

***Watermarking untuk Citra Digital dengan Menggunakan Metode
Discrete Wavelet Transform (DWT)***

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam
bidang Ilmu Komputer

oleh:

DINDA FRISKAYANTI

0610963014-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

WATERMARKING UNTUK CITRA DIGITAL DENGAN
MENGUNAKAN METODE *DISCRETE WAVELET TRANSFORM*
(DWT)

Oleh :
Dinda Friskayanti
0610963014-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 22 Juni 2011
dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana
Komputer dalam bidang Ilmu Komputer

Pembimbing I,

Pembimbing II,

Drs. Muh Arif Rahman M.kom
NIP. 196604231991111001

Dewi Yanti L.S.Kom.M.kom
NIP. 198111162005012004

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, MSc
NIP. 196709071992031001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Dinda Friskayanti
NIM : 0610963014-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis Skripsi berjudul : *Watermarking* dengan Menggunakan Metode *Discrete Wavelet Transform* (DWT)

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 22 Juni 2011
Yang menyatakan,

Dinda Friskayanti
NIM. 0610963014-96

UNIVERSITAS BRAWIJAYA



ABSTRAK

Watermarking merupakan sebuah cara yang digunakan untuk melindungi data multimedia dari pelanggaran hak cipta. Dalam *watermarking*, sebuah tanda yang tak terlihat ditempelkan ke citra aslinya, dengan identitas unik pemilikinya.

Pada penelitian ini, teknik *watermarking* yang digunakan adalah *discrete wavelet transform*. *Discrete wavelet transform* ini bekerja dalam domain frekuensi. Dalam *Discrete wavelet transform* penyisipan *watermark* dilakukan dengan cara memodifikasi koefisien rentang frekuensi LL, LH, HL, atau HH yang merupakan rentang frekuensi hasil dekomposisi citra menggunakan *wavelet*.

Dari penelitian yang dilakukan menunjukkan bahwa penggunaan nilai α yang kecil dan ukuran tanda *watermark* yang kecil pada penyisipan *watermark* akan menghasilkan nilai *Mean Square Error* (MSE) yang kecil dan nilai PSNR yang besar. Hal ini berarti citra *watermark* yang dihasilkan memiliki kualitas gambar yang bagus. Pengujian yang dilakukan dengan menggunakan serangan terhadap citra yang telah disisipi *watermark* juga menunjukkan bahwa tanda *watermark* yang telah disisipkan ke dalam citra *watermark* mampu terdeteksi dengan baik dan tahan terhadap serangan penambahan *Noise*, *cropping*, *sharpening*, *resizing* dan *rotation*.

Pengujian terakhir yang dilakukan adalah dengan menunjukkan hasil citra yang disisipi *watermark* terhadap 10 responden dan diperoleh hasil semua responden tidak melihat adanya perbedaan kualitas gambar yang telah disisipi *watermark* dengan citra asli untuk α 0.0001 dan 0.0003.

Kata kunci: *Watermarking*, *Discrete Wavelet Transform* (DWT), dekomposisi citra, domain frekuensi

UNIVERSITAS BRAWIJAYA



ABSTRACT

Watermarking is a means used to protect multimedia data from copyright infringement. In watermarking, an invisible sign affixed to the original image, with the unique identity of its owner.

In this study, the watermarking technique used is the discrete wavelet transform. Discrete wavelet transform works in the frequency domain. In the discrete wavelet transform watermark embedding is done by modifying the frequency range of coefficients LL, LH, HL, or HH, which is the frequency range images using wavelet decomposition.

From research conducted showed that the use of a small alpha value and the size of a small sign on the watermark embedding watermark will generate value Mean Square Error (MSE) is small and a large PSNR value. This means that the resulting watermarked image has good image quality. Tests carried out by using the attack on the watermarked image has been inserted is also shown that the sign of the watermark that has been inserted into the watermarked image can be detected with good and resistant to attack by adding noise, cropping, sharpening, resizing and rotation.

Final testing is done is by showing the image of the inserted watermark to 10 respondents and obtained the results of all respondents saw no difference in image quality that has been inserted watermark with the original image for the alpha 0.0001 and 0.0003.

Keywords: *Watermarking, Discrete Wavelet Transform (DWT), image decomposition, frequency domain.*

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Alhamdulillah *rabbil 'alamin*. Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas segala rahmat dan limpahan hidayahNya, penulis masih dapat belajar dan mengerjakan Tugas Akhir yang berjudul “**Watermarking untuk Citra Digital dengan Menggunakan Metode Discrete Wavelet Transform (DWT)**”. Tugas Akhir ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW, yang senantiasa memberikan cahaya petunjuk atas keluarganya dan sahabat-sahabatnya.

Dalam penyelesaian tugas akhir ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Drs. Muh. Arif Rahman, M.Kom, dan Dewi Yanti Liliana,S.Kom, M.kom selaku dosen pembimbing skripsi yang telah memberikan saran, kritik, waktu, bimbingan, dan dorongan semangat selama penulisan tugas akhir ini.
2. Drs.Marji MT Selaku Ketua Program Studi Ilmu Komputer Universitas Brawijaya Malang.
3. Drs. A. Ridok, M.Kom selaku Penasihat Akademik yang telah memberikan bimbingan akademik dan dorongan semangatnya.
4. Dr. Abdul Rouf Alghofari, MSc selaku Ketua Jurusan Matematika Universitas Brawijaya
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya.
7. Ayah, Mama, dan Kakak atas cinta, kasih sayang, doa, dukungan dan semangat yang tiada henti.
8. Sahabat – sahabatku yang telah memberikan semangat, inspirasi dan dukungan selama ini.
9. Teman – teman babi inside O6 (epy, imud, mba sulis dan teman teman yang lain) . Terima kasih karena telah memberikan persahabatan yang

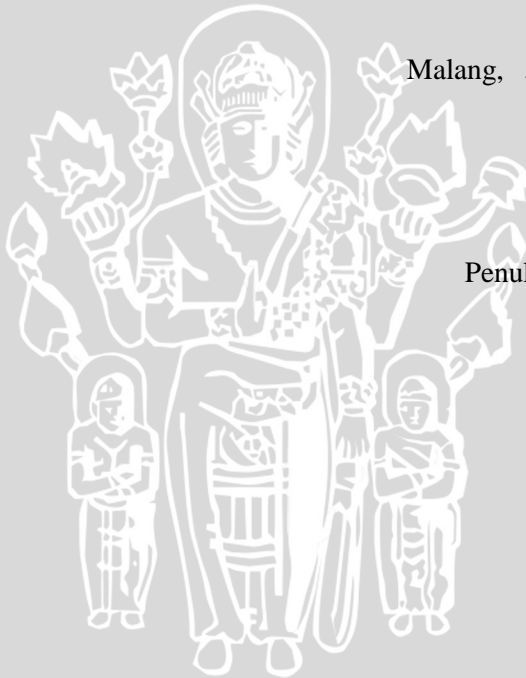
tulus selama ini. Senang bisa bertemu dan melewatkan banyak moment dengan kalian.

10. Pihak lain yang telah membantu terselesaikannya Tugas Akhir ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga penulisan laporan tugas akhir ini bermanfaat bagi pembaca sekalian. Dengan tidak lupa kodratnya sebagai manusia, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, Juni 2011

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
DAFTAR <i>SOURCE CODE</i>	xxi
DAFTAR LAMPIRAN	xxiii

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan masalah.....	3
1.4 Batasan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Pemecahan Masalah	4
1.7 Sistematika Penulisan	5

BAB II TINJAUAN PUSTAKA

2.1 Konsep Dasar Citra Digital.....	7
2.2 Warna dan Ruang Warna.....	8
2.2.1 RGB (<i>red, green, blue</i>).....	8
2.2.2 YCbCr (<i>Luminance – Chrominance</i>)	9
2.3 Digital <i>Watermark</i>	10
2.3.1 Klasifikasi dan Kebutuhan <i>Watermarking</i>	11
2.3.2 Tipe Digital <i>Watermarking</i>	12
2.3.3 Teknik – teknik dalam <i>Watermarking</i>	13
2.3.4 <i>Transform Domain Watermarking</i>	13
2.3.5 Serangan pada <i>Watermarking</i>	14
2.4 <i>Discrete Wavelet Transform</i>	15
2.4.1 Pengertian <i>Discrete wavelet Transform</i>	15
2.4.2 Proses Dekomposisi Citra.....	15

2.4.3 Proses <i>Discrete Wavelet Transform</i>	18
2.4.4 Proses <i>Inverse Discrete wavelet Transform</i>	19
2.5 <i>Mean Square Error (MSE)</i>	20
2.6 <i>Peak Signal Noise to Ratio (PSNR)</i>	20

BAB III METODOLOGI DAN PERANCANGAN

3.1 Deskripsi Umum Perangkat Lunak.....	23
3.1.1 Batasan Perangkat Lunak	24
3.2 Mekanisme Kerja Perangkat Lunak.....	24
3.2.1 Perancangan Proses <i>Input</i> Perangkat Lunak	24
3.2.2 Perancangan Proses Perhitungan Pembuatan <i>Watermark Discrete Wavelet Transform</i>	25
3.2.3 Perancangan Proses Dekomposisi <i>Watermark</i>	25
3.2.4 Perancangan Proses Penyisipan <i>Watermark</i>	26
3.2.5 Perancangan Proses Ekstraksi <i>Watermark</i>	27
3.3 Perancangan Uji Coba	28
3.3.1 Citra Uji.....	28
3.3.2 Lingkungan Pengujian	28
3.3.3 Pengujian Kualitas Citra <i>Watermark</i>	28
3.3.4 Pengujian terhadap Ketahanan <i>Watermark</i>	29
3.3.5 Pengujian <i>Watermark</i> terhadap <i>Unperceptibility</i>	30
3.4 Contoh Perhitungan Manual Penyisipan <i>Watermark</i>	31

BAB IV HASIL DAN PEMBAHASAN

4.1 Lingkungan Sistem.	39
4.1.1 Lingkungan perangkat keras.....	39
4.1.2 Lingkungan perangkat lunak	39
4.2 Implementasi Program.....	39
4.2.1 Input	39
4.2.2 Dekomposisi Transformasi <i>Wavelet</i>	42
4.2.3 <i>Discrete Wavelet Transform</i>	45
4.2.4 <i>Inverse Discrete Wavelet Transform (IDWT)</i>	47
4.2.5 Perhitungan MSE dan PSNR.....	48
4.2.6 Penyimpanan Citra yang Disisipi <i>Watermark</i>	49
4.3 Implementasi Antar Muka.....	50
4.3.1 Tampilan Utama Aplikasi <i>Watermarking</i> Citra	50
4.3.2 Tampilan Proses Penyisipan <i>Watermark</i>	51
4.3.3 Tampilan Proses Ekstraksi <i>Watermark</i>	54
4.4 Implementasi Uji Coba.....	57
4.4.1 Evaluasi Kualitas Citra.....	57

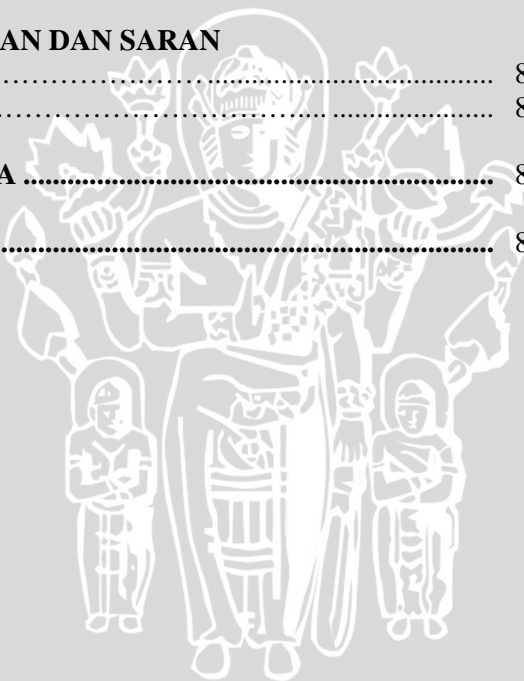
4.4.2 Evaluasi Hasil Citra yang Disisipi <i>Watermark</i>	66
4.4.2.1 Hasil Evaluasi Tanpa Serangan	66
4.4.2.2 Evaluasi Hasil Citra yang Disisipi <i>Watermark</i> dari Sisi <i>Robustness</i> (Ketahanan)	66
4.4.2.2.1 <i>Cropping</i>	66
4.4.2.2.2 Penambahan <i>Noise</i>	68
4.4.2.2.3 <i>Sharpening</i>	69
4.4.2.2.4 <i>Resizing</i>	70
4.4.2.2.5 <i>Rotation</i>	72
4.4.2.3 Evaluasi Hasil Citra yang Disisipi <i>Watermark</i> Tidak Terlihat (<i>Nonperceptibility</i>)	73
4.5 Analisa hasil	77

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	81
5.2 Saran.....	81

DAFTAR PUSTAKA	83
-----------------------------	-----------

LAMPIRAN	87
-----------------------	-----------



UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

Halaman

Gambar 2.1	Citra Digital	7
Gambar 2.2	Ruang Warna RGB	8
Gambar 2.3	Ruang Warna YcbCr	9
Gambar 2.4	Penyisipan <i>Watermark</i>	10
Gambar 2.5	Proses Perbandingan <i>Watermark</i> pada Citra Digital	11
Gambar 2.6	Hasil proses dekomposisi level 1	16
Gambar 2.7	Proses Dekomposisi Level 2	17
Gambar 2.8	Dekomposisi <i>wavelet</i> dengan fungsi dasar $x[n]f=0 \sim \pi$	17
Gambar 3.1	Diagram Alir Pembuatan Perangkat Lunak	23
Gambar 3.2	Diagram Alir Proses Penyisipan <i>Watermark</i>	25
Gambar 3.3	Diagram Alir Proses Ekstraksi <i>Watermark</i>	26
Gambar 4.1	Tampilan utama	50
Gambar 4.2	Tampilan <i>Open Image</i> citra <i>watermark</i>	51
Gambar 4.3	Tampilan Tanda <i>Watermark</i>	52
Gambar 4.4	Tampilan <i>Open Image</i>	52
Gambar 4.5	Tampilan <i>Image Original</i>	53
Gambar 4.6	Tampilan Proses Penyisipan Citra	54
Gambar 4.7	Tampilan citra yang telah disisipi <i>watermark</i>	54
Gambar 4.8	Tampilan <i>Open Image</i> citra <i>watermark</i>	55
Gambar 4.9	Tampilan Tanda <i>Watermark</i>	55
Gambar 4.10	Tampilan citra yang disisipi <i>watermark</i>	56
Gambar 4.11	Tampilan proses Ekstraksi	56
Gambar 4.12	Tampilan Hasil Proses Ekstraksi	57
Gambar 4.13	Grafik pengaruh alpha terhadap nilai PSNR untuk ukuran tanda <i>watermark</i> 20x20 pixel	62
Gambar 4.14	Grafik pengaruh alpha terhadap nilai PSNR untuk ukuran tanda <i>watermark</i> 40x40 pixel	63
Gambar 4.15	Grafik pengaruh alpha terhadap nilai PSNR untuk ukuran tanda <i>watermark</i> 60x60 pixel	63
Gambar 4.16	Grafik tanda <i>watermark</i> terhadap nilai PSNR $\alpha = 0.0001$	64

Gambar 4.17 Grafik tanda *watermark* terhadap nilai PSNR $\alpha = 0.0003$ 65

Gambar 4.18 Grafik tanda *watermark* terhadap nilai PSNR $\alpha = 0.0006$ 65

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

	Halaman	
Tabel 3.1	Rancangan Tabel Hasil Uji Kualitas Citra yang Disisipi <i>Watermark</i>	29
Tabel 3.2	Rancangan Tabel Uji Ketahanan Citra yang Disisipi <i>Watermark</i>	30
Tabel 3.3	Rancangan Tabel Hasil terhadap Uji <i>Nonperceptibility</i> Citra yang Disisipi <i>Watermark</i>	31
Tabel 3.4	Tabel Warna R (<i>Red</i>)	31
Tabel 3.5	Tabel Warna G (<i>Green</i>)	31
Tabel 3.6	Tabel Warna B (<i>Blue</i>)	31
Tabel 3.7	Tabel Warna Y (<i>Luminance</i>)	32
Tabel 3.8	Tabel Warna Cb (<i>Chrominance Blue</i>)	32
Tabel 3.9	Tabel Warna Cr (<i>Chrominance Red</i>)	33
Tabel 3.10	Tabel Transformasi DWT	33
Tabel 3.11	Tabel Transformasi IDWT	34
Tabel 3.12	Tabel Subband LL	35
Tabel 3.13	Tabel Subband LH	36
Tabel 3.14	Tabel Subband HL	36
Tabel 3.15	Tabel Subband HH	36
Tabel 3.16	Tabel Citra yang Disisipi <i>Watermark</i>	36
Tabel 3.17	Tabel Rata-rata Perbedaan Citra Asli dengan Citra <i>Watermark</i>	37
Tabel 4.1	Tabel hasil penyisipan <i>watermark</i> untuk $\alpha = 0.0001$	58
Tabel 4.2	Tabel hasil penyisipan <i>watermark</i> untuk $\alpha = 0,0003$	59
Tabel 4.3	Tabel hasil penyisipan <i>watermark</i> untuk $\alpha = 0,0006$	60
Tabel 4.4	Tabel hasil pendeteksian pada citra yang sudah disisipi <i>watermark</i> dengan serangan <i>cropping</i>	67
Tabel 4.5	Tabel hasil pendeteksian pada citra yang sudah disisipi <i>watermark</i> dengan serangan penambahan noise	68

Tabel 4.6	Tabel hasil pendeteksian pada citra yang sudah yang disisipi <i>watermark</i> dengan serangan penajaman citra	69
Tabel 4.7	Tabel hasil pendeteksian pada citra yang disisipi <i>watermark</i> dengan serangan <i>resizing</i>	71
Tabel 4.8	Tabel hasil pendeteksian pada citra yang disisipi <i>watermark</i> dengan serangan <i>rotation</i>	72
Tabel 4.9	Tabel hasil citra yang disisipi <i>watermark</i> tidak terlihat (<i>Nonperceptibility</i>)	73



DAFTAR SOURCE CODE

		Halaman
<i>Source Code 4.1</i>	<i>Source code</i> Transformasi RGB ke YCbCr	40
<i>Source Code 4.2</i>	<i>Source code class Pixel</i>	41
<i>Source Code 4.3</i>	<i>Source code class YCbCr</i>	42
<i>Source Code 4.4</i>	<i>Source code</i> dekomposisi pada citra	45
<i>Source Code 4.3</i>	<i>Source code</i> Perhitungan DWT	46
<i>Source Code 4.4</i>	<i>Source code</i> Perhitungan IDWT	48
<i>Source Code 4.5</i>	<i>Source code</i> perhitungan MSE dan PSNR	49
<i>Source Code 4.6</i>	<i>Source code</i> penyimpanan citra yang disisipi watermark	50



UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

		Halaman
Lampiran 1	Hasil Penyisipan <i>Watermark</i> dengan Menggunakan Metode <i>Discrete Wavelet Transform</i>	89
Lampiran 2	Hasil Pendeteksian <i>Watermark</i> dengan Menggunakan Metode <i>Discrete Waveket Transform</i>	93



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam perkembangan teknologi internet yang ada pada saat ini telah memberikan kemudahan untuk melakukan akses serta pendistribusian informasi dalam format digital. Kemudahan tersebut akhirnya dapat digunakan secara “negatif” tanpa memperhatikan aspek hak cipta (*Intellectual Property Right*). Perlindungan hak cipta terhadap data digital memang sudah menjadi perhatian orang – orang sejak dulu. Banyak cara yang sudah ditempuh untuk melindungi data digital, seperti : *encryption*, *copy protection*, *visible marking*, *header marking*, dan sebagainya, tetapi semua cara tersebut memiliki kelemahan masing – masing (Suhono H. Supangkat, Kuspriyanto dan Juanda, 2000). Adapun kelemahan tersebut diantaranya adalah (Tamatjita, 2006):

1. *Header Marking*; dengan menggunakan software sejenis Hex Editor, orang lain dapat dengan mudah membuka file berisi data digital tersebut dan menghapus segala informasi yang berguna untuk melindungi keaslian dari data digital tersebut yang tersimpan dalam *header* file tersebut.
2. *Encryption*; Penyebaran data digital dengan kunci untuk *decryption* tidak dapat menjamin penyebarannya yang legal. Maksudnya setelah data digitalnya terenkripsi dan kuncinya telah diberikan kepada pihak yang telah diberi ijin, tidak dapat dijamin penyebaran data digital yang telah terdekripsi tadi oleh pihak lain.
3. *Copy Protection*; Perlindungan sejenis ini biasanya dilakukan secara hardware, seperti halnya perlindungan pada *hardware* DVD, tetapi sebagaimana diketahui, banyak data digital saat ini tidak dapat dilindungi secara *hardware*.
4. *Visible Marking*; penandaan secara eksplisit pada data digital memang memberikan sejenis tanda semi – permanen, tetapi dengan tersedianya software atau metode untuk pengolahan, tanda yang semi – permanen tersebut dapat dihilangkan dari data digitalnya.

Ditengah berbagai kekurangan metode perlindungan terhadap data digital tersebut, *watermarking* timbul sebagai alat untuk melindungi data multimedia dari pelanggaran hak cipta. Dalam *watermarking*, sebuah tanda yang tak terlihat ditempelkan ke citra aslinya, dengan identitas unik pemilikinya. Teknik dalam *watermarking* dapat dibagi ke dalam dua

kategori : yaitu metode domain spasial dan metode domain transformasi. Metode domain spasial tidak serumit metode transformasi, tetapi tidak cukup kuat untuk melawan serangan. Teknik domain transformasi lebih kuat dibandingkan dengan metode domain spasial (Hameed, 2006). Teknik ini didasarkan pada *discrete cosine transform* (DCT), dimana urutan *pseudo – random*, misalnya M - urutan ditambahkan pada koefisien DCT di tengah frekuensi sebagai tanda. Pendekatan ini sesuai dengan standart kompresi citra / video pada saat ini seperti JPEG, MPEG1-2, dan sebagainya.(Xiang-Gen Xia, 2008)

Namun, diantara semua teknik dalam domain transformasi *watermarking*, *discrete wavelet transform* (DWT) merupakan salah satu teknik yang paling banyak digunakan baik dalam jenis *blind watermarking* dan *nonblind watermarking*. DWT membagi sebuah dimensi sinyal menjadi dua bagian, biasanya bagian dengan frekuensi tinggi dan frekuensi rendah, yang disebut dengan dekomposisi. Sebuah sinyal dilewatkan melalui *highpass filter* untuk menganalisis frekuensi tinggi, dan dilewatkan melalui *lowpass filter* untuk menganalisis frekuensi rendah. Keluaran dari *highpass filter* dan *lowpass filter* ini menghasilkan koefisien DWT, dengan menggunakan koefisien ini citra asli dapat direkonstruksi (Alfatwa, 2008).

Watermarking dalam *Discrete Wavelet Transform* (DWT) domain ini dipilih karena beberapa alasan:

1. Metode *watermarking* mempunyai karakteristik multiresolusi dan hierarkis. Dalam kasus ini, ketika penerimaan image tidak menyimpang secara signifikan, korelasi persilangan dengan ukuran penuh dari image mungkin tidak dibutuhkan dan oleh karena itu, beban komputasi yang besar dapat disimpan.
2. DWT merupakan yang paling dekat terhadap *Human Visual System* (HVS). *Human Visual System* (HVS) merupakan tingkat sensitivitas sistem penglihatan manusia terhadap warna dan tingkat kecerahan.
3. *Bit – error rate* yang rendah. *Bit – error rate* merupakan perbandingan antara *bit* yang salah diekstraksi dengan total *bit* yang disisipkan.

Dari *watermarking* dengan menggunakan metode *discrete wavelete transform* (DWT) yang telah dibahas oleh peneliti sebelumnya (Alfatwa,2008), pengujian yang dilakukan masih terbatas pada citra keabuan. Oleh karena itu perlu dilakukan pengujian keefektifan metode DWT terhadap serangan citra ber-*watermark* terhadap citra berwarna.

Dari uraian latar belakang diatas, maka penulis melakukan penelitian mengenai pengujian metode DWT dalam *watermarking* terhadap citra berwarna. Dalam skripsi ini penulis mengambil judul “***Watermarking untuk Citra Digital Menggunakan Metode Discrete Wavelet Transform (DWT)***”.

1.2. Rumusan Masalah

Adapun rumusan permasalahan dalam skripsi ini adalah :

1. Bagaimana perbandingan kualitas citra yang telah disisipi *watermark* dengan citra asli dengan ukuran alpha 0.0001, 0.0003 dan 0.0006 menggunakan *Mean Squared Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR).
2. Bagaimana perbandingan kualitas citra yang telah disisipi *watermark* dengan citra asli dengan ukuran tanda *watermark* 20x20 pixel, 40x40 pixel dan 60x60 pixel menggunakan *Mean Squared Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR).
3. Bagaimana tingkat ketahanan citra yang telah disisipi *watermark* terhadap serangan penambahan *noise*, *cropping*, *sharpening*, *resizing*, dan *rotation*.
4. Bagaimana kualitas citra yang disisipi *watermark* dari segi *Nonperceptibility*(Tidak Terlihat).

1.3. Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah

1. Membandingkan kualitas citra yang telah disisipi *watermark* dengan citra asli dengan ukuran alpha 0.0001, 0.0003 dan 0.0006 menggunakan *Mean Squared Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR).
2. Membandingkan kualitas citra yang telah disisipi *watermark* dengan citra asli dengan ukuran tanda *watermark* 20x20 pixel, 40x40 pixel dan 60x60 pixel menggunakan *Mean Squared Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR).
3. Mengetahui tingkat ketahanan citra yang telah disisipi *watermark* terhadap serangan penambahan *noise*, *cropping*, *sharpening*, *resizing*, dan *rotation*.
4. Mengetahui kualitas citra yang disisipi *watermark* dari segi *Nonperceptibility*(Tidak Terlihat).

1.4. Batasan Masalah

Untuk menghindari meluasnya permasalahan pada skripsi kali ini, maka dibatasi pada hal – hal berikut ini :

1. Citra yang digunakan adalah citra berwarna.
2. Ruang warna yang digunakan dalam penelitian kali ini adalah RGB dan YCbCr

1.5. Manfaat

Adapun manfaat yang dapat diambil dalam penulisan skripsi kali ini adalah mengimplementasikan teknik *watermarking* yang dapat digunakan sebagai proteksi terhadap kepemilikan data digital.

1.6. Metodologi Pemecahan Masalah

Untuk mencapai tujuan dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan skripsi ini adalah :

1. Studi literatur
Mempelajari teori – teori dan konsep – konsep dasar serta identifikasi masalah yang berhubungan dengan *watermarking*, khususnya mengenai metode *discrete wavelet transform* melalui pustaka yang bersangkutan baik berupa buku, jurnal ilmiah, maupun melalui internet
2. Pendefinisian dan analisis masalah
Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan implementasi sistem
Membuat perancangan perangkat lunak dan mengimplementasikan hasil rancangan tersebut, yaitu membuat perangkat lunak *watermarking* metode *discrete wavelet transform* (DWT).
4. Uji coba dan analisis hasil implementasi
Menguji perangkat lunak, dan menganalisis hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnaka.

1.7. Sistematika Penulisan

Pembuatan tugas akhir ini dilakukan dengan pembagian bab sebagai berikut :

BAB I: PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, metodologi pemecahan masalah, serta sistematika penulisan tugas akhir.

BAB II: TINJAUAN PUSTAKA

Bab ini menjelaskan tentang citra digital secara umum, teori dasar *watermarking*, dan teknik *watermarking* menggunakan metode *discrete wavelet transform*. Adapun literatur yang digunakan meliputi buku referensi dan dokumentasi internet.

BAB III: METODOLOGI DAN PERANCANGAN

Pada bab ini, dijelaskan metode – metode yang digunakan dalam menyelesaikan masalah teknik *watermarking* menggunakan metode *discrete wavelete transform*. Beberapa poin yang dibahas adalah : Deskripsi umum sistem, Perancangan kerja sistem, Perancangan uji coba dan Evaluasi hasil.

BAB IV: HASIL DAN PEMBAHASAN

Bab IV menerangkan proses implementasi dari rancangan penelitian yang dijelaskan pada BAB III. Implementasi yang dijelaskan terdiri dari implementasi program. Selain itu, bab IV juga menjelaskan penerapan aplikasi, uji coba, dan analisa hasil percobaan.

BAB V: KESIMPULAN DAN SARAN

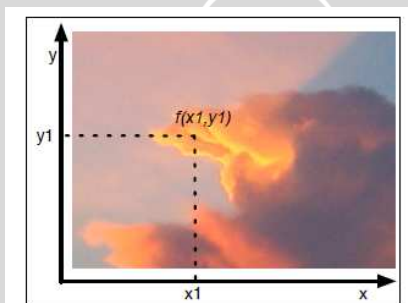
BAB V berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan tugas akhir selanjutnya.



BAB II TINJAUAN PUSTAKA

2.1 Konsep Dasar Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah tingkat kecerahan atau derajat keabuan (*brightness/gray level*) dan informasi citra pada koordinat tersebut, hal tersebut diilustrasikan pada Gambar 2.1. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital didasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue – RGB*) (Gonzalez, 2002).



Gambar 2.1 Citra Digital (Sumber: Permana, 2008)

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya *scanner*, kamera digital, dan *handycam*. Ketika sebuah citra sudah diubah ke dalam bentuk digital (selanjutnya disebut citra digital), bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut (Jahne, 2002).

Pengolahan citra digital dapat dilakukan dengan cara-cara sebagai berikut:

1. Representasi dan pemodelan citra
2. Peningkatan kualitas citra
3. Restorasi citra
4. Analisis citra
5. Rekonstruksi citra
6. Kompresi citra

2.2 Warna dan Ruang Warna

Menurut jurnal yang ditulis oleh Mar (2009) disebutkan bahwa “warna merupakan hasil persepsi dari warna cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dengan panjang gelombang antara 400/nm sampai dengan 700/nm” sedangkan Ford (1998) menyebut warna sebagai “reaksi otak dalam menerima rangsangan visual”.

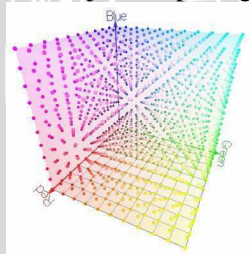
Ruang warna atau yang sering juga disebut sebagai model warna merupakan sebuah metode yang dapat kita gunakan untuk menentukan, membuat dan memvisualisasikan warna (Ford,1998). Dalam skripsi ini, hanya akan dibahas beberapa ruang warna yang biasa digunakan untuk aplikasi *watermarking*. Beberapa ruang warna tersebut antara lain adalah sebagai berikut (Ford, 1998):

1. RGB (*Red, Green, Blue*)
2. HSI (*Hue Saturation Intensity*)
3. YCbCr (*Luminance – Chrominance*)

Dalam tugas akhir ini ruang warna yang digunakan adalah ruang warna RGB dan YCbCr.

2.2.1 RGB (*Red Green Blue*)

Citra berwarna umumnya memiliki ruang warna RGB. Ruang warna RGB dapat divisualisasikan sebagai sebuah kubus, dengan tiga sumbu yang mewakili komponen warna merah (*red*) R, hijau (*green*) G dan biru (*blue*) B. Ruang warna RGB ditunjukkan pada gambar 2.2.



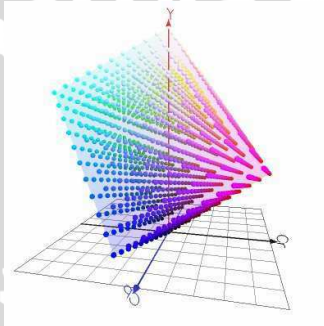
Gambar 2.2 Ruang warna RGB

(Sumber: <http://www.couleur.org/>, 2009)

RGB sering digunakan didalam sebagian besar aplikasi komputer karena dengan ruang warna ini, tidak diperlukan transformasi untuk menampilkan informasi di layar monitor. Hal tersebut juga menyebabkan RGB banyak dimanfaatkan sebagai ruang warna dasar bagi sebagian besar aplikasi (Mar,2009).

2.2.2 YCbCr (*Luminance – Chrominance*)

YCbCr merupakan standart internasional bagi pengkodean digital gambar video dan televisi yang didefinisikan di CCIR *Recommended 601* (Ford, 1998). Y merupakan komponen luminance, C_b dan C_r adalah komponen *chrominance*. Ruang warna YCbCr ditunjukkan pada gambar 2.2.



Gambar 2.3 Ruang Warna $Y C_b C_r$
(Sumber: <http://www.couleur.org/>, 2009)

$Y C_b C_r$ dapat diperoleh dari RGB dengan menggunakan persamaan 2.1-2.3 (Hamilton, 1992):

$$Y = 0.229R + 0.587G + 0.114B \quad (2.1)$$

$$C_b = -0.1687R - 0.3313G + 0.5B + 128 \quad (2.2)$$

$$C_r = 0.5R - 0.418G - 0.0813B + 128 \quad (2.3)$$

Sedangkan untuk konversi $Y C_b C_r$ ke RGB dapat dilakukan dengan persamaan 2.4-2.6 (Hamilton, 1992):

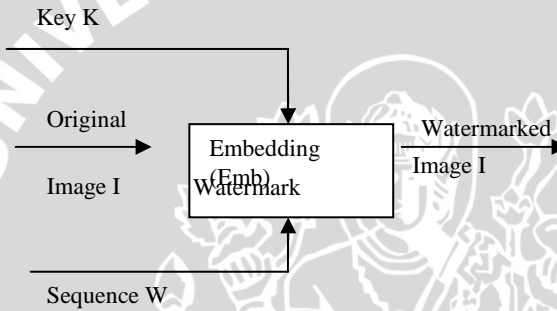
$$R = Y + 1.402(C_r - 128) \quad (2.4)$$

$$G = Y - 0.34414(C_b - 128) - 0.71414(C_r - 128) \quad (2.5)$$

$$B = Y + 1.772(C_b - 128) \quad (2.6)$$

2.3 Digital Watermark

Menurut Barni (1998) digital *watermark* dapat diartikan sebagai kode yang membawa informasi yang diperlukan untuk menangani kepemilikan data digital. Sedangkan dalam jurnal yang ditulis oleh Alfatwa (2008) disebutkan bahwa digital *watermark* merupakan suatu teknik penyembunyian data berupa teks, citra, atau suara pada citra digital yang timbul akibat kebutuhan perlindungan kepemilikan data *digital*. Dari kedua pengertian di tersebut dapat ditarik garis kesimpulan bahwa digital *watermark* adalah teknik penyembunyian informasi pemilik data digital untuk melindungi kepemilikan data digital.



Gambar 2.4 Penyisipan *watermark*.
(Sumber: Alfatwa, 2008)

Secara umum, proses watermarking pada citra digital dipaparkan pada Gambar 2.4 dimana citra digital disisipi dengan *watermark* yang dapat dianggap sebagai rangkaian bilangan w menggunakan kunci sebagai sarana kepemilikan untuk dapat membuka *watermark* yang disisipkan melalui *encoder* yang berisi algoritma penyisipan *watermark* ke dalam citra digital (Alfatwa, 2008).

Proses *watermarking* hanya berpengaruh sedikit terhadap perubahan warna pada citra *digital* asal. Sehingga secara visual citra yang dihasilkan oleh proses *watermark* tidak berbeda jauh dengan citra *digital* asalnya.

Sedangkan proses perbandingan *watermark* dilakukan untuk membuktikan status kepemilikan citra digital yang disengketakan. Perbandingan *watermark* terdiri dari dua sub-proses, yaitu ekstraksi *watermark* dan perbandingan. Proses ekstraksi *watermark* disebut juga *decoding* bertujuan mengungkap *watermark* dari dalam citra. *Decoding* dapat mengikutsertakan citra asal atau tidak sama sekali, dengan tujuan

untuk meningkatkan kinerja yang lebih baik. Proses perbandingan bertujuan untuk membandingkan *watermark* yang diungkap dengan *watermark* asli dan memberi keputusan tentang *watermark* tersebut (Supangkat, 2000). Proses verifikasi dapat dilihat pada Gambar 2.5.



Gambar 2.5 Proses Perbandingan *Watermark* pada Citra Digital (Sumber: Supangkat, 2000)

2.3.1 Klasifikasi dan Kebutuhan *Watermarking*

Digital watermarking dapat diklasifikasikan dan diukur atas dasar beberapa karakteristik termasuk pertahanan dari berbagai macam serangan, kapasitas penyimpanan dari perlawanan dari serangan jahat, kecocokan dengan *watermark* lain, dan kompleksitas dari metode *watermarking*. Secara umum, dapat diuraikan sebagai kepercayaan, ketahanan, kelemahan, kompleksitas, dan pembatasan lainnya. *Digital watermarking* harus memenuhi persyaratan di bawah ini (Juergen, 2004):

1. Ketahanan (*Robustness*)
Robustness dapat diartikan pembatasan hak akses terhadap perubahan informasi *watermarking* dan memodifikasi file asli. Modifikasi yang dimaksud adalah mengubah ukuran, kompresi file, rotasi, dll.
2. Tidak terlihat (*Nonperceptibility*)
 Klasifikasi ini didasarkan pada *Human Visual System* (HVS) dan *Human Audio System* (HAS). *Watermark* tidak terlihat jika manusia normal tidak dapat membedakan antara file yang asli dan pembawa *watermark*.
3. Tidak terdeteksi (*Nondetecable*)
 Materi data dengan membawa informasi *watermark* tidak terdeteksi jika materi data ini konsisten dengan sumber data. *Nondetecability* berhubungan dengan sumber data dan komponen-komponennya.

4. Kapasitas

Kapasitas mengacu pada kemampuan sumber data dalam menyimpan informasi *watermark*.

5. Kompleksitas

Kompleksitas menjelaskan cara yang diperlukan untuk mendeteksi dan menyangkan informasi *watermark*.

2.3.2 Tipe *Digital Watermarking*

Watermark dapat diklasifikasikan dalam beberapa kategori (Juergen, 2004):

1. Tipe sumber atau aplikasi yang digunakan

Tipe sumber atau aplikasi yang dimaksud diatas adalah objek yang digunakan dalam proses *watermark*. Objek tersebut yang akan disisipi oleh suatu tanda identitas unik pemiliknya. Adapun objek yang dapat disisipi *watermark* diantaranya adalah citra, video, audio, text, dll.

2. Persepsi manusia

Dalam kategori ini *watermark* dibedakan menjadi 2 yaitu

a. *Visible watermark*

Dalam *visible watermark*, suatu tanda yang disisipkan dapat dilihat oleh penglihatan mata manusia.

b. *Invisible watermark*

Dalam *invisible watermark*, suatu tanda yang disisipkan dapat dilihat oleh penglihatan mata manusia.

3. Teknik dasar samar dan tidak samar (*Blind and Nonblind watermark*)

Dalam teknik dasar samar dan tidak samar, *watermark* dibagi menjadi 2 yaitu:

a. *Blind watermark*

Dalam *blind watermark*, ekstraksi *watermark* dapat dilakukan tanpa data asli

b. *Nonblind watermark*

Dalam *Non blind watermark*, data asli dan data *watermark* diperlukan dalam ekstraksi *watermark*. Dalam hal ini, data sumber diperiksa dan diperoleh informasi *watermark*.

2.3.3 Teknik-teknik dalam *Watermarking*

Dalam beberapa waktu terakhir ini, beberapa penelitian mengenai teknik-teknik yang digunakan dalam *watermarking* telah banyak dilakukan.

Menurut jurnal yang ditulis oleh Sirait (2007) disebutkan “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 *watermarking* yang bekerja pada kedua domain diatas (*hybrid techniques watermarking*)”

2.3.4 Transform 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) (Swanson,1997).

Teknik yang berbasiskan *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, 1997).

2.3.5 Serangan pada Watermarking

Serangan terhadap citra yang disisipi *watermark* dilakukan untuk menghilangkan *watermark* yang telah disisipkan ke dalam citra tersebut. Dalam subbab ini akan dijelaskan secara singkat beberapa serangan yang digunakan dalam pengujian kualitas *watermark* . Beberapa serangan yang dilakukan adalah *sharpening*, *cropping*, penambahan *noise*, *resizing* dan *rotation* (Juergen, 2004).

1. *Sharpening* (Penajaman Citra)

Penajaman citra bertujuan memperjelas tepi pada objek didalam citra. Penajaman citra dilakukan dengan melewati citra pada penapis lolos-tinggi (high-pass filter). Karena penajaman citra

lebih berpengaruh pada tepi (edge) objek, maka penajaman citra disebut juga penajaman tepi (edge sharpening) atau peningkatan kualitas tepi (edge enhancement). Akibatnya, pinggiran objek terlihat lebih tajam dibandingkan sekitarnya (Munir, 2004).

2. *Cropping*

Cropping adalah istilah yang sering digunakan dalam dunia desain grafis. *Cropping* bertujuan untuk mengambil bagian tertentu saja dari sebuah gambar/foto dan membuang bagian yang tidak diperlukan. Contoh yang paling sederhana adalah hanya mengambil bagian objek yang biasanya dijadikan sebagai *foreground* (misalnya gambar orang, hewan, atau benda).

3. Penambahan *noise*

Dalam jurnal Alfatwa (2008) disebutkan citra digital sangat rentan mendapatkan serangan berbagai macam jenis *noise*. Ada beberapa cara yang menyebabkan suatu *noise* dapat berada didalam sebuah citra tersebut, 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*.

4. *Rotation*

Rotation adalah istilah yang sering digunakan dalam dunia desain grafis. *Rotation* bertujuan untuk memutar posisi gambar. *Rotation* yang dilakukan dapat searah dengan jarum jam atau berlawanan dengan arah jarum jam.

5. *Resize Image*

Resize Image adalah istilah yang sering digunakan dalam dunia desain grafis. *Resize image* dilakukan untuk mengubah ukuran pixel menjadi lebih besar ataupun lebih kecil.

2.4 Discrete Wavelet Transform (DWT)

2.4.1 Pengertian Discrete Wavelet Transform

Transformasi *wavelet* diskrit diperkenalkan pada tahun 1976 dimana pada saat itu teknik untuk mendekomposisi sinyal waktu diskrit ditemukan (Sripathi, 2003).

Transformasi *wavelet* diskrit sendiri secara umum merupakan dekomposisi citra pada frekuensi *subband* dimana komponennya dihasilkan dengan cara penurunan level dekomposisi. Implementasi transformasi *wavelet* diskrit dapat dilakukan dengan cara melewatkan sinyal frekuensi tinggi atau *highpass filter* dan frekuensi rendah atau *lowpass filter* untuk dianalisis. *Highpass filter* digunakan untuk menganalisis frekuensi tinggi dan *lowpass filter* untuk menganalisis frekuensi rendah. Analisis terhadap frekuensi dilakukan dengan menggunakan resolusi hasil frekuensi sinyal. Resolusi merupakan jumlah pixel yang ditampilkan per-unit dari panjang suatu image, biasanya diukur dalam pixel per inch (ppi). Analisis frekuensi yang berbeda dengan menggunakan resolusi yang berbeda disebut dengan *Multiresolution Analysis* (Alfatwa, 2008).

2.4.2 Proses Dekomposisi Citra

Proses dekomposisi dimulai dengan melewatkan sinyal asal melewati *highpass filter* dan *lowpass filter*. Misalkan sinyal asal ini memiliki rentang frekuensi dari 0 sampai dengan π rad/s. Dalam melewati *highpass filter* dan *lowpass filter* ini, rentang frekuensi di-*subsample* menjadi dua, sehingga rentang frekuensi tertinggi pada masing-masing *subsample* menjadi $\pi/2$ rad/s. Setelah filterisasi setengah dari *sample* atau salah satu *subsample* dapat dieliminasi berdasarkan aturan Nyquist. Proses dekomposisi ini dapat melalui satu atau lebih tingkatan (Alfatwa, 2008). Jika suatu citra dilakukan proses transformasi *wavelet* diskrit dua dimensi dengan level dekomposisi satu, maka akan menghasilkan empat buah *subband* yaitu (Novamizanti, 2009):

1. Koefisien Aproksimasi ($CA_j + 1$) atau disebut juga dengan *subband* LL
2. Koefisien Detil Horisontal ($CD(h) + 1$) atau disebut juga dengan *subband* HL
3. Koefisien Detil Vertikal ($CD(v)j + 1$) atau disebut juga dengan *subband* LH
4. Koefisien Detil Diagonal ($CD(d)j + 1$) atau disebut juga dengan *subband* LH

Hasil *subband* hasil proses dekomposisi level 1 ditunjukkan dengan gambar 2.6.

LL	HL
LH	HH

Gambar 2.6 Hasil proses dekomposisi level 1

Dengan Level dekomposisi I *Subband* hasil dari dekomposisi dapat didekomposisi lagi karena level dekomposisi *wavelet* bernilai dari 1 sampai n atau disebut juga transformasi *wavelet multilevel*. Jika dilakukan dekomposisi lagi, maka *subband* LL akan didekomposisi karena *subband* berisi sebagian dari informasi citra. Jika dilakukan dekomposisi dengan level dekomposisi dua maka *subband* LL akan menghasilkan empat buah *subband* baru yaitu *subband* LL2 (koefisien aproksimasi 2), HL2 (koefisien Detil Horisontal 2), LH2 (Koefisien Detil vertical 2), dan (koefisien Detil Diagonal 2). Dan begitu seterusnya jika dilakukan dekomposisi lagi (Novamizanti, 2009).

Hasil *subband* hasil proses dekomposisi level 2 ditunjukkan dengan gambar 2.7.

LL2	HL2	HL1
LH2	HH2	
LH1		HH1

Gambar 2.7 Proses Dekomposisi Level 2

Bila citra asli f dengan $M \times N$ pixel didekomposisi menjadi 4 *subband* sesuai frekuensinya yakni LL, LH, HL, dan HH dengan menggunakan transformasi *wavelet* dengan filter *Haar*, secara matematis dihasilkan dengan menggunakan persamaan 2.7-2.10 (Novamizanti, 2009).

$$ll(x, y) = \frac{1}{4} \sum_{i=0}^1 f(2x + i, 2y + j) \quad (2.7)$$

$$lh(x, y) = \frac{1}{4} \sum_{i=0}^1 f(2x + i, 2y) - \frac{1}{4} \sum_{i=0}^1 f(2x + i, 2y + 1) \quad (2.8)$$

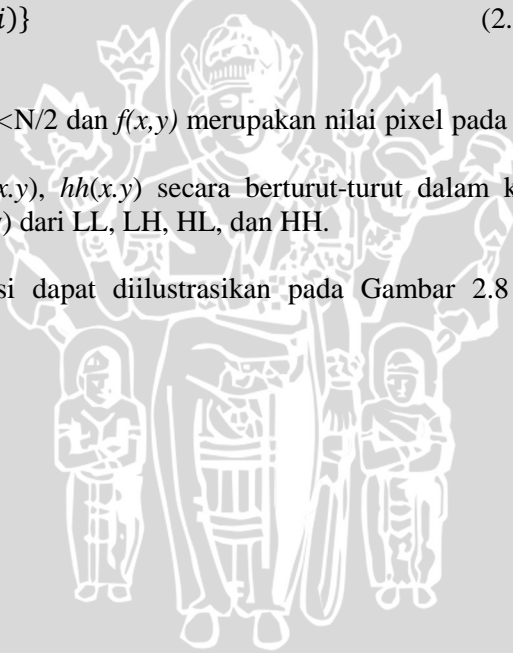
$$hl(x, y) = \frac{1}{4} \sum_{i=0}^1 f(2x, 2y + i) - \frac{1}{4} \sum_{i=0}^1 f(2x + 1, 2y + i) \quad (2.9)$$

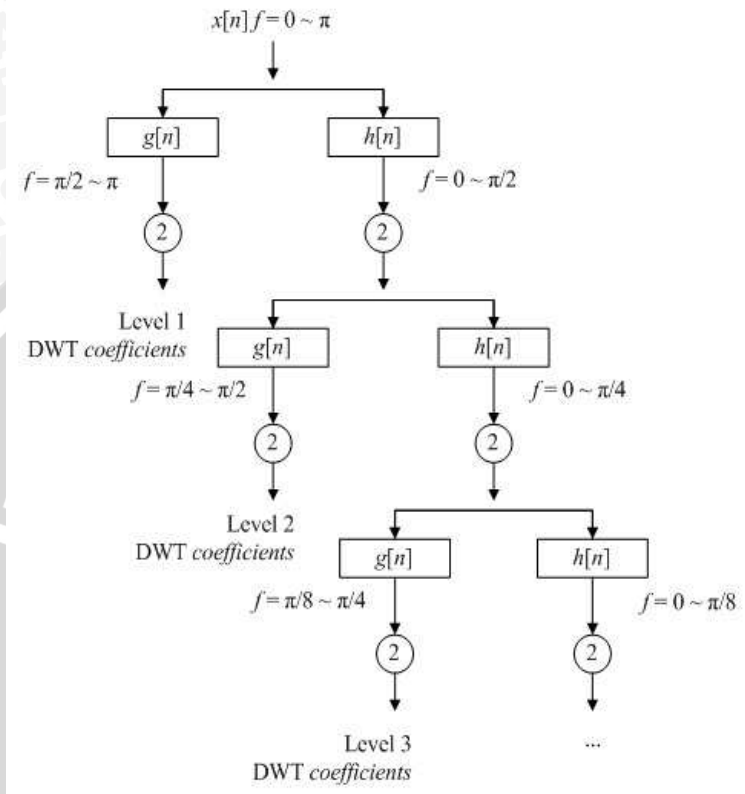
$$hh(x, y) = \frac{1}{4} \{f(2x, 2y) + f(2x + 1, 2y + 1) - f(2x + 1, 2y) - f(2x, 2y + i)\} \quad (2.10)$$

dimana:

- $0 \leq x < M/2, 0 \leq y < N/2$ dan $f(x, y)$ merupakan nilai pixel pada koordinat (x, y) pada citra f
- $ll(x, y), lh(x, y), hl(x, y), hh(x, y)$ secara berturut-turut dalam komponen pada koordinat (x, y) dari LL, LH, HL, dan HH.

Proses dekomposisi dapat diilustrasikan pada Gambar 2.8 (Polikar, 1998).





Gambar 2.8 Dekomposisi wavelet dengan fungsi dasar $x[n] f=0 \sim \pi$
(Sumber: Polikar, 1998)

2.4.3 Proses Discrete Wavelet Transform

Secara umum penyisipan *watermark* dilakukan dengan cara memodifikasi koefisien pada rentang frekuensi LL, LH, HL, atau HH yang merupakan rentang frekuensi hasil dekomposisi citra menggunakan *wavelet*. Data *watermark* ini dapat dianggap sebagai rangkaian bilangan w dengan panjang L , yang disisipkan pada koefisien rentang yang dipilih f (Alfatwa, 2008).

Algoritma umum penyisipan *watermark* pada koefisien ditunjukkan persamaan 2.11 dan 2.12.

$$a. f_{LH}'(x,y) = f_{LH} + \alpha.w(k) + K(k), k=1, \dots, L, \dots \quad (2.11)$$

$$b. f_{HL}'(x,y) = f_{HL} + \alpha \cdot w(k) + K(k), k=1, \dots, L \dots \quad (2.12)$$

dimana:

f_{LH}' = koefisien yang dimodifikasi untuk posisi (x,y)

f_{LH} = koefisien terbesar yang dipilih

f_{HL}' = koefisien yang dimodifikasi untuk posisi (x,y)

f_{HL} = koefisien terbesar yang dipilih

k = sebuah array yang memiliki panjang tertentu dimana panjang maksimumnya adalah sepanjang *watermark*. Kunci k ini ditambahkan dalam proses penyisipan melalui proses penjumlahan pada indeks yang bersesuaian.

α = kekuatan penyisipan yang mengontrol tingkat kekuatan penyisipan *watermark*.

Menurut jurnal “A VLSI Architecture for the Discrete Wavelet Transform” karya Markus Rullmann, disebutkan bahwa *Discrete Wavelet Transform* (DWT) telah memberikan banyak aplikasi dalam proses sinyal digital. Untuk menerima deret persamaan transformasi untuk kasus waktu diskrit, diasumsikan *sample* ditempatkan sama. Oleh sebab itu, mereka membentuk barisan $x(n)$ dimana n adalah sebuah *integer*. Parameter skala berubah ke j , Persamaan dapat dituliskan pada persamaan 2.13.

$$DWT_x(k,j) = (x, \Psi_{k,j}) = 2^{-\frac{j}{2}} \sum_{n=-\infty}^{\infty} x(n) \Psi(2^{-j}n - k) \quad (2.13)$$

Sebagai hasil dari DWT, koefisien *wavelet* diskrit diterima. Koefisien ini menghitung energi sinyal distribusi di setiap *channel* ke parameter skala j pada waktu k .

Jika *wavelet* $(x, \psi_{k,j})$ bentuk fungsi basis *orthonormal*. *Invers* DWT dapat dilakukan dengan persamaan 2.14.

$$x(n) = \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} DWT_x(k,j) \Psi(2^{-j}n - k) \quad (2.14)$$

2.4.4 Proses *Inverse Discrete Wavelet Transform*(IDWT)

Proses IDWT dilakukan untuk membentuk citra ter-*watermark* setelah dilakukan proses penyisipan *watermark* pada citra digital menggunakan DWT (Alfatwa, 2008).

Pada saat proses *Inverse Discrete Wavelet Transform* (IDWT), citra kembali dilakukan proses *upsampling* yang diikuti proses filtering oleh *lowpass filter* rekonstruksi dan *highpass filter* rekonstruksi (Novamizanti, 2009).

2.5 Mean Square Error (MSE)

MSE merupakan salah satu cara untuk mengukur jumlah perbedaan antara nilai perkiraan dengan nilai yang sebenarnya. MSE mengukur rata-rata wilayah kesalahan (*error*). MSE (*Mean Square Error*) merupakan sigma dari jumlah kesalahan (*error*) antara citra hasil *watermark* dan citra asli. Perhitungan nilai MSE dari citra digital berukuran NxM, dilakukan sesuai dengan persamaan 2.15.

$$MSE = \frac{1}{MN} \sum_{Y=1}^M \sum_{X=1}^N [I(x, y) - I'(x, y)]^2 \quad (2.15)$$

Dimana $I(x,y)$ adalah nilai piksel di citra asli,
 $I'(x,y)$ adalah nilai piksel pada citra hasil *watermark*,
 M, N adalah dimensi citra.

Nilai MSE yang rendah akan lebih baik (Linda S., 2005).

2.6 Peak Signal to Noise Ratio (PSNR)

Pada proses kuantisasi, terjadi eliminasi informasi atau adanya penurunan kualitas informasi. Informasi yang hilang akibat proses *watermark* seharusnya seminimal mungkin sehingga kualitas hasil citra *watermark* bagus. PSNR merupakan salah satu cara untuk mengukur tingkat kesalahan akibat hilangnya informasi. PSNR memiliki satuan *decibel* (dB), semakin besar nilai PSNR semakin bagus kualitas hasil *watermarking*. Perhitungan nilai PSNR dari citra digital berukuran N x M, dilakukan sesuai dengan persamaan 2.16.

$$PSNR = 20xLog_{10} \left(\frac{255}{\sqrt{\frac{1}{MN} \sum_{Y=1}^M \sum_{X=1}^N [I(x, y) - I'(x, y)]^2}} \right) \quad (2.16)$$

Dimana $I(x,y)$ adalah nilai piksel di citra asli,
 $I'(x,y)$ adalah nilai piksel pada citra hasil *watermarking*,
 M, N adalah dimensi citra.
Nilai PSNR yang tinggi akan lebih baik (Linda S., 2005).

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



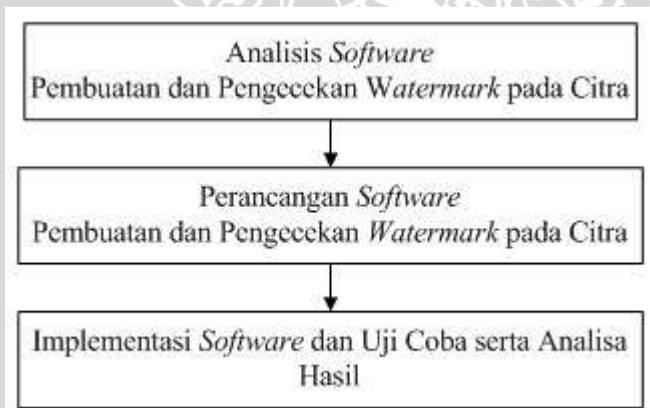
BAB III METODOLOGI DAN PERANCANGAN

Pada bagian bab metode dan perancangan ini, akan dibahas penggunaan metode yang digunakan dalam pembuatan perangkat lunak untuk pembuatan dan pengecekan *watermarking* serta dijelaskan pula langkah-langkah dalam pembuatan perangkat lunak ini.

Tahapan pembuatan perangkat lunak adalah sebagai berikut:

1. Mempelajari metode yang digunakan pada jurnal yang pernah ada, yang telah disinggung pada bab sebelumnya.
2. Membuat perangkat lunak berdasarkan analisis dan perancangan.
3. Uji coba perangkat lunak dengan menggunakan citra berwarna.
4. Evaluasi hasil pengecekan yang dilakukan oleh perangkat lunak untuk mengetahui kehandalan citra yang telah di-*watermark*.

Langkah-langkah pembuatan perangkat lunak dapat digambarkan melalui gambar 3.1.



Gambar 3.1 Diagram Alir Pembuatan Perangkat Lunak

3.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak ini dirancang untuk dapat melakukan suatu penyisipan *watermark* ke dalam sebuah citra digital. *Watermark* yang dipilih berupa citra berwarna yang akan berperan sebagai inputan pada sistem.

Perangkat lunak ini dibangun pada satu lingkungan pengembangan saja yaitu pada *PC (Personal Computer)*. Sehingga proses *watermarking* dan

ekstraksi dilakukan pada lingkungan pengembangan yang sama. Untuk itu *user* harus memilih proses pada awal berjalannya sistem. Proses *watermarking* bertujuan untuk melakukan *watermarking* pada citra *host*, sedangkan untuk proses ekstraksi bertujuan untuk melakukan proses ekstraksi *watermark* dari citra *digital* yang sudah ber- *watermark*.

Ekstraksi *watermark* dirancang untuk dapat melakukan perbandingan antara citra *host* dengan citra ber-*watermark* untuk mendapatkan *watermark*. Citra ber-*watermark* dimasukkan ke dalam sistem sehingga dapat dilakukan proses ekstraksi *watermark*.

3.2 Batasan Perangkat Lunak

Batasan dari perangkat lunak yang akan dikembangkan yaitu:

1. Ukuran citra $N \times M$ dimana $N > 0$, $M > 0$ dan N , M adalah bilangan bulat.
2. Perangkat lunak hanya bisa memproses citra dengan ekstensi *.bmp*.

3.3 Perancangan Sistem Kerja Perangkat Lunak

Berdasarkan deskripsi umum perangkat lunak yang telah dijelaskan sebelumnya, maka pada subbab ini akan dibahas mengenai arsitektur dan proses yang akan terjadi pada perangkat lunak.

3.3.1 Perancangan proses *input* pembuatan perangkat lunak

Contoh citra yang diproses dalam perangkat lunak ini adalah citra berwarna yang berekstensi *.bmp* dan penanaman untuk *watermark* akan diujikan pada kanal warna RGB dan YCbCr.

Pada proses *input* ini, citra yang digunakan dalam perangkat lunak kali ini adalah citra berwarna. Penanaman *watermark* pada citra akan diujikan pada ruang warna RGB untuk seluruh kanal warna (R, G, B) dan ruang YCbCr untuk seluruh kanal warna (Y, Cb, Cr) dengan menyisipkannya hanya pada satu kanal warna untuk setiap proses penyisipan. Untuk proses penanaman pada ruang warna YCbCr dapat dilakukan setelah dilakukan proses transformasi dari ruang warna RGB kedalam ruang warna YCbCr. Kemudian setelah *watermark* disisipkan akan ditransformasikan kembali ke dalam ruang warna RGB.

3.3.2 Perancangan proses perhitungan pembuatan *watermark Discrete Wavelet Transform*

Dalam proses perangkat lunak ini, untuk mengubah citra ke dalam domain frekuensi menggunakan perhitungan yang telah dijelaskan pada bab 2 persamaan 2.13.

Pada DWT ini, sinyal dianalisis pada frekuensi berbeda dengan resolusi yang berbeda melalui dekomposisi sinyal sehingga menjadi detail informasi dan taksiran kasar. DWT bekerja pada dua kumpulan fungsi yang disebut fungsi penskalaan dan fungsi *wavelet* yang masing-masing berhubungan dengan *lowpass filter* dan *highpass filter*.

3.3.3 Perancangan Proses Dekomposisi *Watermark*.

Proses dekomposisi dalam *Watermark* DWT dapat dijelaskan sebagai berikut:

1. Mentransformasikan RGB ke dalam ruang warna YCbCr dan pilih salah satu kanal Y, Cb, atau Cr.
2. Melakukan dekomposisi level satu pada citra untuk proses transformasi *wavelet* diskrit dua dimensi dengan menggunakan persamaan 2.7-2.10 sehingga dihasilkan empat buah *subband* yaitu:
 5. Koefisien Aproksimasi ($CA_j + 1$) atau disebut juga dengan *subband* LL
 6. Koefisien Detil Horisontal ($CD(h) + 1$) atau disebut juga dengan *subband* HL
 7. Koefisien Detil Vertikal ($CD(v)j + 1$) atau disebut juga dengan *subband* LH
 8. Koefisien Detil Diagonal ($CD(d)j + 1$) atau disebut juga dengan *subband* LH
3. Mencari koefisien terbesar dari rentang frekuensi LH atau dari rentang frekuensi HL untuk selanjutnya dilakukan penyisipan *watermark*.
4. Untuk dekomposisi dengan level yang lebih tinggi dilakukan terhadap koefisien aproksimasi dari proses sebelumnya.

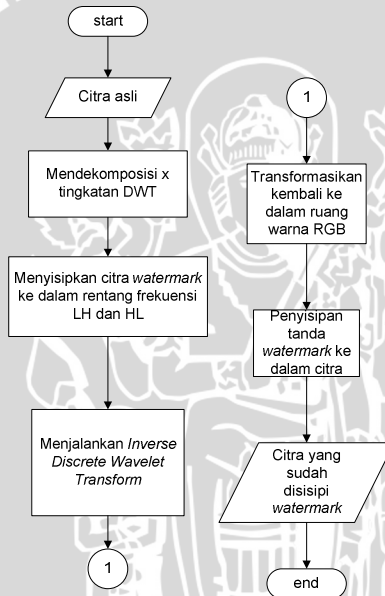
3.3.4 Perancangan Proses Penyisipan *Watermark*

Penyisipan *watermark* dapat dijelaskan dalam langkah-langkah berikut:

1. Setelah proses dekomposisi, *watermark* disisipkan ke dalam rentang frekuensi LH_x dan HL_x

2. Menyisipkan data *watermark* yang dapat dianggap rangkaian bilangan w ke dalam rentang frekuensi LH (f_{LH}) dan frekuensi HL (f_{HL}) menggunakan persamaan 2.11 dan 2.12
3. *Inverse Discrete Wavelet Transform* dijalankan untuk membentuk citra *watermark*
4. Transformasikan kembali ke ruang warna RGB untuk mendapatkan citra yang disisipi *watermark*.

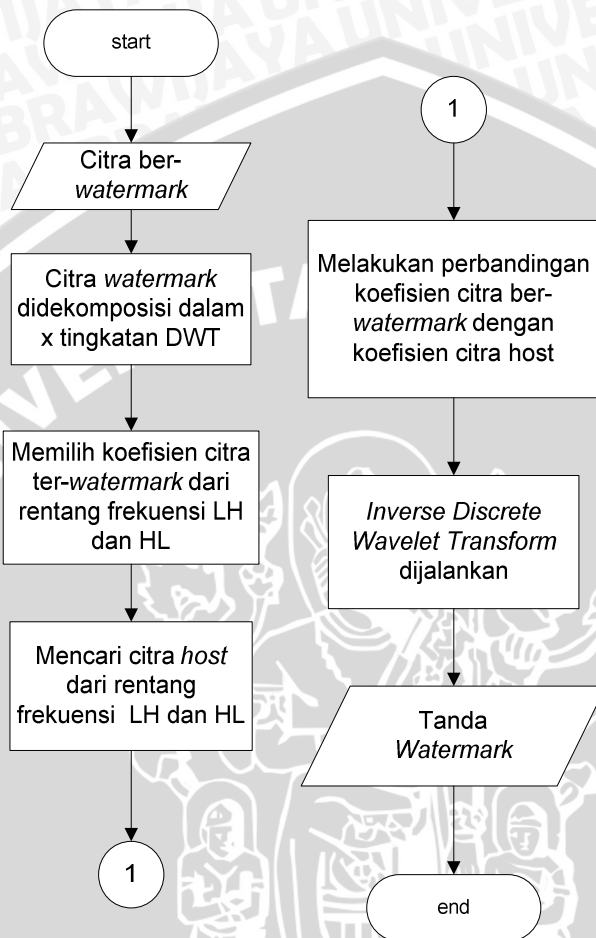
Diagram alir untuk proses penyisipan *watermark* ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Alir Proses Penyisipan *Watermark*

3.3.5 Perancangan proses ekstraksi citra *watermark*

Diagram alir untuk proses Ekstraksi *watermark* ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram Alir Proses Ekstraksi *Watermark*

Pengecekan *watermark* dapat dijelaskan dalam langkah-langkah berikut

1. Citra yang sudah memuat *watermark*.
2. Citra *watermark* didekomposisi dalam x tingkatan DWT
3. Memilih koefisien citra ber – *watermark* dari rentang frekuensi LH dan HL yaitu f_{LH} dan f_{HL} .
4. Mencari citra *host* dari rentang frekuensi LH dan HL yaitu f_{LH} dan f_{HL} .
5. Melakukan perbandingan koefisien citra ber – *watermark* dengan koefisien citra *host* untuk menghasilkan *watermark*

6. *Inverse Discrete Wavelet Transform* dijalankan.

3.4 Perancangan Uji Coba

Pengujian yang dilakukan dalam sistem perangkat lunak ini dilihat kualitas citra *watermark* digital yang dihasilkan dibandingkan dengan citra asli, ketahanan *watermark* dalam citra terhadap beberapa serangan citra di antaranya adalah *sharpening*, *adding noise*, *cropping*, *rotation*, *resizing*. Pengujian dari sisi *nonperceptibility* telah dijelaskan pada Bab 2.3.1.

3.4.1 Citra Uji

Spesifikasi citra yang akan diujikan, yaitu:

1. Ukuran *image* untuk perhitungan manual 4 x 4 pixel
2. Ukuran *image* untuk citra yang disisipi *watermark* adalah 800x600 pixel dan 700x700 pixel
3. Empat citra untuk melihat kualitas citra setelah diberikan *watermark*.
4. Citra yang digunakan sebagai tanda *watermark* adalah citra *mipa2010.bmp* dengan ukuran 20x20, 40x40, dan 60x60 pixel.

3.4.2 Lingkungan pengujian

Perangkat lunak yang digunakan adalah *Microsoft Access* sebagai media penyimpanan data, *Microsoft Excel* untuk pengolahan data serta *Adobe Photoshop CS3* untuk manipulasi citra *Watermark*. Perangkat keras yang digunakan dalam pengujian ini sama dengan proses Implementasi.

3.4.3 Pengujian kualitas citra *watermark*

Pada pengujian kualitas citra *watermark* ini menggunakan *Mean Square Error* (MSE). MSE ini akan melihat citra asli sebagai $I(x,y)$ sedangkan untuk citra *watermark* disimbolkan dengan $I'(x,y)$, dengan ukuran citra $N \times N$.

Pengukuran MSE ini menunjukkan rata-rata perbedaan antara intensitas citra asli $I(x,y)$ dan intensitas citra *watermark* $I'(x,y)$. $N \times N$ adalah ukuran panjang dan lebar citra. Untuk perhitungan MSE ditunjukkan pada bab 2 dalam persamaan 2.15.

Pengujian nilai PSNR menggunakan persamaan 2.16. Semakin besar nilai PSNR maka semakin baik hasil *watermarking* citra. Masukan dari pengujian ini berupa nilai piksel dari citra asli dan citra hasil *watermarking*.

Keluaran dari pengujian ini berupa nilai kesalahan. Makin besar kesalahannya, maka makin kecil nilai PSNR yang dihasilkan dan sebaliknya. Rancangan Tabel Hasil Uji Kualitas Citra yang telah disisipi *watermark* ditunjukkan pada Tabel 3.1.

Tabel 3.1 Rancangan Tabel Hasil Uji Kualitas Citra yang Disisipi *Watermark*

No	Citra Asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR

Keterangan :

Citra Asli	Citra asli yang akan diberi <i>watermark</i>
Citra <i>watermark</i>	Citra yang akan disisipi ke dalam citra asli
Citra yang telah disisipi <i>watermark</i>	Citra hasil yang telah diberi <i>watermark</i>
MSE	Rata – rata kemiripan citra asli (sebelum diberi <i>watermark</i>) dengan citra yang disisipi <i>watermark</i> . MSE dihitung sesuai persamaan(2.15)
PSNR	Tingkat kualitas dari citra yang disisipi <i>watermark</i> . PSNR dihitung sesuai persamaan (2.16)

3.4.4 Pengujian terhadap ketahanan *watermark*

Uji coba ketahanan yang dilakukan adalah dengan memberikan beberapa serangan kepada citra yang disisipi *watermark* dan kemudian dilakukan pengujian apakah *watermark* masih dapat dideteksi atau tidak. Adapun variasi pemrosesan citra yang dilakukan adalah:

- Resizing* citra dengan memperkecil ukuran 25%,50%,75% dari ukuran citra asli
- Pemotongan citra (*cropping*) sebesar 25 dan 40%
- Penambahan *noise* dengan *variance* 5% dan 10% untuk *Gaussian Noise* dan *uniform noise*.
- Penajaman citra yang disisipi *watermark* (*sharpening*) untuk *sharpen* dan *sharpen edge*.
- Rotation* citra yang disisipi *watermark* masing-masing 20⁰CW, 20⁰CCW, 50⁰CW dan 50⁰CCW.

Rancangan Tabel Uji Ketahanan Citra yang disisipi *watermark* ditunjukkan pada Tabel 3.2.

Tabel 3.2 Rancangan Tabel Uji Ketahanan citra yang disisipi *Watermark*

Citra yang disisipi <i>watermark</i>	Perlakuan	α	Deteksi <i>Watermark</i>

Hasil perhitungan untuk uji ketahanan akan disimpan dalam tabel uji ketahanan dimana untuk citra *watermark* yang telah di manipulasi disimpan dalam *field* citra yang disisipi *watermark* dan *field* perlakuan untuk menyimpan jenis manipulasi yang digunakan. Untuk *field* deteksi *watermark* digunakan untuk menyimpan hasil pendeteksian *watermark* dan α merupakan kekuatan penyisipan yang mengontrol tingkat kekuatan penyisipan *watermark*.

3.4.5 Pengujian *Watermark* terhadap *Nonperceptibility*

Seperti yang telah dijelaskan pada Bab II, pengujian ini didasarkan pada *Human Visual System*. Setelah citra asli disisipkan *watermark*, maka akan diujikan pada penglihatan mata manusia normal, apakah dapat membedakan antara citra asli yang belum disisipi *watermark* dan citra yang telah disisipi *watermark*. Pada pengujian ini, dilakukan tes terhadap 10 orang dari mulai usia 20 - 30 tahun. Rancangan tabel hasil uji *nonperceptibility* citra yang disisipi *watermark* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Rancangan Tabel Hasil Uji *Nonperceptibility* Citra yang disisipi *Watermark*

No	Nama Responden	Citra	Hasil

3.5 Contoh Perhitungan Manual Penyisipan *Watermark*

Pada contoh perhitungan manual kasus berikut digunakan matriks 4 x 4 untuk mempermudah proses perhitungan. Pixel citra yang digunakan akan dipisahkan menjadi komponen-komponen *red*, *green*, dan *blue* yang ditunjukkan pada Tabel 3.4 sampai dengan Tabel 3.6.

Tabel 3.4 WarnaR (*Red*)

	0	1	2	3
0	38	57	35	40
1	37	102	50	26
2	25	35	23	17
3	28	13	19	34

Tabel 3.5 WarnaG (*Green*)

	0	1	2	3
0	35	56	39	45
1	39	106	58	36
2	29	40	33	29
3	36	23	33	50

Tabel 3.6 Warna B (*Blue*)

	0	1	2	3
0	37	58	40	46
1	40	107	58	36
2	30	41	33	29
3	36	23	32	49

Perhitungan Manual transformasi RGB ke Ruang Warna *YCbCr* .

Perhitungan manual transformasi RGB ke ruang warna *YCbCr* menggunakan persamaan 2.1 sampai dengan 2.3.

Contoh Perhitungan:

$$Y = 0.299900R + 0.58700G + 0.11400B$$

$$Y = 11.3962 + 20.545 + 4.218$$

$$Y = \mathbf{36.1592}$$

$$Cb = -0.16874R - 0.33126G + 0.50000B$$

$$Cb = -6.41212 - 11.5941 + 18.5$$

$$Cb = \mathbf{0.49378}$$

$$Cr = 0.50000R - 0.41869G - 0.08131B$$

$$Cr = 19 - 14.65415 - 3.00847$$

$$Cr = 1.33738$$

Hasil Lengkap Perhitungan Transformasi RGB ke YCbCr ditunjukkan pada tabel 3.7 sampai dengan 3.9.

Tabel 3.7 WarnaY (*Luminance*)

	0	1	2	3
0	36.1592	56.5783	37.9495	43.655
1	38.5493	105.0098	55.653	33.0334
2	27.9405	38.6505	30.0307	25.4273
3	33.6332	20.0217	28.7171	45.1326

Tabel 3.8 WarnaCb (*Chrominance Blue*)

	0	1	2	3
0	0.49378	0.83126	1.17496	1.3437
1	0.83748	1.17496	1.34992	1.6874
2	1.17496	1.3437	1.6874	2.02488
3	1.34992	1.6874	1.86236	2.19984

Tabel 3.9 WarnaCr (*Chrominance Red*)

	0	1	2	3
0	1.33738	0.3378	-2.08131	-2.58131
1	-1.08131	-2.08131	-4	-5
2	-2.08131	-2.58131	-5	-6
3	-4	-5	-6.91869	-7.91869

1. **Perhitungan Manual** transformasi DWT pada *Luminance* Y dengan $\Psi = 0.1$

Perhitungan manual transformasi DWT pada *Luminance* Y dengan $\Psi = 0.1$ menggunakan persamaan 2.13.

Contoh Perhitungan:

$$DWT_x(k, j) = (x, \Psi_{k, j}) = 2^{-\frac{j}{2}} \sum_{n=-\infty}^{\infty} x(n) \Psi(2^{-j} n - k)$$

$$\begin{aligned} \text{DWT}_x(0,0) &= 2^{\frac{-0}{2}} (36.1592)(0.1)(2^{-0} 36.1592 - 0) \\ &= \mathbf{130.7487} \end{aligned}$$

Hasil lengkap perhitungan transformasi DWT ditunjukkan dengan tabel 3.10.

Tabel 3.10 Transformasi DWT

	0	1	2	3
0	130.7487	113.1761	18.0020	16.6670
1	144.7499	386.1527	38.0200	4.6765
2	72.4790	50.0829	10.5222	2.6326
3	103.0292	12.1837	9.2315	8.4037

2. **Perhitungan Manual** Transformasi IDWT untuk mendapatkan komponen Y baru.

Perhitungan manual transformasi IDWT untuk mendapatkan komponen Y baru menggunakan persamaan 2.14.

Contoh Perhitungan

$$\begin{aligned} x(n) &= \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} \text{DWT}_x(k,j) \Psi(2^{-j}n - k) \\ &= 130.74877 (0.1) (2^{-0} 36.1592 - 0) \\ &= 472.7770 \end{aligned}$$

Hasil perhitungan lengkap transformasi IDWT ditunjukkan pada tabel 3.11.

Tabel 3.11 Transformasi IDWT

	0	1	2	3
0	472.7770	320.1656	17.0792	12.7942
1	543.5258	740.1833	51.9477	1.8725
2	188.0143	91.7702	7.3736	2.6326
3	315.6115	10.3693	5.9351	8.4037

3. **Perhitungan Manual** Transformasi empat *subband* sesuai frekuensinya yaitu LL, LH, HL, dan HH.

Perhitungan manual transformasi empat *subband* sesuai frekuensinya yaitu LL, LH, HL, dan HH menggunakan persamaan 2.7 sampai dengan 2.10.

Contoh Perhitungan :

Perhitungan *subband* LL untuk $x,y (0,1)$

$$\begin{aligned} ll(x,y) &= \frac{1}{4} \sum_{i=0}^1 f(2x+i, 2y+j) \\ &= \frac{1}{4} f(0,2) + f(1,3) \\ &= \frac{1}{4} (70,9829) \\ &= \mathbf{17,745725} \end{aligned}$$

Perhitungan *subband* LH untuk $x,y (0,1)$

$$\begin{aligned} lh(x,y) &= \frac{1}{4} \sum_{i=0}^1 f(2x+i, 2y) - \frac{1}{4} \sum_{i=0}^1 f(2x+i, 2y+1) \\ &= \frac{1}{4} (f(0,1) + f(1,1)) - \frac{1}{4} (f(0,2) - f(1,3)) \\ &= \frac{1}{4} (56,5783 + 105,0098) - \frac{1}{4} (37,9495 - 33,0334) \\ &= \mathbf{22,6513} \end{aligned}$$

Perhitungan *subband* HL untuk $x,y (0,1)$

$$hl(x,y) = \frac{1}{4} \sum_{i=0}^1 f(2x, 2y+i) - \frac{1}{4} \sum_{i=0}^1 f(2x+1, 2y+i)$$

$$\begin{aligned}
 &= \frac{1}{4}(f(0,2) + f(0,3)) - \frac{1}{4}(f(1,1) + f(1,2)) \\
 &= \frac{1}{4}(37,9495 + 61,411) - \frac{1}{4}(105,0098 + 55,653) \\
 &= -15,325575
 \end{aligned}$$

Perhitungan subband HH untuk x,y (0,1)

$$\begin{aligned}
 hh(x,y) &= \frac{1}{4} \{f(2x, 2y) + f(2x + 1, 2y + 1) - f(2x + 1, 2y) - \\
 &\quad f(2x, 2y + i)\} \\
 &= \frac{1}{4} \{f(0,2) + f(1,3) - f(1,2) - f(0,2)\} \\
 &= \frac{1}{4}(37,9495 + 33,0334 - 55,653 - 37,9495) \\
 &= -22,6196
 \end{aligned}$$

Untuk hasil lengkap perhitngan transformasi empat *subband* sesuai frekuensi yaitu LL, LH, HL,dan HH ditunjukkan dengan Tabel 3.12 sampai dengan 3.15.

Tabel 3.12 Subband LL

	0	1	2	3
0	36.1592	17.7457	37.9495	61.411
1	38.5493	105.0098	55.653	33.0334
2	27.9405	38.6505	30.0307	25.4273
3	33.6332	20.0217	28.7171	45.1326

Tabel 3.13 Subband LH

	0	1	2	3
0	36.1592	22.6513	37.9495	61.411
1	38.5493	105.0098	55.653	33.0334
2	27.9405	38.6505	30.0307	25.4273
3	33.6332	20.0217	28.7171	45.1326

Tabel 3.14 Subband HL

	0	1	2	3
0	36.1592	-15.3256	37.9495	61.411
1	38.5493	105.0098	55.653	33.0334
2	27.9405	38.6505	30.0307	25.4273
3	33.6332	20.0217	28.7171	45.1326

Tabel 3.15 Subband HH

	0	1	2	3
0	36.1592	-22.6196	37.9495	61.411
1	38.5493	105.0098	55.653	33.0334
2	27.9405	38.6505	30.0307	25.4273
3	33.6332	20.0217	28.7171	45.1326

4. Hasil Transformasi ke ruang warna *RGB* untuk mendapatkan citra yang disisipi *watermark*, ditunjukkan pada Tabel 3.16.

Tabel 3.16 Citra yang disisipi *watermark*

	0	1	2	3
0	38.0342	53.0272	35.0315	40.036
1	37.0333	102.0918	50.045	26.0234
2	25.02250	35.0315	23.0207	17.0153
3	28.0252	13.0117	19.0171	34.1126

Pada Tabel 3.16 didapatkan hasil bahwa nilai citra yang disisipi *watermark* tidak jauh berbeda dengan citra asli yang belum diberikan *watermark*.

3.5.1 Analisa Ketahanan watermark.

Perhitungan *Mean Square Error* (MSE), ditunjukkan pada Tabel 3.17.

Tabel 3.17 Rata – rata perbedaan citra asli dengan citra watermark

	0	1	2	3
0	0.00116964	15.7831398	0.00099225	0.036
1	0.00110889	0.008427	0.002025	0.00054756
2	0.050625	0.00099225	0.00042849	0.00023409
3	0.00063504	0.00013689	0.0029241	0.012679
Σ	0.05354	15.79269594	0.00373815	0.04946065

Perhitungan nilai MSE menggunakan persamaan 2.15.

$$\sum_{Y=1}^M \sum_{X=1}^N [I(x, y) - I'(x, y)]^2 = \mathbf{15.94889539}$$

$$MSE = \frac{1}{MN} \sum_{Y=1}^M \sum_{X=1}^N [I(x, y) - I'(x, y)]^2$$

$$MSE = \frac{1}{16} (15.94889539)$$

$$= \mathbf{0.996805961}$$

Maka Nilai MSE = 0.9375

Perhitungan nilai PSNR menggunakan persamaan 2.16.

$$PSNR = 10 \log \frac{255^2}{MSE}$$

$$PSNR = 10 \log \frac{255^2}{0.996805961}$$

$$= 18.4111$$

$$\text{Nilai PSNR} = 18.14469734$$

Pada Tabel 3.16 didapatkan hasil bahwa nilai PSNR adalah 18.411 yang menandakan bahwa kualitas citra yang disisipi *watermark* tidak jauh berbeda dengan citra asli yang belum diberikan *watermark*.

UNIVERSITAS BRAWIJAYA



BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Sistem

Lingkungan sistem yang akan dijelaskan dalam subbab ini adalah lingkungan perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi *watermarking* adalah sebagai berikut :

1. Processor Intel (R)Pentium(R)Dual CPU
2. RAM 512 MB
3. Monitor 13”
4. Keyboard
5. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem pembuatan citra *watermark* ini adalah :

1. Sistem operasi *Microsoft Windows XP Profesional*
2. Bahasa Pemograman *Microsoft Office C# 2008*

4.2 Implementasi Program

Berdasarkan perancangan perangkat lunak pada subbab 3.2 maka pada subbab ini akan dibahas mengenai implementasi dari perancangan tersebut.

4.2.1 Input

Proses penyisipan *watermark* bisa dilakukan langsung pada ruang warna RGB atau pada ruang warna YCbCR.. Apabila di dilakukan pada ruang warna YCbCr , perlu dilakukan suatu proses transformasi ruang warna RGB ke ruang warna YCbCr. Proses Transformasi untuk pengubahan ruang warna RGB ke ruang warna YCbCr ditunjukkan pada *sourcecode* 4.1.

```
public void GetYCrCb()  
{  
    //membuat array baru  
    imageYCbCr = new YCrCb[Gambar.Height,  
    Gambar.Width];
```

```

double Yv, Cr, Cb;
byte red, green, blue;
// isi log
parent.tLog.Items.Add(DateTime.Now.ToString() + " :
Starting Get Y Cr Cb");
for (int y = 0; y < Gambar.Height; ++y)
{
    for (int x = 0; x < Gambar.Width; ++x)
    {
        //ambil isi pixel pada posisi xy
        blue = imagePixel[y, x].GetB();
        green = imagePixel[y, x].GetG();
        red = imagePixel[y, x].GetR();
        //perhitungan YcrCb
        Yv = (0.299900 * red) + (0.58700 * green) +
(0.11400 * blue);
        Cb = -(0.16874 * red) - (0.33126 * green) +
(0.50000 * blue) + 128;
        Cr = (0.50000 * red) - (0.41869 * green) -
(0.08131 * blue) + 128;
        //mengisi hasil ke array
        imageYCbCr[y, x] = new YCrCb(Yv, Cr, Cb);
        //
        Console.WriteLine("$$$$$$${0} {1} {2}",
imageYCbCr[y, x].GetY(), imageYCbCr[y, x].GetCb(),
imageYCbCr[y, x].GetCr());
    }
}

parent.tLog.Items.Add(DateTime.Now.ToString() + " :
Complete Get Y Cr Cb");
}

```

Source Code 4.1 Source code transformasi RGB ke YCbCr

Untuk `imagePixel` mengambil nilai dari `class Pixel`.
Sourcecode `class Pixel` ditunjukkan pada sourcecode 4.2. Untuk

imageYCbCr mengambil nilai dari `class YCrCb` yang ditunjukkan pada *sourcecode* 4.3.

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace StegDWT
{
    class Pixel
    {
        byte r, g, b;
        public Pixel(byte R, byte G, byte B)
        {
            this.r = R;
            this.g = G;
            this.b = B;
        }
        public byte GetR() { return r; }
        public byte GetG() { return g; }
        public byte GetB() { return b; }
    }
}
```

Source Code 4.2 Source code class Pixel

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace StegDWT
{
    class YCrCb
    {
        private double Y, Cr, Cb;
        public YCrCb(double Y, double Cr, double
Cb)
        {
            this.Y = Y;
            this.Cb = Cb;
            this.Cr = Cr;
        }
    }
}
```

```

    }
    public double GetY() { return Y; }
    public double GetCr() { return Cr; }
    public double GetCb() { return Cb; }
}
}

```

Source Code 4.3 Source code class YCbCr

4.2.2 Dekomposisi Transformasi Wavelet

Dekomposisi pada citra akan menghasilkan 4 *subband* yang mengandung 4 informasi rentang frekuensi yang berbeda yaitu LL, LH, HH dan HL. Proses dalam dekomposisi citra ditunjukkan dalam *source code* 4.4.

```

private void Filtering()
{
    ArrayList aLL = new ArrayList();
    ArrayList aLH = new ArrayList();
    ArrayList aHL = new ArrayList();
    ArrayList aHH = new ArrayList();
    //isi log
    parent.tLog.Items.Add(DateTime.Now.ToString() + "
: Getting Range HL, LH");
    double LLv=0, LHv=0, HLv=0, HHv=0;
    for (int y = 0; y < Gambar.Height/2; ++y)
    {
        for (int x = 0; x < Gambar.Width/2; ++x)
        {
            LLv = 0;
            LHv = 0;
            HLv = 0;
            HHv = 0;

            for (int i = 0; i < 1; i++){
                // perhitungan posisi
                LLv = LLv + (imageDWT[2*y+i, 2*x+i].GetY())/4;
                LHv = LHv + ((imageDWT[2 * y, 2 * x +
i].GetY()) / 4) - ((imageDWT[2 * y + 1, 2 * x +
i].GetY()) / 4);
                HLv = HLv + ((imageDWT[2 * y + i, 2 * x
].GetY()) / 4) - ((imageDWT[2 * y + i, 2 * x +

```

```

1].GetY()) / 4);
    HHv = HHv + ((imageDWT[2 * y , 2 * x].GetY()
+ imageDWT[2 * y + 1, 2 * x + 1].GetY() -
imageDWT[2 * y , 2 * x + 1].GetY() +
imageDWT[2 * y + 1, 2 * x].GetY()) / 4);
}
    aLL.Add(LLv);
    aLH.Add(LHv);
    aHL.Add(HLv);
    aHH.Add(HHv);
}
}
aLL.Sort();
aLH.Sort();
aHL.Sort();
aHH.Sort();
//mendapatkan range frekuensi
LL = (double) aLL[0];
HL = (double) aHL[aHL.Count - 1];
LH = (double) aLH[0];
HH = (double) aHH[0];
aLL.Clear();
aLH.Clear();
aHL.Clear();
aHH.Clear();

for (int y = 0; y < Gambar.Height / 2; ++y)
{
    for (int x = 0; x < Gambar.Width / 2; ++x)
    {
        LLv = 0;
        LHv = 0;
        HLv = 0;
        HHv = 0;

        for (int i = 0; i < 1; i++)
        {
            LLv = LLv + (imageDWT[2 * y + i, 2 * x +
i].GetCr()) / 4;
            LHv = LHv + ((imageDWT[2 * y, 2 * x +
i].GetCr()) / 4) - ((imageDWT[2 * y + 1, 2 * x +

```

```

i].GetCr()) / 4);
    HLv = HLv + ((imageDWT[2 * y + i, 2 *
x].GetCr()) / 4) - ((imageDWT[2 * y + i, 2 * x +
1].GetCr()) / 4);
    HHv = HHv + ((imageDWT[2 * y, 2 * x].GetCr())
+ imageDWT[2 * y + 1, 2 * x + 1].GetCr()) -
                    imageDWT[2 * y, 2 * x
+ 1].GetCr() + imageDWT[2 * y + i, 2 * x].GetCr())
/ 4);
    }
    aLL.Add(LLv);
    aLH.Add(LHv);
    aHL.Add(HLv);
    aHH.Add(HHv);
    }
}

aLL.Sort();
aLH.Sort();
aHL.Sort();
aHH.Sort();
LL1 = (double)aLL[0];
HL1 = (double)aHL[aHL.Count - 1];
LH1 = (double)aLH[0];
HH1 = (double)aHH[aHH.Count - 1];
aLL.Clear();
aLH.Clear();
aHL.Clear();
aHH.Clear();

for (int y = 0; y < Gambar.Height / 2; ++y)
{
    for (int x = 0; x < Gambar.Width / 2; ++x)
    {
        LLv = 0;
        LHv = 0;
        HLv = 0;
        HHv = 0;

        for (int i = 0; i < 1; i++)
        {
            LLv = LLv + (imageDWT[2 * y + i, 2 * x +

```

```

i].GetCb()) / 4;
    LHv = LHv + ((imageDWT[2 * y, 2 * x +
i].GetCb()) / 4) - ((imageDWT[2 * y + 1, 2 * x +
i].GetCb()) / 4);
    HLv = HLv + ((imageDWT[2 * y + i, 2 *
x].GetCb()) / 4) - ((imageDWT[2 * y + i, 2 * x +
1].GetCb()) / 4);
    HHv = HHv + ((imageDWT[2 * y, 2 * x].GetCb() +
imageDWT[2 * y + 1, 2 * x + 1].GetCb() -
imageDWT[2 * y, 2 * x
+ 1].GetCb() + imageDWT[2 * y + i, 2 * x].GetCb())
/ 4);
}
    aLL.Add(LLv);
    aLH.Add(LHv);
    aHL.Add(HLv);
    aHH.Add(HHv);
}
}
aLL.Sort();
aLH.Sort();
aHL.Sort();
aHH.Sort();
LL2 = (double)aLL[0];
HL2 = (double)aHL[aHL.Count - 1];
    LH2 = (double)aLH[0];
    HH2 = (double)aHH[aHH.Count - 1];

```

Source Code 4.4 Source code dekomposisi pada citra

Untuk `imageDWT` mengambil nilai dari `class YCrCb`. *Sourcecode* `class YCrCb` ditunjukkan pada *sourcecode* 4.3.

4.2.3 Discrete Wavelet Transform (DWT)

Dalam Proses pembuatan dan proses pengecekan citra *watermark* dibutuhkan perhitungan *discrete wavelet transform* dimana dalam proses pembuatan dan pengecekan tersebut bekerja dalam domain frekuensi. Prosedur penghitungan proses *discrete wavelet transform* ditunjukkan pada *Source Code* 4.5.

```

//metod proses DWT
public void DWT()
{
    //isi log

parent.tLog.Items.Add(DateTime.Now.ToString()+" :
Initializing DWT Starting");
    imageDWT = new YCrCb[Gambar.Height, Gambar.Width];
    double Yv, Cr, Cb;
    for (int y = 0; y < Gambar.Height; ++y)
    {
        for (int x = 0; x < Gambar.Width; ++x)
        {
            Yv = Math.Pow(2, -y / 2) * imageYCbCr[y,
x].GetY() * Omega * (Math.Pow(2, -y) *
(imageYCbCr[y, x].GetY() - x));
            Cr = Math.Pow(2, -y / 2) * imageYCbCr[y,
x].GetCr() * Omega * (Math.Pow(2, -y) *
(imageYCbCr[y, x].GetCr() - x));
            Cb = Math.Pow(2, -y / 2) * imageYCbCr[y,
x].GetCb() * Omega * (Math.Pow(2, -y) *
(imageYCbCr[y, x].GetCb() - x));
            if (Yv==0 || Cr==0 || Cb==0)
            {
                Yv=0;
                Cr=0;
                Cb=0;
            }
            //init pixel dwt baru
            imageDWT[y, x] = new YCrCb(Yv,Cr,Cb);
        }
    }

parent.tLog.Items.Add(DateTime.Now.ToString() + " :
Initializing DWT Complete");
}

```

Sourcecode 4.5 Source code perhitungan DWT

Untuk imageDWT mengambil nilai dari class YCrCb. Sourcecode class YCrCb ditunjukkan pada sourcecode 4.3.

4.2.4 Inverse Discrete Wavelet Transform (IDWT)

Teknik *Inverse Discrete Wavelet Transform* digunakan untuk mengembalikan nilai komponen DWT ke dalam domain koordinat. Transformasi IDWT ditunjukkan pada *sourcecode* 4.6.

```
private void IDWT()
{
parent.tLog.Items.Add(DateTime.Now.ToString() + " :
Starting IDWT");
double Yv=0, Cr=0, Cb=0;
for (int y = 0; y < Gambar.Height; ++y)
{
for (int x = 0; x < Gambar.Width; ++x)
{
//Console.WriteLine("=== {0} {1} {2}",
imageDWT[y, x].GetY(), imageDWT[y, x].GetCb(),
imageDWT[y, x].GetCr());
if (imageDWT[y, x].GetY() == 0 || imageDWT[y,
x].GetCr() == 0 || imageDWT[y, x].GetCb() == 0)
{
Yv = imageYCbCr[y, x].GetY();
Cr = imageYCbCr[y, x].GetCr();
Cb = imageYCbCr[y, x].GetCb();
}
else
{
Yv = imageDWT[y, x].GetY() / (Omega *(Math.Pow(2,
-y) * (imageYCbCr[y, x].GetY() - x)) * Math.Pow(2,
-y/ 2));
Cr = imageDWT[y, x].GetCr() / (Omega *(Math.Pow(2,
-y) * (imageYCbCr[y, x].GetCr() - x)) * Math.Pow(2,
-y / 2));
Cb = imageDWT[y, x].GetCb() / (Omega * (Math.Pow(2,
-y) * (imageYCbCr[y, x].GetCb() - x)) * Math.Pow(2,
-y / 2));
}
}
}
}
```

```
// Console.WriteLine(">>> {0} ++++++{1} ",
imageDWT[y, x].GetY(), Yv);

    imageDWT[y, x] = new YCrCb(Yv, Cr, Cb);
}
}

parent.tLog.Items.Add(DateTime.Now.ToString() + " :
IDWT Complete");
}
```

Source Code 4.6 Source code perhitungan IDWT

Untuk imageDWT mengambil nilai dari class YCrCb. Sourcecode class YCrCb ditunjukkan pada sourcecode 4.3. Untuk imageYCbCr mengambil nilai dari class YCrCb yang ditunjukkan pada sourcecode 4.3.

4.2.5 Perhitungan MSE dan PSNR

Perhitungan nilai kualitas citra digital yang merupakan hasil watermarking, terhadap citra digital yang asli, dapat dilakukan dengan menggunakan nilai MSE (Mean Square Error) dan juga nilai PSNR (Peak Signal to Noise Ratio). Perhitungan nilai MSE dari citra digital berukuran NxM, dilakukan sesuai persamaan 2.15. Sedangkan untuk nilai PSNR didapat dengan menghitung nilai algoritma dari hasil pembagian kuadrat nilai bit maksimal yang dimiliki suatu citra dengan nilai MSE. Prosedur penghitungan MSE dan PSNR ditunjukkan oleh source code 4.7.

```
for (int x = 0; x < ori.Length; x++)
{
    temp += Math.Pow(ori[x] - eks[x], 2);
}

MSE = temp / ori.Length;
if (MSE != 0)
{
    PSNR = 10 * Math.Log10(Math.Pow(255, 0) / MSE);
}
else
{
    PSNR = 0;
}
```


Source Code 4.7 Source code perhitungan MSE dan PSNR

4.2.6 Penyimpanan Citra yang Disisipi Watermark

Citra yang telah disisipi *watermark* akan disimpan dalam format .bmp. Peosedur Penyimpanan citra yang disisipi *watermark* ditunjukkan pada *source code* 4.8.

```
private void BtnSaveE_Click(object sender, EventArgs e)
{
    SaveFileDialog SFDialog = new SaveFileDialog();
    SFDialog.Title = "Save Image Result";
    SFDialog.Filter = "Bitmap File (*.Bmp) | *.bmp";
    if (SFDialog.ShowDialog() == DialogResult.OK)
    {
        PicRes.Image.Save(SFDialog.FileName);
        FileInfo fi = new FileInfo(SFDialog.FileName);
        DirectoryInfo di = fi.Directory;

        FileInfo fix = new
        FileInfo(Application.ExecutablePath);
        DirectoryInfo dix = fix.Directory;
        string nama2 = dix.FullName + "\\\" +
        (fi.Name).Substring(0, fi.Name.IndexOf('.')) +
        ".dat";
        fi = new FileInfo(nama2);
        if (fi.Exists)
            fi.Delete();

        string nama = dix.FullName + "\\temp.dat";
        fi = new FileInfo(nama);
        fi.CopyTo(nama2);
        fi.Delete();

    }
}
```

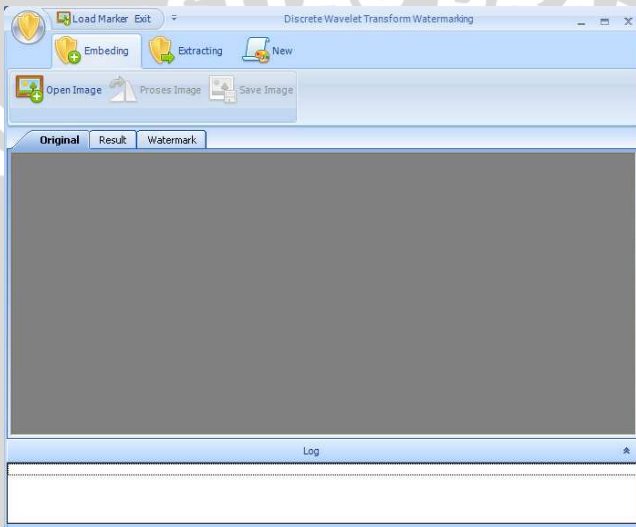
Source Code 4.8 Source code penyimpanan citra yang disisipi *watermark*

4.3 Implementasi Antar Muka

Berdasarkan rancangan antarmuka pada bab sebelumnya maka antarmuka dibuat sebagai berikut :

4.3.1 Tampilan Utama Aplikasi *Watermarking* Citra

Tampilan utama implementasi program *Watermarking* Citra ditunjukkan pada Gambar 4.1.

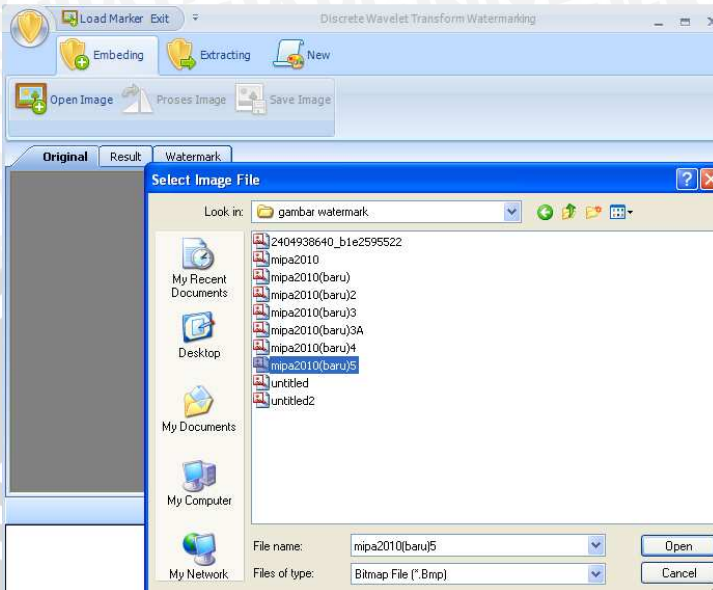


Gambar 4.1 Tampilan utama

Pada tampilan utama diatas terdapat terdapat 2 *Button* untuk proses *Embedding* citra dan proses *Extracting* citra. Selain *Button* untuk proses *Embedding* citra dan proses *Extracting* citra, pada tampilan program terdapat *Button* untuk proses *Load Marker*, *New*, *Open Image*, *Proses Image*, *Save Image*, dan *Tab* untuk *Orginal*, *Result*, dan *Watermark*.

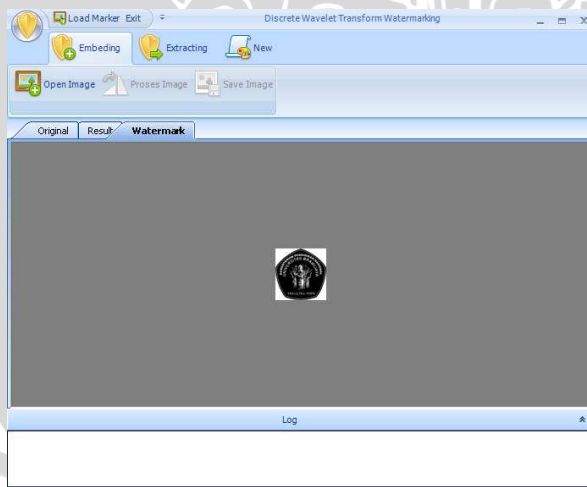
4.3.2 Tampilan Proses Penyisipan *Watermark*

Untuk proses penyisipan *watermark* langkah pertama yang harus dilakukan adalah membuka *file* citra *watermark* yang akan disisipkan ke citra asli dengan menekan *button Load Marker*. Proses membuka *file* citra *watermark* ini ditunjukkan pada gambar 4.2.



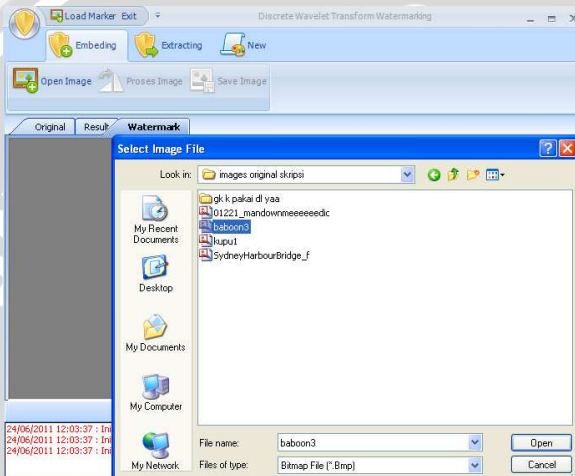
Gambar 4.2 Tampilan *open image* citra watermark

Setelah proses membuka *file* maka citra yang akan disisipkan kedalam citra asli akan tampak setelah kita memilih *Tab watermark*. Untuk tampilan citra yang akan disisipkan ditunjukkan pada gambar 4.3.



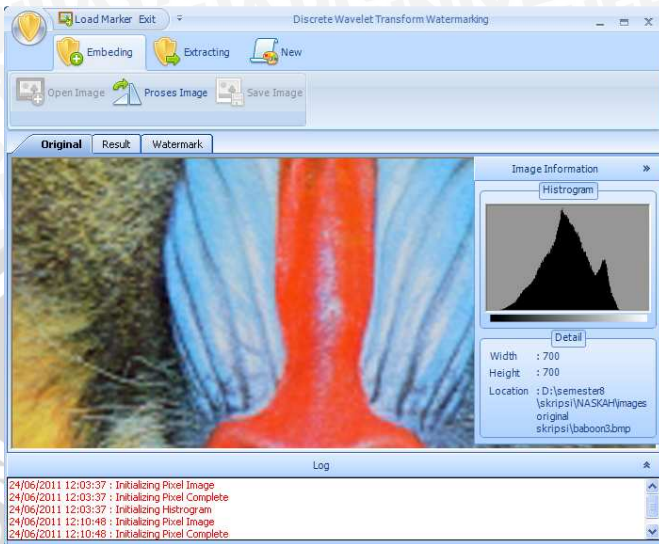
Gambar 4.3 Tampilan Tanda *Watermark*

Selanjutnya yang dilakukan adalah membuka *file* citra yang akan disisipkan *watermark*. Proses membuka *file* citra ini ditunjukkan pada gambar 4.4.



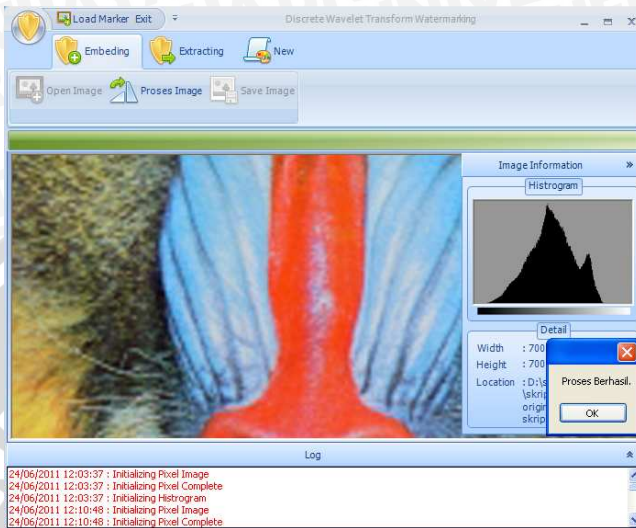
Gambar 4.4 Tampilan *open image*

Setelah proses *open file* maka citra yang akan diwatermark akan tampak setelah kita memilih *Tab* original. Untuk tampilan citra asli dan informasi mengenai citra tersebut ditunjukkan pada gambar 4.5.

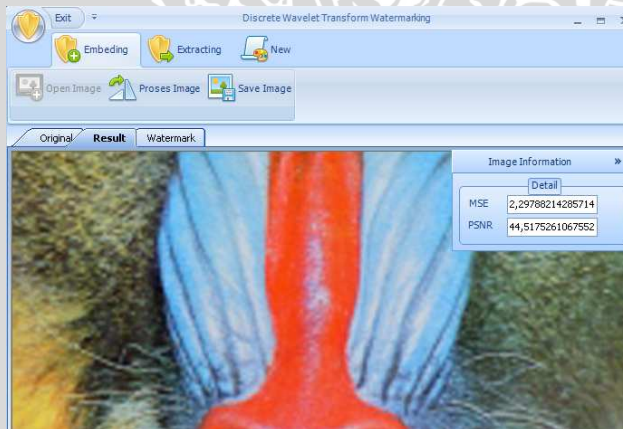


Gambar 4.5 Tampilan *image* original

Setelah *image* original yang akan diwatermark terlihat maka untuk melakukan proses penyisipan *watermark*, dilakukan dengan menekan *Button* proses *image*. Setelah proses selesai akan timbul pesan “proses berhasil” seperti terlihat pada gambar 4.5. Tampilan citra yang telah disisipi *watermark* ditunjukkan pada gambar 4.6 dan 4.7.



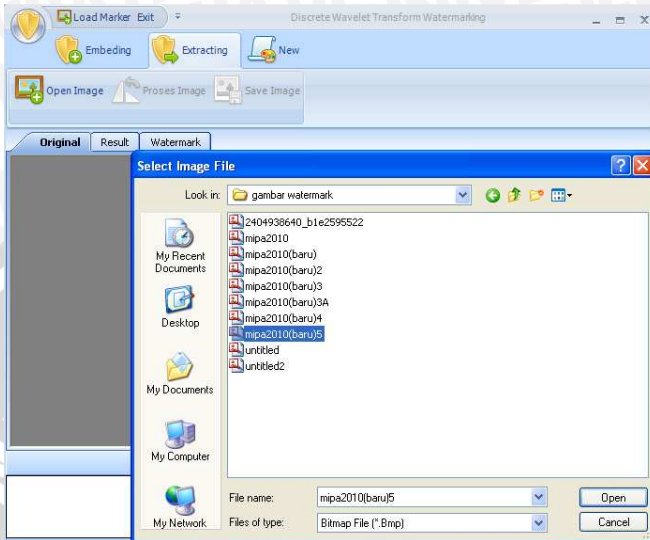
Gambar 4.6 Tampilan proses penyisipan citra



Gambar 4.7 Tampilan citra yang telah disisipi watermark

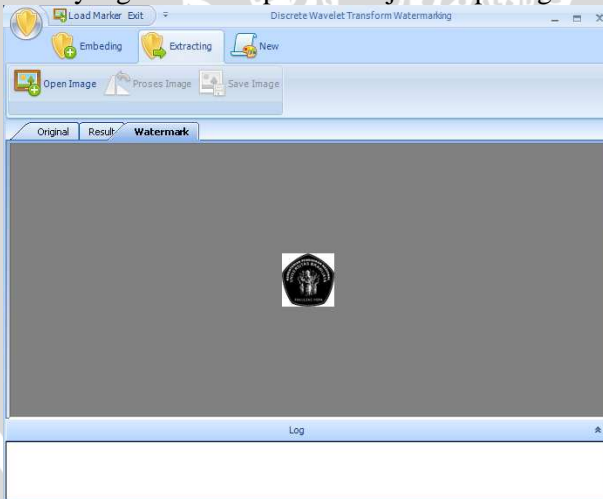
4.3.3 Tampilan Proses Ekstraksi Watermark

Untuk proses ekstraksi watermark langkah pertama yang harus dilakukan adalah membuka file citra tanda watermark yang telah disisipkan ke dalam citra asli dengan menekan button *Load Marker*. Proses membuka file citra watermark ini ditunjukkan pada gambar 4.8.



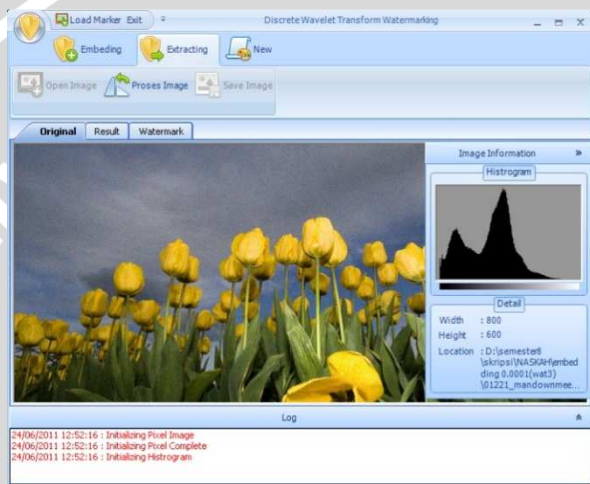
Gambar 4.8 Tampilan *open image* citra watermark

Setelah proses membuka *file* maka citra yang akan disisipkan ke dalam citra asli akan tampak setelah kita memilih *Tab watermark*. Untuk tampilan citra yang akan disisipkan ditunjukkan pada gambar 4.9.

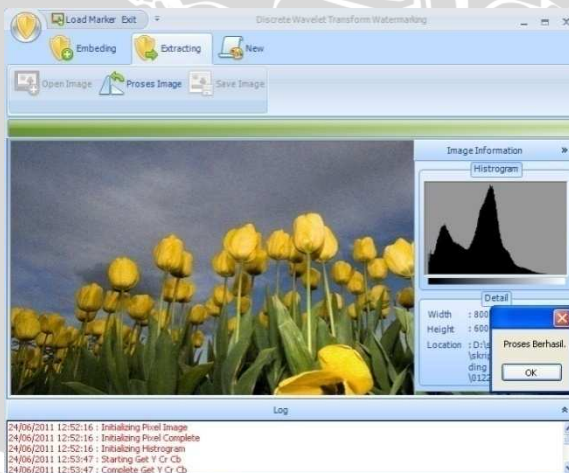


Gambar 4.9 Tampilan Tanda *Watermark*

Langkah selanjutnya yang dilakukan adalah membuka *file* citra yang telah terdapat *watermark* dan dilakukan proses ekstraksi *watermark* untuk mendapatkan tanda *watermark* yang terdapat pada citra tersebut. Tampilan proses ekstraksi *watermark* dapat dilihat pada gambar 4.10 – 4.11.

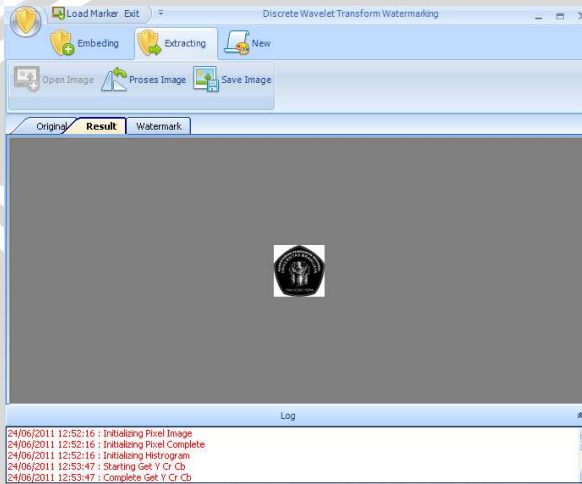


Gambar 4.10 Tampilan citra yang disisipi *watermark*



Gambar 4.11 Tampilan proses ekstraksi

Pada gambar 4.11, terlihat proses ekstraksi *watermark* berhasil dilakukan. Hal ini ditunjukkan dengan adanya pesan “proses berhasil”. Untuk hasil tanda *watermark* yang disisipkan, dapat dilihat pada *Tab Result* seperti yang terlihat pada gambar 4.12.



Gambar 4.12 Tampilan hasil proses ekstraksi

4.4 Implementasi Uji Coba

Pada subbab ini akan dilakukan pembahasan mengenai pengujian yang telah dilakukan pada sistem dan hasil evaluasi dari hasil yang dikeluarkan sistem.

4.4.1 Evaluasi Kualitas Citra

Kualitas citra dihitung dengan MSE, untuk melihat pengaruh pemberian *watermark*. Penyisipan *watermark* ini dilakukan terhadap 4 citra yang berbeda yang masing – masing memiliki ukuran antara 800xS600, 700x700, 800x600, dan 800x600, dan pixel dalam format BMP 24 bit. Setiap citra ditandai *watermark* dengan menggunakan alpha (α) = 0,0001 ; 0,0003 ; 0,0006.

Pada tugas akhir ini juga digunakan tanda *watermark* mipa2010.bmp dengan ukuran pixel yang digunakan adalah 20x20, 40x40 dan 60x60.

Data Penyisipan *watermark* yang telah dilakukan pada 4 citra yang berbeda untuk alpha (α) = 0,0001 ditunjukkan pada tabel 4.1.

Tabel 4.1. Tabel hasil penyisipan *watermark* untuk alpha (α) = 0,0001

Citra asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR
01221_m andownm eeeeedic .bmp	01221_mando wnmeeeeeedic (wat1).bmp	mipa2010.b mp(20x20)	$3,15 \times 10^{-4}$	83,1462
baboon3. bmp	baboon3(wat1) .bmp	mipa2010.b mp(20x20)	$7,28 \times 10^{-4}$	79,5091
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 1).bmp	mipa2010.b mp(20x20)	$4,67 \times 10^{-4}$	81,4407
kupu1.b mp	kupu1(wat1).b mp	mipa2010.b mp(20x20)	$1,63 \times 10^{-4}$	85,9945
01221_m andownm eeeeedic .bmp	01221_mando wnmeeeeeedic (wat2).bmp	mipa2010.b mp(40x40)	$5,92 \times 10^{-2}$	60,4059
baboon3. bmp	baboon3(wat2) .bmp	mipa2010.b mp(40x40)	$4,83 \times 10^{-2}$	61,2889
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 2).bmp	mipa2010.b mp(40x40)	$1,08 \times 10^{-1}$	57,7786
kupu1.b mp	kupu1(wat2).b mp	mipa2010.b mp(40x40)	$7,59 \times 10^{-2}$	59,3303
01221_m andownm eeeeedic .bmp	01221_mando wnmeeeeeedic (wat3).bmp	mipa2010.b mp(60x60)	2,561	44,0459
baboon3. bmp	baboon3(wat3) .bmp	mipa2010.b mp(60x60)	2,298	44,5175
SydneyH	SydneyHarbou	mipa2010.b	2,391	44,3441

arbourBri dge_f.bmp p	rBridge_f(wat 3).bmp	mp(60x60)		
kupu1.b mp	kupu1(wat3).b mp	mipa2010.b mp(60x60)	5,402	40,8051

Data Penyisipan *watermark* yang telah dilakukan pada 25 citra yang berbeda untuk alpha (α) = 0,0003 ditunjukkan pada tabel 4.2.

Tabel 4.2. Tabel hasil penyisipan *watermark* untuk alpha (α) = 0,0003

Citra asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR
01221_m andownm eeeeedic .bmp	01221_mando wnmeeeeeedic (wat1).bmp	mipa2010.b mp(20x20)	$2,85 \times 10^{-3}$	73,5792
baboon3. bmp	baboon3(wat1) .bmp	mipa2010.b mp(20x20)	$6,48 \times 10^{-3}$	70,0116
SydneyH arbourBri dge_f.bmp p	SydneyHarbou rBridge_f(wat 1).bmp	mipa2010.b mp(20x20)	$4,26 \times 10^{-2}$	61,8375
kupu1.b mp	kupu1(wat1).b mp	mipa2010.b mp(20x20)	$1,38 \times 10^{-3}$	76,7346
01221_m andownm eeeeedic .bmp	01221_mando wnmeeeeeedic (wat2).bmp	mipa2010.b mp(40x40)	$1,31 \times 10^{-1}$	56,9595
baboon3. bmp	baboon3(wat2) .bmp	mipa2010.b mp(40x40)	$1,29 \times 10^{-1}$	57,0378
SydneyH arbourBri dge_f.bmp p	SydneyHarbou rBridge_f(wat 2).bmp	mipa2010.b mp(40x40)	$2,52 \times 10^{-1}$	54,1239
kupu1.b mp	kupu1(wat2).b mp	mipa2010.b mp(40x40)	$2,97 \times 10^{-1}$	53,3973
01221_m	01221_mando	mipa2010.b	4,683	41,4254

andownm eeeeeeedic .bmp	wnmeeeeeedic (wat3).bmp	mp(60x60)		
baboon3. bmp	baboon3(wat3) .bmp	mipa2010.b mp(60x60)	4,917	41,2136
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 3).bmp	mipa2010.b mp(60x60)	4,430	41,6666
kupu1.b mp	kupu1(wat3).b mp	mipa2010.b mp(60x60)	9,846	38,1981

Data Penyisipan *watermark* yang telah dilakukan pada 25 citra yang berbeda untuk alpha (α) = 0,0006 ditunjukkan pada tabel 4.3

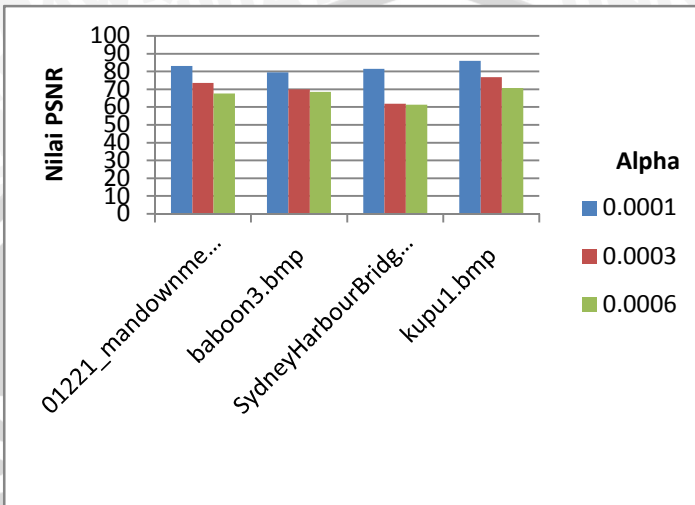
Tabel 4.3. Tabel hasil penyisipan *watermark* untuk alpha (α) = 0,0006

Citra asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR
01221_m andownm eeeeeeedic .bmp	01221_mando wnmeeeeeedic (wat1).bmp	mipa2010.b mp(20x20)	$1,12 \times 10^{-2}$	67,6372
baboon3. bmp	baboon3(wat1) .bmp	mipa2010.b mp(20x20)	$9,26 \times 10^{-3}$	68,4649
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 1).bmp	mipa2010.b mp(20x20)	$4,87 \times 10^{-2}$	61,2472
kupu1.b mp	kupu1(wat1).b mp	mipa2010.b mp(20x20)	$5,62 \times 10^{-3}$	70,6272
01221_m andownm eeeeeeedic .bmp	01221_mando wnmeeeeeedic (wat2).bmp	mipa2010.b mp(40x40)	$2,43 \times 10^{-1}$	54,2601
baboon3. bmp	baboon3(wat2) .bmp	mipa2010.b mp(40x40)	$4,61 \times 10^{-1}$	51,4907
SydneyH	SydneyHarbou	mipa2010.b	$4,91 \times 10^{-1}$	51,2119

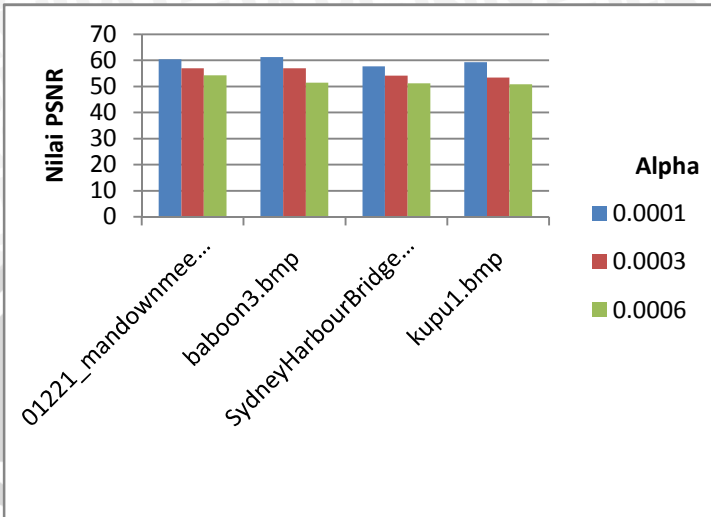
arbourBridge_f.bmp	rBridge_f(wat2).bmp	mp(40x40)		
kupu1.bmp	kupu1(wat2).bmp	mipa2010.bmp(40x40)	$5,31 \times 10^{-1}$	50,8739
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat3).bmp	mipa2010.bmp(60x60)	5,901	40,4217
baboon3.bmp	baboon3(wat3).bmp	mipa2010.bmp(60x60)	8,866	38,6534
SydneyHarbourBridge_f.bmp	SydneyHarbourBridge_f(wat3).bmp	mipa2010.bmp(60x60)	6,198	40,2079
Kupu1	kupu1(wat3).bmp	mipa2010.bmp(60x60)	14,125	36,6309

Nilai PSNR diperoleh dengan menghitung nilai MSE terlebih dahulu. Pada hasil penyisipan *watermark* diatas terlihat bahwa nilai MSE berbanding terbalik dengan nilai PSNR sehingga semakin kecil nilai MSE yang dihasilkan dan semakin besar nilai PSNR-nya maka semakin baik kualitas citra yang disisipi *watermark* yang dihasilkan.

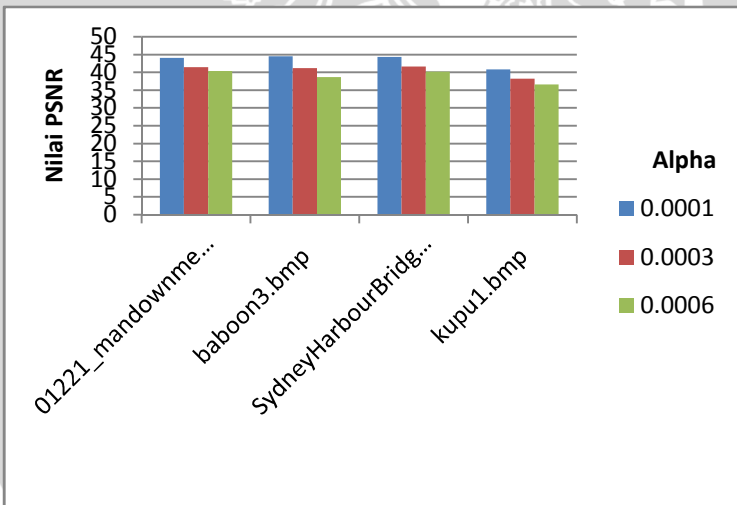
Dari hasil penyisipan *watermark* pada tabel 4.1 dapat dilihat bahwa dengan menggunakan alpha yang kecil yaitu 0.0001 maka MSE yang dihasilkan akan kecil pula dan memiliki nilai PSNR yang besar. Sedangkan semakin alpha ditingkatkan menjadi 0.0003 dan 0.0006 maka MSE yang dihasilkan juga semakin besar dan memiliki nilai PSNR yang semakin kecil. Hal ini ditunjukkan dengan rata – rata MSE untuk alpha 0.0001 sebesar 1,0789, untuk rata – rata MSE dengan nilai alpha 0.0003 dan 0.0006 adalah sebesar 2,0506 dan 3,0745. Untuk rata – rata PSNR dengan nilai alpha 0.0001 sebesar 61,8839 dan untuk rata – rata PSNR dengan nilai alpha 0.0003 dan 0.0006 adalah 55,5154 dan 52,6439. Sehingga dapat disimpulkan bahwa nilai MSE dan PSNR yang baik dihasilkan oleh citra yang disisipi *watermark* dengan nilai alpha yang kecil. Grafik pengaruh alpha untuk nilai PSNR pada masing – masing ukuran tanda *watermark* ditunjukkan pada gambar 4.13 – 4.15.



Gambar 4.13 Grafik pengaruh alpha terhadap nilai PSNR untuk ukuran tanda watermark 20x20 pixel



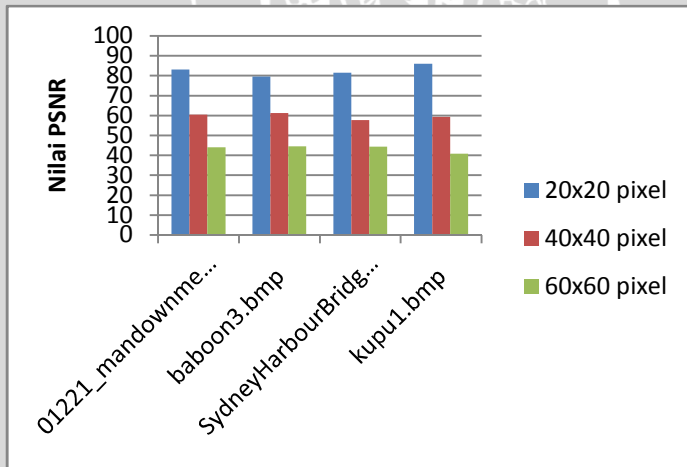
Gambar 4.14 Grafik pengaruh alpha terhadap nilai PSNR untuk ukuran tanda watermark 40x40 pixel



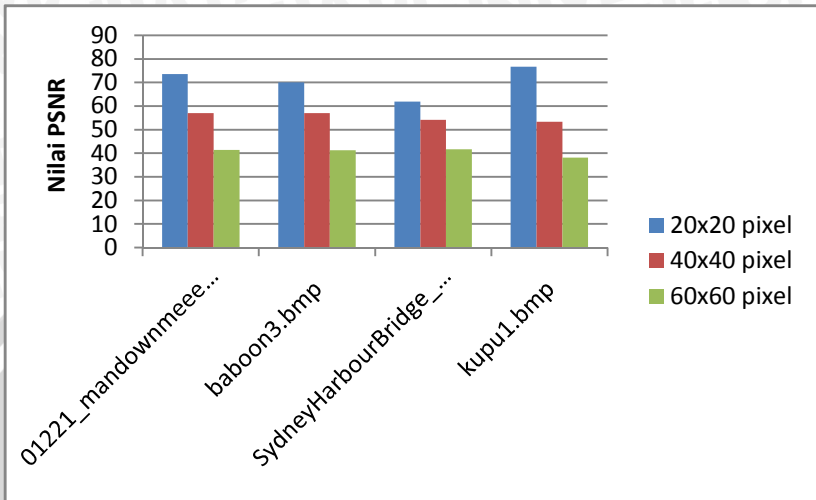
Gambar 4.15 Grafik pengaruh alpha terhadap nilai PSNR untuk ukuran tanda watermark 60x60 pixel

Pada gambar grafik 4.16 sampai dengan 4.18 dapat dilihat bahwa nilai PSNR yang dihasilkan akan semakin menurun ketika nilai alpha dinaikkan untuk semua ukuran tanda *watermark* yang digunakan. Nilai PSNR tertinggi dihasilkan oleh citra yang disisipi *watermark* dengan nilai alpha 0.0001 dan ukuran *watermark* yang digunakan adalah 20x20 pixel yaitu pada gambar kupu1(wat1).bmp dengan nilai MSE $1,63 \times 10^{-4}$ dan nilai PSNR 85,99452.

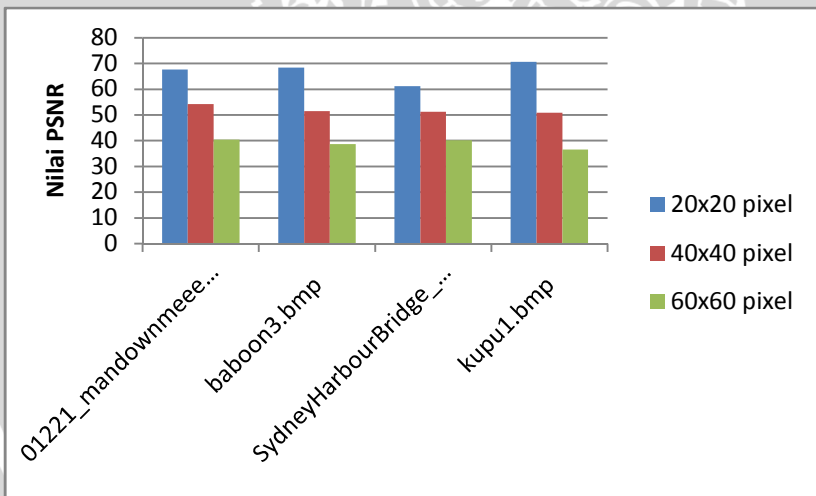
Selain pengaruh alpha terhadap nilai PSNR dan MSE, dilakukan juga uji coba terhadap pengaruh ukuran tanda *watermark* terhadap nilai MSE dan PSNR. Dari tabel hasil penyisipan *watermark* menunjukkan bahwa dengan pemberian ukuran tanda *watermark* 20x20 pixel untuk semua nilai alpha, nilai MSE yang dihasilkan kecil dan nilai PSNR yang dihasilkan besar. Nilai MSE akan semakin besar dan PSNR yang semakin kecil ketika ukuran tanda *watermark* dibesarkan menjadi 40x40 pixel dan 60x60 pixel. Untuk pengaruh ukuran tanda *watermark* terhadap nilai PSNR yang dihasilkan ditunjukkan pada gambar grafik 4.13 – 4.15.



Gambar 4.16 Grafik tanda *watermark* terhadap nilai PSNR $\alpha = 0.0001$



Gambar 4.17 Grafik tanda *watermark* terhadap nilai PSNR $\alpha = 0.0003$



Gambar 4.18 Grafik tanda *watermark* terhadap nilai PSNR $\alpha = 0.0006$

Pada gambar grafik 4.16 sampai dengann 4.17 menunjukkan bahwa dengan penambahan ukuran tanda *watermark* maka nilai PSNR akan semakin menurun yang berarti kualitas citra yang dihasilkan juga akan semakin

menurun. Untuk perubahan nilai MSE, semakin ditingkatkan ukuran tanda *watermark* maka nilai MSE yang dihasilkan akan semakin tinggi. Hasil untuk nilai MSE dapat dilihat pada tabel 4.1 – 4.3.

4.4.2 Evaluasi Hasil Citra yang Disisipi *Watermark*

Dalam bab 2.3.1 sebuah citra *watermark* yang disisipkan harus memiliki persyaratan yang antara lain ketahanan (*robustness*) dan tidak terlihat (*Nonperceptibility*). Untuk itu, perlu dilakukan suatu evaluasi hasil terhadap hasil citra yang disisipi *watermark* .

4.4.2.1 Hasil Evaluasi Tanpa Serangan

Uji coba pertama yang dilakukan terhadap 4 citra ter – *watermark* dengan menggunakan ukuran citra *watermark* 20x20, 40x40, dan 60x60 pixel dan menggunakan α (0.0001 ; 0.0003 ; 0.0006) adalah tanpa serangan pemrosesan citra. Dimana citra asli yang telah di beri tanda *watermark* akan langsung dilakukan proses *extracting watermark* untuk mengetahui apakah tanda *watermark* yang telah disisipkan dapat terdeteksi atau tidak.

Hasil pengujian tanpa serangan yang telah dilakukan menunjukkan bahwa semua citra yang telah disisipi *watermark* untuk semua nilai alpha dan untuk semua ukuran tanda *watermark* mampu terdeteksi dengan baik .

4.4.2.2 Evaluasi Hasil Citra yang Disisipi *Watermark* dari Sisi *Robustness* (Ketahanan)

Evaluasi ketahanan *watermark* ini dilakukan dengan memberikan beberapa serangan terhadap citra *watermark*. Serangan tersebut antara lain *cropping*, *noising*, *resizing*, dan *sharpening*.

4.4.2.1.1 *Cropping*

Percobaan dengan perlakuan pemotongan citra atau *cropping* dilakukan dengan menghilangkan bagian citra ter – *watermark* sebesar 25 %, dan 40% dari ukuran semula. Hasil pengujian serangan citra yang disisipi *watermark* dengan *cropping* ditunjukkan dengan tabel 4.4.

Tabel 4.4 Tabel hasil pendeteksian pada citra yang sudah disisipi *watermark* dengan serangan *cropping*

Citra yang disisipi <i>Watermark</i>		<i>Cropping</i>	
		25%	40%
01221_mandownmeeeeee dic(wat1)			
Alpha	0.0001	√	√
	0.0003	√	√
	0.0006	√	√

Citra yang disisipi <i>Watermark</i>		<i>Cropping</i>	
		25%	40%
01221_mandownmeeeeee dic(wat2)			
Alpha	0.0001	√	√
	0.0003	√	√
	0.0006	√	√

Citra yang disisipi <i>Watermark</i>		<i>Cropping</i>	
		25%	40%
01221_mandownmeeeeee dic(wat3)			
Alpha	0.0001	√	√
	0.0003	√	√
	0.0006	√	√

Ket : √ = *watermark* terdeteksi
 - = *watermark* tidak terdeteksi

Pada tabel 4.4 menunjukkan bahwa ujicoba yang dilakukan terhadap citra yang disisipi *watermark* setelah diberikan berbagai serangan *cropping* untuk penggunaan semua nilai alpha baik yang bernilai kecil (0.0001) ataupun yang bernilai besar (0.0006) masih dapat dideteksi

watermark yang disisipkan didalamnya meskipun mengalami pemotongan hingga sebesar 40% dari ukuran semula.

4.4.2.1.2 Penambahan Noise

Pemberian serangan penambahan noise dilakukan dengan menambahkan sejumlah *pixel* pengganggu ke dalam citra yang sudah disisipi *watermark* sehingga citra terlihat sedikit rusak. *Noise* yang digunakan adalah dua jenis *noise* yang telah disediakan oleh aplikasi *Adobe Photoshop CS3* yaitu *Uniform Noise* dan *Gaussian Noise* dengan variance sebesar 5% dan 10%

Tabel 4.5 Tabel hasil pendeteksian pada citra yang sudah disisipi *watermark* dengan serangan penambahan noise

Citra yang disisipi <i>Watermark</i>		Uniform Noise		Gaussian Noise	
		5%	10%	5%	10%
01221_mandownmeeeeeee dic(wat1)					
Alpha	0.0001	√	√	√	√
	0.0003	√	√	√	√
	0.0006	√	√	√	√

Citra yang disisipi <i>Watermark</i>		Uniform Noise		Gaussian Noise	
		5%	10%	5%	10%
01221_mandownmeeeeeee dic(wat2)					
Alpha	0.0001	√	√	√	√
	0.0003	√	√	√	√
	0.0006	√	√	√	√

Citra yang disisipi <i>Watermark</i>		Uniform Noise		Gaussian Noise	
		5%	10%	5%	10%
01221_mandownmeeeeee dic(wat3)					
Alpha	0.0001	√	√	√	√
	0.0003	√	√	√	√
	0.0006	√	√	√	√

Ket : √ = *watermark* terdeteksi
 - = *watermark* tidak terdeteksi

Percobaan pertama memanfaatkan *Uniform Noise*. Pada tabel 4.5 menunjukkan bahwa ketika citra yang disisipi *watermark* diberikan serangan *Uniform Noise* untuk semua nilai alpha masih dapat dideteksi *watermark* yang disisipkan didalamnya begitu juga dengan percobaan kedua untuk *Gaussian Noise*, tanda *watermark* yang disisipkan masih dapat dideteksi dengan baik untuk semua nilai alpha .

4.4.2.1.3 *Sharpening*

Uji serangan kali ini dilakukan dengan memperjelas tepi pada objek di dalam citra. Citra yang sudah disisipi *watermark* diberi dua jenis serangan penajaman citra (*sharpening*) yang disediakan oleh aplikasi *Adobe Photoshop CS3* yaitu *sharpen* dan *sharpen edge*.

Tabel 4.6. Tabel hasil pendeteksian pada citra yang sudah disisipi *watermark* dengan serangan penajaman citra

Citra yang disisipi <i>watermark</i>		Tipe	
		Sharpen	Sharpen Edge
01221_mandownmeeeeee dic(wat1)			
alpha	0.0001	√	√
	0.0003	√	√
	0.0006	√	√

Citra yang disisipi <i>watermark</i>		Type	
		Sharpen	Sharpen Edge
01221_mandownmeeeeee dic(wat2)			
Alpha	0.0001	√	√
	0.0003	√	√
	0.0006	√	√

Citra yang disisipi <i>watermark</i>		Type	
		Sharpen	Sharpen Edge
01221_mandownmeeeeee dic(wat3)			
Alpha	0.0001	√	√
	0.0003	√	√
	0.0006	√	√

Ket : √ = *watermark* terdeteksi
 - = *watermark* tidak terdeteksi

Pada tabel 4.6 menunjukkan ketika citra yang disisipi *watermark* diberikan serangan dengan teknik sharpen edge dan sharpen untuk semua nilai alpha, *watermark* yang ditanamkan ke dalam citra mampu dideteksi dengan baik. Hal ini menunjukkan bahwa metode yang digunakan pada sistem mampu bertahan dari usaha perusakan citra dengan menajamkan nilai *pixel – pixel* citra.

4.4.2.1.4 *Resizing*

Percobaan ini dilakukan dengan mengubah ukuran citra. Dalam hal ini, percobaan yang dilakukan adalah melakukan pengecilan citra *watermark*. Ukuran citra diubah menjadi 25%, 50%, 75%, dari ukuran citra semula. Saat akan dideteksi, ukuran citra ter – *watermark* yang telah dikecilkan dan dikembalikan terlebih dahulu ke ukuran semula. Untuk hasil pengujian dengan serangan *resizing* dapat dilihat pada tabel 4.7.

Tabel 4.7 Tabel hasil pendeteksian pada citra disisipi *watermark* dengan serangan *resizing*

Citra yang disisipi <i>watermark</i>		Skala Pengecilan		
		25%	50%	75%
01221_mandownmee eeedic(wat1)				
Alpha	0.0001	√	√	√
	0.0003	√	√	√
	0.0006	√	√	√

Citra yang disisipi <i>watermark</i>		Skala Pengecilan		
		25%	50%	75%
01221_mandownmee eeedic(wat2)				
Alpha	0.0001	√	√	√
	0.0003	√	√	√
	0.0006	√	√	√

Citra yang disisipi <i>watermark</i>		Skala Pengecilan		
		25%	50%	75%
01221_mandownmee eeedic(wat3)				
alpha	0.0001	√	√	√
	0.0003	√	√	√
	0.0006	√	√	√

Ket : √ = *watermark* terdeteksi
 - = *watermark* tidak terdeteksi

Pada tabel 4.4 menunjukkan bahwa ujicoba yang dilakukan terhadap citra yang disisipi *watermark* setelah diberikan berbagai serangan *resizing* untuk penggunaan semua nilai alpha baik yang bernilai kecil (0.0001) ataupun yang bernilai besar (0.0006) masih dapat dideteksi *watermark* yang disisipkan di dalamnya meskipun mengalami perubahan ukuran hingga sebesar 75% dari ukuran citra semula.

4.4.2.1.5 Rotation

Percobaan ini dilakukan dengan melakukan rotasi terhadap citra ter- *watermark* sebesar $20^{\circ}CW, 50^{\circ}CW, 20^{\circ}CCW$, dan $50^{\circ}CCW$. Untuk hasil pengujian dengan serangan *rotation* dapat dilihat pada tabel 4.8.

Tabel 4.8 Tabel hasil pendeteksian pada citra disisipi *watermark* dengan serangan *rotation*

Citra yang disisipi <i>watermark</i>		Rotasi			
		20° CW	50° CW	20° CCW	20° CCW
01221_mandownmeeee eedic(wat1)					
Alpha	0.0001	√	√	√	√
	0.0003	√	√	√	√
	0.0006	√	√	√	√

Citra yang disisipi <i>watermark</i>		Rotasi			
		20° CW	50° CW	20° CCW	20° CCW
01221_mandownmeeee eedic(wat2)					
Alpha	0.0001	√	√	√	√
	0.0003	√	√	√	√
	0.0006	√	√	√	√

Citra yang disisipi <i>watermark</i>		Rotasi			
		20° CW	50° CW	20° CCW	20° CCW
01221_mandownmeeee eedic(wat3)					
Alpha	0.0001	√	√	√	√
	0.0003	√	√	√	√
	0.0006	√	√	√	√

Ket : √ = *watermark* terdeteksi

- = *watermark* tidak terdeteksi

Pada tabel 4.5 menunjukkan bahwa ujicoba yang dilakukan terhadap citra yang disisipi *watermark* setelah diberikan berbagai serangan *rotation* untuk penggunaan semua nilai alpha baik yang bernilai kecil (0.0001) ataupun yang bernilai besar (0.0006) masih dapat dideteksi *watermark* yang disisipkan didalamnya meskipun mengalami rotasi hingga 50⁰ baik CW maupun CCW.

4.4.2.1.6 Evaluasi Hasil Citra yang disisipi *watermark* Tidak Terlihat (*Nonperceptibility*).

Salah satu persyaratan yang harus dimiliki oleh citra *watermark* adalah (*Nonperceptibility*) dimana citra asli dan citra yang telah disisipi oleh *watermark* tidak bisa dibedakan oleh mata manusia normal. Untuk evaluasi *Nonperceptibility* ini, dilakukan tes terhadap 10 orang dari mulai usia 20 – 30 tahun. Hasil tes terhadap citra yang disisipi *watermark* menunjukkan bahwa semua responden melihat tidak adanya perbedaan antara citra asli dan citra pembawa *watermark* untuk semua alpha pada tanda *watermark* dengan ukuran 20x20 dan 40x40 pixel. Untuk citra yang disisipi *watermark* dengan tanda *watermark* 60x60 pixel untuk semua nilai alpha, responden melihat ada sedikit goresan pada bagian atas citra. Tabel hasil citra yang disisipi *watermark* tidak terlihat (*Nonperceptibility*) ditunjukkan pada tabel 4.9.

Tabel 4.9. Tabel hasil citra yang disisipi *watermark* tidak terlihat (*Nonperceptibility*) untuk alpha (α) = 0,0001

Nama	Citra	Hasil
Roswati	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-

	SydneyHarbourBridge_f(wat3)	-
Yunika	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Septianingsih	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ruri	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-

Anisa Wima	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yuli	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anung	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Dani	01221_mandownmeeeeedic(wat1)	√

	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ainun	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ida	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-

Ket : √ = Tidak terlihat perbedaan antara citra asli dan citra

- yang telah disisipi *watermark*
- = Terlihat perbedaan antara citra asli dan citra yang disisipi *watermark*

4.5 Analisa Hasil

Pada tugas akhir ini, pengujian dilakukan terhadap 4 citra berwarna yang berbeda dengan memberikan alpha sebesar 0.0001, 0.0003 dan 0.0006. Kualitas citra dihitung dengan menggunakan MSE dan PSNR dan memiliki nilai yang berbeda – beda. Nilai MSE dan PSNR untuk masing – masing citra yang disisipi *watermark* dapat dilihat pada subbab 4.4.1.

Dari hasil penyisipan tanda *watermark* keempat citra didapatkan hasil dengan menggunakan alpha yang kecil yaitu nilai alpha = 0.0001 akan dihasilkan nilai MSE yang kecil dan nilai PSNR yang besar . Sedangkan dengan penambahan alpha maka nilai MSE yang dihasilkan akan semakin besar dan nilai PSNR yang dihasilkan akan semakin kecil. Penambahan nilai MSE dan pengurangan nilai PSNR ditunjukkan dengan rata – rata nilai MSE untuk alpha 0.0001 sebesar 1,0789, untuk rata – rata MSE dengan nilai alpha 0.0003 dan 0.0006 adalah sebesar 2,0506 dan 3,0745. Untuk rata – rata PSNR dengan nilai alpha 0.0001 sebesar 61,8839 dan untuk rata – rata PSNR dnegan nilai alpha 0.0003 dan 0.0006 adalah 55,5154 dan 52,6439. Hal ini dapat diartikan bahwa untuk mendapatkan nilai error dari citra yang disisipi *watermark* yang kecil dan menghasilkan kualitas gambar yang yang bagus diperlukan penggunaan nilai alpha yang kecil. Hal ini terjadi karena nilai *alpha* digunakan saat penyisipan *watermark* ke dalam *pixel* citra. Semakin besar *alpha* maka nilai *watermark* yang disisipkan menjadi semakin besar sehingga akan memperbesar nilai *pixel* citra dari sebelumnya setelah ditambahkan. Nilai *pixel* citra yang terlalu besar akan mengurangi kualitas citra yang disisipi *watermark*.

Untuk pengujian penambahan ukuran tanda *watermark* mulai dari 20x20 *pixel* lalu ditingkatkan menjadi 40x40 dan yang terakhir 60x60 *pixel* hasil yang didapatkan adalah untuk citra *watermark* yang disisipi tanda *watermark* dengan ukuran 20x20 dan 40x40 *pixel* memiliki nilai MSE yang kecil dan nilai PSNR yang besar ini berarti kualitas citra *watermark* yang dihasilkan bagus. Untuk citra *watermark* yang disisipi tanda *watermark* dengan ukuran 60x60 *pixel* pada semua nilai alpha MSE yang dihasilkan cukup besar yaitu mulai dari 2 sampai dengan 14 dan nilai PSNR yang dihasilkan berkisar antara 44 kebawah yang berarti kualitas gambar *watermark* yang dihasilkan kurang baik dibandingkan dengan penyisipan

tanda *watermark* ukuran 20x20 dan 40x40 pixel. Hal ini disebabkan semakin besar ukuran tanda *watermark* yang digunakan maka semakin memperbesar nilai pixel citra yang akan disisipi *watermark* sehingga penyebaran nilai pixel merata sehingga akan mengurangi kualitas citra yang disisipi *watermark*.

Pada subbab 4.4.2 dilakukan percobaan evaluasi terhadap citra yang disisipi *watermark*. Evaluasi terhadap citra yang disisipi *watermark* yang dilakukan terdiri dari uji coba citra yang disisipi *watermark* tanpa serangan dimana hasil uji coba dapat dilihat pada subbab 4.4.2.1. Untuk pengujian terhadap citra yang disisipi *watermark* tanpa serangan, tanda *watermark* yang telah disisipkan dapat dideteksi dengan baik.

Pengujian selanjutnya dilakukan dengan menggunakan serangan terhadap citra yang telah disisipi *watermark*. Serangan yang diberikan meliputi penambahan *noise*, *cropping*, *resizing*, *rotation* dan *sharpen*. Untuk hasil pengujian kelima serangan terhadap citra ter – *watermark* dapat dilihat pada subbab 4.4.2.1 sampai dengan subbab 4.4.2.5. Dari hasil pengujian, didapatkan bahwa tanda *watermark* yang telah disisipkan ke dalam citra yang disisipi *watermark* mampu dideteksi dengan baik walaupun telah diberikan serangan. Dengan kata lain tanda *watermark* yang disisipkan pada domain frekuensi dengan metode *discrete wavelet transform* dapat tahan dari berbagai macam serangan. Hal ini dikarenakan penggunaan nilai alpha sebagai penguat *watermark* dan pembuatan citra *watermark* dalam domain frekuensi tertinggi membuat *attacker* sulit untuk melepaskan *watermark* – nya.

Untuk pengujian terakhir citra yang dilakukan adalah dengan mengujikan citra yang disisipi *watermark* dari sisi *nonperceptibility*. Pengujian ini melibatkan 10 orang dari berbagai usia. Dari hasil penelitian didapatkan bahwa semua responden menyatakan tidak melihat perbedaan antara citra asli maupun citra yang telah disisipi *watermark* untuk semua alpha pada tanda *watermark* dengan ukuran 20x20 dan 40x40 pixel. Untuk citra pembawa *watermark* dengan tanda *watermark* 60x60 pixel dengan semua alpha, responden melihat tampilan citra yang disisipi *watermark* sedikit ada goresan dibagian atas citra. Hal ini menunjukkan bahwa citra yang disisipi *watermark* telah memenuhi syarat Tidak Terlihat (*Nonperceptibility*) untuk ukuran watermark 20x20 pixel dan 40x40 pixel.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan, maka dapat diambil kesimpulan antara lain:

1. Kualitas citra dengan pemberian alpha 0.0001 terlihat lebih baik dibandingkan dengan citra pemberian alpha 0.0003 dan 0.0006. Hal ini dapat dilihat dari nilai rata – rata *Mean Square Error* (MSE), dan rata – rata nilai *Peak Signal Noise to Ratio* (PSNR) yang dihasilkan. Pemberian alpha 0.0001 menghasilkan nilai rata – rata MSE terkecil sebesar 1,0789 dan nilai PSNR terbesar sebesar 61,8839.
2. Kualitas citra yang dihasilkan dengan pemberian tanda *watermark* dengan ukuran 20x20 pixel lebih bagus bila dibandingkan dengan pemberian tanda *watermark* dengan ukuran 40x40 dan 60x60 pixel. Hal ini ditunjukkan dengan nilai MSE dan PSNR yang dihasilkan.
3. Tanda *watermark* yang disisipkan berhasil di deteksi dengan baik setelah dilakukan serangan pemrosesan citra yaitu penambahan *noise*, *cropping*, *sharpening*, *resizing*, *rotation*.
4. Hasil citra yang telah disisipi *watermark* yang diujikan kepada responden menunjukkan bahwa seluruh responden tidak melihat perbedaan antara citra asli maupun citra yang telah disisipi *watermark* untuk semua alpha pada tanda *watermark* dengan ukuran 20x20 dan 40x40 pixel. Hal ini menunjukkan bahwa citra yang disisipi *watermark* telah memenuhi syarat Tidak Terlihat (*Nonperceptibility*) untuk ukuran watermark 20x20 pixel dan 40x40 pixel.

5.2 Saran

Beberapa saran untuk pengembangan penelitian lebih lanjut adalah:

1. Penelitian dapat dikembangkan dengan menggunakan nilai alpha yang lebih besar dan ukuran citra yang lebih besar
2. Penelitian dapat dikembangkan dengan menambahkan *attack* terhadap citra yang disisipi *watermark*.
3. Penelitian dapat dikembangkan untuk dokumen lain yang dapat diberikan *watermark* seperti *audio*, *video*, maupun *text*.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Alfatwa, Dean Fathony (2008). *Watermarking pada citra digital menggunakan discrete wavelet transform*. Program Studi Teknik Informatika. Institut Teknologi Bandung. Bandung
- 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>>.
- Ford, A., dan Roberts, A.(1998), *Colour Space Conversions*, <<http://www.inforamp.net/~poynton/PDFs/coloureq.pdf>>.
- Gonzalez, R.C. dan R. E. Woods (2005). *Digital Image Processing*, 2nd ed., Prentice-Hall Inc., New Jersey.
- Juergen, Seitz (2004), *Digital Watermarking for Digital Media*, Information Science Publishing, Germany.
- Hamilton, E., September (1992), *JPEG File Interchange Format*, Version 1.02.<<http://www.w3.org/Graphics/JPEG/jfif.txt>>.
- Jahne,Bernd (2002).*Digital Image Processing*. Springer. Germany.
- Latifawariq (2007). *Ketahanan DigitalWatermarking pada Citra Terhadap Berbagai Image Distortion Dalam Sebaran Frekuensi dengan Discrete Cosine Transform*. Program Studi Ilmu Komputer FMIPA, Universitas Brawijaya. Malang.
- Linda S., Agustina (2005). *Penerapan Region of Interest (ROI) pada Metode Kompresi JPEG2000*. Departemen Teknik Infomatika, Institut Teknologi Bandung.
- Markus Rullmann.(2000). *A VLSI Architecture for the Discrete Wavelet Transform - Literature Review -* . New Jersey.

- Mar,Eriel.,dkk(2009).*Implementasi dan Analisis Video Watermarking dengan format video MPEG berbasis Wavelet Transform*.Jurusan Teknik Elektro Sekolah Tinggi Teknologi Bandung.
- Novamizanti, Ledy** (2009). *Dekomposisi Transformasi Wavelet*. Perpustakaan Institut Teknologi Telkom. Bandung.
- Permana, sutrisna(2008). *Deteksi Kebakaran Webcam Dengan Pengolahan Citra Digital "Fire Detection Using Webcam With Digital Image Processing"*. IT Telkom.Bandung
- Polikar, Robi (1988), *Multi Resolution Analysis : The Discrete Wavelet Transform*, Durham Computation Center, Iowa state University.
- Munir, Rinaldi (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Penerbit Informatika, Bandung.
- Sirait,Rummi(2007). *Teknologi Watermarking pada Citra Digital*. Penerbit. Informatika, Bandung.
- Sripathi,Deepika(2003).*Efficient Implementations of Discrete Wavelet Transform Using FPGAs*. Florida State University.America.
- Sugiharto,Aris. Arif Wibawa, Helmi (2009). *Ketahanan watermarking terhadap serangan kompresi jpeg*. Jurusan Matematika FMIPA UNDIP. Semarang.
- Supangkat,H.,dkk(2000).*Watermarking sebagai Teknik Penyembunyian Label Hak Cipta pada Data Digital*. Departemen Teknik Elektro, Institut Teknologi Bandung.
- 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.
- Tamatjita (2006). *Kriptografi Untuk Perlindungan Data*.
<http://www.smeapgri>

tng.sch.id/sekolah/html/?tab=amik§ion=artikel&num=001&P
HPSESSID=90305d3781d3f6b0. Terakhir diakses pada tanggal 2
Oktober 2010 pada pukul 16.43 WIB

Xiang – Gen Xia (2008). *A Multiresolution Watermark for Digital
Images*. Department of Electrical Engineering University of
Delaware. Newark

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA

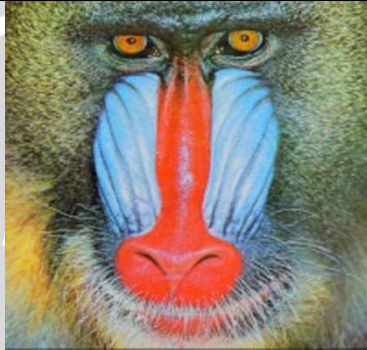


LAMPIRAN

Empat Jenis citra yang digunakan adalah sebagai berikut :



01221_mandownmeeeeeedic.bmp
(800x600)pixel



baboon3.bmp
(700x700)pixel





kupu1.bmp
(800x600)pixel



SydneyHarbourBridge_f
(800x600)pixel

Tanda Watermark yang digunakan adalah :

Citra watermark	Ukuran Citra
	mipa2010.bmp (20x20 pixel)
	mipa2010.bmp (40x40 pixel)



mipa2010.bmp (60x60
pixel)

UNIVERSITAS BRAWIJAYA



Hasil Penyisipan *Watermark* dengan Menggunakan Metode *Discrete Wavelet Transform*

Tabel 1 Hasil Uji Kualitas Citra Disisipi *Watermark* $\alpha = 0.0001$

Citra asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR
01221_m andownm eeeeeedic .bmp	01221_mando wnmeeeeeedic (wat1).bmp	mipa2010.b mp(20x20)	$3,15 \times 10^{-4}$	83,1462
baboon3. bmp	baboon3(wat1) .bmp	mipa2010.b mp(20x20)	$7,28 \times 10^{-4}$	79,5091
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 1).bmp	mipa2010.b mp(20x20)	$4,67 \times 10^{-4}$	81,4407
kupu1.b mp	kupu1(wat1).b mp	mipa2010.b mp(20x20)	$1,63 \times 10^{-4}$	85,9945
01221_m andownm eeeeeedic .bmp	01221_mando wnmeeeeeedic (wat2).bmp	mipa2010.b mp(40x40)	$5,92 \times 10^{-2}$	60,4059
baboon3. bmp	baboon3(wat2) .bmp	mipa2010.b mp(40x40)	$4,83 \times 10^{-2}$	61,2889
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 2).bmp	mipa2010.b mp(40x40)	$1,08 \times 10^{-1}$	57,7786
kupu1.b mp	kupu1(wat2).b mp	mipa2010.b mp(40x40)	$7,59 \times 10^{-2}$	59,3303
01221_m andownm eeeeeedic .bmp	01221_mando wnmeeeeeedic (wat3).bmp	mipa2010.b mp(60x60)	2,561	44,0459
baboon3. bmp	baboon3(wat3) .bmp	mipa2010.b mp(60x60)	2,298	44,5175
SydneyH arbourBri	SydneyHarbou rBridge_f(wat	mipa2010.b mp(60x60)	2,391	44,3441

dge_f.bmp	3).bmp			
kupu1.bmp	kupu1(wat3).bmp	mipa2010.bmp(60x60)	5,402	40,8051

Tabel 2 Hasil Uji Kualitas Citra Disisipi *Watermark* $\alpha = 0.0003$

Citra asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat1).bmp	mipa2010.bmp(20x20)	$2,85 \times 10^{-3}$	73,5792
baboon3.bmp	baboon3(wat1).bmp	mipa2010.bmp(20x20)	$6,48 \times 10^{-3}$	70,0116
SydneyHarbourBridge_f.bmp	SydneyHarbourBridge_f(wat1).bmp	mipa2010.bmp(20x20)	$4,26 \times 10^{-2}$	61,8375
kupu1.bmp	kupu1(wat1).bmp	mipa2010.bmp(20x20)	$1,38 \times 10^{-3}$	76,7346
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat2).bmp	mipa2010.bmp(40x40)	$1,31 \times 10^{-1}$	56,9595
baboon3.bmp	baboon3(wat2).bmp	mipa2010.bmp(40x40)	$1,29 \times 10^{-1}$	57,0378
SydneyHarbourBridge_f.bmp	SydneyHarbourBridge_f(wat2).bmp	mipa2010.bmp(40x40)	$2,52 \times 10^{-1}$	54,1239
kupu1.bmp	kupu1(wat2).bmp	mipa2010.bmp(40x40)	$2,97 \times 10^{-1}$	53,3973
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat3).bmp	mipa2010.bmp(60x60)	4,683	41,4254
baboon3.bmp	baboon3(wat3).bmp	mipa2010.bmp	4,917	41,2136

bmp	.bmp	mp(60x60)		
SydneyHarbourBridge_f.bmp	SydneyHarbourBridge_f(wat3).bmp	mipa2010.bmp(60x60)	4,430	41,6666
kupu1.bmp	kupu1(wat3).bmp	mipa2010.bmp(60x60)	9,846	38,1981

Tabel 3 Hasil Uji Kualitas Citra Disisipi *Watermark* $\alpha = 0.0006$

Citra asli	Citra yang disisipi <i>watermark</i>	Citra <i>watermark</i>	MSE	PSNR
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat1).bmp	mipa2010.bmp(20x20)	$1,12 \times 10^{-2}$	67,6372
baboon3.bmp	baboon3(wat1).bmp	mipa2010.bmp(20x20)	$9,26 \times 10^{-3}$	68,4649
SydneyHarbourBridge_f.bmp	SydneyHarbourBridge_f(wat1).bmp	mipa2010.bmp(20x20)	$4,87 \times 10^{-2}$	61,2472
kupu1.bmp	kupu1(wat1).bmp	mipa2010.bmp(20x20)	$5,62 \times 10^{-3}$	70,6272
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat2).bmp	mipa2010.bmp(40x40)	$2,43 \times 10^{-1}$	54,2601
baboon3.bmp	baboon3(wat2).bmp	mipa2010.bmp(40x40)	$4,61 \times 10^{-1}$	51,4907
SydneyHarbourBridge_f.bmp	SydneyHarbourBridge_f(wat2).bmp	mipa2010.bmp(40x40)	$4,91 \times 10^{-1}$	51,2119
kupu1.bmp	kupu1(wat2).bmp	mipa2010.bmp(40x40)	$5,31 \times 10^{-1}$	50,8739
01221_mandownmeeeeedic.bmp	01221_mandownmeeeeedic(wat3).bmp	mipa2010.bmp(60x60)	5,901	40,4217

eeeeedic .bmp	(wat3).bmp			
baboon3. bmp	baboon3(wat3) .bmp	mipa2010.b mp(60x60)	8,866	38,6534
SydneyH arbourBri dge_f.bm p	SydneyHarbou rBridge_f(wat 3).bmp	mipa2010.b mp(60x60)	6,198	40,2079
Kupu1	kupu1(wat3).b mp	mipa2010.b mp(60x60)	14,125	36,6309



Hasil Pendeteksian *Watermark* dengan Menggunakan Metode *Discrete Waveket Transform*

a. Tanpa Serangan

Tabel 4 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila tidak diberikan Serangan $\alpha = 0.0001$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeedic(wat1).bmp	Tanpa Serangan	Ada
2	baboon3(wat1).bmp	Tanpa Serangan	Ada
3	SydneyHarbourBridg e_f(wat1).bmp	Tanpa Serangan	Ada
4	kupu1(wat1).bmp	Tanpa Serangan	Ada
5	01221_mandownmee eeedic(wat2).bmp	Tanpa Serangan	Ada
6	baboon3(wat2).bmp	Tanpa Serangan	Ada
7	SydneyHarbourBridg e_f(wat2).bmp	Tanpa Serangan	Ada
8	kupu1(wat2).bmp	Tanpa Serangan	Ada
9	01221_mandownmee eeedic(wat3).bmp	Tanpa Serangan	Ada
10	baboon3(wat3).bmp	Tanpa Serangan	Ada
11	SydneyHarbourBridg e_f(wat3).bmp	Tanpa Serangan	Ada
12	kupu1(wat3).bmp	Tanpa Serangan	Ada

Tabel 5 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila tidak diberikan Serangan $\alpha = 0.0003$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeedic(wat1).bmp	Tanpa Serangan	Ada
2	baboon3(wat1).bmp	Tanpa Serangan	Ada
3	SydneyHarbourBridg e_f(wat1).bmp	Tanpa Serangan	Ada
4	kupu1(wat1).bmp	Tanpa Serangan	Ada

5	01221_mandownmee eeeeedic(wat2).bmp	Tanpa Serangan	Ada
6	baboon3(wat2).bmp	Tanpa Serangan	Ada
7	SydneyHarbourBridg e_f(wat2).bmp	Tanpa Serangan	Ada
8	kupu1(wat2).bmp	Tanpa Serangan	Ada
9	01221_mandownmee eeeeedic(wat3).bmp	Tanpa Serangan	Ada
10	baboon3(wat3).bmp	Tanpa Serangan	Ada
11	SydneyHarbourBridg e_f(wat3).bmp	Tanpa Serangan	Ada
12	kupu1(wat3).bmp	Tanpa Serangan	Ada

Tabel 6 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila tidak diberikan Serangan $\alpha = 0.0006$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeeeedic(wat1).bmp	Tanpa Serangan	Ada
2	baboon3(wat1).bmp	Tanpa Serangan	Ada
3	SydneyHarbourBridg e_f(wat1).bmp	Tanpa Serangan	Ada
4	kupu1(wat1).bmp	Tanpa Serangan	Ada
5	01221_mandownmee eeeeedic(wat2).bmp	Tanpa Serangan	Ada
6	baboon3(wat2).bmp	Tanpa Serangan	Ada
7	SydneyHarbourBridg e_f(wat2).bmp	Tanpa Serangan	Ada
8	kupu1(wat2).bmp	Tanpa Serangan	Ada
9	01221_mandownmee eeeeedic(wat3).bmp	Tanpa Serangan	Ada
10	baboon3(wat3).bmp	Tanpa Serangan	Ada
11	SydneyHarbourBridg e_f(wat3).bmp	Tanpa Serangan	Ada
12	kupu1(wat3).bmp	Tanpa Serangan	Ada

b. Serangan Cropping $\alpha = 0.0001$

Tabel 7 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Cropping* $\alpha = 0.0001$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeeedic(wat1).bmp	<i>Cropping</i> 25%	Ada
2	baboon3(wat1).bmp	<i>Cropping</i> 25%	Ada
3	SydneyHarbourBridg e_f(wat1).bmp	<i>Cropping</i> 25%	Ada
4	kupu1(wat1).bmp	<i>Cropping</i> 25%	Ada
5	01221_mandownmee eeeedic(wat2).bmp	<i>Cropping</i> 25%	Ada
6	baboon3(wat2).bmp	<i>Cropping</i> 25%	Ada
7	SydneyHarbourBridg e_f(wat2).bmp	<i>Cropping</i> 25%	Ada
8	kupu1(wat2).bmp	<i>Cropping</i> 25%	Ada
9	01221_mandownmee eeeedic(wat3).bmp	<i>Cropping</i> 25%	Ada
10	baboon3(wat3).bmp	<i>Cropping</i> 25%	Ada
11	SydneyHarbourBridg e_f(wat3).bmp	<i>Cropping</i> 25%	Ada
12	kupu1(wat3).bmp	<i>Cropping</i> 25%	Ada
13	01221_mandownmee eeeedic(wat1).bmp	<i>Cropping</i> 40%	Ada
14	baboon3(wat1).bmp	<i>Cropping</i> 40%	Ada
15	SydneyHarbourBridg e_f(wat1).bmp	<i>Cropping</i> 40%	Ada
16	kupu1(wat1).bmp	<i>Cropping</i> 40%	Ada
17	01221_mandownmee eeeedic(wat2).bmp	<i>Cropping</i> 40%	Ada
18	baboon3(wat2).bmp	<i>Cropping</i> 40%	Ada
19	SydneyHarbourBridg e_f(wat2).bmp	<i>Cropping</i> 40%	Ada
20	kupu1(wat2).bmp	<i>Cropping</i> 40%	Ada
21	01221_mandownmee	<i>Cropping</i> 40%	Ada

	eeeeedic(wat3).bmp		
22	baboon3(wat3).bmp	<i>Cropping 40%</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Cropping 40%</i>	Ada
24	kupu1(wat3).bmp	<i>Cropping 40%</i>	Ada

c. **Serangan *Cropping* $\alpha = 0.0003$**

Tabel 8 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila Serangan *Cropping* $\alpha = 0.0003$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeeeedic(wat1).bmp	<i>Cropping 25%</i>	Ada
2	baboon3(wat1).bmp	<i>Cropping 25%</i>	Ada
3	SydneyHarbourBridge_f(wat1).bmp	<i>Cropping 25%</i>	Ada
4	kupu1(wat1).bmp	<i>Cropping 25%</i>	Ada
5	01221_mandownmee eeeeedic(wat2).bmp	<i>Cropping 25%</i>	Ada
6	baboon3(wat2).bmp	<i>Cropping 25%</i>	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Cropping 25%</i>	Ada
8	kupu1(wat2).bmp	<i>Cropping 25%</i>	Ada
9	01221_mandownmee eeeeedic(wat3).bmp	<i>Cropping 25%</i>	Ada
10	baboon3(wat3).bmp	<i>Cropping 25%</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Cropping 25%</i>	Ada
12	kupu1(wat3).bmp	<i>Cropping 25%</i>	Ada
13	01221_mandownmee eeeeedic(wat1).bmp	<i>Cropping 40%</i>	Ada
14	baboon3(wat1).bmp	<i>Cropping 40%</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Cropping 40%</i>	Ada
16	kupu1(wat1).bmp	<i>Cropping 40%</i>	Ada
17	01221_mandownmee	<i>Cropping 40%</i>	Ada

	eeeeedic(wat2).bmp		
18	baboon3(wat2).bmp	<i>Cropping 40%</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Cropping 40%</i>	Ada
20	kupu1(wat2).bmp	<i>Cropping 40%</i>	Ada
21	01221_mandownmee eeeeedic(wat3).bmp	<i>Cropping 40%</i>	Ada
22	baboon3(wat3).bmp	<i>Cropping 40%</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Cropping 40%</i>	Ada
24	kupu1(wat3).bmp	<i>Cropping 40%</i>	Ada

d. Serangan *Cropping* $\alpha = 0.0006$

Tabel 9 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Cropping* $\alpha = 0.0006$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeeeedic(wat1).bmp	<i>Cropping 25%</i>	Ada
2	baboon3(wat1).bmp	<i>Cropping 25%</i>	Ada
3	SydneyHarbourBridge_f(wat1).bmp	<i>Cropping 25%</i>	Ada
4	kupu1(wat1).bmp	<i>Cropping 25%</i>	Ada
5	01221_mandownmee eeeeedic(wat2).bmp	<i>Cropping 25%</i>	Ada
6	baboon3(wat2).bmp	<i>Cropping 25%</i>	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Cropping 25%</i>	Ada
8	kupu1(wat2).bmp	<i>Cropping 25%</i>	Ada
9	01221_mandownmee eeeeedic(wat3).bmp	<i>Cropping 25%</i>	Ada
10	baboon3(wat3).bmp	<i>Cropping 25%</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Cropping 25%</i>	Ada
12	kupu1(wat3).bmp	<i>Cropping 25%</i>	Ada
13	01221_mandownmee	<i>Cropping 40%</i>	Ada

	eeeeedic(wat1).bmp		
14	baboon3(wat1).bmp	<i>Cropping 40%</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Cropping 40%</i>	Ada
16	kupu1(wat1).bmp	<i>Cropping 40%</i>	Ada
17	01221_mandownmee eeeeedic(wat2).bmp	<i>Cropping 40%</i>	Ada
18	baboon3(wat2).bmp	<i>Cropping 40%</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Cropping 40%</i>	Ada
20	kupu1(wat2).bmp	<i>Cropping 40%</i>	Ada
21	01221_mandownmee eeeeedic(wat3).bmp	<i>Cropping 40%</i>	Ada
22	baboon3(wat3).bmp	<i>Cropping 40%</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Cropping 40%</i>	Ada
24	kupu1(wat3).bmp	<i>Cropping 40%</i>	Ada

e. Serangan Penambahan Noise $\alpha = 0.0001$

Tabel 10 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan Penambahan Noise $\alpha = 0.0001$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmee eeeeedic(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
2	baboon3(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
3	SydneyHarbourBridge_f(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
4	kupu1(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
5	01221_mandownmee eeeeedic(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
6	baboon3(wat2).bmp	<i>Uniform Noise</i>	Ada

	p	5%	
7	SydneyHarbourBri dge_f(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
8	kupu1(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
9	01221_mandownm eeeeedic(wat3).bm p	<i>Uniform Noise</i> 5%	Ada
10	baboon3(wat3).bm p	<i>Uniform Noise</i> 5%	Ada
11	SydneyHarbourBri dge_f(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
12	kupu1(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
13	01221_mandownm eeeeedic(wat1).bm p	<i>Uniform Noise</i> 10%	Ada
14	baboon3(wat1).bm p	<i>Uniform Noise</i> 10%	Ada
15	SydneyHarbourBri dge_f(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
16	kupu1(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
17	01221_mandownm eeeeedic(wat2).bm p	<i>Uniform Noise</i> 10%	Ada
18	baboon3(wat2).bm p	<i>Uniform Noise</i> 10%	Ada
19	SydneyHarbourBri dge_f(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
20	kupu1(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
21	01221_mandownm eeeeedic(wat3).bm p	<i>Uniform Noise</i> 10%	Ada
22	baboon3(wat3).bm p	<i>Uniform Noise</i> 10%	Ada
23	SydneyHarbourBri	<i>Uniform Noise</i>	Ada

	dge_f(wat3).bmp	10%	
24	kupu1(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
25	01221_mandownm eeeeedic(wat1).bm p	<i>Gaussian Noise</i> 5%	Ada
26	baboon3(wat1).bm p	<i>Gaussian Noise</i> 5%	Ada
27	SydneyHarbourBri dge_f(wat1).bmp	<i>Gaussian Noise</i> 5%	Ada
28	kupu1(wat1).bmp	<i>Gaussian Noise</i> 5%	Ada
29	01221_mandownm eeeeedic(wat2).bm p	<i>Gaussian Noise</i> 5%	Ada
30	baboon3(wat2).bm p	<i>Gaussian Noise</i> 5%	Ada
31	SydneyHarbourBri dge_f(wat2).bmp	<i>Gaussian Noise</i> 5%	Ada
32	kupu1(wat2).bmp	<i>Gaussian Noise</i> 5%	Ada
33	01221_mandownm eeeeedic(wat3).bm p	<i>Gaussian Noise</i> 5%	Ada
34	baboon3(wat3).bm p	<i>Gaussian Noise</i> 5%	Ada
35	SydneyHarbourBri dge_f(wat3).bmp	<i>Gaussian Noise</i> 5%	Ada
36	kupu1(wat3).bmp	<i>Gaussian Noise</i> 5%	Ada
37	01221_mandownm eeeeedic(wat1).bm p	<i>Gaussian Noise</i> 10%	Ada
38	baboon3(wat1).bm p	<i>Gaussian Noise</i> 10%	Ada
39	SydneyHarbourBri dge_f(wat1).bmp	<i>Gaussian Noise</i> 10%	Ada
40	kupu1(wat1).bmp	<i>Gaussian Noise</i>	Ada

		10%	
41	01221_mandownm eeeeedic(wat2).bm p	<i>Gaussian Noise</i> 10%	Ada
42	baboon3(wat2).bm p	<i>Gaussian Noise</i> 10%	Ada
43	SydneyHarbourBri dge_f(wat2).bmp	<i>Gaussian Noise</i> 10%	Ada
44	kupu1(wat2).bmp	<i>Gaussian Noise</i> 10%	Ada
45	01221_mandownm eeeeedic(wat3).bm p	<i>Gaussian Noise</i> 10%	Ada
46	baboon3(wat3).bm p	<i>Gaussian Noise</i> 10%	Ada
47	SydneyHarbourBri dge_f(wat3).bmp	<i>Gaussian Noise</i> 10%	Ada
48	kupu1(wat3).bmp	<i>Gaussian Noise</i> 10%	Ada

f. Serangan Penambahan Noise $\alpha = 0.0003$

Tabel 11 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan Penambahan Noise $\alpha = 0.0003$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Uniform Noise</i> 5%	Ada
2	baboon3(wat1).bm p	<i>Uniform Noise</i> 5%	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
4	kupu1(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
5	01221_mandownm eeeeedic(wat2).bm p	<i>Uniform Noise</i> 5%	Ada

6	baboon3(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
8	kupu1(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
9	01221_mandownmeeeeedic(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
10	baboon3(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
12	kupu1(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
13	01221_mandownmeeeeedic(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
14	baboon3(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
16	kupu1(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
17	01221_mandownmeeeeedic(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
18	baboon3(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
20	kupu1(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
21	01221_mandownmeeeeedic(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
22	baboon3(wat3).bmp	<i>Uniform Noise</i> 10%	Ada

23	SydneyHarbourBri dge_f(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
24	kupu1(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
25	01221_mandownm eeeeedic(wat1).bm p	<i>Gaussian Noise</i> 5%	Ada
26	baboon3(wat1).bm p	<i>Gaussian Noise</i> 5%	Ada
27	SydneyHarbourBri dge_f(wat1).bmp	<i>Gaussian Noise</i> 5%	Ada
28	kupu1(wat1).bmp	<i>Gaussian Noise</i> 5%	Ada
29	01221_mandownm eeeeedic(wat2).bm p	<i>Gaussian Noise</i> 5%	Ada
30	baboon3(wat2).bm p	<i>Gaussian Noise</i> 5%	Ada
31	SydneyHarbourBri dge_f(wat2).bmp	<i>Gaussian Noise</i> 5%	Ada
32	kupu1(wat2).bmp	<i>Gaussian Noise</i> 5%	Ada
33	01221_mandownm eeeeedic(wat3).bm p	<i>Gaussian Noise</i> 5%	Ada
34	baboon3(wat3).bm p	<i>Gaussian Noise</i> 5%	Ada
35	SydneyHarbourBri dge_f(wat3).bmp	<i>Gaussian Noise</i> 5%	Ada
36	kupu1(wat3).bmp	<i>Gaussian Noise</i> 5%	Ada
37	01221_mandownm eeeeedic(wat1).bm p	<i>Gaussian Noise</i> 10%	Ada
38	baboon3(wat1).bm p	<i>Gaussian Noise</i> 10%	Ada
39	SydneyHarbourBri dge_f(wat1).bmp	<i>Gaussian Noise</i> 10%	Ada

40	kupu1(wat1).bmp	<i>Gaussian Noise</i> 10%	Ada
41	01221_mandownm eeeeedic(wat2).bm p	<i>Gaussian Noise</i> 10%	Ada
42	baboon3(wat2).bm p	<i>Gaussian Noise</i> 10%	Ada
43	SydneyHarbourBri dge_f(wat2).bmp	<i>Gaussian Noise</i> 10%	Ada
44	kupu1(wat2).bmp	<i>Gaussian Noise</i> 10%	Ada
45	01221_mandownm eeeeedic(wat3).bm p	<i>Gaussian Noise</i> 10%	Ada
46	baboon3(wat3).bm p	<i>Gaussian Noise</i> 10%	Ada
47	SydneyHarbourBri dge_f(wat3).bmp	<i>Gaussian Noise</i> 10%	Ada
48	kupu1(wat3).bmp	<i>Gaussian Noise</i> 10%	Ada

g. Serangan Penambahan *Noise* $\alpha = 0.0006$

Tabel 12 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan Penambahan *Noise* $\alpha = 0.0006$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Uniform Noise</i> 5%	Ada
2	baboon3(wat1).bm p	<i>Uniform Noise</i> 5%	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
4	kupu1(wat1).bmp	<i>Uniform Noise</i> 5%	Ada
5	01221_mandownm eeeeedic(wat2).bm	<i>Uniform Noise</i> 5%	Ada

	p		
6	baboon3(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
8	kupu1(wat2).bmp	<i>Uniform Noise</i> 5%	Ada
9	01221_mandownmeeeeedic(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
10	baboon3(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
12	kupu1(wat3).bmp	<i>Uniform Noise</i> 5%	Ada
13	01221_mandownmeeeeedic(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
14	baboon3(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
16	kupu1(wat1).bmp	<i>Uniform Noise</i> 10%	Ada
17	01221_mandownmeeeeedic(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
18	baboon3(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
20	kupu1(wat2).bmp	<i>Uniform Noise</i> 10%	Ada
21	01221_mandownmeeeeedic(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
22	baboon3(wat3).bmp	<i>Uniform Noise</i>	Ada

	p	10%	
23	SydneyHarbourBri dge_f(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
24	kupu1(wat3).bmp	<i>Uniform Noise</i> 10%	Ada
25	01221_mandownm eeeeedic(wat1).bm p	<i>Gaussian Noise</i> 5%	Ada
26	baboon3(wat1).bm p	<i>Gaussian Noise</i> 5%	Ada
27	SydneyHarbourBri dge_f(wat1).bmp	<i>Gaussian Noise</i> 5%	Ada
28	kupu1(wat1).bmp	<i>Gaussian Noise</i> 5%	Ada
29	01221_mandownm eeeeedic(wat2).bm p	<i>Gaussian Noise</i> 5%	Ada
30	baboon3(wat2).bm p	<i>Gaussian Noise</i> 5%	Ada
31	SydneyHarbourBri dge_f(wat2).bmp	<i>Gaussian Noise</i> 5%	Ada
32	kupu1(wat2).bmp	<i>Gaussian Noise</i> 5%	Ada
33	01221_mandownm eeeeedic(wat3).bm p	<i>Gaussian Noise</i> 5%	Ada
34	baboon3(wat3).bm p	<i>Gaussian Noise</i> 5%	Ada
35	SydneyHarbourBri dge_f(wat3).bmp	<i>Gaussian Noise</i> 5%	Ada
36	kupu1(wat3).bmp	<i>Gaussian Noise</i> 5%	Ada
37	01221_mandownm eeeeedic(wat1).bm p	<i>Gaussian Noise</i> 10%	Ada
38	baboon3(wat1).bm p	<i>Gaussian Noise</i> 10%	Ada
39	SydneyHarbourBri	<i>Gaussian Noise</i>	Ada

	dge_f(wat1).bmp	10%	
40	kupu1(wat1).bmp	<i>Gaussian Noise</i> 10%	Ada
41	01221_mandownm eeeeedic(wat2).bm p	<i>Gaussian Noise</i> 10%	Ada
42	baboon3(wat2).bm p	<i>Gaussian Noise</i> 10%	Ada
43	SydneyHarbourBri dge_f(wat2).bmp	<i>Gaussian Noise</i> 10%	Ada
44	kupu1(wat2).bmp	<i>Gaussian Noise</i> 10%	Ada
45	01221_mandownm eeeeedic(wat3).bm p	<i>Gaussian Noise</i> 10%	Ada
46	baboon3(wat3).bm p	<i>Gaussian Noise</i> 10%	Ada
47	SydneyHarbourBri dge_f(wat3).bmp	<i>Gaussian Noise</i> 10%	Ada
48	kupu1(wat3).bmp	<i>Gaussian Noise</i> 10%	Ada

h. Resizing $\alpha = 0.0001$

Tabel 13 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Resizing* $\alpha = 0.0001$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Resizing</i> 25%	Ada
2	baboon3(wat1).bm p	<i>Resizing</i> 25%	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Resizing</i> 25%	Ada
4	kupu1(wat1).bmp	<i>Resizing</i> 25%	Ada
5	01221_mandownm	<i>Resizing</i> 25%	Ada

	eeeeedic(wat2).bmp		
6	baboon3(wat2).bmp	<i>Resizing 25%</i>	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Resizing 25%</i>	Ada
8	kupu1(wat2).bmp	<i>Resizing 25%</i>	Ada
9	01221_mandownmeeeeeedic(wat3).bmp	<i>Resizing 25%</i>	Ada
10	baboon3(wat3).bmp	<i>Resizing 25%</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Resizing 25%</i>	Ada
12	kupu1(wat3).bmp	<i>Resizing 25%</i>	Ada
13	01221_mandownmeeeeeedic(wat1).bmp	<i>Resizing 50%</i>	Ada
14	baboon3(wat1).bmp	<i>Resizing 50%</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Resizing 50%</i>	Ada
16	kupu1(wat1).bmp	<i>Resizing 50%</i>	Ada
17	01221_mandownmeeeeeedic(wat2).bmp	<i>Resizing 50%</i>	Ada
18	baboon3(wat2).bmp	<i>Resizing 50%</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Resizing 50%</i>	Ada
20	kupu1(wat2).bmp	<i>Resizing 50%</i>	Ada
21	01221_mandownmeeeeeedic(wat3).bmp	<i>Resizing 50%</i>	Ada
22	baboon3(wat3).bmp	<i>Resizing 50%</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Resizing 50%</i>	Ada

24	kupu1(wat3).bmp	Resizing 50%	Ada
25	01221_mandownm eeeeedic(wat1).bm p	Resizing 75%	Ada
26	baboon3(wat1).bm p	Resizing 75%	Ada
27	SydneyHarbourBri dge_f(wat1).bmp	Resizing 75%	Ada
28	kupu1(wat1).bmp	Resizing 75%	Ada
29	01221_mandownm eeeeedic(wat2).bm p	Resizing 75%	Ada
30	baboon3(wat2).bm p	Resizing 75%	Ada
31	SydneyHarbourBri dge_f(wat2).bmp	Resizing 75%	Ada
32	kupu1(wat2).bmp	Resizing 75%	Ada
33	01221_mandownm eeeeedic(wat3).bm p	Resizing 75%	Ada
34	baboon3(wat3).bm p	Resizing 75%	Ada
35	SydneyHarbourBri dge_f(wat3).bmp	Resizing 75%	Ada
36	kupu1(wat3).bmp	Resizing 75%	Ada

i. Resizing $\alpha = 0.0003$

Tabel 14 Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Resizing* $\alpha = 0.0003$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	Resizing 25%	Ada
2	baboon3(wat1).bm p	Resizing 25%	Ada

3	SydneyHarbourBridge_f(wat1).bmp	Resizing 25%	Ada
4	kupu1(wat1).bmp	Resizing 25%	Ada
5	01221_mandownmeeeeedic(wat2).bmp	Resizing 25%	Ada
6	baboon3(wat2).bmp	Resizing 25%	Ada
7	SydneyHarbourBridge_f(wat2).bmp	Resizing 25%	Ada
8	kupu1(wat2).bmp	Resizing 25%	Ada
9	01221_mandownmeeeeedic(wat3).bmp	Resizing 25%	Ada
10	baboon3(wat3).bmp	Resizing 25%	Ada
11	SydneyHarbourBridge_f(wat3).bmp	Resizing 25%	Ada
12	kupu1(wat3).bmp	Resizing 25%	Ada
13	01221_mandownmeeeeedic(wat1).bmp	Resizing 50%	Ada
14	baboon3(wat1).bmp	Resizing 50%	Ada
15	SydneyHarbourBridge_f(wat1).bmp	Resizing 50%	Ada
16	kupu1(wat1).bmp	Resizing 50%	Ada
17	01221_mandownmeeeeedic(wat2).bmp	Resizing 50%	Ada
18	baboon3(wat2).bmp	Resizing 50%	Ada
19	SydneyHarbourBridge_f(wat2).bmp	Resizing 50%	Ada
20	kupu1(wat2).bmp	Resizing 50%	Ada
21	01221_mandownmeeeeedic(wat3).bmp	Resizing 50%	Ada

22	baboon3(wat3).bmp	Resizing 50%	Ada
23	SydneyHarbourBridge_f(wat3).bmp	Resizing 50%	Ada
24	kupu1(wat3).bmp	Resizing 50%	Ada
25	01221_mandownmeeeeedic(wat1).bmp	Resizing 75%	Ada
26	baboon3(wat1).bmp	Resizing 75%	Ada
27	SydneyHarbourBridge_f(wat1).bmp	Resizing 75%	Ada
28	kupu1(wat1).bmp	Resizing 75%	Ada
29	01221_mandownmeeeeedic(wat2).bmp	Resizing 75%	Ada
30	baboon3(wat2).bmp	Resizing 75%	Ada
31	SydneyHarbourBridge_f(wat2).bmp	Resizing 75%	Ada
32	kupu1(wat2).bmp	Resizing 75%	Ada
33	01221_mandownmeeeeedic(wat3).bmp	Resizing 75%	Ada
34	baboon3(wat3).bmp	Resizing 75%	Ada
35	SydneyHarbourBridge_f(wat3).bmp	Resizing 75%	Ada
36	kupu1(wat3).bmp	Resizing 75%	Ada

j. Resizing $\alpha = 0.0006$

Tabel 15. Hasil Uji Ketahanan Citra Disisipi Watermark Apabila diberikan Serangan Resizing $\alpha = 0.0006$

No	Citra yang disisipi watermark	Perlakuan	Deteksi
1	01221_mandownm	Resizing 25%	Ada

	eeeeedic(wat1).bmp		
2	baboon3(wat1).bmp	<i>Resizing 25%</i>	Ada
3	SydneyHarbourBridge_f(wat1).bmp	<i>Resizing 25%</i>	Ada
4	kupu1(wat1).bmp	<i>Resizing 25%</i>	Ada
5	01221_mandownmeeeeeedic(wat2).bmp	<i>Resizing 25%</i>	Ada
6	baboon3(wat2).bmp	<i>Resizing 25%</i>	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Resizing 25%</i>	Ada
8	kupu1(wat2).bmp	<i>Resizing 25%</i>	Ada
9	01221_mandownmeeeeeedic(wat3).bmp	<i>Resizing 25%</i>	Ada
10	baboon3(wat3).bmp	<i>Resizing 25%</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Resizing 25%</i>	Ada
12	kupu1(wat3).bmp	<i>Resizing 25%</i>	Ada
13	01221_mandownmeeeeeedic(wat1).bmp	<i>Resizing 50%</i>	Ada
14	baboon3(wat1).bmp	<i>Resizing 50%</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Resizing 50%</i>	Ada
16	kupu1(wat1).bmp	<i>Resizing 50%</i>	Ada
17	01221_mandownmeeeeeedic(wat2).bmp	<i>Resizing 50%</i>	Ada
18	baboon3(wat2).bmp	<i>Resizing 50%</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Resizing 50%</i>	Ada

20	kupu1(wat2).bmp	<i>Resizing 50%</i>	Ada
21	01221_mandownm eeeeedic(wat3).bm p	<i>Resizing 50%</i>	Ada
22	baboon3(wat3).bm p	<i>Resizing 50%</i>	Ada
23	SydneyHarbourBri dge_f(wat3).bmp	<i>Resizing 50%</i>	Ada
24	kupu1(wat3).bmp	<i>Resizing 50%</i>	Ada
25	01221_mandownm eeeeedic(wat1).bm p	<i>Resizing 75%</i>	Ada
26	baboon3(wat1).bm p	<i>Resizing 75%</i>	Ada
27	SydneyHarbourBri dge_f(wat1).bmp	<i>Resizing 75%</i>	Ada
28	kupu1(wat1).bmp	<i>Resizing 75%</i>	Ada
29	01221_mandownm eeeeedic(wat2).bm p	<i>Resizing 75%</i>	Ada
30	baboon3(wat2).bm p	<i>Resizing 75%</i>	Ada
31	SydneyHarbourBri dge_f(wat2).bmp	<i>Resizing 75%</i>	Ada
32	kupu1(wat2).bmp	<i>Resizing 75%</i>	Ada
33	01221_mandownm eeeeedic(wat3).bm p	<i>Resizing 75%</i>	Ada
34	baboon3(wat3).bm p	<i>Resizing 75%</i>	Ada
35	SydneyHarbourBri dge_f(wat3).bmp	<i>Resizing 75%</i>	Ada
36	kupu1(wat3).bmp	<i>Resizing 75%</i>	Ada

k. *Sharpen* $\alpha = 0.0001$

Tabel 16. Hasil Uji Ketahanan Citra ter – *Watermark* Apabila diberikan Serangan *Sharpen* $\alpha = 0.0001$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Sharpen</i>	Ada
2	baboon3(wat1).bm p	<i>Sharpen</i>	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Sharpen</i>	Ada
4	kupu1(wat1).bmp	<i>Sharpen</i>	Ada
5	01221_mandownm eeeeedic(wat2).bm p	<i>Sharpen</i>	Ada
6	baboon3(wat2).bm p	<i>Sharpen</i>	Ada
7	SydneyHarbourBri dge_f(wat2).bmp	<i>Sharpen</i>	Ada
8	kupu1(wat2).bmp	<i>Sharpen</i>	Ada
9	01221_mandownm eeeeedic(wat3).bm p	<i>Sharpen</i>	Ada
10	baboon3(wat3).bm p	<i>Sharpen</i>	Ada
11	SydneyHarbourBri dge_f(wat3).bmp	<i>Sharpen</i>	Ada
12	kupu1(wat3).bmp	<i>Sharpen</i>	Ada
13	01221_mandownm eeeeedic(wat1).bm p	<i>Sharpen Edge</i>	Ada
14	baboon3(wat1).bm p	<i>Sharpen Edge</i>	Ada
15	SydneyHarbourBri dge_f(wat1).bmp	<i>Sharpen Edge</i>	Ada

16	kupu1(wat1).bmp	<i>Sharpen Edge</i>	Ada
17	01221_mandownm eeeeedic(wat2).bm p	<i>Sharpen Edge</i>	Ada
18	baboon3(wat2).bm p	<i>Sharpen Edge</i>	Ada
19	SydneyHarbourBri dge_f(wat2).bmp	<i>Sharpen Edge</i>	Ada
20	kupu1(wat2).bmp	<i>Sharpen Edge</i>	Ada
21	01221_mandownm eeeeedic(wat3).bm p	<i>Sharpen Edge</i>	Ada
22	baboon3(wat3).bm p	<i>Sharpen Edge</i>	Ada
23	SydneyHarbourBri dge_f(wat3).bmp	<i>Sharpen Edge</i>	Ada
24	kupu1(wat3).bmp	<i>Sharpen Edge</i>	Ada

l. Sharpening $\alpha = 0.0003$

Tabel 17. Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Sharpen* $\alpha = 0.0003$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Sharpen</i>	Ada
2	baboon3(wat1).bm p	<i>Sharpen</i>	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Sharpen</i>	Ada
4	kupu1(wat1).bmp	<i>Sharpen</i>	Ada
5	01221_mandownm eeeeedic(wat2).bm p	<i>Sharpen</i>	Ada
6	baboon3(wat2).bm p	<i>Sharpen</i>	Ada

7	SydneyHarbourBri dge_f(wat2).bmp	<i>Sharpen</i>	Ada
8	kupu1(wat2).bmp	<i>Sharpen</i>	Ada
9	01221_mandownm eeeeedic(wat3).bm p	<i>Sharpen</i>	Ada
10	baboon3(wat3).bm p	<i>Sharpen</i>	Ada
11	SydneyHarbourBri dge_f(wat3).bmp	<i>Sharpen</i>	Ada
12	kupu1(wat3).bmp	<i>Sharpen</i>	Ada
13	01221_mandownm eeeeedic(wat1).bm p	<i>Sharpen Edge</i>	Ada
14	baboon3(wat1).bm p	<i>Sharpen Edge</i>	Ada
15	SydneyHarbourBri dge_f(wat1).bmp	<i>Sharpen Edge</i>	Ada
16	kupu1(wat1).bmp	<i>Sharpen Edge</i>	Ada
17	01221_mandownm eeeeedic(wat2).bm p	<i>Sharpen Edge</i>	Ada
18	baboon3(wat2).bm p	<i>Sharpen Edge</i>	Ada
19	SydneyHarbourBri dge_f(wat2).bmp	<i>Sharpen Edge</i>	Ada
20	kupu1(wat2).bmp	<i>Sharpen Edge</i>	Ada
21	01221_mandownm eeeeedic(wat3).bm p	<i>Sharpen Edge</i>	Ada
22	baboon3(wat3).bm p	<i>Sharpen Edge</i>	Ada
23	SydneyHarbourBri dge_f(wat3).bmp	<i>Sharpen Edge</i>	Ada
24	kupu1(wat3).bmp	<i>Sharpen Edge</i>	Ada

m. Sharpening $\alpha = 0.0006$

Tabel 18. Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Sharpen* $\alpha = 0.0006$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Sharpen</i>	Ada
2	baboon3(wat1).bm p	<i>Sharpen</i>	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Sharpen</i>	Ada
4	kupu1(wat1).bmp	<i>Sharpen</i>	Ada
5	01221_mandownm eeeeedic(wat2).bm p	<i>Sharpen</i>	Ada
6	baboon3(wat2).bm p	<i>Sharpen</i>	Ada
7	SydneyHarbourBri dge_f(wat2).bmp	<i>Sharpen</i>	Ada
8	kupu1(wat2).bmp	<i>Sharpen</i>	Ada
9	01221_mandownm eeeeedic(wat3).bm p	<i>Sharpen</i>	Ada
10	baboon3(wat3).bm p	<i>Sharpen</i>	Ada
11	SydneyHarbourBri dge_f(wat3).bmp	<i>Sharpen</i>	Ada
12	kupu1(wat3).bmp	<i>Sharpen</i>	Ada
13	01221_mandownm eeeeedic(wat1).bm p	<i>Sharpen Edge</i>	Ada
14	baboon3(wat1).bm p	<i>Sharpen Edge</i>	Ada
15	SydneyHarbourBri dge_f(wat1).bmp	<i>Sharpen Edge</i>	Ada

16	kupu1(wat1).bmp	<i>Sharpen Edge</i>	Ada
17	01221_mandownm eeeeedic(wat2).bm p	<i>Sharpen Edge</i>	Ada
18	baboon3(wat2).bm p	<i>Sharpen Edge</i>	Ada
19	SydneyHarbourBri dge_f(wat2).bmp	<i>Sharpen Edge</i>	Ada
20	kupu1(wat2).bmp	<i>Sharpen Edge</i>	Ada
21	01221_mandownm eeeeedic(wat3).bm p	<i>Sharpen Edge</i>	Ada
22	baboon3(wat3).bm p	<i>Sharpen Edge</i>	Ada
23	SydneyHarbourBri dge_f(wat3).bmp	<i>Sharpen Edge</i>	Ada
24	kupu1(wat3).bmp	<i>Sharpen Edge</i>	Ada

n. *Rotation $\alpha = 0.0001$*

Tabel 19. Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Rotation $\alpha = 0.0001$*

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Rotation 20⁰CW</i>	Ada
2	baboon3(wat1).bm p	<i>Rotation 20⁰CW</i>	Ada
3	SydneyHarbourBri dge_f(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
4	kupu1(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
5	01221_mandownm eeeeedic(wat2).bm p	<i>Rotation 20⁰CW</i>	Ada
6	baboon3(wat2).bm p	<i>Rotation 20⁰CW</i>	Ada

7	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
8	kupu1(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
9	01221_mandownmeeeeedic(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
10	baboon3(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
12	kupu1(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
13	01221_mandownmeeeeedic(wat1).bmp	<i>Rotation 50⁰CW</i>	Ada
14	baboon3(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
16	kupu1(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
17	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
18	baboon3(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
20	kupu1(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
21	01221_mandownmeeeeedic(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
22	baboon3(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
24	kupu1(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
25	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada

26	baboon3(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
27	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
28	kupu1(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
29	01221_mandownmeeeeedic(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
30	baboon3(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
31	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
32	kupu1(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
33	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
34	baboon3(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
35	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
36	kupu1(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada

o. *Rotation* $\alpha = 0.0003$

Tabel 20. Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Rotation* $\alpha = 0.0003$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownmeeeeedic(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
2	baboon3(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
3	SydneyHarbourBridge_f(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
4	kupu1(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
5	01221_mandownm	<i>Rotation 20⁰CW</i>	Ada

	eeeeedic(wat2).bmp		
6	baboon3(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
8	kupu1(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
9	01221_mandownmeeeeeedic(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
10	baboon3(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
12	kupu1(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
13	01221_mandownmeeeeeedic(wat1).bmp	<i>Rotation 50⁰CW</i>	Ada
14	baboon3(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
16	kupu1(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
17	01221_mandownmeeeeeedic(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
18	baboon3(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
20	kupu1(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
21	01221_mandownmeeeeeedic(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
22	baboon3(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada

24	kupu1(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
25	01221_mandownm eeeeedic(wat2).bm p	<i>Rotation 50⁰CCW</i>	Ada
26	baboon3(wat2).bm p	<i>Rotation 50⁰CCW</i>	Ada
27	SydneyHarbourBri dge_f(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
28	kupu1(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
29	01221_mandownm eeeeedic(wat3).bm p	<i>Rotation 50⁰CCW</i>	Ada
30	baboon3(wat3).bm p	<i>Rotation 50⁰CCW</i>	Ada
31	SydneyHarbourBri dge_f(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
32	kupu1(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
33	01221_mandownm eeeeedic(wat2).bm p	<i>Rotation 50⁰CCW</i>	Ada
34	baboon3(wat2).bm p	<i>Rotation 50⁰CCW</i>	Ada
35	SydneyHarbourBri dge_f(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
36	kupu1(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada

p. *Rotation* $\alpha = 0.0006$

Tabel 21. Hasil Uji Ketahanan Citra Disisipi *Watermark* Apabila diberikan Serangan *Rotation* $\alpha = 0.0006$

No	Citra yang disisipi <i>watermark</i>	Perlakuan	Deteksi
1	01221_mandownm eeeeedic(wat1).bm p	<i>Rotation 20⁰CW</i>	Ada
2	baboon3(wat1).bm p	<i>Rotation 20⁰CW</i>	Ada

3	SydneyHarbourBridge_f(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
4	kupu1(wat1).bmp	<i>Rotation 20⁰CW</i>	Ada
5	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
6	baboon3(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
7	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
8	kupu1(wat2).bmp	<i>Rotation 20⁰CW</i>	Ada
9	01221_mandownmeeeeedic(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
10	baboon3(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
11	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
12	kupu1(wat3).bmp	<i>Rotation 20⁰CW</i>	Ada
13	01221_mandownmeeeeedic(wat1).bmp	<i>Rotation 50⁰CW</i>	Ada
14	baboon3(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
15	SydneyHarbourBridge_f(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
16	kupu1(wat1).bmp	<i>Rotation 20⁰CCW</i>	Ada
17	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
18	baboon3(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
19	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
20	kupu1(wat2).bmp	<i>Rotation 20⁰CCW</i>	Ada
21	01221_mandownmeeeeedic(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada

22	baboon3(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
23	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
24	kupu1(wat3).bmp	<i>Rotation 20⁰CCW</i>	Ada
25	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
26	baboon3(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
27	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
28	kupu1(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
29	01221_mandownmeeeeedic(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
30	baboon3(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
31	SydneyHarbourBridge_f(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
32	kupu1(wat3).bmp	<i>Rotation 50⁰CCW</i>	Ada
33	01221_mandownmeeeeedic(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
34	baboon3(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
35	SydneyHarbourBridge_f(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada
36	kupu1(wat2).bmp	<i>Rotation 50⁰CCW</i>	Ada

Hasil Pendeteksian *Watermark Nonperceptibility* dengan Menggunakan Metode *Discrete Waveket Transform*

Tabel 22. Tabel hasil citra disisipi *watermark* tidak terlihat (*Nonperceptibility*) untuk alpha (α) = 0,0001

Nama	Citra	Hasil
Roswati	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yunika	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Septianingsih	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√

	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ruri	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anisa Wima	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yuli	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√

	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anung	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Dani	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ainun	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-

	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ida	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-

Tabel 23. Tabel hasil citra yang disisipi *watermark* tidak terlihat (*Nonperceptibility*) untuk alpha (α) = 0,0003

Nama	Citra	Hasil
Roswati	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yunika	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√

	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Septianingsih	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ruri	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anisa Wima	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√

	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yuli	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anung	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Dani	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√

	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ainun	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ida	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-

Tabel 24. Tabel hasil citra yang dsisipi *watermark* tidak terlihat (*Nonperceptibility*) untuk alpha (α) = 0,0006

Nama	Citra	Hasil
Roswati	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√

	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yunika	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Septianingsih	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ruri	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√

	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anisa Wima	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Yuli	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Anung	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√

	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Dani	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ainun	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√
	01221_mandownmeeeeedic(wat3)	-
	baboon3(wat3)	-
	kupu1(wat3)	-
	SydneyHarbourBridge_f(wat3)	-
Ida	01221_mandownmeeeeedic(wat1)	√
	baboon3(wat1)	√
	kupu1(wat1)	√
	SydneyHarbourBridge_f(wat1)	√
	01221_mandownmeeeeedic(wat2)	√
	baboon3(wat2)	√
	kupu1(wat2)	√
	SydneyHarbourBridge_f(wat2)	√

01221_mandownmeeeeeedic(wat3)	-
baboon3(wat3)	-
kupu1(wat3)	-
SydneyHarbourBridge_f(wat3)	-

UNIVERSITAS BRAWIJAYA

