DIGITAL WATERMARKING PADA CITRA DALAM SEBARAN FREKUENSI MENGGUNAKAN HAAR WAVELET TRANSFORM

SKRIPSI

Oleh:

EDWIN IBNU KAUTSAR 0410960019-96



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

DIGITAL WATERMARKING PADA CITRA DALAM SEBARAN FREKUENSI MENGGUNAKAN HAAR WAVELET TRANSFORM

SKRIPSI

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

Oleh:

EDWIN IBNU KAUTSAR 0410960019-96



PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2010



LEMBAR PENGESAHAN SKRIPSI

DIGITAL WATERMARKING PADA CITRA DALAM SEBARAN FREKUENSI MENGGUNAKAN HAAR WAVELET TRANSFORM

Oleh : EDWIN IBNU KAUTSAR 0410960019-96

Setelah dipertahankan di depan Majelis Penguji pada tanggal 6 Januari 2011 dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,

Pembimbing II,

Edy Santoso, S.Si., M.Kom NIP, 197404142003121004 Dian Eka R., S.Si., M.Kom NIP. 197306192002122001

Mengetahui, Ketua Jurusan Matematika Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc NIP. 196908071994121001



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Edwin Ibnu Kautsar
NIM : 0410960019-96
Jurusan : Matematika
Program Studi : Ilmu Komputer

Penulis Skripsi beriudul : Digital Watermarking Pada

Citra Dalam Sebaran Frekuensi Menggunakan Haar Wavelet

Transform

Dengan ini menyatakan bahwa:

- 1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain namanama 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, 6 Januari 2011 Yang menyatakan,

Edwin Ibnu Kautsar NIM. 0410960019-96

ERSITAS BRAWN

DIGITAL WATERMARKING PADA CITRA DALAM SEBARAN FREKUENSI MENGGUNAKAN HAAR WAVELET TRANSFORM

ABSTRAK

Perkembangan teknologi citra digital telah berkembang pesat baik dari segi kualitas maupun ketersediaan sejak kamera digital pertama kali diperkenalkan pada Februari 1994. Salah satu keunggulan teknologi citra digital dalam hal ketahanan dibandingkan dengan teknologi citra analog telah banyak dimanfaatkan oleh sebagian pihak yang tidak bertanggung jawab untuk kepentingan sendiri tanpa memperhatikan aspek hak cipta (*Intellectual Property Right*). Salah satunya adalah maraknya penggandaan citra digital yang dilakukan oleh pihak-pihak yang tidak mempunyai hak atas konten tertentu membuat kekhawatiran yang besar bagi pemilik citra digital.

Watermark merupakan salah satu solusi untuk melindungi hak cipta atas citra digital yang dihasilkan. Dengan diterapkannya *Digital Image Watermarking* ini maka hak cipta citra digital yang dihasilkan akan terlindungi dengan cara menyisipkan informasi ke dalam citra digital tersebut. *Haar Wavelet Transform (HWT)* merupakan salah satu kakas yang banyak digunakan dalam teknik *watermarking* dalam *domain transform*.

Penyisipan *watermark* ke dalam citra digital menggunakan HWT ini dibangun menggunakan Delphi 7. Pemberian beberapa jenis serangan terhadap citra ber*watermark* telah mengubah *watermark* yang disisipkan. Serangan yang mengubah watermark tersebut antara lain filterisasi, rotasi, perubahan format dan *cropping*.

Kata kunci : citra digital, watermarking, Haar Wavelet Transform (HWT)

ERSITAS BRAWIUM viii

DIGITAL IMAGE WATERMARKING IN FREQUENCY DISTRIBUTION USING HAAR WAVELET TRANSFORM

ABSTRACT

Digital image technology has developed rapidly both in quality as well as availability since it was introduced on February 1994. One of the digital image technology superiority compared to analog image technology, its endurance, has been utilized by irresponsible party for their own interest without considering Intellectual Property Right. The booming of digital image reduplication done by paties who do not have the right of particular content is an illustration which causing a great worry to the digital image owner.

Watermark becomes one of the solution to protect the Intellectual Property Right of the digital image produced. By applying Digital Image Watermarking, the digital image intellectual property right will be protected by inserting information into the digital image. One of the most used technique of watermarking in transform domain is Haar Wavelet Transform (HWT).

The watermark inserting into the digital image using HWT is built using Delphi 7. The extending of some standard attack to watermarked image have changed the inserted watermark. The standard attack which change the watermark namely filteritation, rotation, format change and cropping.

Keyword : digital image, watermarking, Haar Wavelet Transform (HWT)

JERSITAS BRAWIUM

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, yang dengan rahmat dan karunia-Nya penulis dapat menyelesaikan Skripsi ini. Skripsi yang berjudul "Digital Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar Wavelet Transform" ini dibuat sebagai syarat kelulusan tingkat sarjana.

Selama melaksanakan Skripsi ini, penulis mendapat bantuan dan dukungan dari banyak pihak. Untuk itu, penulis ingin memberikan terima kasih kepada:

- 1. Edy Santoso, S.Si., M.Kom., selaku pembimbing utama dalam penulisan tugas akhir.
- 2. Dian Eka Ratnawati, S.Si., M.Kom., selaku pembimbing pendamping dalam penulisan tugas akhir.
- 3. Drs. Marji, MT selaku dosen penasehat akademik dan Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
- 4. Dr. Agus Suryanto, M.Sc, selaku Ketua Jurusan Matematika, FMIPA Universitas Brawijaya.
- 5. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
- 6. Ayah dan ibu penulis, yang mendukung serta mencurahkan kasih sayangnya.
- 7. Rekan-rekan di Program Studi Ilmu Komputer FMIPA Universitas Brawijaya yang telah memberikan bantuannya.
- 8. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat penulis sebutkan satu per satu.

Semoga Allah SWT membalas budi baik semua pihak dengan rahmat-Nya yang meliputi segala sesuatu.

Akhir kata, penulis menyadari bahwa Skripsi ini bukanlah tanpa kelemahan, untuk itu kritik dan saran sangat diharapkan.

Malang, Januari 2011

Penulis

ERSITAS BRAWA xii

DAFTAR ISI

	i
Lembar Pengesahan	iii
Lembar Pernyataan	v
Abstrak	vii
Abstract	ix
Kata Pengantar	xi
Kata Pengantar	xvi
Daftar Gambar	xvi
Daftar Tabel	xix
Daftar Sourcecode	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian1.6 Metodologi Penelitian	4
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	7
2.1 Citra Digital	7
2.1.1 Ciri Suatu Citra	8
2.1.2 Warna dan Tata Warna	9
2.2 Model Warna RGB	9
2.3 Transformasi Wavelet (Wavelet Transform)	10
2.3.1 Haar Wavelet Transform (HWT)	11
2.3.1.1 Dekomposisi Koefisien Haar Wavelet	16
2.3.1.2 Rekonstruksi Koefisien Haar Wavelet	16
2.3.1.3 Transformasi <i>Haar Wavelet</i> Pada Citra	17
2.4 Digital Image Watermarking	19
2.4.1 Karakteristik Digital Watermarking	19
2.4.2 Prinsip Dasar Watermarking	20
2.4.3 Serangan terhadap citra ber-watermark	21
2.5 Format Citra Bitmap	
2.6 Means Sauare Error (MSE)	26

BAB III METODOLOGI DAN PERANCANGAN	27
3.1 Analisis Perangkat Lunak	28
3.1.1 Deskripsi umum perangkat lunak	28
3.1.2 Batasan Perangkat Lunak	28
3.2 Perancangan Perangkat Lunak	30
3.2.1 Perancangan Proses Input Penyisipan Watermark	31
3.2.1.1 Haar Wavelet Transform	31
3.2.1.2 Proses Penyisipan Watermark	32
3.2.1.2 Proses Penyisipan Watermark	37
3.2.2 Proses Pendeteksian Watermark	38
3.2.2.1 Proses Recovery Watermark	42
3.3 Contoh Perhitungan	44
3.3.1 Proses Dekomposisi Citra	44
3.3.2 Proses Rekonstruksi Citra	48
3.4 Perancangan Antarmuka3.5 Perancangan Pengujian	49
3.5 Perancangan Pengujian	51
3.5.1 Perangkat Lunak dalam Pengujian	51
3.5.2 Bahan Pengujian	51
3.5.3 Pengujian Kebenaran Perangkat Lunak	51
3.5.4 Pengujian Kinerja Perangkat Lunak	52
3.5.5 Pengujian Ketahanan Citra Berwatermark	52
BAB IV IMPLEMENTASI DAN PEMBAHASAN	53
4.1 Lingkungan Implementasi	53
4.1.1 Lingkungan Implementasi Perangkat Keras	53
4.1.2 Lingkungan Implementasi Perangkat Lunak	53
4.2 Implementasi Perangkat Lunak	53
4.2.1 Struktur Data	54
4.2.2 Implementasi Watermarking	55
4.2.2.1 Haar Wavelet Transform	55
4.2.2.2 Proses Penyisipan Watermark	57
4.2.2.3 Proses Rekonstruksi Citra Berwatermark	58
4.2.3 Implementasi Pendeteksian <i>Watermark</i>	60
4.2.3.1 Proses Pengambilan Watermark	61
4.2.4 Implementasi MSE (Means Square Error)	62
4.3 Implementasi Antarmuka (Interface)	62
4.4 Implementasi Uji Coba	64
4.4.1 Skenario Pengujian	64
4.5 Penguijan Dan Analisis Hasil	65

4.5.1 Hasil Pengujian & Analisis Kebenaran Perangkat Lunak.	66
4.5.2 Hasil Pengujian & Analisis Kinerja Perangkat Lunak	66
4.5.3 Hasil Pengujian & Analisis Ketahanan Citra	
Berwatermark	73
4.6.4 Analisis Umum Hasil Uji	76
BAB V KESIMPULAN DAN SARAN	79
5.1 Kesimpulan	79
5.2 Saran	79
1 2511- 1141	
DAFTAR PUSTAKA	81



ERSITAS BRAWN xvi

DAFTAR GAMBAR

Gambar 2.1 Representasi spatial citra–pixel	8
Gambar 2.2 Skema kubus tiga warna	10
Gambar 2.3 Keluarga wavelet	11
Gambar 2.4 Haar Scaling Function dan Haar wavelet Function	12
Gambar 2.5 Fungsi ϕ (2 ^j t-i)	13
Gambar 2.6 Fungsi Ψ (2 ² t-i)	14
Gambar 2.7 Hubungan antara ruang funsi scalling dan wavelet	14
Gambar 2.8 Piramida Mallat untuk dekomposisi	16
Gambar 2.9 Matriks L dan H	16
Gambar 2.10 Piramida Mallat untuk rekonstruksi	17
Gambar 2.11 Matriks L* dan H*	17
Gambar 2.12 Algoritma Mallat untuk dekomposisi citra	18
Gambar 2.13 (a) citra asli, (b) dekomposisi level, (c)spektrum &	
penataan komponen	18
Gambar 2.14 Proses encoder watermark secara umum	21
Gambar 2.15 Proses decoder watermark secara umum	21
Gambar 2.16 Bentuk Matriks berukuran N x M	23
Gambar 2.17 Format berkas bitmap	24
Gambar 2.18 Format citra 8-bit (256 warna)	25
Gambar 2.19 Format citra 24-bit (16,7 juta warna)	26
Gambar 3.1 Diagram Alir Perancangan Perangkat Lunak	27
Gambar 3.2 Flowchart proses penyisipan Watermark	29
Gambar 3.3 Flowchart proses pendeteksian Watermark	30
Gambar 3.4 Flowchart proses Haar Wavelet Transfrom	32
Gambar 3.5 Struktur piramida Haar Wavelet Transform	33
Gambar 3.6 Flowchart penyisipan watermark (\omega)	34
Gambar 3.7 Flowchart pencarian koefisien terbesar matrik A	35
Gambar 3.8 Flowchart penyisipan watermark	36
Gambar 3.9 Flowchart Inverse Haar Wavelet Transform	38
Gambar 3.10 Flowchart proses pendeteksian watermark	39
Gambar 3.11 Flowchart proses cari koefisien pada rentang LH / I	HL
yang lebih besar atau sama dengan key	40
Gambar 3.12 Flowchart proses mengambil watermark dari koefis	ien
LH / HL terpilih	41
Gambar 3.13 Flowchart merubah binary watermark menjadi strin	g/
karakter	43

Gambar 3.14 Nilai Citra berukuran 8x8	44
Gambar 3.15 Filter Dekomposisi LH	44
Gambar 3.16 Perkalian kolom 1 dengan filter LH	45
Gambar 3.17 Matriks D2	45
Gambar 3.18 Matriks D3	46
Gambar 3.19 Perkalian Baris 1 dengan matriks LH	46
Gambar 3.20 Matriks D4	46
Gambar 3.21 Matriks D5	47
Gambar 3.22 Matrik dekomposisi 1 tingkat	47
Gambar 3.23 Matrik dekomposisi 1 tingkat	48
Gambar 3.24 Perkalian Baris 1 dengan filter L* dan H*	48
Gambar 3.25 Rancangan form utama	49
Gambar 3.26 Rancangan menu Embedding Watermark	50
Gambar 3.27 Rancangan menu Recovery Watermark	50
Gambar 4.1 Struktur Data	55
Gambar 4.2 Gambar citra setelah proses dekomposisi (Citra	
Transformasi)	59
Gambar 4.3 Antarmuka Utama	64
Gambar 4.4 Antarmuka menu penyisipan watermark serta tampil	an
citra setelah disisipi watermark	64
Gambar 4.5 Antarmuka menu pendeteksian watermark serta tamp	oilan
watermark yang terdeteksi	65
Gambar 4.6 Grafik rata-rata uji kinerja perangkat lunak	80
Gambar 4.7 Grafik prosentase gallat/error	81

DAFTAR TABEL

Tabel 3.1 Pengujian kebenaran perangkat lunak	.51
Tabel 3.2 Pengujian kinerja perangkat lunak	.52
Tabel 4.1 Daftar Struktur Data	.56
Tabel 4.2 Daftar Berkas Citra	.66
Tabel 4.3 Daftar Teks Untuk Penelitian Watermarking	.68
Tabel 4.4 Hasil pengujian Pengujian Kebenaran Perangkat Lunak	.68
Tabel 4.5 Hasil pengujian kinerja perangkat lunak dengan teks:	
Edwin Ibnu Kautsar	.71
Tabel 4.6 Hasil pengujian kinerja perangkat lunak dengan teks:	
Program Studi Ilmu Komputer FMIPAUniversitas	
Brawijaya	.72
Tabel 4.7 Hasil pengujian kinerja perangkat lunak dengan teks :	
Digital Image Watermarking Pada Citra Dalam Sebaran	
Frekuensi Menggunakan Haar wavelet transform	.74
Tabel 4.8 Hasil Pengujian Ketahanan Citra Berwatermark	.77

ERSITAS BRAWNURLA XX

DAFTAR SOURCECODE

Sourcecode 4.1 Potongan kode proses dekomposisi citra asli	58
Sourcecode 4.2 Potongan kode proses penyisipan Watermark	59
Sourcecode 4.3 Potongan kode proses rekonstruksi citra	
berwatermark	61
Sourcecode 4.4 Potongan kode proses pengambilan watermark.	63
Sourcecode 4.5 Prosedur proses penghitungan MSE	63



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi citra digital saat ini telah berkembang pesat baik dari segi kualitas maupun ketersediaan sejak kamera digital pertama kali diperkenalkan pada Februari 1994. Seiring dengan perkembangan diatas, teknologi analog pada pencitraan khususnya penggunaan bahan seluloid mulai ditinggalkan. Sesuai dengan salah satu sifat dari bahan seluloid yang mahal dalam pemrosesan dan penyimpanannya, yang berkebalikan dengan sifat citra digital yang murah dan mudah diduplikasi dan disimpan, membuat teknologi digital saat ini marak digunakan. Salah satunya dalam hal ketahanan, teknologi analog lebih banyak beresiko daripada teknologi digital. Keunggulan teknologi digital dalam hal ketahanan inilah yang akhirnya dimanfaatkan oleh sebagian pihak yang tidak bertanggung jawab untuk kepentingannya sendiri tanpa memperhatikan aspek hak cipta (Intellectual Property Right). Salah satunya adalah maraknya penggandaan citra digital yang dilakukan oleh pihak-pihak yang tidak mempunyai hak atas konten tertentu, membuat kekhawatiran yang besar bagi pemilik citra digital. Oleh karena itu seiring dengan perkembangannya, kebutuhan pengamanan data digital sangatlah penting (Rummi, 2003).

Salah satu solusi pengamanan citra digital adalah Watermarking. Watermarking merupakan salah satu bentuk dari steganography, yang dapat diartikan sebagai suatu teknik penyembunyian data atau informasi lain kedalam suatu data lainnya untuk "ditumpangi" (kadang disebut host data). Orang lain tidak menyadari kehadiran adanya data tambahan pada data host-nya. Jadi seolah-olah tidak ada perbedaan signifikan antara data host sebelum dan sesudah proses watermarking (Suhono, 2000).

Pada umumnya, setelah data di-watermark, maka data asli tidak disebarluaskan, data yang disebarluaskan adalah data yang telah diberi watermark. Dalam permasalahan ini, citra yang telah ber-watermark apabila telah tersebar luas tidak menutup kemungkinan akan digunakan oleh pihak-pihak yang tidak bertanggung jawab dan dilakukan beberapa bentuk sabotase atau perubahan, misalnya perbaikan sinyal seperti penajaman citra atau operasi geometri seperti rotasi, translasi, cropping dan scaling.

Walaupun data yang telah mengalami perubahan sudah tidak persis sama seperti data ber-watermark, namun kedua data tersebut memiliki kemiripan tertentu apabila dihitung secara statistik maka dapat dikatakan citra tersebut telah dipergunakan tanpa izin pemilik.

Digital Image Watermarking sendiri memiliki beberapa jenis teknik yang memiliki keunggulan dan kelemahan masing-masing. Pada umumnya, watermarking yang kokoh memiliki kualitas citra ber-watermark yang kurang memuaskan, demikian juga sebaliknya. Teknik penyembunyian data sendiri terbagi menjadi dua domain, yaitu domain spasial dan domain frekuensi.

Haar Wavelet Transfrom (HWT) yang merupakan salah satu dari sekian banyak keluarga Discrete Wavelet Transform (DWT) adalah metode yang banyak digunakan dalam watermarking yang menggunakan domain frekuensi. Metode ini berbasis wavelet, suatu pendekatan yang populer karena kekuatannya melawan malicious attack. Secara umum penyisipan watermark ke dalam citra dilakukan dengan cara membandingkan koefisien HWT dari dekomposisi citra, dimana koefisien yang memiliki nilai terbesar adalah tempat paling signifikan untuk menyisipkan watermark. Beberapa keunggulan dimiliki HWT dalam watermarking diantaranya HWT merupakan metode yang paling dekat dengan HVS (Human Visual System), distorsi yang rendah dalam domain wavelet apabila dilakukan kompresi tinggi dibandingkan dengan domain lain dengan bit rate yang sama, dan bit error rate yang rendah membuat HWT merupakan salah satu pilihan unggul dalam pemilihan metode watermarking.

Penelitian watermarking citra digital yang dilakukan pada skripsi ini, mengacu pada penelitian yang pernah dilakukan oleh Kamran Hameed, Adeel Mumtaz, and S.A.M. Gilani dengan judul "Digital Image Watermarking in the Wavelet Transform Domain" tahun 2006 dimana pada penelitian ini watermark yang disisipkan adalah citra biner. Sedangkan pada skripsi ini penulis mencoba menggunakan teks sebagai watermark yang akan disisipkan.

Berdasarkan pada penjelasan sebelumnya untuk mengatasi meningkatnya kebutuhan akan pengamanan citra digital, maka judul yang penulis ambil adalah "Digital Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar Wavelet Transform".

1.2 Rumusan Masalah

Dalam penulisan skripsi ini, data yang digunakan adalah citra digital dengan rumusan masalah yang dibahas adalah :

- 1. Bagaimana membuat perangkat lunak penyisipan dan pendeteksian *watermark* pada citra dimana *watermark* yang disisipkan tidak disadari oleh indera manusia.
- 2. Bagaimana proses pendeteksian watermark pada citra.
- 3. Bagaimana kualitas citra setelah diberikan watermark.
- 4. Bagaimana pengaruh *Standard attack* terhadap *watermark* pada citra.

1.3 Batasan Masalah

Agar tidak terjadinya kesalahan persepsi dan mencegah melebarnya pokok bahasan, maka ditentukan batasan-batasan masalah yang diantaranya sebagai berikut :

- 1. Citra yang diproses adalah citra berwarna (*true colour*) 24 bit, dengan resolusi 320x320, 480x480, dan 500x500 piksel.
- 2. Format citra yang digunakan adalah Bitmap (.bmp).
- 3. Penerapan metode berbasis frekuensi menggunakan *Haar Wavelet Transform* dan pengaruh rentang nilai frekuensi yang digunakan terhadap hasil *watermark*.
- 4. Data yang disisipkan ke dalam citra adalah *text*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yang akan dicapai adalah:

- 1. Membuat perangkat lunak penyisipan dan pendeteksian watermark pada citra dimana watermark yang disisipkan tidak disadari oleh indera manusia.
- 2. Melakukan proses pendeteksian *watermark* pada citra berwatermark dan merekonstruksi citra kembali.
- 3. Membandingkan kualitas citra ber-watermark dengan citra asli.
- 4. Menganalisa pengaruh *standard attack* terhadap *watermark* yang telah disisipkan ke citra.

1.5 Manfaat Penelitian

Manfaat yang bisa didapatkan dari penelitian ini adalah:

- 1. Menyediakan model perangkat lunak (software) untuk menyisipkan dan mendeteksi watermark pada citra menggunakan Haar Wavelet Transform.
- 2. Menyediakan referensi mengenai teknik watermarking menggunakan Haar Wavelet Transform bagi peneliti teknik watermarking yang lain.

1.6 Metodologi Penelitian

1. Studi Kepustakaan

Studi ini dilakukan dengan cara mencari sekaligus mempelajari beberapa literatur dan artikel mengenai *watermarking* dan pemrograman *file* citra. Disamping itu juga mempelajari program aplikasi yang sudah ada untuk memberikan gambaran yang jelas mengenai aplikasi *watermarking*, sebagai acuan dalam perencanaan dan pembuatan Tugas Akhir.

- Pendefinisan dan Analisis Masalah
 Mendefinisikan dan menganalisis permasalahan dan mencari
 solusi yang tepat.
- 3. Perancangan dan Implementasi Sistem Membuat rancangan model perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut yaitu membuat perangkat lunak pembuatan dan pendeteksian watermark pada citra menggunakan metode Haar Wavelet Transform.
- 4. Uji Coba dan Analisa Hasil Implementasi Menguji coba perangkat lunak tersebut dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievalusi dan disempurnakan.

1.1 Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi menjadi lima bab dengan masing-masing bab diuraikan sebagai berikut:

1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, metodologi pemecahan masalah, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai teori-teori penunjang yang membahas konsep dasar dari citra, serta beberapa macam representasinya, lalu pengaruh manipulasi citra terhadap suatu citra, serta beberapa teori statisik yang mendukung metode *watermark* ini.

3. BAB III METODE DAN PERANCANGAN

Bab ini membahas mengenai penggunaan teori-teori dalam perancangan perangkat lunak ini. Serta membahas bagaimana perangkat lunak ini direncanakan, dibuat dan diimplementasikan.

4. BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan implementasi sistem, serta analisa hasil yang dikeluarkan oleh perangkat lunak, untuk selanjutnya dilakukan analisa hasil untuk mengevaluasi kulaitas citra setelah mengalami *watermark* serta ketahanan *watermark* terhadap beberapa image distortion.

5. BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari proses pembuatan sampai proses pengimplementasian perangkat lunak serta untuk pengembangan perangkat lunak lebih lanjut.

WERSITAS BRAWN

BAB II TINJAUAN PUSTAKA

2.1 Citra Digital

Citra digital sebenarnya bukanlah sebuah data digital yang normal, melainkan sebuah representasi dari citra asal yang bersifat analog. Citra digital ditampilkan pada layar komputer dengan berbagai macam susunan warna dan tingkat kecerahan. Hal ini disebabkan karena susunan warna yang dimiliki dalam sebuah citra mengandung jumlah warna dan tingkat kecerahan yang tidak terbatas. Citra digital dibentuk dengan mengekstraksi informasi pada citra analog asli dan mengirimkannya ke komputer dalam bentuk biner (Anonimus, 2005). Proses ini disebut dengan digitasi. Digitasi dapat dilakukan oleh alat seperti kamera digital, *scanner*, atau *webcam*. Selain dapat mengambil atau menangkap sebuah citra, alatalat tersebut juga dapat bertindak sebagai alat input (masukan) bagi komputer.

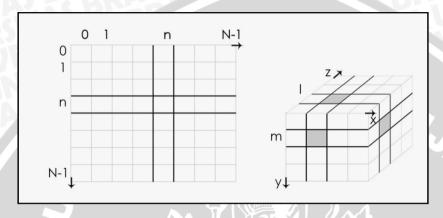
Citra dapat dinyatakan sebagai fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi yaitu f(x,y), x dan y menyatakan koordinat ruang, dan nilai f pada suatu koordinat (x,y) menyatakan tingkat kecerahan atau derajat keabuan $(brightness/gray\ level)$ dan informasi warna citra (Gultom, 2007). Secara matematis persamaan untuk fungsi intensitas f(x,y) ditunjukkan pada persamaan 2.1.

$$0 \le f(x, y) \le \infty \tag{2.1}$$

Citra digital adalah representasi citra dalam bentuk diskrit, baik pada koordinat ruang maupun intensitas cahayanya. Citra digital dapat direpresentasikan dengan suatu matriks $M \times N$ (baris dan kolom) yang menyatakan koordinat sebuah titik pada citra tersebut dan nilai masing-masing elemen menyatakan intensitas cahaya pada titik tersebut. Suatu titik pada sebuah citra digital sering disebut sebagai *image element, picture element,* atau *pixel*.

Representasi spatial dari citra berupa *pixel*, yang berupa koordinat dari tiap citra. Didalam tiap *pixel* mengandung informasi tentang tingkat *irradiance* atau tingkat kecerahan. Susunan *pixel* sama seperti susunan matriks, hanya saja, awal koordinat dimulai dari atas ke kanan (kolom) untuk *x* ke bawah (baris) untuk *y*. Sama seperti array, index *pixel* dimulai dari 0, untuk pixel yang berada

pada koordinat X bergerak dari 0 hingga *x-1*, begitu juga halnya dengan pixel pada koordinat Y bergerak dari 0 hingga *y-1* (Jähne, 2002).



Gambar 2.1 Representasi spatial citra – *pixel*. Gambar kiri untuk citra 2D sedangkan gambar kanan untuk citra 3D Sumber : Jähne, 2002

2.1.2 Ciri Suatu Citra

Ciri merupakan suatu tanda yang khas, yang membedakan antara satu dengan yang lain. Ciri – ciri dasar dari citra adalah :

1. Warna

• Ciri warna suatu citra dapat dinyatakan dalam bentuk histogram dari citra tersebut yang dituliskan dengan: H(r,g,b), dimana H(r,g,b) adalah jumlah munculnya pasangan warna r (red), g (green) dan b (blue) tertentu.

2. Bentuk

- Ciri bentuk suatu citra dapat ditentukan oleh tepi (sketsa), atau besaran momen dari suatu citra. Pemakaian besaran momen pada ciri bentuk ini banyak digunakan orang dengan memanfaatkan nilai-nilai transformasi fourier dari citra.
- Proses yang dapat digunakan untuk menentukan ciri bentuk adalah edge detection, threshold, segmentasi dan perhitungan momen seperti (mean, median dan standar deviasi dari setiap lokal citra).

3 Tekstur

- Ciri tekstur dari suatu citra dapat ditentukan dengan menggunakan filter.
- Ciri tekstur ini sangat handal dalam menentukan informasi suatu citra bila digabungkan dengan ciri warna citra (Djuhari, 2006).

2.1.3 Warna dan Tata warna

Warna pada dasarnya merupakan hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400nm sampai dengan 700nm (Lestari, 2003).

Tata warna atau yang sering juga disebut sebagai model warna merupakan sebuah cara atau metode untuk mengatur, membuat dan memvisualisasikan warna (Lestari, 2003). Untuk aplikasi yang berbeda tata warna yang dipakai bisa juga berbeda, hal ini dikarenakan beberapa peralatan tertentu memang membatasi secara ketat ukuran dan jenis tata warna yang dapat digunakan.

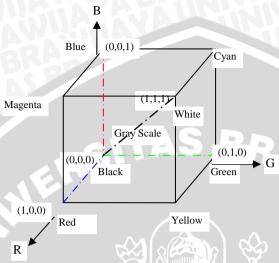
Beberapa model tata warna antara lain (Lestari, 2003):

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

2.2 Model Warna RGB

Model warna RGB adalah model warna yang digunakan pada sistem komputer. Secara fisika dalam dunia warna dikenal adanya warna primer. Dari kombinasi warna primer ini bisa menghasilkan warna-warna yang lain. Untuk cahaya warna primernya adalah merah, hijau, dan biru yang disingkat RGB (*Red,Green,Blue*). Untuk non cahaya memiliki warna primer yang berbeda.

Model RGB direpresentasikan dengan kubus tiga dimensi dengan merah, hijau dan biru pada setiap sudut masing-masing aksis (gambar 2.2). Warna hitam berada di pangkal (0,0,0), sedangkan warna putih terletak pada sudut yang berlawanan (1,1,1) dengan sudut hitam (Nugroho, 2008).



Gambar 2.2 Skema kubus tiga warna Sumber: Nugroho, 2008

Nilai RGB dalam format gambar 24 bit ditulis secara spesifik menggunakan tiga angka integer mulai 0 sampai 255, masing-masing merepresentasikan intensitas *red*, *green*, dan *blue*. Sebagai contoh:

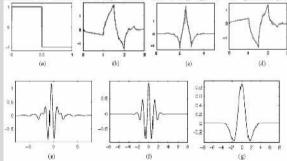
- (0, 0, 0) adalah hitam (black)
- (255, 255, 255) adalah putih (*white*)
- (255, 0, 0) adalah merah (*red*)
- (0, 255, 0) adalah hijau (*green*)
- (0, 0, 255) adalah biru (*blue*)
- (255, 255, 0) adalah kuning (*yellow*)
- (0, 255, 255) adalah *cyan*
- (255, 0, 255) adalah *magenta*

2.3 Transformasi Wavelet (Wavelet Transform)

Transformasi sinyal merupakan bentuk lain dari penggambaran sinyal yang tidak merubah isi informasi dalam sinyal tersebut. Transformasi *Wavelet (Wavelet Transform)* menyediakan penggambaran frekuensi dari sinyal. Pada awalnya, transformasi wavelet digunakan untuk menganalisis sinyal bergerak (nonstationary signals). Sinyal bergerak ini dianalisis dalam transformasi

wavelet dengan menggunakan teknik multi-resolution analysis. Secara umum teknik multi-resolution analysis adalah teknik yang digunakan untuk menganalisis frekuensi dengan cara menganalisa frekuensi yang berbeda dengan resolusi yang berbeda pula. Resolusi dari sinyal merupakan ukuran jumlah informasi di dalam sinyal yang dapat berubah melalui operasi filterisasi (Polikar, 1998).

Dalam pengembangannya, transformasi wavelet memiliki dua seri, yaitu Continous Wavelet Transform (CWT) dan Discrete Wavelet Transform (DWT). Semua fungsi yang digunakan dalam tranformasi CWT dan DWT diturunkan dari mother wavelet melalui translasi (pergeseran) dan penskalaan (kompresi). Mother wavelet merupakan fungsi dasar yang digunakan dalam transformasi wavelet (Sripati, 2003). Karena mother wavelet menghasilkan semua fungsi wavelet yang digunakan dalam transformasi melalui translasi dan mother wavelet juga penskalaan, maka akan menentukan karakteristik dari transformasi wavelet yang dihasilkan. Perlu pencatatan secara teliti terhadap penerapan wavelet dan pemilihan yang tepat terhadap mother wavelet harus dilakukan agar dapat menggunakan taransformasi wavelet secara efisien. Fungsi-fungsi yang termasuk dalam keluarga wavelet dipaparkan pada gambar 2.3.



 $\begin{array}{l} \textbf{Gambar 2.3} \ \ \text{Keluarga} \ \textit{wavelet} \ (a) \ \ \text{Haar, (b) Daubechies, (c) Coiflet,} \\ \text{(d) Symlet, (e) Meyer, (f) Morlet, (g) Mexican Hat. Dengan sumbu x} \\ \text{merupakan waktu dan sumbu y} \ \psi \ (t) \end{array}$

Sumber: Sripati, 2003

2.3.1 Haar Wavelet Transform (HWT)

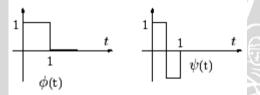
Transformasi *wavelet* dalam konteks pemrosesan sinyal adalah suatu metode yang digunakan untuk mendekomposisi suatu sinyal

masukan ke dalam bentuk suatu gelombang yang disebut *wavelet* induk, dan menganalisis sinyal dengan memberi perlakuan terhadap koefisien *wavelet*.(Walidainy, 2004)

Haar wavelet transform adalah salah satu jenis transformasi wavelet yang paling lama dikenal. Fungsi Haar sudah digunakan sejak 1910 oleh matematikawan Hungaria bernama Alfred Haar. Haar Transform merupakan transformasi simetris dengan ukuran data N=2ⁿ. (Porwik, 2004)

Wavelet merupakan sebuah basis. Basis wavelet berasal dari sebuah fungsi penskalaan (scaling function). Scaling function memiliki sifat dapat disusun dari sejumlah salinan dirinya yang telah didilasikan, ditranslasikan, dan diskalakan. Fungsi ini diturunkan dari persamaan dilasi (dilation equation), yang dianggap sebagai dasar dari teori wavelet. Dari persamaan scaling function dapat dibentuk persamaan wavelet yang pertama (atau disebut juga mother wavelet). Dari mother wavelet dapat dibentuk wavelet-wavelet berikutnya (ψ^1 , ψ^2 dan seterusnya) dengan cara mendilasikan (memampatkan atau meregangkan) dan menggeser mother wavelet. (Murni, 2003)

Haar transform menggunakan Haar scaling function dan Haar wavelet function yang ditunjukkan pada gambar 2.4.



Gambar 2.4. *Haar scaling function* dan *Haar wavelet function* Sumber: Salomon, 2004

a. Haar scaling function

Fungsi dilasi Haar (scaling) didefinisikan dari rumus (Porwik, 2004):

$$\Phi(t) = \begin{cases} 1, & \text{for } t \in [0,1), \\ 0, & \text{for } t \notin [0,1). \end{cases}$$
(2.2)

Untuk sembarang fungsi Haar, maka akan menghasilkan fungsi:

$$\Phi_{i}^{j}(t) = \sqrt{2^{j}} \Phi(2^{j}t - i)$$
 (2.3)

Untuk
$$i = 0,1,...,2^{j}-1$$

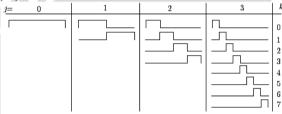
 $j = 0,1,..., log_2N-1$

Konstanta $\sqrt{2^j}$ dipilih agar hasil skalar. $\langle \Phi_i^j, \Phi_i^j \rangle = 1, \Phi_i^j \in L^2(R)$.

Indek j menyatakan dilasi dan indek i menyatakan translasi. Sehingga didapatkan rentang linier ruang vektor \mathbf{V}^{j} :

$$V^{j} = span \left\{ \Phi_{i}^{j} \right\}_{i=0,\dots,2^{j}-1}$$
 (2.4)

Pada gambar 2.5 diberikan contoh fungsi $\Phi(2^jt-i)$ dengan j= 0,1,2,3 dan i= 0,1,...,7.



Gambar 2.5. Fungsi Φ(2^jt-i) Sumber: Salomon, 2004

b. Haar wavelet function

Fungsi wavelet Haar didefinisikan dari rumus (Porwik, 2004):

$$\Psi(t) = \begin{cases}
1, & \text{for } t \in [0, \frac{1}{2}), \\
-1, & \text{for } t \in [\frac{1}{2}, 1), \\
0, & \text{otherwise.}
\end{cases}$$
(2.5)

Untuk sembarang fungsi Haar, maka akan menghasilkan fungsi:

$$\Psi_i^j(t) = \sqrt{2^j} \Psi(2^j t - i)$$
 (2.6)

Untuk
$$i = 0,1,...,2^{j}-1$$

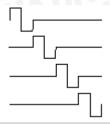
 $j = 0,1,..., log_2N-1$

Konstanta $\sqrt{2^j}$ dipilih agar hasil skalar $\langle \Psi_i^j, \Psi_i^j \rangle = 1, \Psi_i^j \in L^2(R)$.

Indek j menyatakan dilasi dan indek i menyatakan translasi. Sehingga didapatkan rentang linier ruang vektor W^j:

$$W^{j} = span\{\Psi_{i}^{j}\}_{i=0, 2^{j}-1}$$
 (2.7)

Sebagai contoh pada gambar 2.6 ditunjukkan 4 *Haar wavelet* $\Psi(2^2t-i)$ dengan i=0,1,2,3.

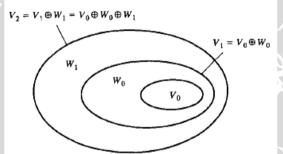


Gambar 2.6. Fungsi $\Psi(2^2t-i)$ Sumber: Salomon, 2004

Dalam multiresolusi analisis, basis *Haar* memiliki sifat yang sangat penting (Sandberg, 2000):

$$V^{j+1} = V^j \oplus W^j \tag{2.8}$$

Misalnya $V^3=V^2 \oplus W^2$. Hal ini dapat dijelaskan pada gambar 2.7:



Gambar 2.7. Hubungan antara ruang fungsi *scaling* dan *wavelet* Sumber: Gonzales, 2002

Haar wavelet bekerja pada data dengan metode transformasi vektor yang disebut averaging dan differencing. Hasil dari operasi averaging disebut approximation coefficients. Hasil dari operasi differencing disebut detail coefficients. (Ames, 2002)

Berdasarkan persamaan 2.3 dan 2.6 mengenai fungsi Haar dan fakta bahwa V^j dan W^j merentang seperti pada persamaan 2.4 dan 2.7. Maka fungsi Φ dan fungsi Ψ dapat dinyatakan sebagai kombinasi linier fungsi basis dari ruang V dan W berturut-turut. Dalam *Haar wavelet*, fungsi Φ dan Ψ dapat ditulis (Porwik, 2004):

$$\Phi(t) = h(0) \sqrt{2} \Phi(2t) + h(1) \sqrt{2} \Phi(2t-1)$$
 (2.9)

$$\Psi(t) = g(0) \sqrt{2} \Phi(2t) + g(1) \sqrt{2} \Phi(2t-1)$$
 (2.10)

Dimana $\{h(0),h(1)\}\$ dan $\{g(0),g(1)\}\$ mendefinisikan tapis(filter) lowpass dan high-pass berturut-turut. Dalam hal ini, h(k), k=0,1 koefisiennya diketahui (Porwik, 2004):

$$h(0) = 1/2, h(1) = 1/2, g(0) = 1/2, g(1) = -1/2$$
 (2.11)

jika C menyatakan matrik citra, maka:

$$A(i) = 1/2 C(2i) + 1/2 C(2i+1)$$
(2.12)

$$A(i) = 1/2 C(2i) + 1/2 C(2i+1)$$
 (2.12)
 $D(i) = 1/2 C(2i) - 1/2 C(2i+1)$ (2.13)

Dimana C(i) = vektor berukuran N, berisi baris atau kolom matrik C, $i \in \{0,1,...,N/2 - 1\}$, A(i) = vektor berukuran N/2, berisi approximation coefficients, D(i) = vektor berukuran N/2, berisi detail coefficients. (Porwik, 2004)

rekonstruksi citra (Invers Haar Wavelet Transform) dinyatakan sebagai berikut (Porwik, 2004):

$$C(2i) = 1/2 A(i) + 1/2 D(i)$$
 (2.14)

$$C(2i+1) = 1/2 A(i) - 1/2 D(i)$$
 (2.15)

Hal tersebut juga dapat dijelaskan melalui perkalian matrik. Berdasarkan persamaan 2.8, maka dapat dibentuk matrik wavelet sebagai berikut. Untuk langkah ke 1, persamaan Vⁿ = Vⁿ⁻¹ \oplus Wⁿ⁻¹. matrik wavelet M₁ berbentuk:

$$\mathbf{M}_1 = [\; \Phi^{n-2}_{j=0,\dots,2^{n-2}-1} \subset V^{n-1}, \; \Psi^{n-1}_{j=0,\dots,2^{n-1}-1} \subset W^{n-1} \; \;]^T$$

Untuk langkah ke 2, persamaan $V^{n-1} = V^{n-2} \oplus W^{n-2} \oplus W^{n-1}$, matrik wavelet M₂ berbentuk:

$$\mathbf{M}_{2} = [\Phi_{j=0,\dots,2^{n-2}-1}^{n-2} \subset V^{n-2}, \Psi_{j=0,\dots,2^{n-2}-1}^{n-2} \subset W^{n-2}, \Psi_{j=0,\dots,2^{n-1}-1}^{n-1} \subset W^{n-1}]^{\mathrm{T}}$$

Untuk langkah ke n, persamaan $V^1 = V^0 \oplus W^0 \oplus W^1 \oplus W^2 \oplus ... \oplus W^{n-1}$ matrik wavelet M_n berbentuk (Porwik, 2004):

$$\mathbf{M}_{\mathbf{n}} = [\Phi_{0}^{0} \subset V^{0}, \Psi_{0}^{0} \subset W^{0}, \Psi_{j=0,1}^{1} \subset W^{1} ... \Psi_{j=0,...2^{n-1}-1}^{n-1} \subset W^{n-1}]^{\mathsf{T}}$$

Jika M adalah matrik wavelet dan I adalah matrik citra asli, maka matrik transformasi T didapatkan dengan persamaan (Ames, 2002):

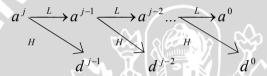
$$T = M^{T}. I. M$$
 (2.17)

Sedangkan untuk penyusunan/rekonstruksi citranya menggunakan persamaan(Ames, 2002):

 $I = (M^{-1})^{T}$. I. M^{-1} (2.18)

2.3.1.1 Dekomposisi Koefisien Haar Wavelet

Stephane Mallat memperkenalkan cara mudah menghitung koefisien *Haar Wavelet* yang dikenal dengan algoritma piramida Mallat. Algoritma tersebut ditunjukkan dengan gambar 2.8. (Murni, 2003)



Gambar 2.8. Piramida Mallat untuk dekomposisi Sumber : Murni, 2003

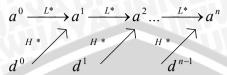
Dengan a^j adalah vektor awal dengan ukuran 2^j, dan koefisien *Haar Wavelet* dapat diperoleh dari aproksimasi a⁰ detail-detail d⁰, d¹ dan seterusnya. Dari Gambar 2.8 terlihat bahwa vektor awal dilewatkan filter *lowpass* (L) dan *highpass* (H) untuk mendapatkan nilai aproksimasi dan detail pada level selanjutnya. Matriks L dan H masing-masing adalah matriks *lowpass* (*averaging*) dan *highpass* (*differencing*) yang koefisiennya didefinisikan pada persamaan 2.11. Matrik LH ditunjukkan pada gambar 2.9.

$$L = \begin{bmatrix} \frac{1}{\sqrt{\sqrt{2}}} & \frac{1}{\sqrt{\sqrt{2}}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{\sqrt{2}}} & \frac{1}{\sqrt{\sqrt{2}}} \end{bmatrix} \qquad H = \begin{bmatrix} \frac{1}{\sqrt{\sqrt{2}}} & -\frac{1}{\sqrt{\sqrt{2}}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{\sqrt{2}}} & -\frac{1}{\sqrt{\sqrt{2}}} \end{bmatrix}$$

Gambar 2.9. Matriks L dan H Sumber : Murni, 2003

2.3.1.2 Rekonstruksi Koefisien Haar Wavelet

Proses kebalikan dari dekomposisi ini disebut proses rekonstruksi, dan algoritma piramida Mallat untuk rekonstruksi ditunjukan pada gambar 2.10.



Gambar 2.10. Piramida Mallat untuk rekonstruksi Sumber: Murni, 2003

Matriks pengali L* dan H* memiliki hubungan dengan matriks L dan H pada proses dekomposisi yaitu LL* = HH* = I (L merupakan invers dari L*). Isi dari L* dan H* untuk basis Haar ditunjukkan pada gambar 2.11. (Murni, 2003)

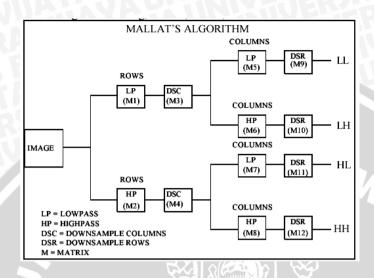
$$L^* = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \qquad H^* = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Gambar 2.11. Matriks L* dan H* Sumber: Murni, 2003

2.3.1.3 Transformasi *Haar Wavelet* Pada Citra

Suatu citra merupakan matriks dua dimensi. Sehingga dapat dilakukan transformasi terhadap baris-baris pada citra, dan dilanjutkan dengan transformasi terhadap kolom-kolom pada citra, seperti pada Gambar 2.12.

Sinyal pertama-tama dilewatkan pada rangkain filter *high-pass* dan *low-pass*, kemudian setengah dari masing-masing keluaran diambil sebagai *sample* melalui operasi *sub-sampling*. Proses ini disebut sebagai proses dekomposisi satu tingkat. Keluaran dari filter *low-pass* digunakan sebagai masukkan di proses dekomposisi tingkat berikutnya. Proses ini diulang sampai tingkat proses dekomposisi yang diinginkan. Gabungan dari keluaran-keluaran filter *high-pass* dan satu keluaran filter *low-pass* yang terakhir, disebut sebagai koefisien *wavelet*, yang berisi informasi sinyal hasil transformasi (Walidainy, 2004). Gambar 2.13 adalah contoh dekomposisi pada citra.



HL HH

LL: hasil lowpass terhadap baris dan kolom

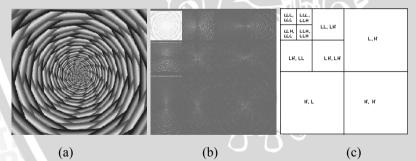
LH: hasil lowpass terhadap baris diteruskan dengan Highpass terhadap kolom

HL: hasil highpass terhadap baris diteruskan dengan

lowpass terhadap kolom

HH: hasil highpass terhadap baris dan kolom

Gambar 2.12. Algoritma Mallat untuk dekomposisi citra Sumber : Saha, 2004



Gambar 2.13 (a) citra asli, (b)dekomposisi level 3, (c)spektrum dan penataan dekomposisi Sumber : Sanberg, 2000

2.4 Digital Image Watermarking

Watermark merupakan sebuah informasi yang disisipkan pada media lain dengan tujuan melindungi media yang disisipi oleh informasi tersebut dari pembajakan, penyalahgunaan hak cipta, dan sebagainya, watermarking sendiri adalah cara untuk menyisipkan watermark kedalam media yang ingin dilindungi hak ciptanya (Munir, 2006).

Digital watermarking adalah proses penambahan kode identifikasi secara permanen ke dalam data digital, kode identifikasi tersebut dapat berupa teks, gambar, audio, atau, video. Selain tidak merusak data digital produk yang akan dilindungi, kode yang disisipkan seharusnya memiliki ketahanan (robustness) dari berbagai pemrosesan lanjutan seperti pengubahan, transformasi, geometri, kompresi, enkripsi, dan sebagainya. Sifat robustness berarti data watermark tidak rusak akibat pemrosesan lebih lanjut (Munir, 2006).

Salah satu prinsip dalam digital watermarking adalah informasi yang disisipkan pada media digital tidak boleh mempengaruhi kualitas media digital tersebut. Jadi pada citra digital, mata manusia tidak bisa membedakan apakah citra tersebut disisipi watermark atau tidak. Sehingga pada digital watermarking terdapat persyaratan bahwa digital watermark atau informasi digital yang disisipkan ke dalam media digital harus imperceptible atau tidak dapat dipersepsi oleh indera manusia baik penglihatan atau Human Visual System (HVS). (Agung, 2001).

2.4.1 Karakteristik Digital Watermarking

Beberapa karaktersitik atau sifat khusus harus dimiliki oleh sebuah *watermark*. Sifat-sifat tersebut tergantung dari aplikasi *watermarking* yang akan dibuat. Namun ada beberapa sifat umum yang dimiliki oleh aplikasi *watermarking* yang diantaranya sebagai berikut (Munir, 2006).

- Imperceptibility: Keberadaan watermark tidak dapat dipersepsi oleh indera visual, hal ini untuk menghindari terjadinya gangguan pengamatan visual.
- Key Uniqueness: Kunci yang berbeda seharusnya menghasilkan watermark yang berbeda. Hal ini berarti

penggunaan kunci yang salah dapat menyebabkan hasil deteksi watermark yang salah pula.

- *Nonintervibility* : Secara komputasi sangat sukar menemukan *watermark* jika diketahui hanya citra ber*watermark* saja.
- Robustness : Watermark seharusnya tetap kokoh terhadap berbagai serangan yang dilakukan pada citra berwatermark

2.4.2 Prinsip Dasar Watermarking

Prinsip dasar dari sistem *watermarking* ialah sistem yang dapat membandingkan dengan enkripsi simetrik, yaitu dengan menggunakan kunci yang sama dalam proses *encoding* maupun *decoding watermark*. Sistem *watermark* terdiri dari 2 subsistem, yaitu *encoder* (E_k) dan *decoder watermarking* (D_k). Sistem *watermarking* dapat dideskripsikan dengan tuple { O,W,K,E_k,D_k,C_τ }. Dimana O adalah data asli, W adalah *watermark* , dan W adalah Kunci (Arnold, 2003). Tuple tersebut dapat di ilustrasikan pada persamaan 2.19 yang merupakan fungsi embedding dan persamaan 2.20 adalah fungsi proses deteksi.

$$E_k: O \times W \times K \to O$$
 (2.19)

$$D_k: O \times K \to W$$
 (2.20)

Sedangkan fungsi pembandingnya dapat dilihat pada persamaan 2.21.

$$C_{\tau}: W^2 \to \{0,1\}$$
 (2.21)

Data yang di ekstrak dibandingkan dengan data watermark yang disisipkan dengan menggunakan threshold yang dilambangkan dengan τ . Paramater input yang digunakan untuk proses penyisipan adalah carrier object atau data asli c_o , selanjutnya watermark disisipkan sebagai kunci publik atau kunci rahasia K.

$$E_k(\mathbf{c}_0, \mathbf{W}) = \mathbf{C}_{\mathbf{w}} \tag{2.22}$$

Proses encoder hingga dapat menghasilkan data yang telah diberi *watermark* dapat dilihat pada gambar 2.14.



Gambar 2.14 Proses *Encoder Watermark* secara umum Sumber: Munir, 2006

Dalam proses pendeteksian, data yang diberi *watermark* dan kemungkinan data dimanipulasi dilambangkan dengan \hat{c}_w , *watermark* adalah c_w , dan kunci K merupakan parameter input untuk proses memperoleh kembali waterwark yang telah disisipkan kedalam data asli. Proses memperoleh kembali *watermark* atau proses decoder dapat dilihat pada gambar 2.15.



Gambar 2.15 Proses *Decoder Watermark* secara umum Sumber : Munir 2006

Berbagai jenis sistem *watermarking* memiliki perbedaaan dalam proses pembacaan parameter input. Umumnya data *watermark* yang telah di ekstraksi $\hat{\mathbf{w}}$ berbeda dari data *watermark* asal w, hal ini dikarenakan adanya kemungkinan manipulasi data *watermark*. Agar kedua data *watermark* $\hat{\mathbf{w}}$ dan w dapat memiliki kecocokan, digunakan fungsi pembanding pada persamaan 2.23 yaitu C_{τ} sehingga dapat membandingkan *watermark* yang diinginkan dengan mengembalikan nilai threshold τ .

$$C_{\tau}(\hat{\mathbf{w}}, \mathbf{w}) = \begin{cases} 1, & c \ge \tau \\ 0, & c < \tau \end{cases}$$
 (2.23)

Nilai threshold τ tergantung pada algoritma yang dipilih dan yang dapat mengidentifikai *watermark* dengan benar.

2.4.3 Serangan Terhadap Citra Ber-watermark

Secara umum, jenis serangan terhadap citra berwatermark dibagi dua, yaitu serangan standar (Standard Attack) dan malicious attack. Pengujian citra ber-watermark terhadap bentuk sabotase hanya dapat menggunakan standard attack saja, hal ini disebabkan karena malicious attack adalah bentuk serangan yang memiliki tujuan menghilangkan watermark dimana umumnya pihak penyerang mencari algoritma penyisipan dan kunci yang digunakan saat penyisipan watermark. Hal ini menyebabkan bentuk serangan ini tidak dapat diujikan karena algoritma dan kunci yang digunakan tentunya sudah diketahui oleh penyisip watermark.

Serangan standar biasanya merupakan serangan yang tidak disengaja untuk merusak atau mendapatkan *watermark* di dalam citra ber*watermark*. Beberapa bentuk serangan ini diantaranya adalah :

1. Cropping

Cropping merupakan serangan yang umum karena banyak orang sering menginginkan bagian tertentu dari sebuah citra saja. Untuk dapat mengatasi serangan ini dapat dilakukan dengan cara menyebarkan *watermark* pada tempat-tempat yang memungkinkan terjadinya serangan (Terzija, 2006).

- 2. Serangan Geometris (*Geometrical Attack*)
 Beberapa yang termasuk dalam serangan geometris ini adalah rotasi citra, penskalaan ulang citra, pengubahan aspect ratio, translasi, dan sebagainya (Terzija, 2006).
- 3. Kompresi

Serangan ini juga merupakan serangan yang dilakukan secara tidak sengaja. Kompresi sering dilakukan pada file multimedia seperti citra, audio, dan video. *Watermark* yang disisipkan biasanya lebih tahan terhadap kompresi yang memiliki domain sama dengan domain yang dipakai pada saat *watermarking*. (Terzija, 2006).

2. Penambahan Derau (*Noise*)

Citra digital sangat rentan mendapatkan berbagai macam jenis derau (*Noise*). Ada beberapa cara yang menyebabkan suatu derau dapat berada dalam sebuah citra digital. Sebagai contoh, jika citra merupakan hasil scan foto yang berasal film negatif (seluloid), maka negatif film ini merupakan sumber derau.

3. Filterisasi

Beberapa filter yang sering digunakan yaitu gaussian filter (blur) dan sharpening filter. Watermark dapat disisipkan pada frekuensi yang paling sedikit berubah jika terjadi kompresi, dengan memperkirakan filterisasi apa saja yang umum digunakan (Terzija, 2006).

2.5 Format Citra Bitmap

Matriks adalah struktur data yang tepat untuk merepresentasikan citra digital (Munir, 2006). Elemen-elemen matriks dapat diakses secara langsung melalui indeksnya (baris dan kolom).

Citra digital yang berukuran $N \times M$ (tinggi = N, lebar = M) lazim dinyatakan dengan matriks N baris dan M kolom yang ditunjukkan pada gambar 2.16:

$$f(0,0) \qquad f(0,1) \qquad \dots \qquad f(0,M)$$

$$f(1,0) \qquad f(1,1) \qquad \dots \qquad f(1,M)$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$f(N-1,0) \qquad f(N-1,1) \qquad \dots \qquad f(N-1,M-1)$$

Gambar 2.16 Bentuk Matriks berukuran N x M Sumber : Munir, 2006

Citra dalam format BMP ada tiga macam : citra biner, citra berwarna, dan citra hitam-putih (*grayscale*). Citra biner hanya mempunyai dua nilai keabuan, 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai pixel. Citra berwarna adalah citra yang lebih umum. Warna yang terlihat pada citra bitmap merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru. Setiap pixel disusun oleh tiga komponen warna : R (*red*),

G(green), dan B (blue). Kombinasi dari ketiga warna RGB tersebut menghasilkan warna yang khas untuk pixel yang bersangkutan. Pada citra 256 warna, setiap pixel panjangnya 8 bit, tetapi komponen warna RGB-nya disimpan di dalam tabel RGB yang disebut palet. Setiap komponen panjangnya 8 bit, jadi ada 256 nilai keabuan untuk warna merah, 256 nilai keabuan untuk warna hijau, dan 256 warna keabuan untuk warna biru. Nilai setiap pixel tidak menyatakan derajat keabuannya secara langsung, tetapi nilai pixel menyatakan indeks tabel RGB yang memuat nilai keabuan merah (R), nilai keabuan hijau (G), dan nilai keabuan biru (B) untuk pixel yang bersangkutan. Pada citra hitam-putih, nilai R = G = B untuk menyatakan bahwa citra hitam-putih hanya mempunyai satu kanal warna. Citra hitam-putih umumnya dalah citra 8-bit.

Citra yang lebih kaya warna adalah citra 24-bit. Setiap pixel panjangnya 24 bit, karena setiap pixel langsung menyatakan komponen warna merah, komponen warna hijau, dan komponen warna biru. Masing-masing komponen panjangnya 8 bit. Citra 24-bit disebut juga citra 16 juta warna, karena mampu menghasilkan 2^{24} = 16.772.216 kombinasi warna.

Setiap berkas bitmap terdiri atas header berkas, header bitmap, informasi palet, dan data bitmap (gambar 2.17).

Header	Header bitmap	Informasi palet	Data
berkas	Treader bitiliap	iliformasi palet	bitmap

Gambar 2.17 Format berkas bitmap Sumber: Munir, 2006

Informasi palet warna terletak sesudah header bitmap. Informasi palet warna dinyatakan dalam suatu tabel RGB. Setiap entry pada tabel terdiri atas tiga buat field, yaitu R(red), G(green), dan B(blue).

Data bitmap diletakkan sesudah informasi palet. Penyimpanan data bitmap di dalam berkas disusun terbalik dari bawah ke atas dalam bentuk matriks yang berukuran Height x Width. Baris ke-0 pada matriks data bitmap menyatakan data pixel di citra baris terbawah, sedangkan baris terakhir pada matriks menyatakan data pixel di citra baris teratas.

Format citra 4-bit (16 warna) serupa dengan format citra 8-bit. Pada citra 4-bit dan citra 8-bit, warna suatu pixel diacu dari tabel informasi palet pada entry ke-k (k merupakan nilai dengan rentang 0 – 15 untuk citra 16 warna dan 0-255 untuk citra 256 warna). Sebagai contoh pada gambar 2.18, pixel pertama bernilai 2; warnapixel pertama ini ditentukan oleh komponen RGB pada tabel palet warna entry ke-2, yaitu R=14, G=13, dan B=16. Pixel kedua serupa dengan pixel pertama. Pixel ketiga bernilai 1, warnanya ditentukan oleh komponen RGB pada tabel warna entry ke-1, yaitu R=20, G=45, dan B=24. Demikian seterusnya untuk pixel-pixel lainnya. Khusus untuk citra hitam-putih (8 bit), komponen R, R, dan R0 suatu pixel bernilai sama dengan data bitmap pixel tersebut. Jadi, pixel dengan nilai data bitmap R129, memiliki nilai R129, R129, dan R129.

	er berkas er bitma			
 <palet< td=""><td>warna F</td><td>RGB></td><td></td><td></td></palet<>	warna F	RGB>		
_	R	G	В	
1	20	45	24	
2	14	13	16	
3	12	17	15	
256	46	78	25	
	bitmap>			

Gambar 2.18 Format citra 8-bit (256 warna) Sumber : Munir 2006

Berkas citra 24-bit (16,7 juta warna) tidak mempunyai palet RGB, karena nilai RGB langsung diuraikan dalam data bitmap. Setiap elemen data bitmap panjangnya 3 byte, masing-masing byte menyatakan komponen R, G, dan B. Contoh format citra 24-bit (16 juta warna) kira-kira seperti pada gambar 2.19.

<header berkas>
<header bitmap>

<data bitmap>
20 19 21 24 24 23 24 ...

Gambar 2.19 Format citra 24-bit (16,7 juta warna) Sumber: Munir, 2006

2.6 Mean Square Error (MSE)

Dalam statistika, 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 watermark dan citra asli. Perhitungan nilai MSE dari citra digital berukuran N x M, dilakukan sesuai dengan persamaan 2.24.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{x=1}^{N} [I(x, y) - I'(x, y)]^{2}$$
 (2.24)

Dengan: I (x,y) adalah nilai piksel di citra asli,

I' (x,y) adalah nilai piksel pada citra berwatermark

M, N adalah dimensi citra

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

BAB III METODOLOGI DAN PERANCANGAN

Pada bab metode dan perancangan ini akan dibahas penggunaan metode yang digunakan dalam perancangan perangkat lunak pembuatan dan pengecekan *watermark* serta langkah-langkah yang dilakukan dalam pembuatan perangkat lunak ini.

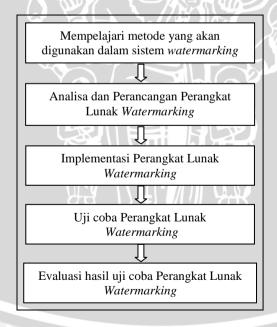
Tahapan pembuatannya adalah sebagai berikut:

- 1. Mempelajari metode yang digunakan dari jurnal yang pernah ada, yang telah disinggung pada bab 1.
- 2. Menganalisa dan merancang perangkat lunak dengan menggunakan penggabungan metode pada penelitian sebelumnya.
- 3. Membuat perangkat lunak berdasarkan analisis dan perancangan yang dilakukan.
- 4. Uji coba perangkat lunak dengan menggunakan citra.

3.1:

5. Evaluasi hasil pengecekan yang dilakukan oleh perangkat lunak untuk mengetahui kehandalan citra yang telah di*watermark*.

Langkah-langkah pembuatan dapat digambarkan pada Gambar



Gambar 3.1 Diagram Alir Perancangan Perangkat Lunak

3.1 Analisis Perangkat Lunak

Pada subbab analisis akan dibahas mengenai semua hal yang diperlukan dalam proses pembuatan dan pendeteksian *watermark* pada citra.

3.1.1 Deskripsi umum perangkat lunak

Perangkat lunak yang akan dibuat ini diperuntukkan bagi seorang *user* yang ingin mempublikasikan hasil karya miliknya berupa citra digital dan melindunginya agar tidak dipergunakan oleh pihak-pihak lain serta *user* tersebut ingin memeriksa citra temuan yang dianggap miliknya dan melakukan pembandingan dengan citra aslinya.

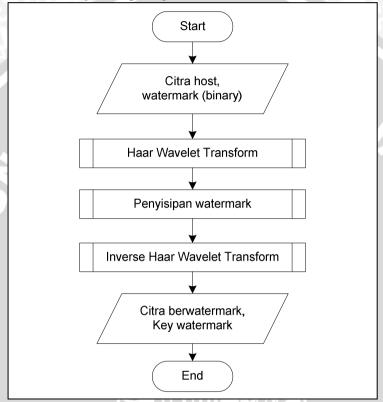
Maka perangkat lunak yang dibuat berupa sistem yang dirancang untuk dapat menyisipkan suatu informasi yang berguna sebagai tanda pengenal pada suatu citra digital dimana informasi tersebut akan selalu berada dalam citra. Citra digital yang akan disisipi watermark adalah citra berwarna dan keabuan dengan format .bmp. Informasi yang disisipkan berupa text yang digunakan sebagai watermark dan berfungsi sebagai identitas dari citra milik user. Masukan yang berupa citra yang akan diberi watermark dan text watermark akan melalui serangkaian proses watermarking hingga dihasilkan citra berwatermark. Sedangkan pada proses pendeteksian watermark tidak dibutuhkan citra asli, hanya dibutuhkan citra berwatermark dan key.

Metode yang digunakan adalah Discrete Wavelet Transform, tepatnya menggunakan Haar wavelet. Adapun proses-proses yang dilakukan dalam proses pembuatan watermarking adalah:

- 1. *User* memasukkan citra asli yang akan disisipi *watermark*, text yang akan digunakan sebagai *watermark*.
- 2. Text yang akan digunakan sebagai *watermark* dari bentuk *string* dirubah menjadi bentuk *binary*.
- 3. Memproses citra menggunakan *Haar Wavelet Transform* untuk menghasilkan rentang frekuensi LL, LH, HL, dan HH.
- 4. Melakukan proses penyisipan *watermark* pada rentang LH atau HL yang telah dihasilkan pada proses sebelumnya.
- 5. Menjalankan proses *Inverse Haar Wavelet Transform* (IHWT) untuk membentuk kembali citra ber-watermark.

6. Keluaran berupa citra yang telah berwatermark dan key watermark.

Diagram langkah-langkah proses pembuatan citra berwatermark ditunjukkan pada gambar 3.2.

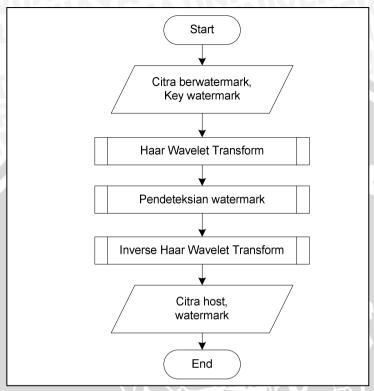


Gambar 3.2 Flowchart proses penyisipan Watermark

Sedangkan proses-proses yang dilakukan untuk mendeteksi watermark adalah :

- 1. User memasukkan citra ber-watermark.
- 2. Memproses citra menggunkan *Haar Wavelet Transform* untuk menghasilkan rentang frekuensi LL, LH, HL, dan HH.
- 3. Melakukan proses pendeteksian watermark.
- 4. Keluaran berupa watermark yang ada pada citra.

Diagram langkah-langkah proses pendeteksian *watermark* ditunjukkan pada gambar 3.3



Gambar 3.3 Flowchart proses pendeteksian Watermark

3.1.2 Batasan Perangkat Lunak

Batasan dari perangkat lunak yang dikembangkan adalah:

- 1. Perangkat lunak hanya bisa memproses citra dengan ekstensi .bmp.
- 2. Batasan proses dekomposisi citra hanya sekali proses.
- 3. Panjang text watermark dibatasi hanya 50 karakter.
- 4. Citra hasil manipulasi disesuaikan dengan citra asli dan dilakukan terpisah dengan perangkat lunak lain.

3.2 Perancangan Perangkat Lunak

Berdasarkan analisis yang telah ditentukan, berikut ini akan dibahas mengenai arsitektur dan proses yang terjadi pada perangkat lunak yang akan dibangun ini.

3.2.1 Perancangan Proses Input Penyisipan Watermark

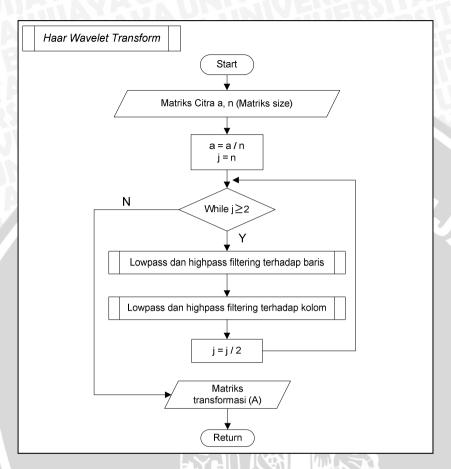
3.2.1.1 Haar Wavelet Transform

Pada proses ini, citra awal atau citra *Host* yang berbentuk matriks (array 2 dimensi) yang akan disisipi *watermark* terlebih dahulu ditransformasi dari bentuk diskrit ke dalam bentuk frekuensi dengan cara mendekomposisi citra *Host* tersebut. Proses dekomposisi tersebut dilakukan satu tingkat sehingga menghasilkan LL (*low-low frequency*), LH (*low-high frequency*), HL (*high-low frequency*), HH (*high-high frequency*). Tiap tingkat dekomposisi akan dilakukan transformasi baris terlebih dahulu melalui filter *lowpass* dan *highpass*. Kemudian dilakukan transformasi serupa pada kolom.

Langkah-langkah yang dilakukan pada tahap ini adalah:

- 1. Memasukkan komponen matrik dari citra yang akan disisipi watermark.
- 2. Membagi matrik dengan akar ukuran array.
- 3. Mendefinisikan ukuran array.
- 4. Melakukan conditional while ukuran array ≥ 2 .
- 5. Melakukan lowpass dan highpass filtering pada baris matrik.
- 6. Melakukan lowpass dan highpass filtering pada kolom matrik.
- 7. Membagi dua ukuran array untuk transformasi pada level selanjutnya (kembali ke point 4).

Lowpass dan highpass filtering merupakan tahap melewatkan koefisien baris atau kolom melalui persamaan 2.3 dan 2.4 untuk mendapatkan rentang frekuensi LL, LH, HL, dan HH. Flowchart proses Haar Wavelet Transform ditunjukkan pada gambar 3.4.



Gambar 3.4 Flowchart proses Haar Wavelet Transfrom

Setelah proses selesai maka akan menghasilkan informasi rentang frekuensi yang akan membentuk struktur piramida dari sebuah citra. Rentang frekuensi LL merupakan rentang taksiran penskalaan, sedangkan rentang LH, HL, dan HH merupakan rentang frekuensi detil informasi dari citra tersebut.

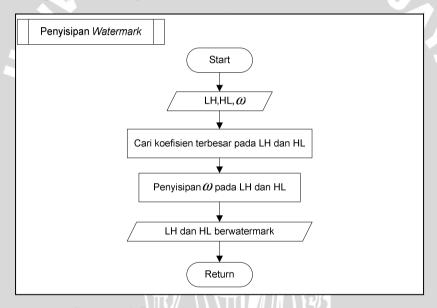
3.2.1.2 Proses Penyisipan Watermark

Pada tahap ini akan dilakukan proses penyisipan watermark (ω) . Dengan sebelumnya merubah watermark yang berbentuk

text/string ke dalam bentuk binary. Langkah-langkah yang dilakukan pada tahap ini adalah :

- 1. Masukan pada tahap ini adalah rentang frekuensi LH dan HL dari citra *host* yang telah didekomposisi dan *binary watermark*.
- 2. Mencari koefisien terbesar pada rentang LH dan HL.
- 3. Menyisipkan *binary watermark* pada koefisien terbesar pada rentang LH dan HL.
- 4. Hasil berupa rentang frekuensi LH dan HL yang telah disisipi binary watermark.

Gambar 3.5 merupakan *Flowchart* penyisipan *watermark* yang mana didalamnya terdapat proses pencarian koefisien terbesar yang terdapat pada rentang frekuensi LH dan HL.

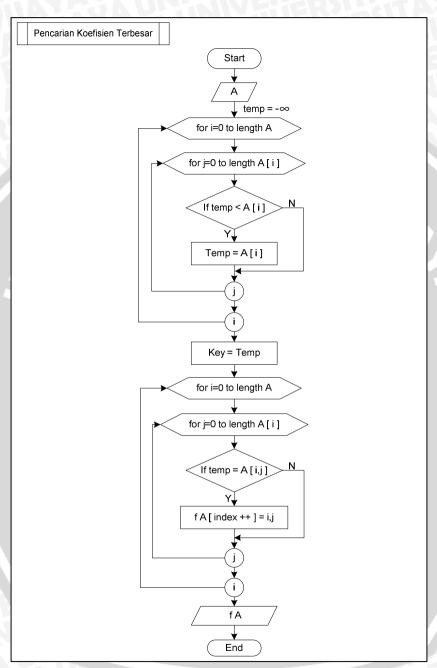


Gambar 3.5 Flowchart penyisipan watermark (ω)

Citra *Host* yang telah melalui proses dekomposisi, pada rentang frekuensi LH dan HL dicari koefisien yang terbesar. Proses pencarian dilakukan per baris yang kemudian hasil pencarian akan disimpan dalam *key*. *Key* adalah informasi nilai koefisien terbesar dari rentang LH dan HL yang nantinya digunakan sebagai kunci utama dalam mendeteksi *watermark* dan merekonstruksi citra *Host*. Gambar 3.6 adalah *Flowchart* dari pencarian koefisien terbesar pada rentang frekuensi LH dan HL, sedangkan proses-prosesnya adalah:

- 1. Input pada proses ini adalah matrik citra (A) yang telah didekomposisi dan telah ditentukan hanya pada rentang frekuensi LH dan HL.
- 2. Melakukan perulangan pada dimensi matrik rentang frekuensi LH dan HL.
- 3. Membandingkan hasil perulangan pada baris matrik dengan nilai baris matrik sehingga didapat koefisien yang terbesar pada baris matrik tersebut yang disimpan pada temporary.
- 4. Menyimpan koefisien terbesar baris matrik pada key.
- 5. Mengulangi proses yang sama pada poin 3 dan 4 sehingga menghasilkan indeks dari koefisien terbesar pada matrik.
- 6. Keluaran yang dihasilkan adalah *key* dan indeks dari koefisien terbesar pada matriks (fA).



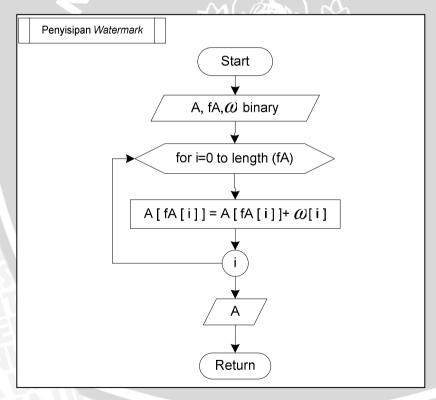


Gambar 3.6 Flowchart pencarian koefisien terbesar pada matrik A

Setelah menemukan koefisien terbesar pada matrik LH dan HL (A) beserta indeks posisi koefisien tersebut, proses selanjutnya adalah penyisipan *watermark* yang telah berbentuk *binary* ke dalam matrik LH dan HL. Adapun langkah-langkah pada proses ini adalah:

- 1. Masukan pada proses ini adalah matrik LH dan HL pada citra hasil dekomposisi, indeks posisi koefisien pada matrik LH dan HL, dan *binary watermark*.
- 2. Melakukan perulangan i sepanjang indeks posisi koefisien pada matrik LH dan HL.
- 3. Menambahkan *binary watermark* ke dalam koefisien terpilih yang sesuai dengan indeks posisi (fA).
- 4. Keluaran adalah matrik citra A yang telah disisipi *watermark*.

 Gambar 3.7 adalah *Flowchart* dari penyisipan *watermark* pada koefisien yang terbesar.



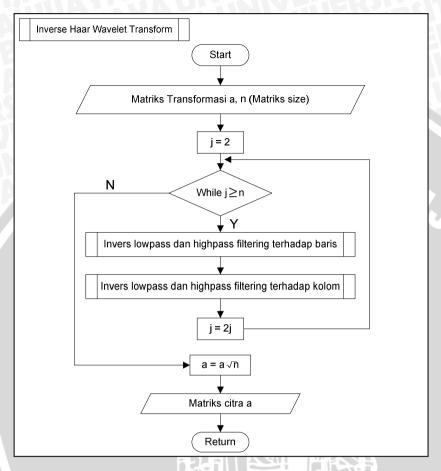
Gambar 3.7 Flowchart penyisipan watermark

3.2.1.3 Inverse Haar Wavelet Transform

Proses ini adalah proses merekonstruksi citra dari bentuk frekuensi ke dalam bentuk spasial sehingga diperoleh kembali matrik komponen warna RGB. Proses ini merupakan kebalikan dari proses *Haar Wavelet Transfrom* yang dilakukan dengan cara melewatkan koefisien matrik transformasi melalui *Inverse* filter seperti pada persamaan 2.3 dan persamaan 2.4. transformasi ini dilakukan sebanyak maksimal dua tingkat. Langkah langkah yang dilakukan pada tahap ini adalah:

- 1. Memasukkan matriks transformasi yang telah disisipi watermark.
- 2. Mendefinisikan ukuran array.
- 3. Melakukan conditional while ukuran array ≤ ukuran matriks transformasi
- 4. Melakukan lowpass dan highpass filtering pada baris matrik.
- 5. Melakukan lowpass dan highpass filtering pada kolom matrik.
- 6. Mengalikan dua ukuran *array* untuk transformasi pada level selanjutnya (kembali ke point 3).

Flowchart proses Inverse Haar Wavelet Transform ini ditunjukkan pada gambar 3.8.



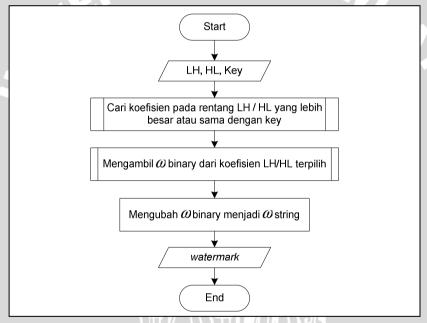
Gambar 3.8 Flowchart Inverse Haar Wavelet Transform

3.2.2 Proses Pendeteksian Watermark

Proses ini adalah suatu mekanisme yang digunakan untuk mengembalikan dan menampilkan watermark yang telah disisipkan ke dalam citra Host. Citra berwatermark yang berdomain spasial didekomposisi kedalam domain frekuensi mengunakan Haar Wavelet Transform untuk mendapatkan rentang frekuensi LH dan HL dimana pada rentang ini watermark disisipkan. Langkah-langkah pada proses ini adalah:

- 1. Melakukan proses pencarian koefisien yang lebih besar atau sama dengan yang tersimpan pada *key* pada rentang frekuensi LH atau HL.
- 2. Mengambil nilai *binary watermark* dari tiap-tiap koefisien yang terpilih.
- 3. Mengubah nilai *binary watermark* yang telah diambil ke dalam bentuk *string* sehingga *watermark* asli didapatkan.
- 4. Keluaran berupa *watermark* yang telah berbentuk text /*string*.

 Gambar 3.9 adalah *Flowchart* proses pendeteksian *watermark* dengan input berupa matrik hasil dekomposisi dan *key*.



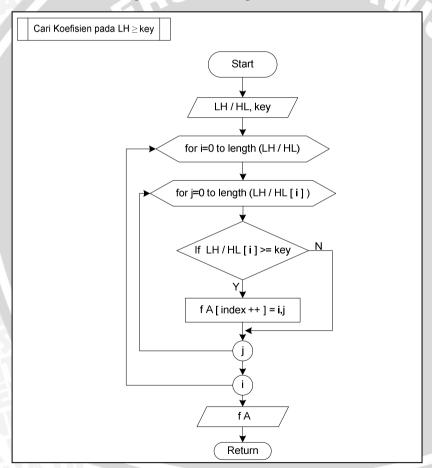
Gambar 3.9 Flowchart proses pendeteksian watermark

Proses awal dari pendeteksian watermark adalah mencari koefisien yang nilanya sama atau lebih besar dari key pada rentang LH atau HL. Sesuai dengan proses penyisipan, key menyimpan informasi nilai-nilai koefisien sebelum disisipi watermark. Proses pencarian ini dilakukan per baris yang nantinya akan menghasilkan indeks posisi dari koefisien yang lebih besar atau sama dengan key. Langkah-langkah yang dilakukan pada tahap ini adalah:

1. Input pada tahap ini berupa matrik LH atau HL, dan key.

- 2. Melakukan perulangan pada dimensi matrik rentang frekuensi LH atau HL
- 3. Membandingkan hasil perulangan pada baris matrik dengan nilai baris matrik sehingga didapat koefisien pada rentang LH atau HL yang nilainya lebih besar atau sama dengan nilai koefisien yang tersimpan dalam *key*.
- 4. Keluaran yang dihasilkan indeks dari koefisien terbesar pada matriks (fA).

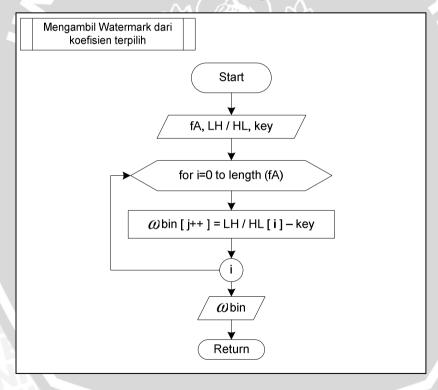
Gambar 3.10 adalah *Flowchart* pencarian koefisien yang lebih besar atau sama dengan *key* dari rentang LH atau HL.



Gambar 3.10 *Flowchart* proses cari koefisien pada rentang LH / HL yang lebih besar atau sama dengan *key*.

Setelah koefisien yang lebih besar atau sama dengan koefisien yang tersimpan dalam *key* ditemukan, dimana pada koefisien tersebut *watermark* disisipkan, maka proses selanjutnya adalah membangun kembali *watermark* (*generate watermark*). Langkah-langkah pada proses ini adalah:

- 1. Masukan berupa indeks dari posisi koefisien LH atau HL terpilih, *key*, dan matrik rentang frekuensi LH atau HL.
- 2. Melakukan perulangan i sepanjang indeks posisi koefisien pada matrik LH dan HL.
- 3. Mengurangi nilai-nilai koefisien yang telah ditemukan tadi dengan nilai-nilai koefisien pada *key*.
- 4. Keluaran dari proses ini adalah *watermark* dalam bentuk *binary*. Gambar 3.11 adalah *Flowchart* dari proses mengambil *watermark* dari koefisien LH atau HL terpilih.



Gambar 3.11 *Flowchart* proses mengambil *watermark* dari koefisien LH / HL terpilih.

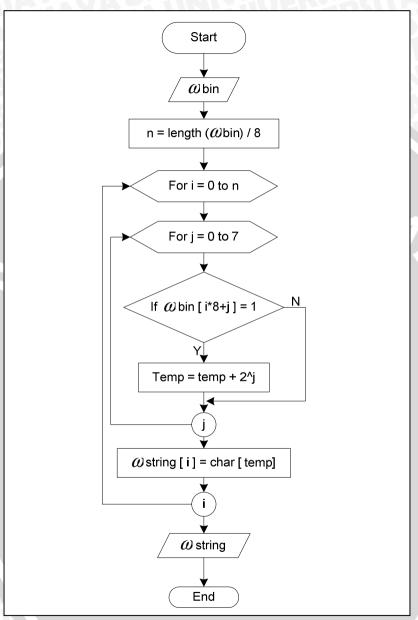
3.2.2.1 Proses Recovery Watermark

Proses *Recovery watermark* ini adalah proses mengembalikan *binary watermark* kedalam bentuk karakter ASCII. Penjelasan mengenai proses ini adalah sebagai berikut :

- 1. Masukan dari proses ini adalah *binary watermark* yang telah ditemukan pada proses sebelumnya.
- 2. Melakukan inisialisasi dengan membagi 8 rentang *binary* watermark.
- 3. Melakukan perulangan pada hasil bagi binary watermark.
- 4. Melakukan perulangan pada hasil bagi *binary watermark* dari 0 sampai 7.
- 5. Melakukan pengecekan apabila kondisi array i*8+j = 1, maka pada temporary akan menghasilkan temp+2^j, apabila hasil i*8+j≠1, maka proses kembali pada poin 3.
- 6. setelah proses diatas, *string watermark* akan disimpan didalam temporary dan proses kembali pada point 3 sampai semua *binary watermark* terproses
- 7. hasil dari proses ini adalah *string watermark* yang pada awalnya disisipkan pada citra *Host*.

Gambar 3.12 adalah *Flowchart* proses merubah *binary* watermark ke dalam bentuk string / karakter.





Gambar 3.12 *Flowchart* merubah *binary watermark* menjadi *string* / karakter.

3.3 Contoh Perhitungan

3.3.1 Proses Dekomposisi Citra

Misalkan ada citra berukuran 8x8 dengan nilai yang ditunjukkan pada gambar 3.13.

	1	2	3	4	5	6	7	8
1	65	69	90	31	47	29	76	74
2	55	72	41	54	59	88	66	11
3	47	40	11	5	54	67	8	92
4	83	68	92	27	70	73	49	66
5	15	23	94	36	12	98	30	5
6	10	86	55	69	10	66	1	30
7	90	53	32	44	90	34	35	48
8	7	85	76	18	92	49	12	89

AWINAL

Gambar 3.13 Nilai Citra berukuran 8x8

Untuk lebih memudahkan pemahaman dalam perhitungan maka digunakan pendekatan perkalian matrik. Langkah-langkah dekomposisi wavelet Haar terhadap potongan citra tersebut adalah:

 Menentukan filter dekomposisi LH yang ditunjukkan pada gambar 3.14

						11	PAFA 4
1/2	1/2	0	0	0	0	0	0
1/2	- 1/2	0	0	0	0	0	0
0	0	1/2	1/2	0	0	0	0
0	0	1/2	- 1/2	0	0	0	0
0	0	0	0	1/2	1/2	0	0
0	0	0	000	1/2	- 1/2	0	0
0	0	0	0	0 (0	1/2	1/2
0	0	0	0	0	0	1/2	- 1/2

Gambar 3.14 Filter Dekomposisi LH

2. Mengalikan setiap kolom citra pada gambar 3.13 dengan matriks dekomposisi pada gambar 3.14 yang contohnya dapat dilihat pada gambar 3.15.

	1/2	1/2	0	0	0	0	0	0		65		57
3	1/2	- 1/2	0	0	0	0	0	0		55		-10
į	0	0	1/2	1/2	0	0	0	0		47		69
	0	0	1/2	- 1/2	0	0	0	0	*	83	=	2.5
	0	0	0	0	1/2	1/2	0	0		15		45
	0	0	0	0	1/2	- 1/2	0	0		10		- 6
4	0	0	0	0	0	0	1/2	1/2		90		75
	0	0	0	0	0	0	1/2	- 1/2		7		-11

Gambar 3.15 Perkalian kolom 1 dengan filter LH

Hasil dekomposisi perbaris ini akan menghasilkan matriks D2 ditunjukkan pada gambar 3.16

57.00	71.00	46.50	36.00	54.00	31.50	25.50	11.00	
-10.00	2.00	0.50	-6.00	-9.00	5.50	1.50	-3.00	
69.50	37.50	65.50	44.00	56.50	23.00	22.50	70.00	
2.50	2.50	-6.50	-8.00	5.50	-12.00	-7.50	-13.00	
45.00	53.50	59.50	64.50	39.00	28.00	67.50	79.00	
-6.00	-16.50	5.50	-3.50	2.00	-8.00	-43.50	-9.00	
75.00	48.00	47.00	43.50	39.00	67.00	109.00	146.50	
-11.00	2.00	-3.00	6.50	5.00	-19.00	10.00	1.50	

Gambar 3.16 Matriks D2

Keterangan:

- warna biru adalah hasil aproksimasi
- warna merah adalah hasil detail
- 3. Mengatur hasil pada gambar 3.16 supaya bagian aproksimasi berkumpul di bagian atas dan bagian detail mengumpul di bagian bawah sehingga terbentuk matriks D3 yang ditunjukkan pada gambar 3.17

									$\overline{}$
l	57.00	71.00	46.50	36.00	54.00	31.50	25.50	11.00	
	69.50	37.50	65.50	44.00	56.50	23.00	22.50	70.00	
	45.00	53.50	59.50	64.50	39.00	28.00	67.50	79.00	
l	75.00	48.00	47.00	43.500	39.00	67.00	109.00	146.50	
	-10.00	2.00	0.50	-6.00	-9.00	5.50	1.50	-3.00	
	2.50	2.50	-6.50	-8.00	5.50	-12.00	-7.50	-13.00	
۱	-6.00	-16.50	5.50	-3.50	2.00	-8.00	-43.50	-9.00	
۱	-11.00	2.00	-3.00	6.50	5.00	-19.00	10.00	1.50	

Gambar 3.17 Matriks D3

4. Mengalikan tiap baris matrik D3 pada gambar 3.17 dengan matriks filter dekomposisi LH yang ditunjukkan pada gambar 3.14. Contoh untuk baris pertama ditunjukkan pada gambar 3.18

1/2	1/2	0	0	0	0	0	0 = 0		57	\mathcal{M}	65
1/2	- 1/2	0	0	0	0	0	0		71		-5.3
0	0	1/2	1/2	0	0	0	0		46.5		54
0	0	1/2	- 1/2	0	0	0	0	*	36	作式	11.5
0	0	0	0	1/2	1/2	0	0	17	54		55.8
0	0	0	0	1/2	- 1/2	0	0	// }	31.5		-2.8
0	0	0	0	0	0	1/2	1/2		25.5		58.8
0	0	0	0	0	0	1/2	- 1/2		(11)		14.3

Gambar 3.18 Perkalian Baris 1 dengan matriks LH

Hasil dekomposisi pada baris ini akan menghasilkan matriks D4 yang ditunjukkan pada gambar 3.19

```
65.30
       -5.30
              54.00
                    11.50
                            55.80
                                   -2.80 58.80
                                                14.30
59.50
       5.50
             33.80
                    17.80
                            66.00
                                  -4.00 53.80
                                                -25.30
33.50
       -21.0
             63.50 11.00
                            46.50 -35.5 16.50
                                                -1.00
58.80
       -10.3
             42.50
                     11.5
                            66.30
                                   24.8 46.00
                                                -22.5
                                  -7.25 18.3
 1.80
       3.30
              6.5
                     3.25
                           -17.80
                                                2.25
-16.00 -2.00 -25.8 -14.80 -5.50
                                   -2.50 -3.80
                                                -16.8
-14.50 17.00
             1.50
                    18.00
                            8.50
                                   -7.50
                                        1.00
                                                13.50
             -4.50
                    -17.50
                                         -4.50
12.80 28.80
                            -4.30
                                   3.30
                                                16.00
```

Gambar 3.19 Matriks D4

5. Mengatur hasil pada gambar 3.19 agar bagian aproksimasi berkumpul di bagian kiri dan bagian detail mengumpul di bagian kanan sehingga terbentuk matriks D5 yang ditunjukkan pada gambar 3.20.

									_
	65.30	54.00	55.80	58.80	-5.30	11.50	-2.80	14.30	
3	59.50	33.80	66.00	53.80	5.50	17.80	-4.00	-25.30	
	33.50	63.50	46.50	16.50	-21.0	11.00	-35.5	-1.00	
4	58.80	42.50	66.30	46.00	-10.3	11.5	24.8	-22.5	
	1.80	6.5	-17.80	18.3	3.30	3.25	-7.25	2.25	
1	-16.00	-25.8	-5.50	-3.80	-2.00	-14.80	-2.50	-16.8	
	-14.50	1.50	8.50	1.00	17.00	18.00	-7.50	13.50	
	12.80	-4.50	-4.30	-4.50	28.80	-17.50	3.30	16.00	

Gambar 3.20 Matriks D5

6. Telah dilakukan dekomposisi wavelet Haar 1 tingkat terhadap matrik citra. Matrik yang dihasilkan ditunjukkan pada gambar 3.21

65.30	54.00	55.80	58.80	-5.30	11.50	-2.80	14.30	
59.50	33.80	66.00	53.80	5.50	17.80	-4.00	-25.30	
33.50	63.50	46.50	16.50	-21.0	11.50	-35.5	-1.00	
58.80	42.50	66.30	46.00	-10.3	11.5	24.8	-22.5	
1.80	6.5	-17.80	18.3	3.30	3.25	-7.25	2.25	
-16.00	-25.8	-5.50	-3.80	-2.00	-14.80	-2.50	-16.8	
-14.50	1.50	8.50	1.00	17.00	18.00	-7.50	13.50	
12.80	-4.50	-4.30	-4.50	28.80	-17.50	3.30	16.00	

Gambar 3.21. Matrik dekomposisi 1 tingkat

Keterangan:

- warna biru adalah bagian aproksimasi (LL)
- warna merah adalah bagian detail horizontal (LH)
- warna coklat adalah bagian detai vertikal (HL)
- warna hijau adalah bagian detail diagonal (HH)

3.3.2 Proses Binerisasi Watermark

Masukan watermark yang berupa *text (string)* harus di konversi menjadi bentuk biner sesuai dengan kode ASCII dari tiaptiap karakter agar dapat disisipkan ke dalam matriks dekomposisi. Sebagai contoh digunakan *text* "Ilmu Komputer". Gambar 3.22 menunjukkan proses konversi dari teks ke kode ASCII dan menjadi bentuk biner.

Teks	Ilkom
ASCII	73 108 107 111 109
Biner	01001001 01101100 01101011 01101111 011011

Gambar 3.22. Proses Konversi teks ke biner

3.3.3 Proses Penyisipan Biner Watermark & Generating Key

Proses penyisipan watermark dilakukan setelah citra asli didekomposisi satu tingkat dimana akan terbentuk 4 bagian matrik yaitu LL, LH, HL, dan HH. Bagian paling signifikan untuk disisipi biner watermark adalah bagian LH dan HL. Kemudian dilakukan pencarian nilai tertinggi pada LH dan HL dan diurutkan dari yang tertinggi sampai yang terkecil. Biner watermark langsung ditambahkan ke nilai tersebut. Untuk biner bernilai 0 proses tidak diberlakukan. Gambar 3.22 menunjukkan proses penyisipan biner watermark

							Λ	70	7/
No	Val	Row	Col	Bin	\mathcal{L}		\star	d	
1	24.8	4	7	0		17	-3.8	6	4
2	18.3	5	4	1		18	-4.0	2	7
3	17.8	2	6	0	14	19	-4.3	8	3
4	14.3	1	8	0		20	-4.5	8	4
5	12.8	8	1	1	1	21	-4.5	8	2
6	11.5	4	6	0		22	-5.3	1	5
7	11.5	1	6	0		23	-5.5	6	3
8	11.0	3	6	1		24	-10.3	4	5
9	8.5	7	3	0		25	-14.5	7	1
10	6.5	5	2	1		26	-16.0	6	1
11	5.5	2	5	1		27	-17.8	5	3
12	1.8	5	1	0		28	-21.0	3	5
13	1.5	7	2	1		29	-22.5	4	8
14	1.0	7	4	1		30	-25.3	2	8
15	-1.0	3	8	0		31	-25.8	6	2
16	-28	1	7	0		32	-35.5	3	7

Gambar 3.22. Proses Penyisipan Biner Watermark

Pada Gambar 3.22 tampak bahwa proses penyisipan watermark dilakukan dengan mengurutkan nilai dari yang terbesar menuju terkecil pada bagian LH dan HL. Nilai tersebut masuk pada kolom "val". Sedangkan indeks posisi dari nilai tersebut ditunjukkan pada kolom "row" dan "col". Kolom "bin" menunjukkan nilai biner dari watermark. tabel inilah yang selanjutnya disimpan dalam bentuk key, dimana merupakan kunci dari watermark yang disisipkan yang akan digunakan untuk mendeteksi watermark.

3.3.4 Proses Rekonstruksi Citra

I.*=

Matrik yang akan direkonstruksi adalah matrik D5. Berikut ini adalah langkah-langkah dalam merekonstruksi citra dekomposisi kembali menjadi citra semula

1. Menentukan filter rekonstruksi L* dan H*, yaitu pada gambar 3.22.

-1 H*= -1 -1

Gambar 3.23 Filter Rekonstruksi L* dan H*

2. Untuk setiap baris hasil dekomposisi pada matriks D5 yang ditunjukkan pada gambar 3.20, matrik L* dan H* dikalikan dengan bagian aproksimasi dari matriks D5 seperti ditunjukkan pada gambar 3.23:

1	0	0	0		Jul	ムた	V_{\parallel}	0		0		
1	0	0	0		7		-1	0	0	0		
0	1	0	0		64		0	1	0	0		-7
0	1	0	0	*	41.25	+	0	-1	0	0	*	5.25
0	0	1	0		42.75		0	0	1	0		11.25
0	0	1	0		18.25		0	0	-1	0		7.25
0	0	0	1	'.			0	0	0	1		
0	0	0	1				0	0	0	-1		

	64		-7		57
	64		7		71
	41.25		5.25		46.5
=	41.25	+	-5.25	=	36
	42.75		11.25		54
	42.75		-11.25		31.5
	18.25		7.25		25.5
	18.25		-7.25		11

Gambar 3.24 Perkalian Baris 1 dengan filter L* dan H*

Kemudian hasilnya kembali diletakkan secara perbaris di matriks hasil, yang nantinya akan sama persis dengan matriks D3.

- 3. Dengan cara yang sama, dilakukan perkalian pada tahap 2 terhadap setiap kolom dari matriks yang diperoleh pada langkah sebelumnya.
- 4. Langkah ini akan menghasilkan kembali citra awal dan proses rekonstruksi selesai.

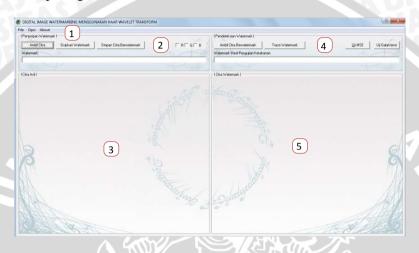
Merujuk pada gambar 3.23, matriks L* dan H* sebenarnya tidak perlu dibuat secara eksplisit, cukup lakukan beberapa operasi perkalian saja. Gambar 3.25 menunjukkan matriks rekonstruksi citra dimana berasal dari matriks dekomposisi yang telah disisipi watermark.

	1	2	3	4	5	6	7	8
1	2	3	88	20	29	67	33	16
2	37	40	8	45	7	82	5	27
3	92	37	77	32	70	83	71	30
4	15	31	46	24	83	26	48	14
5	88	27	78	96	49	88	84	2
6	15	13	51	1	59	0	77	63
7	77	70	56	19	70	59	96	65
8	99	24	68	28	8	75	48	87

Gambar 3.25. Matrik Rekonstruksi Citra

3.4 Perancangan Antarmuka

Berdasarkan analisa perancangan sebelumnya, secara umum rancangan antarmuka perangkat lunak *watermarking* citra dapat dilihat pada gambar 3.26.



Gambar 3.26 Rancangan Form Utama

Keterangan tentang rancangan antar muka pada gambar 3.26 adalah sebagai berikut :

- 1. Nomor 1 adalah tab menu dari perangkat lunak yang terdiri dari 3 menu utama diantaranya *File, opsi,* dan *about*.
- 2. Nomor 2 adalah tab dari penyisipan watermark yang digunakan dalam proses penyisipan watermark.
- 3. Nomor 3 adalah tab yang digunakan untuk menampilkan citra asli yang akan diwatemark dan citra watermark yang akan mengalami proses pendeteksian watermark.
- 4. Nomor 4 adalah tab dari pendeteksian watermark yang digunakan untuk mendeteksi watermark yang telah disisipkan.
- 5. Nomor 5 adalah tab yang digunakan untuk menampilkan citra yang telah disisipi watermark.

Pada masing masing tab, terdapat konten-konten yang akan dijelaskan sebagai berikut :

1. Menu Penyisipan Watermark



Gambar 3.27 Rancangan menu Penyisipan Watermark

Gambar 3.27 adalah konten dari menu Penyisipan watermark dimana pada nomor 1 adalah konten menu untuk memasukkan citra Host dan informasi resolusi dari citra masukan, perintah penyisipan watermark dan penyimpanan citra yang telah diwatermark. Pada nomor 2 adalah masukan dari teks yang akan digunakan sebagai watermark.

2. Menu Pendeteksian Watermark

Trace Watermark 1	<u>Uji</u> MSE	Uj Galat/error
n		1
	2	

Gambar 3.28 Rancangan menu Pendeteksian Watermark

Gambar 3.28 adalah konten dari menu *Recovery watermark* dimana pada nomor 1 adalah konten untuk masukan citra ber*watermark*, perintah *Trace Watermark*, dan perintah pengujian. nomor 2 adalah masukan dari teks watermark yang diingat oleh user yang akan digunakan untuk pengujian galat/error.

3.5 Perancangan Pengujian

Pada sub bab ini, akan dijelaskan mengenai perancangan pengujian perangkat lunak yang akan dibuat. Baik pengujian dari segi perangkat lunak maupun dari segi keluaran citra berwatermark.

3.5.1 Perangkat Lunak dalam Pengujian

Perangkat Lunak yang digunakan dalam pengujian citra watermark adalah Microsoft Excel untuk pengolahan data dan Adobe Photoshop CS3 untuk manipulasi citra watermark.

3.5.2 Bahan Pengujian

Berkas-berkas yang akan yang akan diujikan mempunyai spesifikasi sebagai berikut :

- 1. Citra berekstensi .bmp, 32 bit, RGB.
- 2. Resolusi citra yang digunakan adalah 320x320, 480x480, dan 500x500 piksel.
- 3. Ukuran panjang watermark maksimal 100 karakter.
- 4. 10 citra uji dan 3 teks *watermark* digunakan untuk melihat kebenaran proses penyisipan secara bervariasi.
- 5. 10 citra digunakan untuk melihat kualitas citra setelah melalui proses *watermarking*.

3.5.3 Pengujian Kebenaran Perangkat Lunak

Untuk menguji kebenaran perangkat lunak pengujian yang dilakukan adalah mencoba menyisipkan *watermark* ke dalam citra *Host* dan mencoba mendapatkan kembali *watermark* yang telah disisipkan. Tabel 3.1 merupakan tabel pengujian kebenaran perangkat lunak.

Tabel 3.1 Pengujian kebenaran perangkat lunak.

No	Citra Asal	watermark	Hasil penyisipan	Hasil ekstraksi watermark
		一倍 /	いるとい	

3.5.4 Pengujian Kinerja Perangkat Lunak

Skenario pengujian kinerja perangkat lunak antara lain:

- 1. Variasi ukuran dari citra Host.
- 2. Variasi watermark yang digunakan.
- 3. Menghitung MSE (*Mean Square Error*) dari tiap-tiap skenario pengujian.

Penghitungan MSE digunakan untuk mengetahui seberapa besar *noise* yang terjadi akibat proses *watermark* dimana nilai MSE menentukan seberapa dekat hasil *watermarking* terhadap HVS (*Human Visual System*).

Tabel 3.2 adalah tabel pengujian kinerja perangkat lunak dengan variasi dari citra *Host*

Tabel 3.2 Pengujian Kinerja Perangkat Lunak dengan Variasi Ukuran dari Citra *Host* dan Variasi Teks *Watermark*

No	Citra	Citra	Hasil	Hasil	Nilai
	Asal	watermark	penyisipan	ekstraksi	MSE
111				3D.	

3.5.5 Pengujian Ketahanan Citra Berwatermark

Pengujian ketahanan *watermark* ini dilakukan untuk menguji apakah *watermark* yang disisipkan dapat tetap bertahan setelah mengalami beberapa perubahan. Perubahan pada citra ber*watermark* ini menggunakan standart attack yang telah dijelaskan pada bab sebelumnya. Proses perubahan yang dilakukan pada citra ber*watermark* antara lain *cropping*, rotasi 10⁰ dan 90⁰, kompresi atau perubahan format, dan filterisasi (*bluring* dan *sharpening*).

Tabel 3.4 adalah tabel hasil pengujian ketahanan perangkat lunak

Tabel 3.3 Pengujian Ketahanan Citra Berwatermark

No	Jenis	Hasil	Hasil Ekstraksi	Presentase
	Serangan	Serangan		Galat/error

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Lingkungan implementasi pada subbab ini akan menjelaskan tentang lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan perangkat lunak pembuatan dan pengecekan *watermark* ini adalah :

- 1. Prosesor Intel Pentium D 2.8 Ghz
- 2. Memori 512 MB
- 3. Harddisk dengan kapasitas 80 GB
- 4. Monitor 15"
- 5. Keyboard
- 6. Mouse

4.1.2 Lingkungan Implementasi Perangkat Lunak

Beberapa perangkat lunak yang digunakan untuk pengembangan perangkat lunak pembuatan dan pengecekan watermark ini adalah :

- 1. Sistem Operasi yang digunakan adalah *Microsoft Windows XP* Professional Service Pack 2
- 2. Perangkat lunak yang digunakan dalam mengembangkan sistem adalah *Borland Delphi 7*
- 3. Perangkat lunak yang digunakan untuk menyiapkan citra uji adalah *Adobe Photoshop CS3 Extended*
- 4. Perangkat lunak yang digunakan untuk mengolah data adalah MS Excel 2007

4.2 Implementasi Perangkat Lunak

Berdasarkan analisa dan perancangan proses yang terdapat pada subbab 3.2, maka pada subbab ini akan dijelaskan implementasi dari proses-proses tersebut.

4.2.1 Struktur Data

Struktur data yang digunakan pada perangkat lunak ini direpresentasikan pada gambar 4.1.

```
type

matrixFilter: array of array of real;
matrixR: array of array of real;
matrixG: array of array of real;
matrixB: array of array of real;
matrixTR: array of array of real;
matrixTG: array of array of real;
matrixTB: array of array of real;
matrixTB: array of array of real;

sortR: array of double
sortG: array of double
sortR: array of double
sortR: array of integer
sortRY: array of integer
sortRY: array of integer
sortGY: array of integer
sortGY: array of integer
sortBX: array of integer
sortBX: array of integer
sortBY: array of integer
```

Gambar 4.1. Struktur Data

Keterangan mengenai struktur data pada gambar 4.1 dijelaskan pada tabel 4.1.

Tabel 4.1 Daftar Struktur Data

Struktur Data	Keterangan
matrixFilter: array of array of	Matriks yang digunakan
<pre>real; matrixR: array of array of real;</pre>	untuk menampung matriks
matrixG: array of array of real;	filter, matriks nilai dari
<pre>matrixB: array of array of real; matrixTR: array of array of real;</pre>	komponen warna R, G, dan
matrixTG: array of array of real;	B serta matriks temporary
matrixTB: array of array of real;	dari setiap komponen
	warna yang digunakan
	untuk perhitungan proses
	dekomposisi citra

str : string; nameFile : string;		Tipe data yang digunakan untuk karakter watermark dan nama file citra yang telah diberi watermark
sortRX sortRY sortGX sortGY sortBX sortBY	: array of integer: array of integer: array of integer	pada matriks dekomposisi mulai dari nilai tertinggi dan <i>Array</i> untuk menyimpan posisi x dan y dari nilai-nilai diurutkan
sortR sortG sortB		Array yang digunakan untuk mengurutkan nilai

4.2.2 Implementasi Watermarking

Langkah-langkah yang dilakukan oleh pengguna untuk melakukan proses penyisipan watermark (watermarking) pertamatama adalah memasukkan citra inputan, kemudian memasukkan watermark yang berbentuk teks, menyisipkan watermark dengan menggunakan metode haar wavelet transform, dan menentukan letak dimana file ber-watermark dan key disimpan. Setelah itu akan dilakukan proses seperti yang telah dijelaskan pada subbab 3.2.1

4.2.2.1 Haar Wavelet Transform

Tahap awal dari proses *watermarking* ini adalah melakukan proses dekomposisi citra asli, yaitu proses dimana citra ditransformasi dari bentuk diskrit ke dalam bentuk frekuensi. Transformasi citra dilakukan pada masing-masing komponen citra RGB menggunakan filter yang telah ditentukan. Potongan kode untuk melakukan proses dekomposisi ditunjukkan pada *source code* 4.1.

```
//perkalian kolom
    for idx:=0 to n-1 do
        for i:=0 to n-1 do
        begin
        tR:=0; tG:=0; tB:=0;
        for j:=0 to n-1 do
        begin
        tR:=tR+(matrixR[j,idx]*matrixFilter[i,j]);
```

```
tG:=tG+(matrixG[j,idx]*matrixFilter[i,j]);
      tB:=tB+(matrixB[j,idx]*matrixFilter[i,j]);
   matrixTR[i,idx]:=tR;
   matrixTG[i,idx]:=tG;
   matrixTB[i,idx]:=tB;
  end;
                                        BRAWUA
//pengelompokan hasil perkalian kolom
for i:=0 to n-1 do
begin
  if i \mod 2 = 0 then
    idx:=i div 2
  else idx:=(i div 2)+(n div 2);
  for j := 0 to n-1 do
 begin
   matrixR[idx,j]:=matrixTR[i,j];
   matrixG[idx,j]:=matrixTG[i,j];
   matrixB[idx,j]:=matrixTB[i,j];
  end;
end;
//perkalian baris
for idx:=0 to n-1 do
  for i := 0 to n-1 do
 begin
    tR:=0; tG:=0; tB:=0;
    for j := 0 to n-1 do
   begin
      tR:=tR+(matrixR[idx,j]*matrixFilter[j,i]);
      tG:=tG+(matrixG[idx,j]*matrixFilter[j,i]);
      tB:=tB+(matrixB[idx,j]*matrixFilter[j,i]);
   matrixTR[idx,i]:=tR;
   matrixTG[idx,i]:=tG;
   matrixTB[idx,i]:=tB;
  end;
//pengelompokan hasil perkalian baris
for i:=0 to n-1 do
begin
  if i \mod 2 = 0 then
    idx:=i div 2
  else idx:=(i div 2)+(n div 2);
  for j := 0 to n-1 do
  begin
    matrixR[j,idx]:=matrixTR[j,i];
    matrixG[j,idx]:=matrixTG[j,i];
   matrixB[j,idx]:=matrixTB[j,i];
  end;
end;
```

Sourcecode 4.1 Potongan kode proses dekomposisi citra asli

Pada proses diatas, matriks citra RGB diproses menggunakan matriks filter yang telah ditentukan terlebih dahulu. Proses transformasi diatas akan membentuk citra transformasi yang terbagi menjadi 4 bagian yaitu LL, LH, HL, dan HH yang ditunjukkan pada gambar 4.2.



Gambar 4.2. Gambar citra setelah proses dekomposisi (Citra Transformasi)

4.2.2.2 Proses Penyisipan Watermark dan Generating Key

Setelah citra melalui proses dekomposisi, proses selanjutnya adalah penyisipan *watermark*. *Watermark* yang sebelumnya berbentuk text / *string* terlebih dahulu dirubah menjadi binary. Binary *watermark* tersebut disisipkan pada matrik dekomposisi. Proses dilanjutkan sesuai dengan yang telah dijelaskan pada subbab 3.2.3. Potongan kode proses penyisipan *watermark* ditunjukkan pada sourcecode 4.2.

```
//tambahkan nilai ke ArySort berdasar representasi biner
if (str[i+1]='1') then
begin
matrixR[sortRX[i],sortRY[i]]:=matrixR[sortRX[i],sortRY[i]]+1;
matrixG[sortGX[i],sortGY[i]]:=matrixG[sortGX[i],sortGY[i]]+1;
matrixB[sortBX[i],sortBY[i]]:=matrixB[sortBX[i],sortBY[i]]+1;
end;
//ShowMessage(FloatToStr(matrixR[sortRX[i],sortRY[i]]));
end;
```

Sourcecode 4.2 Potongan kode proses penyisipan Watermark

Pada proses diatas, dilakukan pencarian nilai tertinggi pada rentang frekuensi LH dan HL. Nilai-nilai yang dicari disesuaikan

dengan panjang binary *watermark*, dan *watermark* tersebut disisipkan pada nilai-nilai hasil pencarian sesuai dengan urutan.

Setelah watermark disisipkan, pada proses penyimpanan citra dilakukan pula proses penciptaan kunci (*generating key*) dimana pada proses ini dilakukan penyimpanan nilai tertinggi pada rentang frekuensi LH dan HL beserta indeks posisinya dan biner watermark yang digunakan. Key ini akan digunakan untuk mendeteksi kembali watermark yang telah disisipkan dengan cara menyesuaikan indeks posisi pada rentang LH dan HL lalu mengurangi nilainya dengan biner yang tersimpan dalam key. Potongan kode proses generating key ditunjukkan pada sourcecode 4.3.

```
if (SD1.Execute) then

begin
   if (SD2.Execute) then
   begin
    AssignFile(fB,SD2.FileName);
   Rewrite(fB);
   writeln(fB,nBit);
   for i:=0 to nBit-1 do
   begin
    writeln(fB,SortRX[i],' ',SortRY[i],' ',SortR[i]);
   writeln(fB,SortGX[i],' ',SortGY[i],' ',SortG[i]);
   writeln(fB,SortGX[i],' ',SortGY[i],' ',SortG[i]);
   writeln(fB,SortBX[i],' ',SortBY[i],' ',SortB[i]);
   end;
   CloseFile(fB);
   Image3.Picture.SaveToFile(SD1.FileName);
   end;
end;
```

Sourcecode 4.3 Potongan kode proses generating key

4.2.2.3 Proses Rekonstruksi Citra Ber-watermark

Proses rekonstruksi citra adalah proses mengembalikan citra ke bentuk asli atau sesuai dengan susunan bit RGB yang sesuai dengan aslinya. Proses ini adalah kebalikan dari proses *haar wavelet transform* (Invers *Haar wavelet transform*). Potongan kode proses rekonstruksi citra ber-*watermark* ditunjukkan pada sourcecode 4.4.

```
n:=Image1.Width;
  for i := 0 to n-1 do
    for i := 0 to n-1 do
    begin
      if j \mod 2 = 0 then
      begin
        MatrixTR[i,j]:=MatrixR[i,j div 2]+MatrixR[i,(j div
2)+(n div 2)];
        MatrixTG[i,j]:=MatrixG[i,j div 2]+MatrixG[i,(j div
2)+(n div 2)];
        MatrixTB[i,j]:=MatrixB[i,j div 2]+MatrixB[i,(j div
2)+(n div 2)];
      end
      else
      begin
        MatrixTR[i,j]:=MatrixR[i,j div 2]-MatrixR[i,(j div
2)+(n div 2)];
        MatrixTG[i,j]:=MatrixG[i,j div 2]-MatrixG[i,(j div
2)+(n div 2)];
        MatrixTB[i,j]:=MatrixB[i,j div 2]-MatrixB[i,(j div
2)+(n div 2)];
      end;
    end;
  //rekons. kolom
  for i:=0 to n-1 do
    for j := 0 to n-1 do
    begin
      if j \mod 2 = 0 then
      begin
        MatrixR[j,i]:=MatrixTR[j div 2,i]+MatrixTR[(j div
2)+(n div 2),i];
        MatrixG[j,i]:=MatrixTG[j div 2,i]+MatrixTG[(j div
2)+(n div 2),i];
        MatrixB[j,i]:=MatrixTB[j div 2,i]+MatrixTB[(j div
2)+(n div 2),i];
//ShowMessage('['+IntToStr(j)+','+IntToStr(i)+']=['+IntToStr(
j div 2)+','+IntToStr(i)+']+['
                      +IntToStr((j div 2)+(n div
2))+','+IntToStr(i)+']');
      end
      else
      begin
        MatrixR[j,i]:=MatrixTR[j div 2,i]-MatrixTR[(j div
2)+(n div 2),i];
        MatrixG[j,i]:=MatrixTG[j div 2,i]-MatrixTG[(j div
2)+(n div 2),i];
        MatrixB[j,i]:=MatrixTB[j div 2,i]-MatrixTB[(j div
2)+(n div 2),i];
//ShowMessage('['+IntToStr(j)+','+IntToStr(i)+']=['+IntToStr(
i div 2)+','+IntToStr(i)+']-['
                      +IntToStr((j div 2)+(n div
```

```
2))+','+IntToStr(i)+']');
    end;
end;

//Masukkan Matrix R G B -> Image
for i:=0 to n-1 do
    for j:=0 to n-1 do

Image3.Canvas.Pixels[i,j]:=RGB(round(MatrixR[i,j]),round(MatrixG[i,j]),round(MatrixB[i,j]));
end;
```

Sourcecode 4.4 Potongan kode proses rekonstruksi citra berwatermark

Proses rekonstruksi ini sebenarnya membutuhkan filter sebagaimana proses dekomposisi, akan tetapi dalam hal ini cukup dilakukan operasi perkalian dan penambahan.

4.2.3 Implementasi Pendeteksian Watermark

Langkah awal yang akan dilakukan user apabila akan mendeteksi *watermark* adalah memasukan citra ber-*watermark*. Kemudian dilakukan proses yang sama dengan proses penyisipan *watermark*, yaitu proses *Haar wavelet transform* (HWT). Setelah itu akan dilakukan proses pendeteksian *watermark* pada citra menggunakan *key* dan dilanjutkan proses recovery *watermark* seperti yang telah dijelaskan pada subbab 3.2.5.1

4.2.3.1 Proses Pengambilan Watermark

Setelah citra ber-watermark melalui proses transformasi Haar wavelet transform, dilakukan proses pengambilan watermark pada citra menggunakan key yang disimpan. Didalam key tersimpan data nilai bit RGB yang akan diberi watermark dan indeks posisi dari nilai-nilai tersebut. Watermark didapatkan dengan cara mengambil selisih dari nilai bit RGB yang telah diberi watermark dengan nilai yang tersimpan pada key. Kemudian nilai tersebut akan berbentuk binary yang nantinya akan dikonversi menjadi bentuk ASCII atau string. Kode yang digunakan untuk proses pengambilan watermark ditunjukkan pada source code 4.5.

```
str:='';
 for i:=0 to nBit-1 do
   //ShowMessage(FloatToStr(matrixR[sortRX[i],sortRY[i]])+'-
'+FloatToStr(SortR[i]));
   selisih:=abs(matrixG[sortGX[i],sortGY[i]]-SortG[i]);
   if (selisih>0) then
   begin
                                      BRAWINA
     if selisih=63 then selisih:=1
     else if selisih=64 then selisih:=0;
   str:=str+(FloatToStr(selisih));
 end;
 showMessage(str);
 watermark:=''; idx:=1;
 repeat
 begin
   str1:=copy(str,idx,8);
   inc(idx,8);
   //ShowMessage(str1);
   val:=0;
   for i:= length(str1) downto 1 do
     if str1[i] = '1' then
       val:=val+(1 shl (length(str1)-i));
   watermark:=watermark+Char(val);
 end;
 until (idx>length(str));
 showMessage('watermark: '+QuotedStr(watermark));
```

Sourcecode 4.5 Potongan kode proses pengambilan watermark

4.2.4 Implementasi MSE (*Means Square Error*)

Proses MSE ini digunakan untuk menghitung tingkat kesalahan / noise yang ditimbulkan akibat proses *watermarking* antara citra asli dengan citra ber-*watermark*. Oleh karena itu proses ini membutuhkan inputan berupa dimensi citra, citra asli dan citra ber-*watermark*. Proses ini menghasilkan nilai MSE. Prosedur untuk melakukan proses MSE ditunjukkan pada source code 4.6.

```
procedure TForm1.Button9Click(Sender: TObject);
var x,y,hi,wi:integer; r,g,b:byte;
   MSE,MSEr,MSEg,MSEb, TotR,TotG,TotB:real;
begin
   hi:=Image1.Height;
   wi:=Image1.Width;
   TotR:=0;
   TotB:=0;
```

```
TotG:=0;
  for v:=0 to hi-1 do
 for x:=0 to wi-1 do
 begin
   r:=GetRValue(Image1.Canvas.Pixels[x,y]);
   g:=GetGValue(Image1.Canvas.Pixels[x,y]);
   b:=GetBValue(Image1.Canvas.Pixels[x,y]);
   MSEr:=sgr(r-GetRValue(Image3.Canvas.Pixels[x,y]));
   TotR:=TotR+MSEr;
   MSEg:=sqr(g-GetGValue(Image3.Canvas.Pixels[x,y]));
   TotG:=TotG+MSEq;
   MSEb:=sqr(b-GetBValue(Image3.Canvas.Pixels[x,y]));
   TotB:=TotB+MSEb;
 end;
 MSE:=(TotR+TotG+TotB)/3;
 MSE:=MSE/(wi*hi);
 caption:=formatfloat('0.00',MSE);
end;
```

Sourcecode 4.6 Prosedur proses penghitungan MSE

4.3 Implementasi Antarmuka (interface)

Berdasarkan rancangan antarmuka pada subbab 3.4 dihasilkan antarmuka proses *watermarking* pada citra. Gambar 4.3 adalah tampilan antarmuka utama.



Gambar 4.3. Antarmuka utama

Pada proses *watermarking*, user perlu menginputkan citra inputan dan *text watermark* yang akan disisipkan. Tampilan pada saat proses penyisipan *watermark* ditunjukkan pada gambar 4.4



Gambar 4.4. Antarmuka menu penyisipan *watermark* serta tampilan citra setelah disisipi *watermark*

Sedangkan pada proses pendeteksian *watermark*, user menginputkan citra *watermark* dan *key* yang digunakan. Tampilan pada saat proses pendeteksian *watermark* ditunjukkan pada gambar 4.5.



Gambar 4.5. Antarmuka menu pendeteksian *watermark* serta tampilan *watermark* yang terdeteksi

4.4 Implementasi Uji Coba

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

4.4.1 Skenario Pengujian

Pengujian terhadap perangkat lunak dibagi menjadi tiga, yaitu pengujian kebenaran perangkat lunak, pengujian kinerja perangkat lunak, dan pengujian ketahanan *watermark* terhadap seranganserangan. Dalam pengujian ini, semua berkas citra BMP mempunyai kedalaman 24 bit dan resolusi sebesar 72 dpi. Citra uji yang akan digunakan dapat dilihat pada lampiran 1.

Tabel 4.2 Daftar Berkas Citra

Nama File Res

No	Nama File	Resolusi
1	Berdoa.bmp	320 x 320
2	Sign.bmp	320 x 320
3	Sepi.bmp	320 x 320
4	Bromo.bmp	320 x 320
5	Pagi.bmp	480 x 480
6	Menanti.bmp	480 x 480
7	Balai Kota Malang.bmp	480 x 480
8	Seroja.bmp	500 x 500
9	Menari.bmp	500 x 500
10	Vidy.bmp	500 x 500

Pada penelitian ini ada beberapa teks (*watermark*) yang akan disisipkan pada citra asli. Setiap berkas citra asli akan di uji menggunakan tiga teks yang berbeda. Tabel 4.3 menunjukkan teks yang akan digunakan dalam penelitian ini.

Tabel 4.3 Daftar Teks untuk Penelitian Watermarking

No.	Teks Watermarking			
1.	Edwin Ibnu Kautsar			
2.	Program Studi Ilmu Komputer FMIPA Universitas			
1017	Brawijaya			
3.	Digital Image Watermarking Pada Citra Dalam Sebaran			
	Frekuensi Menggunakan Haar wavelet transform			

4.5 Pengujian Dan Analisis Hasil

Berdasarkan perancangan pengujian yang telah dijelaskan pada bab 3.5, maka hasil pengujian beserta analisisnya akan dijelaskan menjadi beberapa subbab, yaitu hasil pengujian dan analisis kebenaran perangkat lunak, hasil pengujian dan analisis kinerja perangkat lunak, serta hasil pengujian dan analisis ketahanan watermark terhadap serangan-serangan.

4.5.1 Hasil Pengujian Dan Analisis Kebenaran Perangkat Lunak

Pengujian penyisipan *watermark* dinyatakan berhasil apabila proses penyisipan teks ke dalam citra tidak mengalami pesan kegagalan eksekusi dari perangkat lunak dan *watermark* dapat diperoleh kembali. Pada tabel 4.4 merupakan hasil pengujian kebenaran perangkat lunak.

Tabel 4.3 Hasil Pengujian Kebenaran Perangkat Lunak

No	Citra Asal	Proses penyisipan	Proses Ekstraksi
1	Berdoa. bmp	berhasil	berhasil
2	Sign.bmp	berhasil	berhasil
3	Sepi.bmp	berhasil	berhasil
4	Bromo.bmp	berhasil	berhasil
5	Pagi.bmp	berhasil	berhasil
6	Menanti.bmp	berhasil	berhasil
7	Balai Kota Malang. bmp	berhasil	berhasil
8	Seroja. bmp	berhasil	berhasil
9	Menari. bmp	berhasil	berhasil
10	Vidy. bmp	berhasil	berhasil

Hasil uji menunjukkan bahwa perangkat lunak *image* watermarking yang telah dibuat pada penelitian ini telah memenuhi kebutuhan perangkat lunak yang telah dijelaskan pada bab 3. Hal ini dibuktikan dengan keberhasilan perangkat lunak dalam melakukan proses penyisipan watermark dan proses pendeteksian watermark tanpa mengalami pesan kegagalan dan watermark dapat dideteksi

sesuai dengan *watermark* yang disisipkan menggunakan 10 citra yang berbeda dan 3 variasi teks *watermark* dengan panjang teks yang berbeda.

4.5.2 Hasil Pengujian Dan Analisis Kinerja Perangkat Lunak

Tabel 4.5 menunjukkan hasil pengujian kinerja perangkat lunak dengan teks *watermark* "Edwin Ibnu Kautsar". Pada hasil pengujian ini dapat dijelaskan bahwa setelah citra asli disisipi teks *watermark* tidak terjadi perubahan bentuk maupun ukuran citra secara visual maupun secara teknis seperti perubahan resolusi citra.

Tabel 4.5 Hasil pengujian kinerja perangkat lunak dengan teks: "Edwin Ibnu Kautsar"

No	Citra Asal	Citra watermark	Hasil ekstraksi	Nilai MSE
1	Berdoa. bmp	Berdoa_mark.bmp	Edwin Ibnu Kautsar	2,54
2	Sign.bmp	Sign_mark. Bmp	Edwin Ibnu Kautsar	0,00
3	Sepi.bmp	Sepi_mark. Bmp	Edwin Ibnu Kautsar	1,06
4	Bromo.bmp	Bromo_mark.bmp	Edwin Ibnu Kautsar	0,21
5	Pagi.bmp	Pagi_mark. Bmp	Edwin Ibnu Kautsar	1,04
6	Menanti.bmp	Menanti_mark. bmp	Edwin Ibnu Kautsar	1,51
7	Balai Kota Malang. bmp	Balai Kota Malang_mark.bmp	Edwin Ibnu Kautsar	0,19
8	Seroja. bmp	Seroja_mark.bmp	Edwin Ibnu Kautsar	1,74
9	Menari. bmp	Menari_mark.bmp	Edwin Ibnu Kautsar	0,09
10	Vidy. bmp	Vidy_mark. bmp	Edwin Ibnu Kautsar	0,78

Pada pengujian ini digunakan metode *Means Square Error* (MSE) untuk mengetahui perbandingan tingkat kesalahan dari citra asli dengan citra *watermark*. nilai MSE pada tabel 4.5 menunjukkan tingkat noise tertinggi terdapat pada citra berdoa_mark.bmp yang merupakan citra hitam putih dengan banyak detail. Hal ini dikarenakan setiap piksel pada citra berdoa.bmp memiliki nilai R, G dan B yang sama sehingga beberapa nilai piksel yang sama pada citra apabila diproses haar wavelet menjadi nilai tertinggi. Sedangkan tingkat noise terendah terdapat pada citra tersebut hanya ada warna biru dan putih. Dengan demikian, nilai piksel R pada citra tersebut rendah menyebabkan sebaran nilai tertinggi pada hasil proses haar wavelet menyebar rata ke seluruh daerah LH dan HL.

Pada Tabel 4.6 menunjukkan hasil pengujian kinerja perangkat lunak dengan teks *watermark* "Program Studi Ilmu Komputer FMIPA Universitas Brawijaya".

Tabel 4.6 Hasil pengujian kinerja perangkat lunak dengan teks: "Program Studi Ilmu Komputer FMIPA Universitas Brawijaya"

No	Citra Asal	Citra watermark	Hasil ekstraksi	Nilai MSE
1	Berdoa. bmp	Berdoa_mark2 .bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	14,61
2	Sign. bmp	Sign_mark2.	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	0,01
3	Sepi.bmp	Sepi_mark2. Bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	1,49
4	Bromo. bmp	Bromo_mark2. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	2,34

5	Pagi.bmp	Pagi_mark2. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	2,45
6	Menanti.bmp	Menanti_mark 2.bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	4,05
7	Balai Kota Malang. bmp	Balai Kota Malang_mark 2.bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	1,23
8	Seroja. bmp	Seroja_mark2. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	3,82
9	Menari. bmp	Menari_mark2 .bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	0,18
10	Vidy. bmp	Vidy_mark2. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	2,86

Dengan metode yang sama, perbandingan tingkat kesalahan citra asli dengan citra watermark dihitung menggunakan MSE. Nilai MSE tertinggi masih pada terdapat pada citra berdoa.bmp dan nilai MSE terendah pada citra sign.bmp. Pada pengujian ini, text watermark yang digunakan lebih panjang daripada text watermark sebelumnya. Selisish nilai MSE pada citra berdoa.bmp cukup besar dikarenakan derau yang timbul pada citra yang telah disisipi watermark kontras dengan piksel yang ada disekitar derau tersebut.

Pada tabel 4.7 menunjukkan hasil pengujian kinerja perangkat lunak dengan teks *watermark* "Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform".

Tabel 4.7 Hasil pengujian kinerja perangkat lunak dengan teks: "Digital Image *Watermarking* Pada Citra Dalam Sebaran Frekuensi Menggunakan *Haar wavelet transform*"

No	Citra Asal	Citra watermark	Hasil ekstraksi	Nilai MSE
1	Berdoa. bmp	Berdoa_mark3 .bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	20,96
2	Sign. bmp	Sign_mark3. bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	0,22
3	Sepi.bmp	Sepi_mark3. Bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	1,50
4	Bromo. bmp	Bromo_mark3.	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	6,58
5	Pagi.bmp	Pagi_mark3. bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	3,96
6	Menanti.bmp	Menanti _mark3.bmp	Digital Image Watermarking Pada	7,72

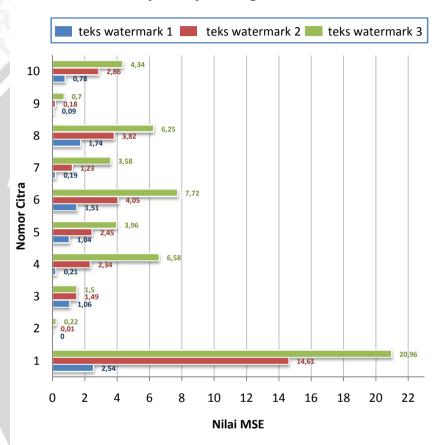
	3RAW 3RAW 5 BRA		Citra Dalam Sebaran Frekuensi Menggunakan <i>Haar</i> wavelet transform	
7	Balai Kota Malang. bmp	Balai Kota Malang_mark 3.bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	3,58
8	Seroja. bmp	Seroja_mark3. bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	6,25
9	Menari. bmp	Menari_mark3	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	0,70
10	Vidy. bmp	Vidy_mark3. bmp	Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	4,34

Pada pengujian ini, teks *watermark* yang digunakan lebih panjang daripada teks sebelumnya. Hampir semua citra mengalami kenaikan nilai MSE yang signifikan. Hal ini disebabkan karena nilai biner yang digunakan semakin panjang. Oleh karena itu semakin luas pula cakupan piksel yang digunakan digunakan untuk disisipi *watermark*.

Pada gambar 4.6 disajikan grafik nilai perbandingan kesalahan citra asli dengan citra ber-*watermark* menggunakan metode MSE

secara keseluruhan dimana pada grafik ini diketahui tingkat kenaikan nilai MSE seiring dengan bertambahnya panjang karakter *watermark* yang digunakan.

Grafik Uji Kinerja Perangkat Lunak



Gambar 4.6. Grafik uji kinerja perangkat lunak

Secara umum, pada pengujian ini nilai MSE tertinggi terdapat pada citra berdoa.bmp yang merupakan citra keabuan dengan tingkat detail cukup tinggi. Hal ini dikarenakan pada citra keabuan nilai R, G dan B bernilai sama pada tiap-tiap piksel. Apabila dilakukan proses haar wavelet nilai piksel yang berada pada rentang tengah akan berubah menjadi nilai tinggi atau rendah sehingga piksel

tersebut menjadi kontras dengan piksel disekitarnya. Hal inilah yang menyebabkan terjadinya derau (noise).

Sedangkan nilai MSE terendah pada pengujian kinerja perangkat lunak ini terdapat pada citra sign.bmp yang merupakan citra berwarna dengan warna dominan biru dan putih. Dengan demikian nilai piksel R pada citra ini lebih rendah daripada nilai piksel G dan B sehingga berpengaruh pada penempatan watermark pada koefisien wavelet di lapisan warna tersebut. Citra ini merupakan citra dengan tingkat detail terendah dibandingkan dengan citra uji yang lain, yang mana berpengaruh pada hasil pencarian nilai tertinggi pada koefisien LH dan HL yang digunakan untuk penempatan watermark sehingga apabila citra direkonstruksi, watermark yang disisipkan lebih merata.

4.5.3 Hasil Pengujian Dan Analisis Ketahanan Citra Berwatermark

Pada pengujian ketahanan citra ber-watermark ini, citra yang digunakan adalah citra menanti_mark2.bmp dengan resolusi citra sebesar 320x320 piksel dan dilakukan beberapa perubahan pada citra ber-watermark menggunakan standart attack. Diantara perubahan perubahan yang akan dilakukan adalah kesalahan memasukkan key, cropping, rotasi, perubahan format dan filterisasi (bluring dan sharpening). Teks watermark yang disisipkan adalah teks "Program Studi Ilmu Komputer FMIPA Universitas Brawijaya" dengan panjang 55 karakter.

Tabel 4.8 adalah tabel hasil pengujian ketahanan citra berwatermark.

Tabel 4.8 Hasil Pengujian Ketahanan Citra Ber-watermark

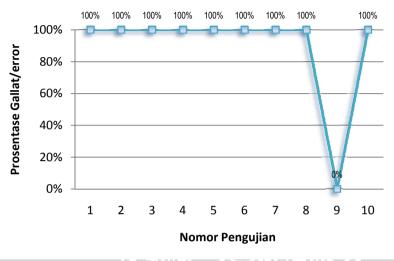
Prosent

No	Jenis Serangan	Hasil Ekstraksi	Prosentase Galat/ error
1	Kesalahan memasukkan <i>key</i>	-	100%
2	Bluring citra 1 Pixel	ÿÿÿÿ^2?øx~°ÿ Ä ÿãâ	100%

3	Cropping citra watermark	ÿÿyÿyÿyÿyyü□ÿyÿyÄ□ øÿÿþnýüÿÿ"ûÿÃÿø□ ÿþ‡ÿã̇ü>óüñÄ Yÿïçá ÜŽž□	100%
4	Pemberian Derau / Noise 10%	ñ‡ÿìq-8ñ‡	100%
5	Rotate citra 10 ⁰	Ÿÿÿÿÿyÿyÿyyÿ ÿÿÿyyyyyÿyyü ?ÿÿyâÿÿyyyyyyyyyïyÿ xÿÿ÷ÿÿ	100%
6	Rotate citra 90°	Ÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿp?yŸyÿÿÿyÇÿyÿ ÿÿÿÿÿÿÿÿ ÿÿyÿÿÿÿÏyÿÿÿ	100%
7	Sharpening citra 1 Pixel	ÿü□ÿ¡Ä!Á# 0ð%8qÚx³Œñðþ o ñ#¢÷Xñq™æ'- ã#Â,,Æ9Ì£ž	100%
8	Perubahan format bmp ke jpeg kembali ke bmp		100%
9	Perubahan format bmp ke png kembali ke bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	0%
10	Penurunan Resolusi citra 50%	Ÿ <u>ŷŷŷŷŷŷŷŷŷŷ</u> ŷŷŷŷyŷÿÿÿÿŷŷŷ ÿŷÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿ	100%

Dengan asumsi semakin tinggi nilai MSE yang didapat, maka semakin kokoh *watermark* yang disisipkan, maka pada pengujian ini penulis menggunakan citra berwarna dengan nilai MSE tertinggi yaitu citra menanti_mark2.bmp dengan panjang *watermark* yang digunakan 55 karakter. Gambar 4.7 menunjukkan grafik dari hasil pengujian ketahanan *watermark* menggunakan citra menanti_mark.bmp.

Grafik Prosentase Gallat/error



Gambar 4.7. Grafik prosentase gallat/error

Dari hasil uji ketahanan citra watermark dapat disimpulkan bahwa citra watermark hanya tahan terhadap proses perubahan format dari bmp ke png dan kembali ke bmp. Hal ini dapat dilihat dari nilai galat/error yang dihasilkan menunjukkan bahwa watermark yang disisipkan pada citra sign.bmp kembali secara utuh. Sedangkan pada pengujian-pengujian lain yang berupa proses kesalahan memasukkan key, cropping, rotasi, perubahan format dan filterisasi (bluring dan sharpening), teks watermark yang disisipkan tidak dapat kembali secara utuh dan benar sehingga nilai galat/error mencapai 100%, dimana tidak ada satupun karakter watermark yang kembali ke asal. Namun watermark masih tetap melekat pada citra tersebut meskipun telah melalui proses pengujian.

Bentuk *standart attack* merupakan bentuk serangan yang secara tidak langsung merubah nilai piksel R, G dan B sehingga berdampak besar pada hasil dekomposisi tiap lapisan warna. Oleh karena itu selisih yang didapat dari nilai piksel pada koefisien *wavelet* dengan nilai yang telah tersimpan pada key tidak sesuai dengan nilai bit *watermark*. Dimana nilai bit *watermark* merupakan

angka biner yang didalamnya hanya kombinasi dari 0 dan 1. Selisih nilai yang lebih dari 1 oleh sistem akan dianggap bernilai 1. Oleh karena itu citra yang telah mendapat serangan apabila diekstraksi rata-rata karakter yang muncul adalah "ÿ".

Pengujian yang dilakukan terhadap citra watermark merupakan bagian dari standart attack yang biasa digunakan attacker untuk merusak citra atau mengkomersialkan citra. Sehingga dapat dikatakan bahwa Digital Image Watermarking menggunakan metode Haar wavelet transform lemah terhadap serangan standar namun mempunyai invisibilitas tinggi yang ditunjukkan dengan nilai-nilai MSE yang rendah. Dimana hal ini sesuai dengan kaidah umum watermark, yaitu semakin nampak suatu watermark, maka semakin kokoh dan semakin susah untuk dihilangkan. Dan semakin tidak nampak suatu watermark, maka semakin lemah dan semakin mudah untuk dihilangkan.

4.6.4 Analisis Umum Hasil Uji

Perangkat lunak secara umum dinilai bekerja cukup baik. Pada uji kebenaran, perangkat lunak bekerja cukup baik dimana pada proses penyisipan maupun proses pendeteksian *watermark* tidak terjadi suatu kesalahan yang dibuktikan dengan munculnya pesan kesalahan.

Pada subbab 4.6.2 dilakukan uji kinerja perangkat lunak dengan menggunakan perhitungan MSE. Pada pengujian yang menggunakan watermark dengan panjang 18 karakter, nilai MSE menunjukkan noise yang dihasilkan akibat proses watermark cukup kecil, dengan rata-rata nilai dibawah 3. Nilai tertinggi terdapat pada citra berdoa_mark.bmp dimana citra tersebut adalah citra keabuan dengan banyak detail. Sedangkan nilai terendah terdapat pada citra sign_mark.bmp yang merupakan citra berwarna namun didalamnya hanya terdapat warna biru dan putih. Pada pengujian-pengujian berikutnya, dengan menggunakan panjang karakter watermark yang semakin besar, skema perubahan yang terjadi pada citra uji tidak berubah drastis. Kenaikan nilai MSE sesuai dengan panjang karakter watermark yang digunakan. Namun terjadi perubahan nilai MSE yang cukup signifikan seiring dengan semakin panjangnya karakter watermark, yang mana terjadi pada citra berdoa_mark.bmp.

Dengan nilai-nilai MSE yang telah tersaji diatas, maka dapat disimpulkan bahwa penyisipan watermark menggunakan metode Haar wavelet transform termasuk dalam kategori watermark invisible karena dengan mengacu pada nilai-nilai diatas, derau yang timbul akibat proses penyisipan watermark tidak disadari oleh Human Visual System. Dan ini telah sesuai dengan salah satu syarat dari sistem watermark, yaitu keberadaan watermark tidak disadari oleh Human Visual System.



BAB V KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan dari seluruh pengerjaan skripsi yang telah dilakukan dan juga saran untuk pengembangan lebih lanjut.

5.1 Kesimpulan

Setelah melakukan skripsi ini maka disimpulkan :

- 1. Proses penyisipan teks *watermark* ke dalam citra menggunakan *Haar Wavelet Transform* pada domain frekuensi menyebarkan bit watermark ke seluruh citra dengan tingkat perubahan piksel yang kecil sehingga keberadaan watermark tidak disadari oleh indera manusia.
- 2. Proses dekomposisi citra watermark menggunakan Haar Wavelet Transform dilakukan dengan cara menggunakan koefisien wavelet yang terbentuk dari proses tersebut.
- 3. Penggunaan karakter *watermark* yang pendek menghasilkan citra yang mendekati *Human Visual System* dengan ditunjukkannya nilai MSE yang rendah.
- 4. Bentuk serangan *standart attack* mampu merusak *watermark* yang disisipkan pada citra namun tidak menghilangkannya.

5.2 Saran

Untuk pengembangan lebih lanjut perangkat lunak maka ada beberapa saran yang dapat diberikan :

- 1. Untuk proses *watermarking* dapat digunakan teknik *wavelet* yang lain seperti *Daubechies, Coiflet, Symlet, Meyer, Morlet* dan *Mexican Hat*. Dimana teknik-teknik ini mungkin lebih tahan terhadap serangan.
- 2. Pengembangan selanjutnya dapat menggunakan citra berukuran n x m dan beresolusi besar.
- 3. Proses *watermarking* diharapkan tidak hanya diterapkan pada citra, namun dapat diterapkan pada *audio* maupun *video* untuk pengembangan lebih lanjut.

ERSITAS BRAWIUM 80

DAFTAR PUSTAKA

- Agung, W. P. 2001. *Digital Watermarking: Teknologi Pelindung HAKI Multimedia*. Elektro Indonesia Nomor 35 Tahun VI.
- Ames, Greg. 2002. Image compression. Jurnal Fall2002.
- Anonimus. 2005. Technical Advisory Service for Images: The Digital Image. http://www.tasi.ac.uk. Tanggal akses: 20 Januari 2009.
- Arnold, M. 2003. Tecniques and Applications of Digital Watermarking and Content Protection. Artech House.
- Djuhari, O. 2007. Pintu Otomatis Dengan Menggunakan Kode Tangan. Teknologi Informasi ITS. Surabaya.
- Gonzales, Rafael C. dan Richard E. Woods. 2005. *Digital Image Processing*, 2nd ed., Prentice-Hall Inc., New Jersey.
- Gultom dan Bernadus Surya Perdana. 2006. Analisis Kinerja Algoritma RSA dalam Pengacakan Citra Watermark pada Imagewatermarking Menggunakan Transformasi Wavelet. STT Telkom Bandung.
- Hameed K., A. Mumtaz, dan S.A.M. Gilani. 2006. Digital Image Watermarking in the Wavelet Transform Domain.

 Proceedings Of World Academy Of Science, Engineering And Technology. Volume 13.
- Jähne, B. 2002. Digital Image Processing. Springer. Germany.
- Lestari, D. A. 2003. *Implementasi Teknik Watermarking Digital Pada Domain DCT Untuk Citra Berwarna*. Universitas Gadjah Mada. Yogyakarta.
- Munir, R. 2006. *Diktat Kuliah IF 5054 Kriptografi*. Institut Teknologi Bandung.

- Murni, Aniati dan Dina Cahyati. 2003. *Pengolahan Citra Digital*. Fakultas Ilmu Komputer Universitas Indonesia.
- Nugroho, E. 2008. *Pengenalan Teori Warna*. Penerbit ANDI. Yogyakarta.
- Polikar, R. 1998. *Multi Resolution Analysis: The Discrete Wavelet Transform.* Durham Computation Center. Iowa State University.
- Porwik, Piotr, dan Agniezka Lisowska. 2004. The Haar-Wavelet Transform in Digital Image Processing: Its Status and Achievements. Jurnal Machine Graphic & Vision.
- Saha, Subhasis.2004. *Image Compression from DCT to Wavelets A Review*. http://www.acm.org/crossroads/xrds6-3/sahaimgcoding. html. 21 Januari 2009
- Salomon, David. 2004. *Data Compression: The Complete Reference*. New York: Springer-Verlag, Inc.
- Sandberg, Kristian. 2000. *The Haar wavelet transform*. http://amath.colorado.edu/courses/4720/2000Spr/Labs/haar.html. 21 Januari 2009
- Sirait, R. 2005. *Teknologi Watermarking Pada Citra Digital*. Institut Teknologi Bandung.
- Sripati, D. 2003. Efficient Implementations of Discrete Wavelet Transform using FPGAs. Florida State University.
- Terzija, N. 2006. Robust Digital Image Watermarking Algorithms for Copyright Protection. Universität Duisburg-Essen.
- Walidainy, Hubbul dan Nazlun. 2004. *Simulasi Menghapus Derau Pada Sinyal Suara*. Jurusan Elektro Fakultas Teknik Universitas Syiah Kuala.

LAMPIRAN

Lampiran 1 Tabel Daftar Berkas citra asli.

No	Nama File	Citra	Resolusi
1	Berdoa.bmp		320 x 320
2	Sign.bmp		320 x 320
3	Sepi,bmp		320 x 320
4	Bromo.bmp		320 x 320
5	Pagi.bmp		480 x 480

	V IA TA I I IN	
6	Menanti.bmp	480 x 480
7	Balai Kota Malang.bmp	480 x 480
8	Seroja.bmp	500 x 500
9	Menari.bmp	500 x 500
10	Vidy.bmp	500 x 500

Lampiran 2 Tabel Hasil pengujian kebenaran perangkat lunak

teks : Edwin Ibnu Kautsar

Gu I	teks : Edwin	Hasil	Hasil ekstraksi
Citra asal	Watermark	penyisipan	watermark
Berdoa.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar
Sign.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar
Sepi.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar
Bromo.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar
Pagi.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar
Menanti.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar
Balai Kota Malang.bmp	Edwin Ibnu Kautsar		Edwin Ibnu Kautsar

Seroja.bmp	Edwin Ibnu Kautsar	Edwin Ibnu Kautsar
Menari.bmp	Edwin Ibnu Kautsar	Edwin Ibnu Kautsar
Vidy.bmp	Edwin Ibnu Kautsar	Edwin Ibnu Kautsar

Teks: Program Studi Ilmu Komputer FMIPA Universitas Brawijaya

Citra asal	Watermark	Hasil penyisipan	Hasil ekstraksi watermark
	Program Studi	The state of the s	Program Studi
	Ilmu Komputer		Ilmu Komputer
Berdoa.bmp	FMIPA	1. 1.	FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya
	Program Studi		Program Studi
	Ilmu Komputer		Ilmu Komputer
Sign.bmp	FMIPA	Y	FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya
	Program Studi		Program Studi
	Ilmu Komputer	1	Ilmu Komputer
Sepi.bmp	FMIPA		FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya
	Program Studi		Program Studi
	Ilmu Komputer		Ilmu Komputer
Bromo.bmp	FMIPA		FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya

	Program Studi	-	Program Studi
(11/11) Le-1	Ilmu Komputer		Ilmu Komputer
Pagi.bmp	FMIPA	The same of the sa	FMIPA
28/25/	Universitas	-	Universitas
	Brawijaya	The second	Brawijaya
	Program Studi		Program Studi
	Ilmu Komputer		Ilmu Komputer
Menanti.bmp	FMIPA	1	FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya
	Program Studi		Program Studi
Balai Kota	Ilmu Komputer		Ilmu Komputer
	FMIPA		FMIPA
Malang.bmp	Universitas	And the same	Universitas
	Brawijaya		Brawijaya
	Program Studi		Program Studi
	Ilmu Komputer		Ilmu Komputer
Seroja.bmp	FMIPA .		FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya
	Program Studi	The state of the s	Program Studi
	Ilmu Komputer		Ilmu Komputer
Menari.bmp	FMIPA		FMIPA
•	Universitas		Universitas
	Brawijaya		Brawijaya
	Program Studi		Program Studi
	Ilmu Komputer		Ilmu Komputer
Vidy.bmp	FMIPA		FMIPA
	Universitas		Universitas
	Brawijaya		Brawijaya

Teks : Program Studi Ilmu Komputer FMIPA Universitas Brawijaya

Citra asal	Watermark	Hasil penyisipan	Hasil ekstraksi watermark
	Digital Image	T Y W	Digital Image
Berdoa.b	Watermarking Pada	Fizz i	Watermarking
mp	Citra Dalam		Pada Citra Dalam
	Sebaran Frekuensi		Sebaran Frekuensi
	Menggunakan Haar	10000000000000000000000000000000000000	Menggunakan Haar

	wavelet transform	NIKHTO	wavelet transform
AUVIII	Digital Image	THREE	Digital Image
	Watermarking Pada		Watermarking
G: 1	Citra Dalam		Pada Citra Dalam
Sign.bmp	Sebaran Frekuensi	Later .	Sebaran Frekuensi
	Menggunakan Haar		Menggunakan Haar
GIV.	wavelet transform		wavelet transform
	Digital Image		Digital Image
	Watermarking Pada		Watermarking
Casi haan	Citra Dalam		Pada Citra Dalam
Sepi.bmp	Sebaran Frekuensi		Sebaran Frekuensi
	Menggunakan Haar		Menggunakan Haar
	wavelet transform		wavelet transform
	Digital Image		Digital Image
	Watermarking Pada		Watermarking
Bromo.b	Citra Dalam		Pada Citra Dalam
mp	Sebaran Frekuensi		Sebaran Frekuensi
	Menggunakan Haar		Menggunakan Haar
	wavelet transform	X SATE	wavelet transform
	Digital Image		Digital Image
	Watermarking Pada		Watermarking
Pagi.bmp	Citra Dalam		Pada Citra Dalam
1 agr.omp	Sebaran Frekuensi	8	Sebaran Frekuensi
	Menggunakan Haar		Menggunakan Haar
	wavelet transform	77 38 6	wavelet transform
	Digital Image	A PARTY	Digital Image
	Watermarking Pada		Watermarking
Menanti.	Citra Dalam		Pada Citra Dalam
bmp	Sebaran Frekuensi		Sebaran Frekuensi
	Menggunakan Haar		Menggunakan Haar
	wavelet transform		wavelet transform
	Digital Image	7 (1) (1) (2)	Digital Image
Balai	Watermarking Pada		Watermarking
Kota	Citra Dalam		Pada Citra Dalam
Malang.b	Sebaran Frekuensi	THE REAL PROPERTY.	Sebaran Frekuensi
mp	Menggunakan Haar		Menggunakan Haar
	wavelet transform	Section 1	wavelet transform

	Digital Image	Digital Image
41111	Watermarking Pada	Watermarking
Seroja.b	Citra Dalam	Pada Citra Dalam
mp	Sebaran Frekuensi	Sebaran Frekuensi
24 31	Menggunakan Haar	Menggunakan Haar
	wavelet transform	wavelet transform
3112	Digital Image	Digital Image
	Watermarking Pada	Watermarking
Menari.b	Citra Dalam	Pada Citra Dalam
mp	Sebaran Frekuensi	Sebaran Frekuensi
	Menggunakan Haar	Menggunakan Haar
	wavelet transform	wavelet transform
	Digital Image	Digital Image
	Watermarking Pada	Watermarking
Vidy.bm	Citra Dalam	Pada Citra Dalam
p	Sebaran Frekuensi	Sebaran Frekuensi
	Menggunakan Haar	Menggunakan Haar
	wavelet transform	wavelet transform

Lampiran 3 Tabel pengujian kinerja perangkat lunak

Teks: Edwin Ibnu Kautsar

No	Citra Asal	Hasil penyisipan	Hasil ekstraksi	Nilai MSE
1	Berdoa. bmp		Edwin Ibnu Kautsar	2,54
2	Sign.bmp		Edwin Ibnu Kautsar	0,00
3	Sepi.bmp		Edwin Ibnu Kautsar	1,06

4	Bromo.bmp		Edwin Ibnu Kautsar	0,21
5	Pagi.bmp		Edwin Ibnu Kautsar	1,04
6	Menanti.bmp		Edwin Ibnu Kautsar	1,51
7	Balai Kota Malang. bmp		Edwin Ibnu Kautsar	0,19
8	Seroja. bmp		Edwin Ibnu Kautsar	1,74
9	Menari. bmp		Edwin Ibnu Kautsar	0,09
10	Vidy. bmp	4.4.0	Edwin Ibnu Kautsar	0,78

Teks : Program Studi Ilmu Komputer FMIPA Universitas Brawijaya

No	Citra Asal	Hasil penyisipan	Hasil ekstraksi	Nilai MSE
1	Berdoa. bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	14,61
2	Sign. bmp	5	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	0,01
3	Sepi.bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	1,49
4	Bromo. bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	2,34
5	Pagi.bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	2,45
6	Menanti.bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	4,05
7	Balai Kota Malang. bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	1,23

8	Seroja. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	3,82
9	Menari. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	0,18
10	Vidy. bmp	Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	2,86

Teks: Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform

No	Citra Asal	Hasil penyisipan	Hasil ekstraksi	Nilai MSE
1	Berdoa. bmp		Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	20,96
2	Sign. bmp		Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	0,22
3	Sepi.bmp		Digital Image Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan	1,50

MP	HAVA		Haar wavelet	
		VALATIN	transform	108
	MILLE	HAVE	Digital Image	14
			Watermarking	AVIT
			Pada Citra Dalam	177
4	Bromo. bmp		Sebaran Frekuensi	6,58
375			Menggunakan	
		AND THE RESERVE TO SERVE THE PARTY OF THE PA	Haar wavelet	
			transform	
		PITAS	Digital Image	
	.0		Watermarking	
			Pada Citra Dalam	
5	Pagi.bmp	Secretary and the second	Sebaran Frekuensi	3,96
		The same of the sa	Menggunakan	
		Contract to the last	Haar wavelet	
		TO OPEN	transform	
	,		Digital Image	
	7		Watermarking	
	<i>§</i>		Pada Citra Dalam	
6	Menanti.bmp	10 10 1 1 1 2	Sebaran Frekuensi	7,72
			Menggunakan	
	7		Haar wavelet	
			transform	
	V		Digital Image	
			Watermarking	
	Balai Kota		Pada Citra Dalam	
7	Malang. bmp		Sebaran Frekuensi	3,58
	Withing: omp		Menggunakan	
			Haar wavelet	
			transform	
			Digital Image	
8	Seroja. bmp		Watermarking	
			Pada Citra Dalam	
			Sebaran Frekuensi	6,25
			Menggunakan	
			Haar wavelet	
			transform	

9	Menari. bmp	Watermarking Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform Digital Image Watermarking	0,70
10	Vidy. bmp	Pada Citra Dalam Sebaran Frekuensi Menggunakan Haar wavelet transform	4,34

Lampiran 4 Tabel Hasil pengujian ketahanan citra berwatermark

Citra: Menanti.bmp

N o	Jenis Serangan	Hasil Serangan	Hasil Ekstraksi	Presentase Galat/error
1	Kesalahan memasukkan key untuk pendeteksian			100%
2	Bluring citra 1 Pixel		ÿÿÿÿ^2?øx~°ÿ Ä ÿãâ	100%
3	Cropping citra watermark		ÿÿyÿyÿyÿyÿü□ ÿÿÿÿÄ□øÿÿhn- ýüÿÿ,,ûÿÃÿø□ ÿþ‡ÿã̇ü>óüñÄ Yÿïçá ÜŽž□	100%

4	Pemberian Derau / Noise 10%		ñ‡ÿìq-8ñ‡	100%
5	Rotate citra 10 ⁰		Ÿyyyyyyyyyyy yyyyyyyyyyy ?yyyâyyyyyyy yyyñyyxyy÷yy	100%
6	Rotate citra 900		Ÿyyyyyyyyyyy yyyyyp?yŸyyyy yyÇyyyyyyyyy y yyyyyyyYyyyy	100%
7	Sharpening citra 1 Pixel		ÿü□ÿ¡Ä!Á# 0ð%8qÚx³Œñðþ o ñ#¢÷Xñq™æ 'ã#Â,,Æ9Ì£ž	100%
8	Perubahan format bmp ke jpeg kembali ke bmp	No. of the last of		100%
9	Perubahan format bmp ke png kembali ke bmp		Program Studi Ilmu Komputer FMIPA Universitas Brawijaya	0%
10	Penurunan Resolusi citra 50%		Ÿÿyyyyÿyyyyy ÿÿÿyyÿyyyyyy ÿ ÿÿyyyyyyyyy yyyyyyyy	100%