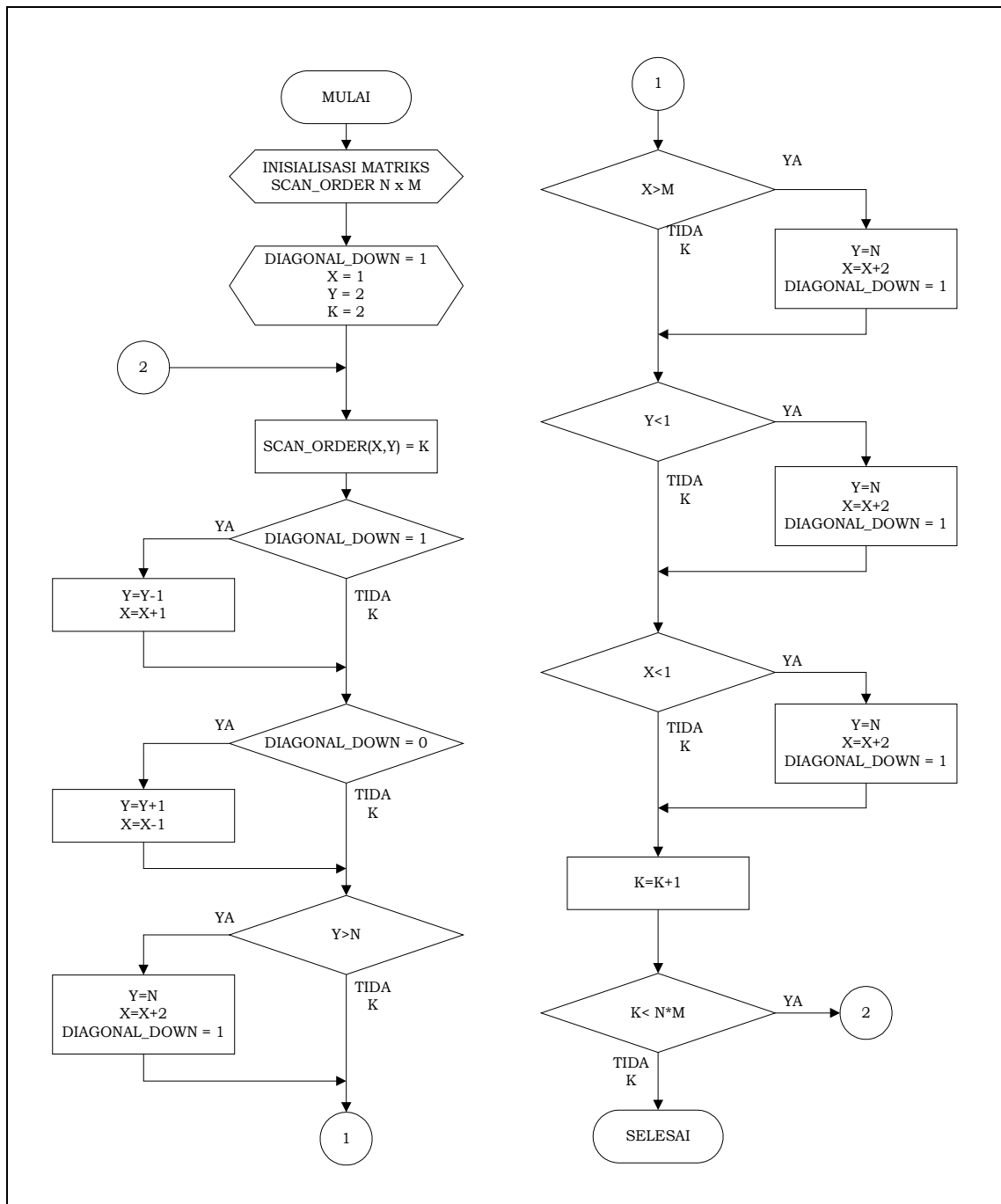
Gambar IV-20 : Histogram empat buah watermark dengan $n = 10000$ dan masing-masing *state*-nya adalah 7,25,46,79.

IV.1.1.4 Fungsi Scan Zig-zag

Didalam ulasan algoritma penanaman dan pendeteksian watermark pada bab 3, disinggung adanya proses pengurutan koefisien DCT menggunakan scan zig-zag sebagaimana yang telah digunakan didalam algoritma kompresi citra JPEG (Wallace, 1991). Scan zig-zag dapat dilihat dalam gambar IV-2. Implementasi dari scan zig-zag ini penulis lakukan berdasarkan pada sumbangan ide dari Cagatay Candan, seorang mahasiswa doktoral dari *School of ECE Georgia Institute of Technology*.

Matlab merupakan bahasa matriks, artinya bahwa ia dirancang untuk operasi vektor dan matriks, sehingga didalam pemrograman dengan Matlab penggunaan struktur perulangan (loop) harus diminimalkan. Berdasarkan pemikiran inilah fungsi scan zig-zag ditujukan untuk membuat keluaran berupa sebuah matriks yang berisi indeks scan zig-zag. Pembuatan fungsi yang penulis namai `jpeg_scan` ini didasarkan pada flowchart pada gambar IV-2. Fungsi ini memerlukan masukan ukuran baris dan kolom matriks zig-zag yang akan dibuat, dan menghasilkan keluaran matriks dua dimensi dengan ukuran $N \times M$. Berikut ini adalah kode sumber fungsi pemindaian zig-zag secara lengkap:

```
function out=jpeg_scan(N,M);  
  
scan_order = zeros(N,M);  
scan_order(1,1) = 1;  
diag_down = 1; x = 1; y = 2;  
for k = 2:N*M,  
    scan_order(x,y) = k;  
    if diag_down==1, y=y-1;x=x+1; end;  
    if diag_down==0, y=y+1;x=x-1; end;  
    if y>N, y=N; x=x+2; diag_down=1; end;  
    if x>M, x=M; y=y+2; diag_down=0; end;  
    if y<1, y=1; diag_down=0; end;  
    if x<1, x=1; diag_down=1; end;  
end;  
out=scan_order;
```



Gambar IV-21 : Flowchart scan zig-zag.

IV.1.1.5 Fungsi Penyisipan Watermark

Untuk mengimplementasikan algoritma penyisipan watermark diatas, penulis terlebih dahulu membuat diagram alir (*flowchart*) seperti pada gambar IV-3. Kemudian dengan menggunakan perintah-perintah yang ada pada program Matlab penulis mengimplementasikan algoritma tersebut kedalam fungsi `WM_EMBED`. Fungsi ini membutuhkan sebuah parameter wajib (harus ada) yang berupa sebuah matriks citra berwarna, dan empat parameter opsional (tidak harus ada), yaitu `L`, `n`, `a` dan `seed`. `L` merupakan banyaknya koefisien DCT pertama yang dilewati pada saat penyisipan watermark, `n` adalah panjang watermark yang disisipkan, `a` mewakili parameter koefisien penguat dan `seed` adalah kunci pembangkit watermark yang akan disisipkan. Khusus untuk untuk parameter `seed`, nilai yang bisa diinputkan pada parameter ini sebenarnya tidak dibatasi pada bilangan bulat positif tertentu, tetapi karena pada prosedur pendeteksian watermark (subbab 4.1.2.4 Prosedur pengujian keberadaan watermark) pendeteksian respon nilai korelasi terhadap watermark terbatas pada bilangan bulat positif antara 1 sampai dengan 1000, maka nilai `seed` yang dianjurkan adalah bilangan bulat positif pada jangkauan tersebut (penulis memberi nilai default parameter `seed` dengan angka 200). Hasil dari fungsi ini adalah sebuah citra berwarna yang telah tertanami watermark.

Penjelasan lebih lanjut akan cara kerja fungsi ini adalah seperti berikut ini: Fungsi pada awalnya melakukan pengecekan parameter yang dimasukkan oleh pengguna. Jika parameter opsional yang dibutuhkan kurang, maka fungsi akan menggunakan nilai default dari masing-masing parameter yang telah disediakan

oleh fungsi. Untuk melakukan pengecekan parameter masukan fungsi penulis menggunakan fungsi `nargin` berikut ini:

```
if nargin < 5,    seed = 200; end
if nargin < 4,    a = 0.2; end
if nargin < 3,    n = 20000; end
if nargin < 2,    L = 20000; end
```

Setelah proses pengecekan dijalankan, fungsi mengkonversi citra masukan dari ruang warna RGB kedalam ruang warna YCbCr dengan memanfaatkan fungsi `rgb2ycbcr`.

```
yCbCr = rgb2ycbcr(cRGB_wm);
```

Kemudian fungsi mengambil kanal *luminance*(Y) dari ruang warna YCbCr, dan membuat matriks yang berisi indeks urutan zig-zag array dua dimensi, sesuai dengan ukuran matriks kanal Y yang diambil sebelumnya. Pembuatan matriks ini menggunakan fungsi `jpeg_scan` yang telah dijelaskan pada bagian 4.1.2.2 **Prosedur Pemindaian Zig-zag**.

```
Y = double(yCbCr(:, :, 1));
[xm, xn] = size(Y);
scan_order = jpeg_scan(xm, xn);
```

Selanjutnya watermark dengan panjang n dibangkitkan berdasarkan parameter kunci `seed`. Penciptaan watermark ini menggunakan fungsi `randn`, sebagaimana yang telah dijelaskan pada bagian 4.1.2.1 **Pembuatan Watermark**.

```
randn('state', seed);
Wm = randn(n, 1);
```

Langkah berikutnya adalah mentransformasikan matriks kanal Y dengan DCT kedalam komponen frekuensi dasarnya dengan menggunakan fungsi `dct2`.

```
Y_dct = dct2(Y);
```

Matriks hasil transformasi DCT kemudian diurutkan dengan menggunakan matriks indeks diatas, dan menghasilkan sebuah vektor koefisien DCT yang telah diurutkan secara zig-zag.

```
temp = Y_dct(:);
zz_dct_koeff = temp;
zz_dct_koeff(scan_order(:)) = temp;
```

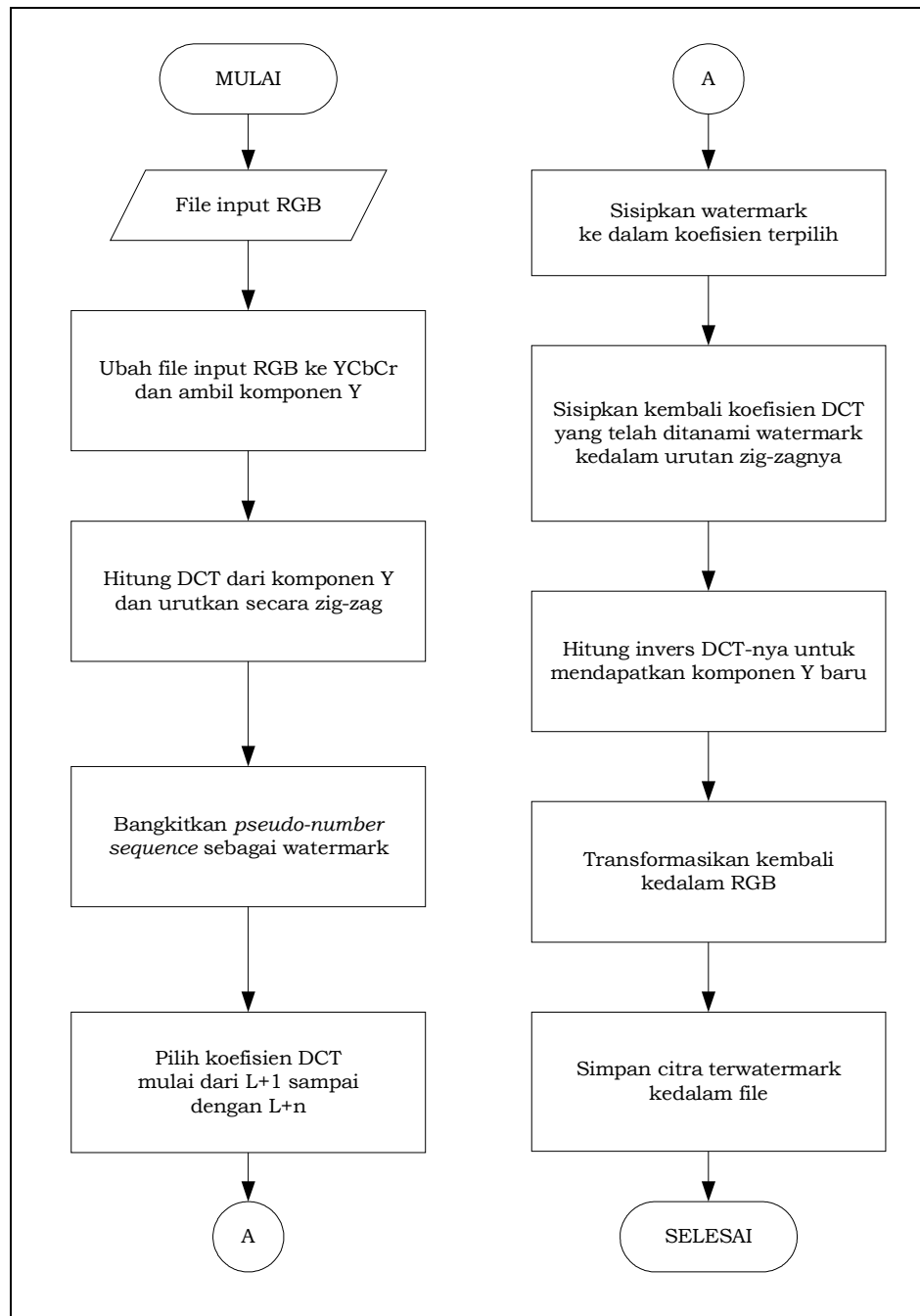
Sekumpulan koefisien DCT dari vektor diatas kemudian dipilih untuk disisipi watermark sesuai persamaan (3.1).

```
V = zz_dct_koeff(L+1:L+n);
V = V + a*abs(V).*Wm;
```

Selanjutnya, tinggal membalik proses-proses diatas untuk mendapatkan citra yang telah ditanami watermark.

```
zz_dct_koeff(L+1:L+n) = V;
temp = zz_dct_koeff(scan_order(:));
Yx_dct = reshape(temp, xm, xn);
Yx = uint8(idct2(Yx_dct));
yCbCr(:, :, 1) = Yx;
cRGB_out = ycbcr2rgb(yCbCr);
```

cRGB_out merupakan matriks hasil citra berwarna yang telah disisipi watermark, yang juga merupakan hasil keluaran dari fungsi WM_EMBED.



Gambar IV-22 : Diagram alir prosedur penanaman watermark

IV.1.1.6 Prosedur Pengujian Keberadaan Watermark

Implementasi prosedur pengujian eksistensi watermark didasarkan pada algoritma pendeteksian watermark yang telah diulas diatas (Algoritma 3.2). Seperti pada implementasi prosedur penanaman watermark, penulis menyusun diagram

alir prosedur yang akan diimplementasikan ini seperti pada gambar IV-4. Selanjutnya berdasarkan diagram alir ini penulis mengimplementasikannya pada program Matlab dalam bentuk sebuah fungsi. Fungsi ini penulis namakan WM_TEST. Fungsi WM_TEST membutuhkan masukan sebuah citra terwatermark, dan beberapa parameter lain yang diperlukan dalam pendeteksian watermark. Parameter-parameter ini haruslah sama dengan parameter opsional yang digunakan dalam prosedur penanaman watermark. Hasil keluaran dari fungsi ini adalah vektor korelasi dengan panjang 1000, yang merepresentasikan nilai korelasi setiap watermark dengan kumpulan koefisien DCT yang terpilih, dan nilai *threshold* yang merupakan nilai batas pendeteksian eksistensi watermark.

Cara kerja fungsi ini akan diulas lebih lanjut berikut ini: Pertama fungsi melakukan pengecekan terhadap beberapa parameter yang diperlukan fungsi, jika parameter yang bersangkutan tidak ada maka program menggunakan nilai default yang telah disediakan oleh fungsi. Fungsi `nargin` yang telah disediakan Matlab, digunakan penulis untuk melakukan pekerjaan ini.

```
if nargin < 4,    a = 0.2; end
if nargin < 3,    n = 20000; end
if nargin < 2,    L = 20000; end
```

Kemudian dilakukan transformasi terhadap citra masukan dari ruang warna asalnya RGB ke ruang warna YCbCr. Selanjutnya fungsi memilih kanal *luminance* (Y) dari hasil transformasi ruang warna diatas.

```
yCbCr = rgb2ycbcr(cRGB_wm);
Y = double(yCbCr(:, :, 1));
```

Proses berikutnya adalah membuat matriks urutan zig-zag dengan ukuran seperti pada matriks kanal *luminance*(Y) diatas.

```
[xm,xn] = size(Y);
scan_order = jpeg_scan(xm,xn);
```

Penghitungan nilai DCT dari matriks kanal *luminance*(Y) kemudian dilakukan dengan menggunakan fungsi *DCT2* dari Matlab. Dan hasilnya yang merupakan matriks dua dimensi dari koefisien DCT, selanjutnya ditransformasikan kedalam sebuah vektor sesuai dengan urutan zigzagnya.

```
Y_dct = dct2(Y);
temp = Y_dct(:);
zz_dct_koeff = temp;
zz_dct_koeff(scan_order(:)) = temp;
```

Berikutnya dipilih serangkaian koefisien DCT dari vektor yang dibuat diatas dan kemudian proses penghitungan nilai korelasi untuk setiap watermark yang dibangkitkan dimulai. Ada 1000 watermark yang dibangkitkan dengan kunci masukan bilangan bulat antara 1 sampai dengan 1000, dan setiap watermark ini selanjutnya dihitung nilai korelasinya dengan vektor koefisien DCT yang terpilih.

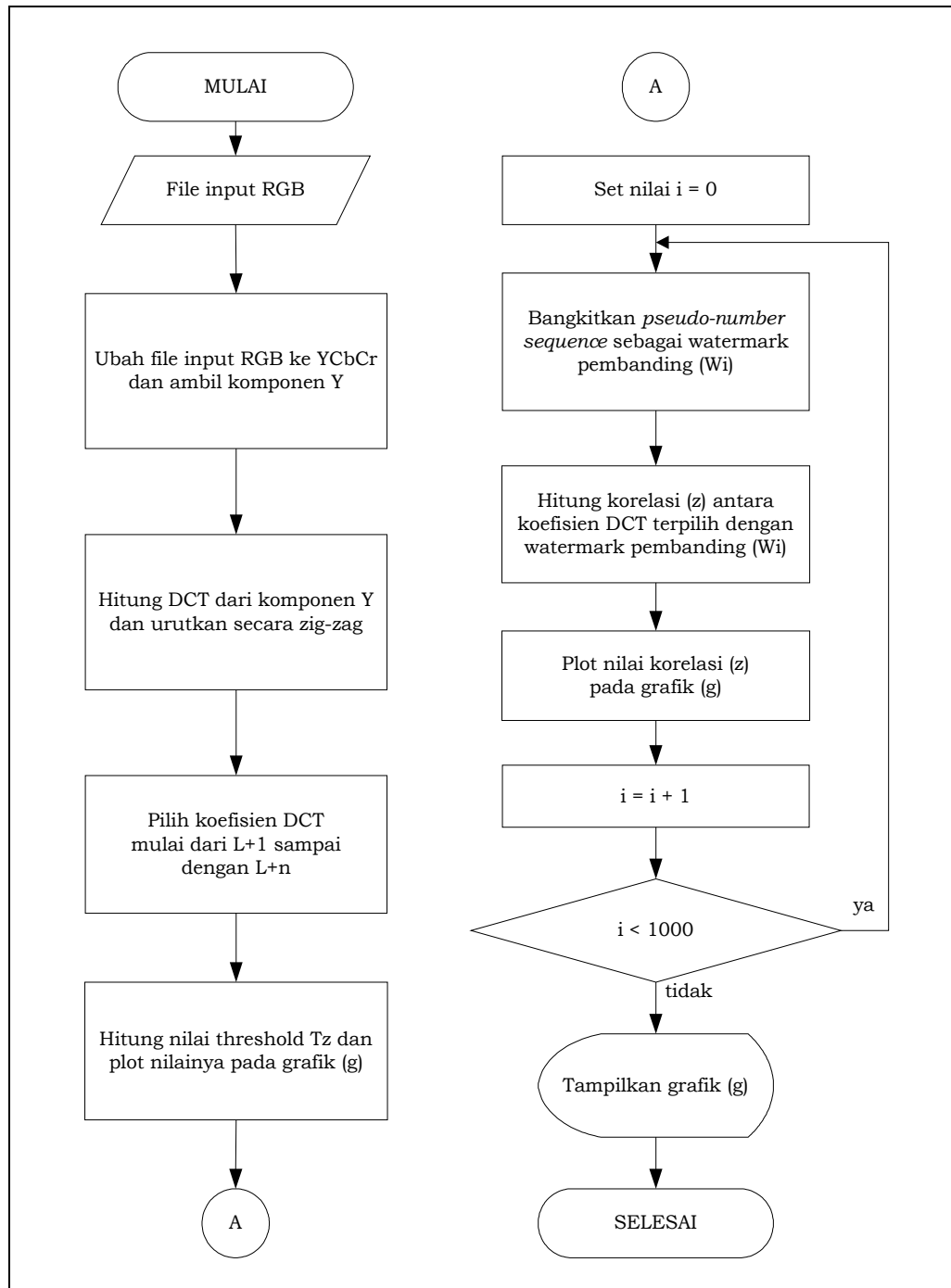
```
V = zz_dct_koeff(L+1:L+n);
for (i=1:1000),
    randn('state',i);
    W = randn(n,1);
    z(i) = mean(V.*W);
end;
```

Proses terakhir dari prosedur pendeteksian keberadaan watermark adalah menghitung nilai *threshold* berdasarkan persamaan (3.14), (3.16), dan (3.19) pada bab sebelumnya diatas.

```

th0 = 3.3*sqrt((2*(std(V)*std(V)))/n)*ones(1,1000);
th1 = (1/2)*a*mean(abs(V))*ones(1,1000);
th2 = (1/3)*a*mean(abs(V))*ones(1,1000);

```



Gambar IV-23 : Diagram alir prosedur pengujian keberadaan watermark

Uji Coba Hasil Implementasi

Berikut ini akan diulas beberapa hasil uji coba terhadap implementasi algoritma watermarking digital yang dibahas dalam penulisan skripsi ini. Pada bagian awal adalah hasil uji coba terhadap citra yang belum dikenai serangan, kemudian dilanjutkan terhadap citra yang dikenakan serangan. Dalam uji coba ini hasil pengujian ditunjukkan oleh grafik korelasi respon detektor antara watermark pembanding dari citra yang diuji. Tiga perhitungan harga *threshold* pada pembahasan bab tiga (persamaan 3.14, 3.16, dan 3.19) juga ditampilkan pada grafik untuk mengetahui kinerja dari ketiga *threshold* tersebut.

Secara umum, semua citra yang diuji berukuran 512x512 citra berwarna RGB 24 bit dalam format file BMP, TIFF atau JPEG. Setiap citra ditandai watermark dengan $n = 20000$, $L = 20000$ dan koefisien penguat $\alpha = 0.2$. Hasil pengujian terhadap serangan, berupa grafik respon detektor terhadap 1000 watermark pembanding yang diujikan pada citra terwatermark baik yang dikenai serangan atau tidak, selain itu kinerja ketiga *threshold* yang dibahas sebelumnya juga dapat diamati pada grafik.

BAB XXVII Hasil Uji Coba tanpa Serangan

Uji coba pertama dilakukan terhadap ketujuh citra berwarna yang penulis dapatkan dari (Petitcolas, 2003), karena citra-citra ini awalnya memiliki ukuran yang berbeda-beda, maka sebelum watermark ditanamkan, penulis menyeragamkan ukuran citra-citra tersebut kedalam ukuran 512x512 piksel, tetap dalam format citra aslinya TIFF (*Tag Image File Format*). Operasi yang dilakukan untuk menyesuaikan ukuran adalah dengan *cropping* dan *resizing* dengan aplikasi

pengolah gambar ACD FotoCanvas Lite versi 2.0 dari ACD System Ltd. Gambar IV-7 menunjukkan respon detektor terhadap 1000 watermark pembanding yang diujikan pada citra terwatermark pada gambar IV-6. Pada gambar IV-7 tersebut juga dapat dilihat kinerja ketiga *threshold* Tr_0, Tr_1 dan Tr_2 masing masing dari persamaan (3.14), (3.16), dan (3.19). Gambar IV-6 merupakan versi citra yang telah ditanami watermark dari citra aslinya pada gambar IV-5. Hasil pengujian pertama ini terhadap seluruh ini dapat dilihat pada tabel IV-3. Secara umum, pengujian pertama terhadap citra terwatermark ini menunjukkan kinerja yang sangat baik, atau dengan kata lain semua watermark yang ditanamkan dapat terdeteksi dengan baik oleh ketiga *threshold*.

Tabel IV-3: Hasil pengujian pada citra terwatermark tanpa serangan

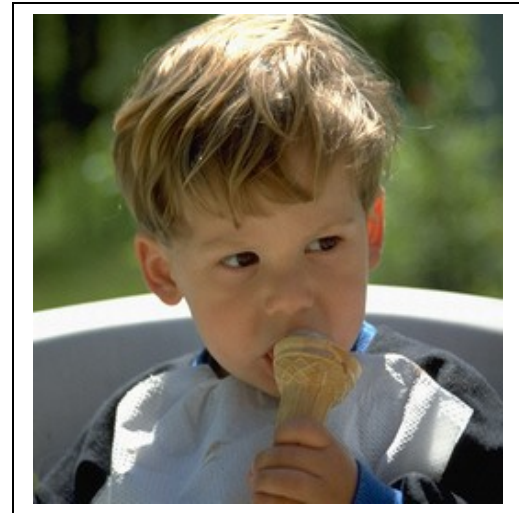
No	Citra	r_1	r_2	Tr_0	Tr_1	Tr_2
1	Skyline Arch	2.5387	0.4258	0.6007	1.3769	0.9179
2	Bear	1.8066	0.3225	0.4445	0.9878	0.6585
3	Baboon	5.3304	1.0001	1.1659	2.6819	1.7880
4	Newyork	5.2909	0.7577	1.2214	2.7911	1.8607
5	F16	3.0358	0.5960	0.7177	1.5938	1.0626
6	Lena	2.6233	0.3385	0.5617	1.2372	0.8248
7	Kid	2.5474	0.4123	0.5272	1.2174	0.8116

Keterangan : r_1 = respon korelasi tertinggi;

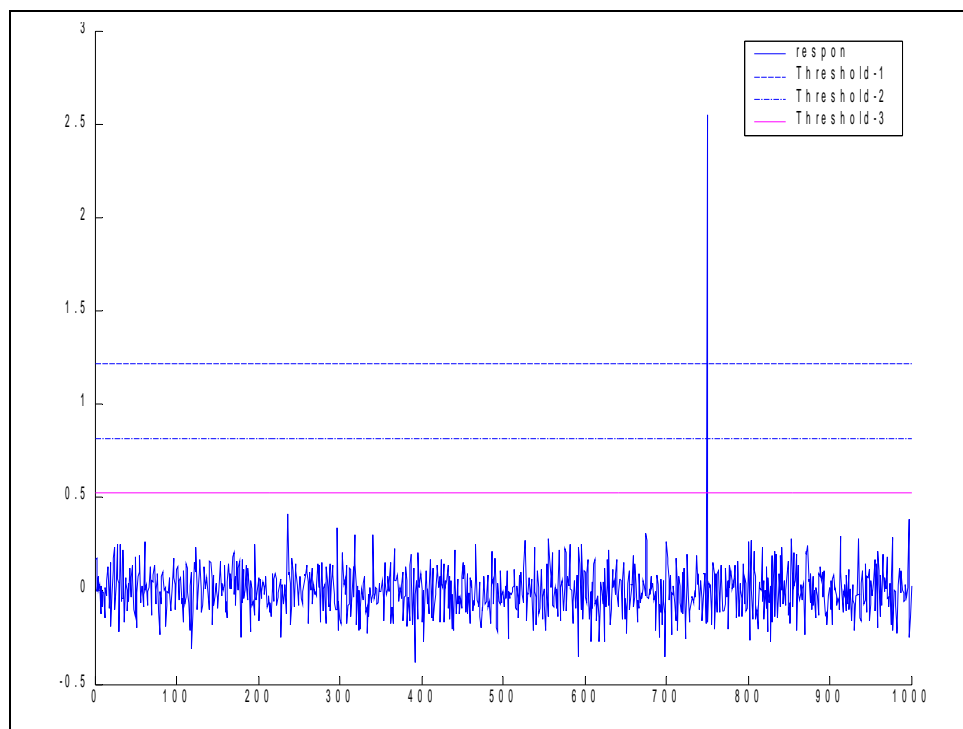
r_2 = respon korelasi tertinggi kedua;



Gambar IV-24 : Citra asli Kid.
Original image courtesy of Karel de Gendre



Gambar IV-25 : Versi terwatermark dari
citra Kid



Gambar IV-26 : Respon detektor terhadap 1000 watermark pengujian pada citra Kid terwatermark dan kinerja ketiga *threshold*

BAB XXVIII Hasil Uji Coba dengan Serangan

Uji coba selanjutnya adalah dengan melakukan beberapa simulasi serangan terhadap citra terwatermark. Beberapa serangan tersebut antara lain adalah kompresi JPEG, perataan histogram, median *filtering*, *window averaging filtering*, penambahan *noise* dan *resizing*. Pada percobaan dengan serangan *threshold* yang menjadi acuan adalah *threshold* Tr_0 yang berasal dari persamaan (3.19).

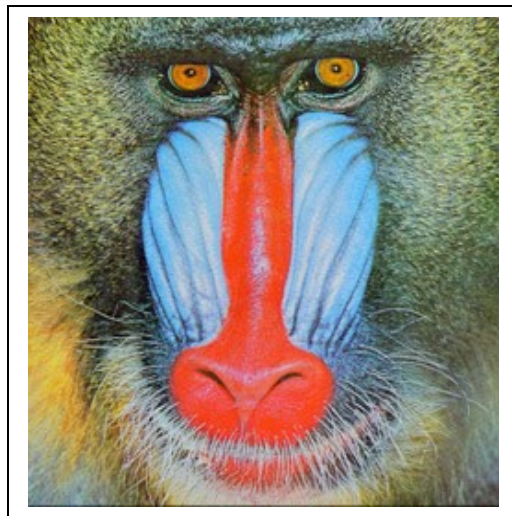
IV.1.1.7 Kompresi JPEG

JPEG dapat dikatakan sebagai kompresi citra yang sangat populer saat ini, oleh sebab itu sebuah algoritma watermarking haruslah tahan terhadap kompresi citra yang satu ini. Penulis melakukan percobaan kompresi JPEG pada citra terwatermark dengan beragam kualitas kompresi. Secara umum dari hasil percobaan terhadap ketujuh citra tersebut menunjukkan detektor masih mampu mendeteksi keberadaan watermark pada citra dengan kualitas kompresi 10% atau dengan rasio kompresi rata-rata 1 : 70,2. Untuk kualitas kompresi yang lebih kecil lagi satu persatu watermark pada citra yang diuji mulai tidak dapat dideteksi. Meskipun demikian, ternyata dengan kualitas kompresi 0%, masih ada citra terwatermark yang bisa dideteksi oleh *threshold* Tr_0 , watermark tersebut tertanam pada citra Baboon. Pada citra Baboon ini rasio kompresi mencapai 1 : 121,4. Tabel IV-4 menunjukkan Hasil kerja detektor terhadap citra terwatermark dengan beragam kualitas kompresi JPEG.

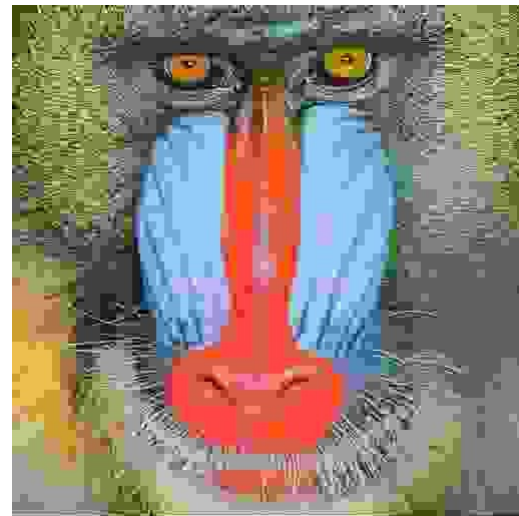
Tabel IV-4 : Hasil pengujian pada citra terwatermark dengan beragam kualitas kompresi JPEG

No .	Citra	Kualitas kompresi JPEG					
		0%	5%	7%	8%	9%	10%
1	Skyline Arch	-	-	-	√	-	√
2	Bear	-	-	-	√	√	√
3	Baboon	√	√	√	√	√	√
4	Newyork	-	√	√	√	√	√
5	F16	-	-	-	-	-	√
6	Lena	-	-	-	√	√	√
7	Kid	-	-	√	√	√	√

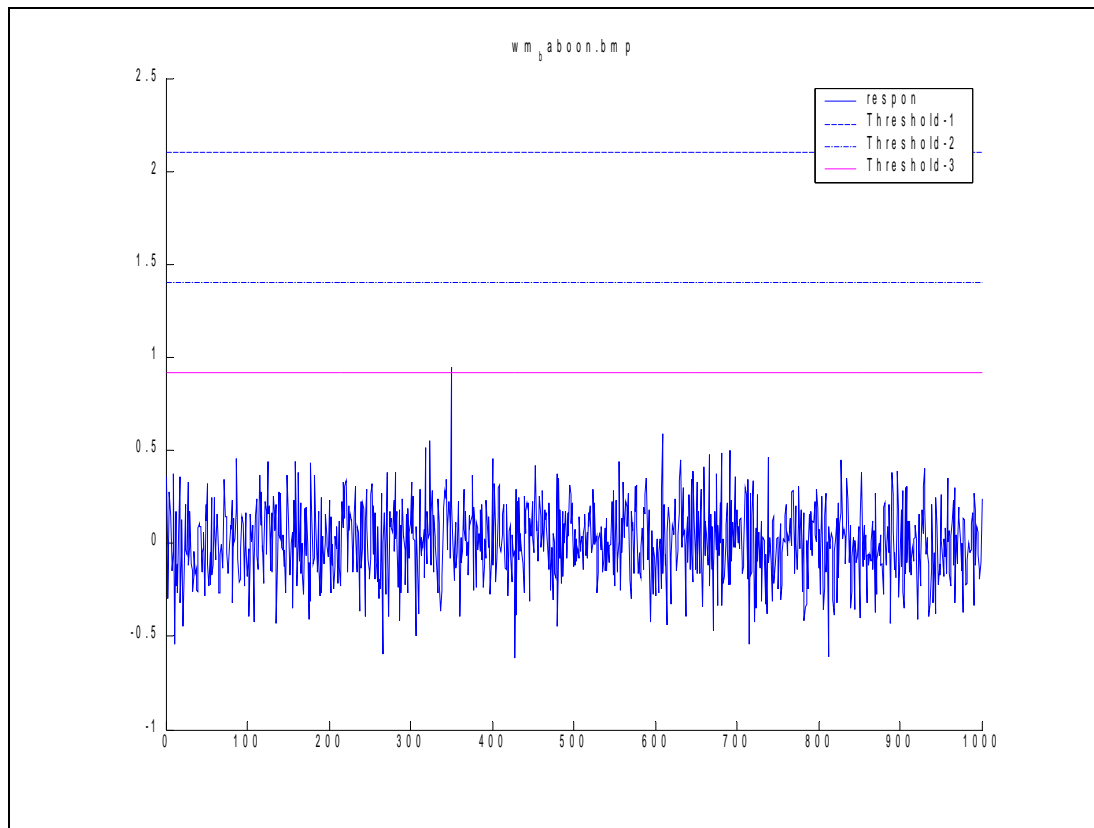
Keterangan : √ : watermark terdeteksi oleh Tr_0 .
 - : watermark tidak terdeteksi Tr_0 .



Gambar IV-27 : Citra Baboon terwatermark. *Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.*



Gambar IV-28 : Hasil kompresi JPEG dengan kualitas 0% dari citra Baboon terwatermark



Gambar IV-29 : Respon detektor terhadap 1000 watermark penguji pada citra Baboon terwatermark dengan kompresi JPEG 0%.

IV.1.1.8 Perataan Histogram (*Histogram Equalization*)

Percobaan dengan perlakuan perataan histogram memberikan hasil sebagai berikut: Respon detektor terhadap watermark ternyata terletak cukup jauh di atas *threshold* $\boxed{Tr_3}$, bahkan *threshold* $\boxed{Tr_1}$ dan $\boxed{Tr_2}$ dapat dengan mudah mendeteksi keberadaan watermark. Fenomena menarik terlihat pada citra New York, hasil perataan histogram pada citra ini menampakkan beberapa distorsi yang disebabkan oleh penanaman watermark. Meskipun demikian secara umum hasil perataan histogram pada citra terwatermark bertekstur, memperlihatkan hasil yang cukup baik.

Tabel IV-5 : Hasil pengujian pada citra terwatermark dengan Perataan Histogram

No	Citra	r_1	r_2	Tr_0	Tr_1	Tr_2
1	Skyline Arch	5.1579	0.3595	0.5029	1.0846	0.7231
2	Bear	2.0313	0.2695	0.4574	1.0270	0.6846
3	Baboon	6.9247	1.4406	1.6361	3.7917	2.5278
4	Newyork	13.6937	0.8914	1.3855	2.9977	1.9985
5	F16	11.8377	0.8502	1.2990	2.9534	1.9690
6	Lena	3.2563	0.4511	0.7304	1.6135	1.0756
7	Kid	4.4295	0.5009	0.6483	1.4747	0.9832

Keterangan : r_1 = respon korelasi tertinggi;

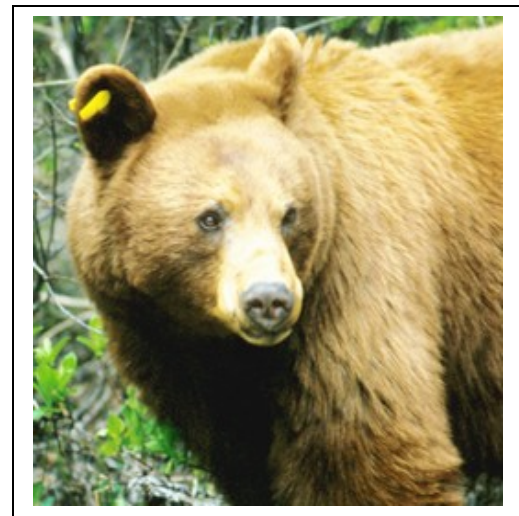
r_2 = respon korelasi tertinggi kedua;

$$T\rho_0 = 3.3\sqrt{\frac{2\sigma^2}{n}}; \quad T\rho_1 = \frac{\alpha}{2}\mu_{|v|}; \quad T\rho_2 = \frac{\alpha}{3}\mu_{|v|}$$

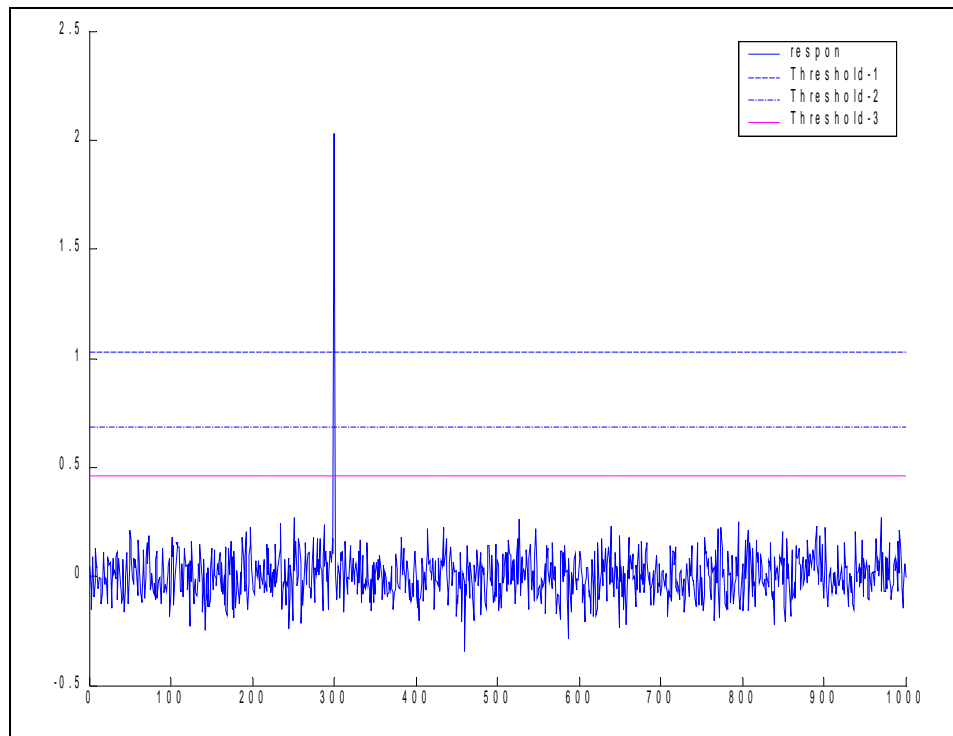


Gambar IV-30 : Citra Bear terwatermark.

Original image courtesy of Robert E. Barber, Barber Nature Photography.



Gambar IV-31 : Hasil perataan histogram pada citra Bear terwatermark



Gambar IV-32 : Respon detektor terhadap 1000 watermark penguji pada citra Bear terwatermark dengan perlakuan perataan histogram.

IV.1.1.9 Window averaging

Percobaan *window averaging* dengan beragam ukuran *window*, yaitu : 3x3, 5x5 dan 7x7, dilakukan terhadap ketujuh citra terwatermark. Hasil percobaan menunjukkan dengan menggunakan ukuran *window* 3x3 dan 5x5 , detektor masih mampu menunjukkan kemampuannya dalam mendeteksi keberadaan watermark, meskipun dengan *window* 5x5 citra sudah mulai berkurang kualitasnya. Sedangkan hasil beberapa percobaan lainnya, dengan *window* 7x7, keberadaan watermark tidak dapat dideteksi.

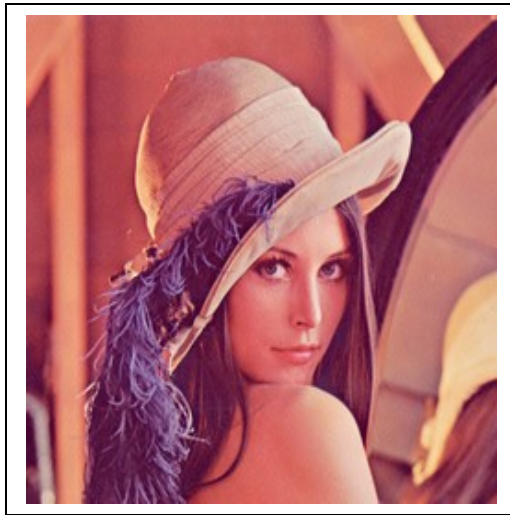
Tabel IV-6 : Hasil pengujian *window averaging* pada citra terwatermark.

No	Citra	Window	r_1	r_2	Tr_0	Tr_1	Tr_2
1	Skyline Arch	3 x 3	1.4014	0.2913	0.3937	0.8654	0.5769
		5 x 5	0.3091	0.1069	0.1633	0.3003	0.2002
		7 x 7	0.0585	0.0509	0.0810	0.1622	0.1081
2	Bear	3 x 3	1.0832	0.1771	0.3024	0.6400	0.4267
		5 x 5	0.2833	0.1032	0.1369	0.2454	0.1636
		7 x 7	0.0469	0.0386	0.0587	0.1180	0.0787
3	Baboon	3 x 3	3.0413	0.5587	0.7030	1.5731	1.0487
		5 x 5	0.5432	0.2009	0.2590	0.4813	0.3209
		7 x 7	0.1145	0.1029	0.1575	0.3140	0.2094
4	Newyork	3 x 3	2.8472	0.4585	0.7157	1.6066	1.0710
		5 x 5	0.3672	0.1914	0.2804	0.4953	0.3302
		7 x 7	0.1206	0.1126	0.1994	0.3770	0.2514
5	F16	3 x 3	1.7880	0.3725	0.4814	0.9528	0.6352
		5 x 5	0.3768	0.1236	0.2134	0.3137	0.2091
		7 x 7	0.0661	0.0640	0.0986	0.1903	0.1269
6	Lena	3 x 3	1.5646	0.2160	0.3921	0.8017	0.5344
		5 x 5	0.3748	0.1021	0.1825	0.2925	0.1950
		7 x 7	0.0459	0.0454	0.0775	0.1451	0.0967
7	Kid	3 x 3	1.5630	0.2960	0.3543	0.7778	0.5186
		5 x 5	0.3881	0.1233	0.1551	0.2786	0.1857
		7 x 7	0.0582	0.0442	0.0723	0.1452	0.0968

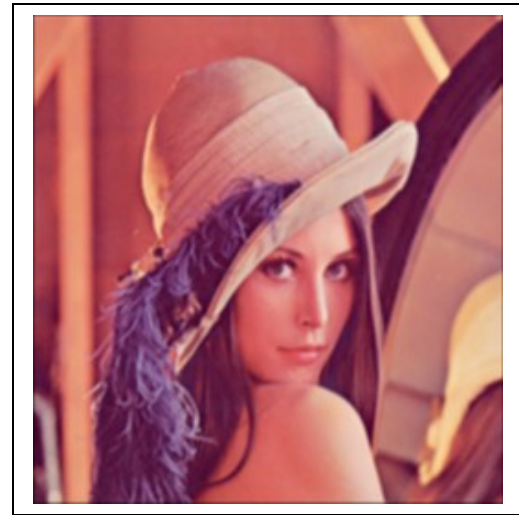
Keterangan : r_1 = respon korelasi tertinggi;

r_2 = respon korelasi tertinggi kedua;

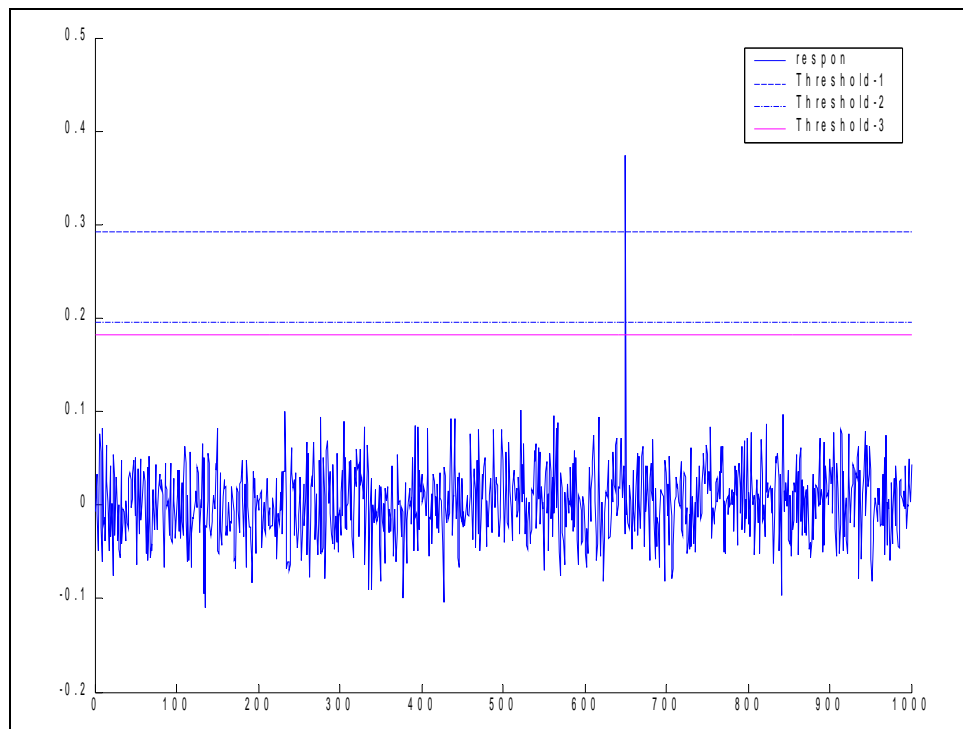
$$T\rho_0 = 3.3\sqrt{\frac{2\sigma^2}{n}}; \quad T\rho_1 = \frac{\alpha}{2}\mu_{|v|}; \quad T\rho_2 = \frac{\alpha}{3}\mu_{|v|}$$



Gambar IV-33 : Citra Lena terwatermark. *Original image courtesy of the Signal and Image Processing Institute at the University of Southern California*



Gambar IV-34 : Hasil filter *window averaging* (5x5) pada citra Lena terwatermark



Gambar IV-35 : Respon detektor terhadap 1000 watermark penguji pada citra F16 terwatermark dengan perlakuan filter *window averaging* (5x5).

IV.1.1.10 Median filtering

Seperti pada percobaan sebelumnya, *window averaging*, percobaan dengan *median filtering* menggunakan beragam ukuran *window*, yaitu : 3x3, 5x5 dan 7x7, juga dilakukan terhadap ketujuh citra terwatermark. Dengan menggunakan ukuran *window* 3x3 dan 5x5, detektor masih mampu mendeteksi keberadaan watermark. Sedangkan hasil percobaan dengan *window* 7x7, memperlihatkan keberadaan watermark tidak mampu dideteksi.

Tabel IV-7: Hasil pengujian *median filtering* pada citra terwatermark.

No	Citra	Window	r_1	r_2	Tr_0	Tr_1	Tr_2
1	Skyline Arch	3 x 3	1.6620	0.3592	0.4874	1.1032	0.7354
		5 x 5	0.5155	0.2427	0.3433	0.7589	0.5060
		7 x 7	0.2033	0.1910	0.2819	0.6213	0.4142
2	Bear	3 x 3	1.2823	0.2388	0.3513	0.7659	0.5106
		5 x 5	0.4935	0.1506	0.2221	0.4809	0.3206
		7 x 7	0.1095	0.1020	0.1649	0.3802	0.2534
3	Baboon	3 x 3	3.2033	0.6508	0.8411	1.8909	1.2606
		5 x 5	0.7590	0.2987	0.4144	0.8031	0.5354
		7 x 7	0.2282	0.2207	0.3153	0.5486	0.3658
4	Newyork	3 x 3	3.2119	0.5050	0.8178	1.9002	1.2668
		5 x 5	0.5102	0.2473	0.4023	0.9396	0.6264
		7 x 7	0.2078	0.1847	0.3071	0.7216	0.4811
5	F16	3 x 3	1.9112	0.5018	0.5968	1.2722	0.8481
		5 x 5	0.4815	0.3383	0.4541	0.8907	0.5938
		7 x 7	0.2748	0.2619	0.3975	0.7487	0.4992
6	Lena	3 x 3	1.7197	0.2678	0.4475	0.9683	0.6455
		5 x 5	0.6157	0.1959	0.3003	0.6439	0.4292
		7 x 7	0.1821	0.1529	0.2357	0.5188	0.3458
7	Kid	3 x 3	1.6920	0.3171	0.4117	0.9373	0.6249
		5 x 5	0.5745	0.2283	0.2790	0.6257	0.4171
		7 x 7	0.1718	0.1644	0.2322	0.5333	0.3555

Keterangan : r_1 = respon korelasi tertinggi;

r_2 = respon korelasi tertinggi kedua;

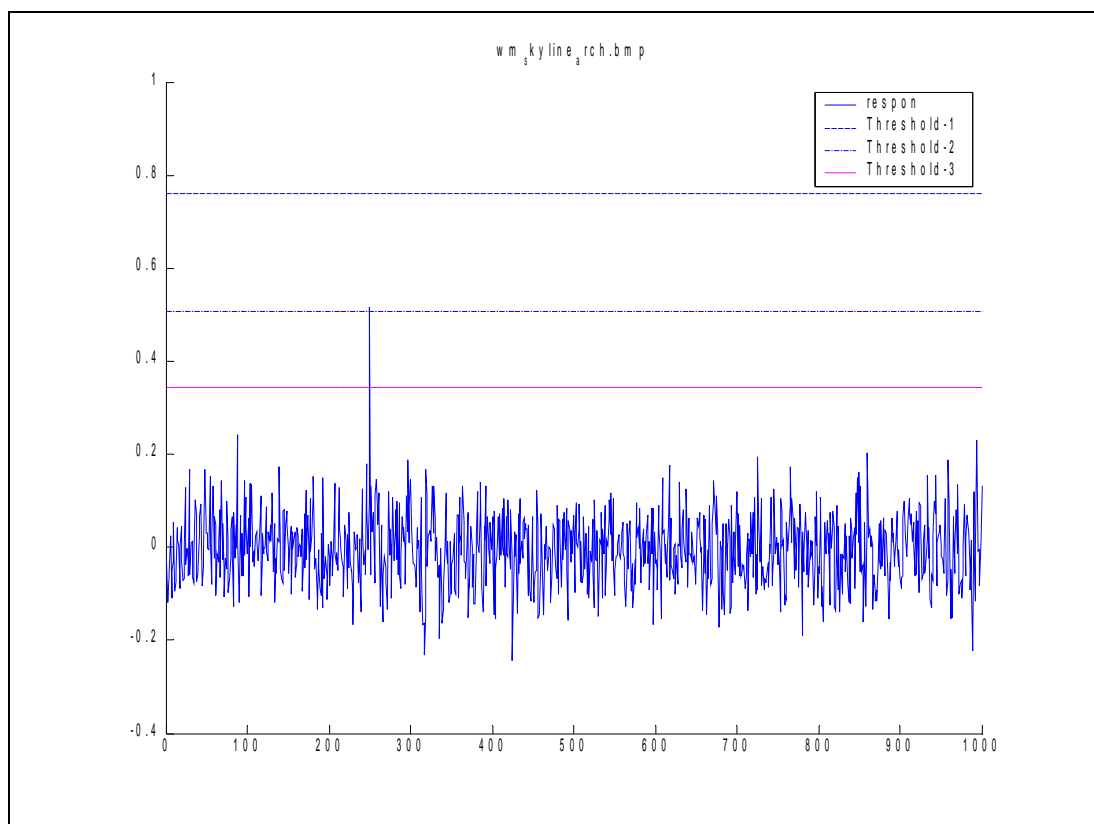
$$T\rho_0 = 3.3\sqrt{\frac{2\sigma_{v'}^2}{n}}; \quad T\rho_1 = \frac{\alpha}{2}\mu_{|v|}; \quad T\rho_2 = \frac{\alpha}{3}\mu_{|v|}$$



Gambar IV-36 : Citra Skyline Arch terwatermark. *Original image courtesy of Robert E. Barber, Barber Nature Photography.*



Gambar IV-37 : Hasil *median filtering* dengan *window 5x5* pada citra Skyline Arch terwatermark.



Gambar IV-38 : Respon detektor terhadap 1000 watermark penguji pada citra Skyline Arch terwatermark dengan *median filtering* (*window 5x5*).

IV.1.1.11 Penambahan *noise*

Pada percobaan penambahan *noise* ini, penulis menggunakan ketiga macam *noise* yang telah disediakan oleh program Matlab, yaitu: *gaussian noise*, '*salt & pepper*' *noise* dan *speckle noise*.

Percobaan penambahan *noise* yang pertama memanfaatkan *gaussian noise* dengan *mean* = 0 dan beragam harga *variance*, menunjukkan untuk nilai *variance* = 0.09 detektor mampu mendeteksi keberadaan watermark. Kemudian dengan menaikkan harga *variance* ini pada angka 0.15, hanya citra Bear saja yang tidak dapat dideteksi. Dan terakhir dengan nilai *variance* 0.79 hanya watermark pada citra Baboon dan citra New York saja yang masih mampu dideteksi.

Percobaan selanjutnya dengan menggunakan '*salt & pepper*' *noise* dengan nilai kepadatan (*density*) = 0.25. Hasilnya memperlihatkan detektor masih mampu mendeteksi keberadaan watermark. Kemudian dengan nilai *density* = 0.35 hanya citra Skyline Arch dan Bear saja yang tidak dapat dideteksi. Untuk nilai kepadatan 0.5 pada penambahan *noise* ini memperlihatkan hasil watermark masih dapat dideteksi pada citra Baboon dan New York.

Percobaan dengan penambahan *noise* ini diakhiri dengan mencoba penambahan *speckle noise* pada citra terwatermark. Angka pertama untuk *variance* dari *noise* yang diujicobakan adalah 0.24, hasilnya menunjukkan semua watermark dapat terdeteksi dengan baik. Kemudian dengan nilai *variance* 0.7, memperlihatkan watermark pada citra F16 dan Lena tidak dapat dideteksi keberadaannya oleh detektor. Dan untuk percobaan dengan harga *variance* 5, ternyata menunjukkan citra Baboon dan New York masih dapat dideteksi keberadaan watermarknya oleh detektor.

Secara umum pengujian dengan menambahkan *noise* memperlihatkan algoritma ini masih mampu bertahan pada penambahan *noise* dengan kuantitas yang tidak terlalu berat, bahkan untuk kasus tertentu, dalam hal ini citra Baboon dan New York, memperlihatkan hasil yang sangat baik, karena kualitas citra setelah penambahan *noise* sudah sangat menurun (lihat gambar IV-21). Meskipun demikian, detektor masih mendeteksi eksistensi watermark pada citra tersebut.

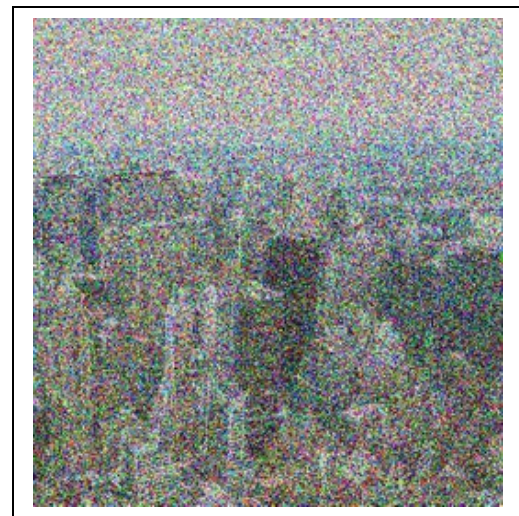
Tabel IV-8 : Hasil pengujian penambahan *noise* pada citra terwatermark

No	Citra	Gaussian (mean=0)			Salt & pepper			Speckle		
		Variance			Density			Variance		
		0.09	0.15	0.79	0.25	0.35	0.5	0.24	0.7	5
1	Skyline Arch	√	√	-	√	-	-	√	√	-
2	Bear	√	-	-	√	-	-	√	√	-
3	Baboon	√	√	√	√	√	√	√	√	√
4	New York	√	√	√	√	√	√	√	√	√
5	F16	√	√	-	√	√	-	√	-	-
6	Lena	√	√	-	√	√	-	√	-	-
7	Kid	√	√	-	√	√	-	√	√	-

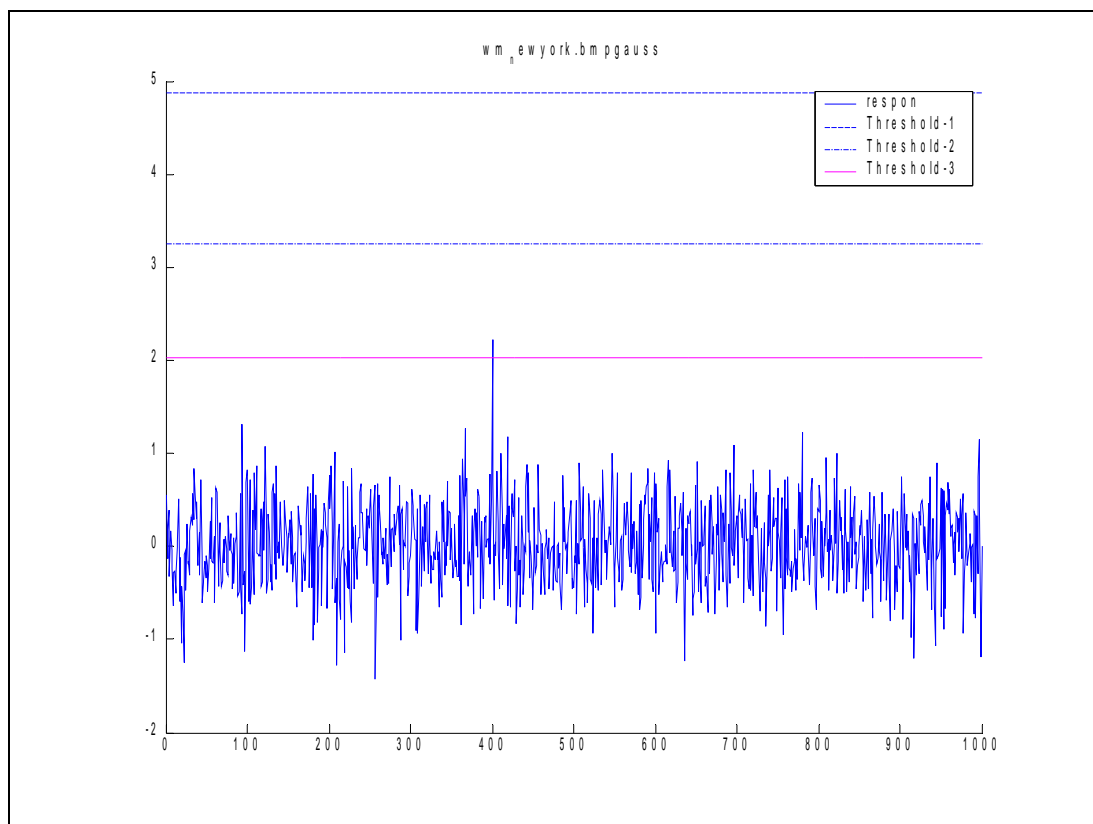
Keterangan : √ : watermark terdeteksi oleh Tr_0 .
 - : watermark tidak terdeteksi Tr_0 .



Gambar IV-39 : Citra New York terwatermark. *Original image courtesy of Patrick Loo, University of Cambridge*



Gambar IV-40 : Hasil penambahan *Gaussian noise* dengan $mean = 0$ dan $variance = 0,79$ dari citra New York terwatermark



Gambar IV-41 : Respon detektor terhadap 1000 watermark penguji pada citra New York terwatermark dengan penambahan *gaussian noise* ($mean=0, var=0.79$).

IV.1.1.12 Resizing

Percobaan dengan pengubahan ukuran (*resizing*) citra terwatermark dilakukan penulis untuk menguji ketahanan algoritma ini terhadap teknik-teknik *resizing*. Percobaan lebih difokuskan pada pengecilan ukuran citra (*shrinking*), meskipun demikian penulis juga sempat menguji dengan citra terwatermark dengan teknik perbesaran *nearest neighbor*, dan hasil yang ditunjukkan sangat baik, karena semua watermark dapat terdeteksi. Kembali pada pengujian dengan *shrinking* penulis mencoba ketiga teknik interpolasi yang sudah dibahas pada bab II, yaitu : *nearest neighbor*, *bilinear* dan *bicubic*. Hasil percobaan memperlihatkan dengan ketiga teknik interpolasi ini detektor ternyata masih mampu mendeteksi keberadaan watermark.

Percobaan perbesaran ukuran citra dilakukan dengan memperbesar citra dari ukuran asalnya 512 x 512 ke ukuran 768x768, dengan teknik *nearest neighbor*. Sedangkan percobaan dengan pengecilan ukuran citra dilakukan dengan memperkecil citra dari ukuran asalnya 512 x 512 ke ukuran 256 x 256, dengan teknik *nearest neighbor*, *bilinear* dan *bicubic*.

Tabel IV-9 : Hasil pengujian pada citra terwatermark dengan *resizing*

No	Citra	768x768 - NN	256x256 - NN	256x256 - BL	256x256 - BC
1	Skyline Arch	√	√	√	√
2	Bear	√	√	√	√
3	Baboon	√	√	√	√
4	New York	√	√	√	√
5	F16	√	√	√	√
6	Lena	√	√	√	√
7	Kid	√	√	√	√

Keterangan : √ : watermark terdeteksi oleh Tr_0 .

- : watermark tidak terdeteksi Tr_0 .

NN : teknik *nearest neighbor*

BL : teknik *Bilinear*

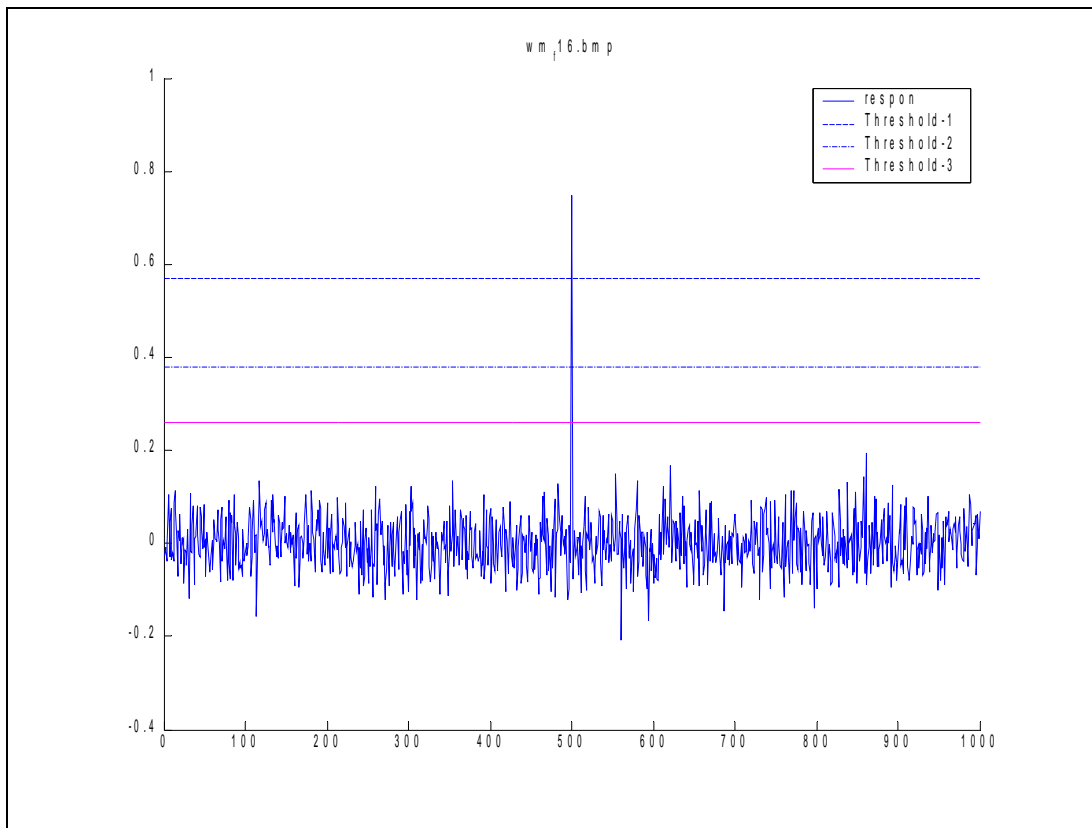
BC : teknik *Bicubic*



Gambar IV-42 : Citra F16 terwatermark.
Original image courtesy of the Signal and Image Processing Institute at the University of Southern California.



Gambar IV-43 : Hasil *bilinear resizing* dari citra F16 terwatermark. Ukuran asli citra 512x512 dan hasil *resizing* 256x256.



Gambar IV-44 : Respon detektor terhadap 1000 watermark pengujian *resizing* pada citra F16 terwatermark dengan teknik *bilinear* dari 512x512 menjadi 256x256.

BAB V PENUTUP

Kesimpulan

Kesimpulan yang dapat diambil penulis setelah mengimplementasikan teknik watermarking digital pada domain DCT ini adalah :

1. Implementasi pada citra berwarna menunjukkan watermark yang ditanamkan tidak tampak oleh indera penglihatan manusia.
2. Teknik yang diimplementasikan ini ternyata memiliki ketahanan (*robustness*) yang cukup baik terhadap serangan kompresi citra JPEG dan juga beberapa usaha pemrosesan citra, seperti peningkatan kualitas citra, penambahan *noise* dan pengubahan ukuran (*resizing*). Khusus untuk kompresi JPEG, teknik ini masih mampu mendeteksi eksistensi watermark sampai dengan kualitas kompresi 10%, dan bahkan pada salah satu kasus watermark masih dapat terdeteksi meskipun dilakukan kompresi sampai kualitas 0%. Satu hal yang perlu digaris bawahi adalah detektor masih dapat mengenali watermark meskipun kualitas citra sudah sangat menurun dibandingkan citra aslinya.
3. Teknik ini juga tidak memerlukan citra aslinya dalam mendeteksi keberadaan watermark. Sehingga dapat juga disebut sebagai *blind watermarking* atau *public watermarking*.

Saran

Setelah mengimplementasikan teknik watermarking digital pada skripsi ini, penulis ingin menyampaikan beberapa saran berikut ini :

1. Meskipun secara umum teknik ini dapat bekerja dengan baik pada semua jenis citra berwarna, beberapa percobaan menunjukkan, perlakuan teknik perataan histogram pada citra terwatermark dengan tekstur yang halus ternyata menampakkan distorsi citra akibat penanaman watermark. Untuk itulah sebaiknya diutamakan menggunakan citra yang cukup bertekstur didalam aplikasinya.
2. Secara komputasi teknik ini memerlukan waktu eksekusi yang cukup tinggi, terutama dalam pengurutan zig-zag dan penghitungan DCT dan *invers*-nya. Sehingga diperlukan teknik pengembangan lain yang memiliki waktu eksekusi yang lebih cepat dalam mengimplementasikannya.
3. Perlu dikembangkan juga penelitian untuk citra dengan ukuran yang berbeda-beda (panjang dan lebarnya tidak sama).

DAFTAR PUSTAKA

Agung, I., W., P., Februari 2001, **Digital Watermarking : Teknologi Pelindung HAKI Multimedia**, *ELEKTRO INDONESIA*, Nomor 35, Tahun VI. <<http://www.elektroindonesia.com/elektro/ut35a.html>> (Didownload pada tanggal 23 Februari 2003).

Barni, M., Bartolini, F., Cappellini, V., dan Piva, A., 1998, **A DCT-Domain System for Robust Image Watermarking**, *Signal Processing, Special Issue in "Copyright Protection and Access Control for Multimedia Services"*, 66 (3), pp 357-372. <<http://lci.det.unifi.it/Publications/sp98.zip>>. (Didownload pada tanggal 11 Februari 2003).

Bender, W., Gruhl, D., Morimoto, N., dan Lu, A., 1996, **Techniques for data hiding**, *IBM Systems Journal*, Vol 35, No. 3&4, pp 313-336. <<http://www.research.ibm.com/journal/sj/353/sectiona/bender.pdf>>. (Didownload pada tanggal 7 Januari 2003).

CCITT, 1993, *Information Technology – Digital Compression and Coding of Continuous-tone Still Images – Requirements and Guidelines*. CCITT Recommendation T.81, The International Telegraph and Telephone Consultative Committee (CCITT), International Telecommunication Union (ITU).

Cox, I., J., Kilian, J., Leighton, F., T., dan Shamoon, T., Desember 1997, **Secure Spread Spectrum Watermarking for Multimedia**, *Proc. IEEE Transactions on Image Processing*, Vol. 6, No. 12, pp 1673-1687. <<ftp://ftp.nj.nec.com/pub/ingemar/papers/ip97.zip>> (Didownload pada tanggal 29 Januari 2003).

Cuturicu, C., 1999, *A note about the JPEG decoding algorithm*, <<http://www.opennet.ru/docs/formats/jpeg.txt>>.

Duan, F., Y., King, I., 1999, **A Short Summary of Digital Watermarking Techniques for Multimedia Data**, In *Proceedings of the 1999 Hong Kong International Computer Conference (HKICC'99)*, Hong Kong, Hong Kong Computer Society. <<http://www.cse.cuhk.edu.hk/~king/PUB/hkicc991.pdf>>

Ford, A., dan Roberts, A., Agustus 1998, *Colour Space Conversions*, <<http://www.inforamp.net/~poynton/PDFs/coloureq.pdf>>.

Fotopolous, V., Kavathas, P., dan Skodra, A., N., Oktober 2000, **Bit Signature Casting in the DCT Domain**, *Technical Report No. TR2000/10/01*. <<http://www.upatras.gr/ieee/skodras/pubs/ans-e12.pdf>>.

Gilani, S., A., M., dan Skodras, A., N., Juni 2000, **DLT-Based Digital Image Watermarking**, *First IEEE Balkan Conference on Signal Processing, Communications, Circuits and Systems, Istanbul, Turkey* <<http://www.upatras.gr/ieee/skodras/pubs/ans-c34.pdf>>

Gonzales, R., C., dan Wintz, P., 1987, *Digital Image Processing*, Addison Wesley-Wesley.

Hamilton, E., September 1992, *JPEG File Interchange Format, Version 1.02*. <<http://www.w3.org/Graphics/JPEG/jfif.txt>>.

Johnson, N., F., dan Jajodia, S., Februari 1998, **Exploring Staganography: Seeing the Unseen**, *IEEE Computer*, pp 26-34. <<http://isse.gmu.edu/~njohnson/pub/r2026.pdf>>

Koch, E., dan Zhao, J., 1995, **Towards Robust and Hidden Image Copyright Labeling**, *Proc. of 1995 IEEE Workshop on Non Linear Signal and Image Processing, Neos Marmaras, Greece*. <http://www.crcg.edu/Staff/jzhao/pubs/IEEE_Hidden.ps>

Langelaar, G., Setyawan, I., dan Lagendijk, L., R., September 2000, **Watermarking Digital Image and Video Data: A State-of-the-Art Overview**, *IEEE Signal Processing Magazine*, Vol 17, pp 20-43. <<http://www-it.et.tudelft.nl/~iwan/WatermarkOverview.pdf>>.

Lee, C., H., dan Lee, T., K., November 1999, **An Adaptive Digital Image Watermarking Technique for Copyright Protection**, *IEEE Trans. on Consumer Electronics*, Vol. 45, No. 4, pp. 1005-1015. <<http://debut.cis.nctu.edu.tw/~yklee/Personal/IS99-Watermarking.pdf>>.

Mohanty, S., P., 1999, *Digital Watermarking : A Tutorial Review*. <<http://www.csee.usf.edu/~smohanty/research/Reports/WMSurvey1999Mohanty.pdf>>

Miller, M., L., Cox, I., J., Linnartz, J., P., M., G., dan Kalker, T., 1999, **A review of watermarking principles and practices**, *Digital Signal Processing in Multimedia Systems* : pp 461-485, Marcell Dekker Inc. <<http://www.ee.ucl.ac.uk/~icox/papers/1999/bookchapter99.pdf>>.

Pocket Oxford Dictionary, Maret 1994, Oxford University Press, Oxford.

Petitcolas, F., A., P., Anderson, R., J., dan Khun, M., G., Juli 1999, **Information Hidding - A Survey**, *Proceedings of the IEEE*, Vol 87, No 7, pp. 1062-1078. <<http://www.cl.cam.ac.uk/~fapp2/publications/ieee99-fohiding.pdf>>.

Petitcolas, F., A., P., 2003, *Photo Database*, <http://www.cl.cam.ac.uk/~fapp2/watermarking/image_database/index.html>.

Piva, A., Barni, M., Bartolini, F., Cappellini, V., Juni 1998, **Threshold Selection for Correlation-Based Watermark Detection**, *Proceedings of COST 254 Workshop on Intelligent Communications*, L'Aquila, Italy, pp. 67-72. <<http://lci.die.unifi.it/Publications/cost98.ps.gz>> (Didownload pada tanggal 9 Februari 2003)

Poynton, C., A., Maret 1997, *Frequently Asked Questions about Color*, <<ftp://ftp.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.pdf>>. (Didownload pada tanggal 26 Mei 2003).

Supangkat, S., H., Kuspriyanto, dan Juanda, 2000, **Watermarking sebagai Teknik Penyembunyian Label Hak Cipta pada Data Digital**, *TEKNIK ELEKTRO*, Vol. 6, No. 3, pp 19-27.

Swanson, M., D., Kobayashi, M., dan Tewfik, A., H., Juni 1998, **Multimedia Data Embedding and Watermarking Technologies**, *Proceedings of the IEEE*, Vol 86, No 6,

pp. 1064-1087. <http://debut.cis.nctu.edu.tw/~ykleee/Research/Watermarking/Ahmed_H_Tewfik/IEEE-Proc-1998-06.pdf>. (Didownload pada tanggal 7 Januari 2003).

Swanson, M., D., Zhu, B., Chau, B., dan Tewfik, A., H., 1997, **Multiresolution Video Watermarking using Perceptual Models and Scene Segmentation**, *Proceedings of the IEEE International Conference of Image Processing 1997: Vol II*, pp. 558-561. <http://debut.cis.nctu.edu.tw/~ykleee/Research/Watermarking/Ahmed_H_Tewfik/IEEE-ICIP-1997-1.pdf>. (Didownload pada tanggal 16 Mei 2003)

The Mathworks Inc., 2000a , *Image Processing Toolbox For Use with MATLAB : User's Guide*, Version 2, Revised for Version 2.2.2, September 2000, The MathWorks Inc., Natick, MA.

The Mathworks Inc., 2000b, *MATLAB Function Reference, Volume 3: P - Z*, Version 6, MATLAB-The Language of Technical Computing, November 2000 The MathWorks Inc., Natick, MA.

Wallace, G., K., April 1991, **The JPEG Still Picture Compression Standard**, *Communications of the ACM*, vol 34, no. 4, pp. 30-40. <<http://www.xfig.org/jpeg/wallace.ps.gz>>. (Didownload pada tanggal 11 Februari 2003).

Watson, A., B., 1994, **Image Compression Using the Discrete Cosine Transform**, *Mathematica Journal*, 4(1), pp. 81-88. <<http://vision.arc.nasa.gov/publications/mathjournal94.pdf>>. (Didownload pada tanggal 26 Mei 2003).

Webopedia, 2003, *Digital Watermark*, <http://www.webopedia.com/TERM/D/digital_watermark.html> (Didownload pada tanggal 21 Mei 2003).

Watermarking World, 2002, *Digital Watermarking Frequently Asked Questions (FAQ)*, Watermarking World, <<http://www.watermarkingworld.org/faq.html>> (Didownload pada tanggal 17 April 2003).

Zinneman, T., Maret 2002, *Interpolation*, TechBrief Articles, <http://www.nut-n-but.net/CCPCUG/TechBrief%20Articles/TB0204.html>, (Didownload pada tanggal 30 Juni 2003).

LAMPIRAN