



TUGAS AKHIR – SM141501

**TEKNIK *INVISIBLE WATERMARKING*
MENGUNAKAN METODE DWT (*DISCRETE
WAVELET TRANSFORM*)**

**Stephanie Gani
NRP 1211 100 116**

**Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



FINAL PROJECT – SM141501

***INVISIBLE DIGITAL WATERMARKING TECHNIQUE
USING DWT (DISCRETE WAVELET TRANSFORM)
METHOD***

**Stephanie Gani
NRP 1211 100 116**

**Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT**

**DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2017**

LEMBAR PENGESAHAN

**TEKNIK *INVISIBLE WATERMARKING* MENGGUNAKAN
METODE DWT (*DISCRETE WAVELET TRANSFORM*)**

***INVISIBLE DIGITAL WATERMARKING TECHNIQUE
USING DWT (DISCRETE WAVELET TRANSFORM) METHOD***

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi ilmu komputer
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
Stephanie Gani
NRP. 1211100116

Menyetujui,
Dosen Pembimbing,

Dr. Budi Setiyono, S.Si, MT
NIP. 19720207 199702 1 001

Mengetahui,

Ketua Jurusan Matematika
MIPA ITS

Dr. Imam Mukhlash, S.Si, MT
NIP. 1960831 199403 1 003
Surabaya, 20 Desember 2017



TEKNIK INVISIBLE WATERMARKING DIGITAL MENGUNAKAN METODE DWT (DESCRETE WAVELET TRANSFORM)

Nama Mahasiswa : Stephanie Gani
NRP : 1211 100 116
Departemen : Matematika
Dosen Pembimbing : Dr. Budi Setiyono, S.Si, MT

Abstrak

Kemudahan penyebaran citra *digital* melalui internet memiliki sisi positif dan negatif terutama bagi pemilik asli citra *digital* tersebut. Sisi positif dari kemudahan penyebaran adalah dengan cepatnya pemilik citra tersebut menyebarkan *file* citra *digital* ke berbagai alamat situs di dunia. Sedangkan sisi negatifnya adalah jika tidak ada hak cipta yang berfungsi sebagai pelindung citra yang disebarakan tersebut, maka citra *digital* ini, yang misalkan adalah hasil foto komersil, atau hasil karya lukisan *digital*, akan sangat mudah diakui kepemilikannya oleh pihak lain. *Watermark* merupakan salah satu solusi untuk melindungi hak cipta terhadap foto *digital* yang dihasilkan. Dengan diterapkannya *Digital Image Watermarking* ini maka hak cipta foto *digital* yang dihasilkan akan terlindungi dengan cara menyisipkan informasi tambahan seperti informasi pemilik, keaslian, dsb ke dalam foto *digital* tersebut. *Discrete Wavelet Transform (DWT)* merupakan salah satu kakas yang banyak digunakan dalam teknik *watermarking* dalam *domain transform*.

Kata Kunci - citra *digital*, *watermarking*, *Discrete Wavelet Transform (DWT)*.

INVISIBLE DIGITAL WATERMARKING TECHNIQUE USING DWT (DESCRETE WAVELET TRANSFORM) METHOD

Name : ***Stephanie Gani***
NRP : ***1211 100 116***
Department : ***Mathematics***
Supervisor : ***Dr. Budi Setiyono, S.Si, MT***

Abstract

Ease of deployment of digital imagery through the internet has positive and negative sides, especially for owners of the original image. The positive side of the ease is the owner of that image can deploy digital image files to various sites address in the world. While the downside is that if there is no copyright that uses as protector of the transmitted image, the ownership of the digital image, such as a commercial photograph, or works of digital paintings, will be very easily claimed by other parties. Watermark is one solution to protect the copyright of the digital photo produced. With the implementation of the Digital Image Watermarking, the copyright of digital photos produced will be protected by inserting additional information such as owner information, authenticity, etc. into the digital photo. Discrete Wavelet Transform (DWT) is one option which is widely used in watermarking techniques in the domain transform.

Keywords: *Digital image processing, watermarking, Discrete Wavelet Transform (DWT).*

KATA PENGANTAR

Segala Puji bagi Allah Yang Maha Kuasa yang telah memberikan karunia, rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul: **“Teknik *Invisible Watermarking* Menggunakan Metode DWT (*Discrete Wavelet Transform*)”** yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Studi S-1 pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan berkat kerjasama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis mengucapkan terima kasih kepada:

1. Dr. Budi Setiyono, S.Si, MT selaku dosen pembimbing yang senantiasa membimbing dengan sabar dan memberikan kritik dan saran dalam penyusunan Tugas Akhir ini.
2. Dr. Imam Mukhlash, S.Si, MT selaku Ketua Jurusan Matematika.
3. Prof. Dr. Erna Apriliani, M.Si selaku Dosen Wali.
4. DR. Dwi Ratna Sulistyaningrum, S.Si, M.T, Drs. Soetrisno, MI.Komp. dan Dr. Mahmud Yunus, M.Si selaku dosen penguji Tugas Akhir ini.
5. Drs. Iis Herismsn, M.Si selaku Koordinator Tugas Akhir.
6. Seluruh jajaran dosen dan staf jurusan Matematika ITS.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga Tugas Akhir ini bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Desember 2017

Penulis

special thanks to

Selama proses pembuatan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan dan dukungan untuk penulis. Penulis mengucapkan terima kasih dan apresiasi secara khusus kepada:

1. Ayah Matius Gani Gan dan Ibu Liem Siek Djing yang telah memberikan dukungan dalam bentuk tempat tinggal di daerah Kampus ITS selama proses pengerjaan tugas akhir ini dan yang senantiasa dengan ikhlas memberikan semangat, perhatian, kasih sayang, doa, motivasi dan nasihat yang sangat berarti bagi penulis.
2. Kakak Margareth Gani Gan dan Adik Cynthia Gani yang menjadi salah satu motivasi bagi penulis untuk segera menyelesaikan Tugas Akhir ini.
3. Rosalina Dewi Intan yang telah meluangkan waktu untuk mengingatkan penulis untuk bimbingan dan mengerjakan tiap *step* tugas akhir ini.
4. Kusuma Wahaninggar, Tyara Herdha, Hasanuddin Al-Habib, Tuffahatul yang sudah membantu dalam mengerjakan program dan hitungan manual dalam tugas akhir ini.
5. Teman-teman diluar Perguruan Tinggi ITS: Lydia Kusnadi, Meyer Fossetta Hansory, Irene Lie Mei Fong, Margareth Stella Martha, Candra Gunawan, Henry Wijaya, dan Joshua. Semoga sukses kedepannya dan selalu diberikan kesehatan dan perlindungan dari Allah.
6. Gresela, Zeta, Zulfar, Rosy, Firdo, dan Teman-teman Matematika ITS serta teman-teman satu dosen pembimbing: Muhammad Azlansyah tetap semangat ya, Danar dan Hasanuddin AL-Habib yang di Program Studi S-2 tetap semangat dan semoga bisa satu periode wisuda.
7. Kepada Fedric Fernando dan Ginanjar yang kebetulan teman satu kelas dalam mata kuliah Pengolahan Citra *Digital* (PCD) sehingga ada beberapa file pada saat mata kuliah PCD yang saya dapatkan dari mereka.

8. Kepada Theater Print di jl. Gebang, yang sangat membantu saya dalam keperluan untuk mencetak selama proses tugas akhir ini. Dimana yang sangat membantu karena ada jasa pengiriman (dikala masih sibuk jadi tidak sempat mengambil).

Tentu saja masih banyak pihak lain yang turut andil dalam penyelesaian tugas akhir ini yang tidak bisa penulis sebutkan satu persatu. Semoga Allah membalas dengan balasan yang lebih baik bagi semua pihak yang telah membantu penulis.

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
DAFTAR LAMPIRAN	xxiii

BAB I. PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan Tugas Akhir	4

BAB II. KAJIAN TEORI

2.1	Penelitian Sebelumnya.....	7
2.2	Citra Digital	7
2.3	<i>Watermark</i>	8
	2.3.1 <i>Watermarking</i>	8
	2.3.2 <i>Framework Watermarking</i>	10
2.4	Transformasi Citra	12
2.5	<i>Discrete Wavelet Transform (DWT)</i>	12
	2.5.1 <i>Wavelet</i>	12
	2.5.2 <i>Transformasi Wavelet</i>	13
	2.5.3 <i>Discrete Wavelet Transform (DWT)</i>	14
	2.5.4 <i>Inverse Discrete Wavelet Transform (IDWT)</i>	16
2.6	PSNR	17

BAB III. METODOLOGI PENELITIAN

3.1 Studi Literatur	19
3.2 Analisis	19
3.2.1 Proses Penyisipan <i>Watermark</i> pada Citra <i>Digital</i>	19
3.2.2 Proses Ekstraksi <i>Watermark</i> pada Citra <i>Digital</i>	21
3.3 Implementasi	22
3.4 Pengujian	22
3.5 Penarikan Kesimpulan	23
3.6 Penulisan Tugas Akhir	23

BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM

4.1 Analisis Sistem.....	25
4.1.1 Analisis Sistem Perangkat Lunak	25
4.1.2 Analisis Kebutuhan Sistem.....	27
4.2 Perancangan Sistem	28
4.2.1 Gambaran Sistem.....	28
4.2.2 Penjelasan Umum Sistem	30
4.2.3 Perancangan Proses Algoritma	31
4.2.3.1 Perancangan Proses <i>Watermarking</i>	31
4.2.3.2 Perancangan Proses Rekonstruksi	33
4.2.3.3 Pengujian Citra dengan MSE.....	34
4.2.3.4 Pengujian Citra dengan PSNR.....	34
4.2.3.5 Pengujian Citra terhadap <i>Noise</i>	35
4.2.4 Perancangan Antar Muka Sistem.....	35
4.2.4.1 Perancangan Halaman Utama.....	35
4.3 Implementasi Sistem.....	38
4.3.1 Implementasi <i>Input</i> Citra	38
4.3.2 Proses Dekomposisi DWT2 pada Citra	39
4.3.3 Proses Penyisipan Citra	40
4.3.4 Proses Rekonstruksi <i>Watermark</i>	40
4.3.5 Proses Pengujian terhadap <i>Noise</i>	41

BAB V. PENGUJIAN DAN PEMBAHASAN	
5.1 Data Uji Coba	43
5.2 Pengujian Hasil <i>Watermark</i>	45
5.2.1 Pengujian PSNR	48
5.2.2 Pengujian Ketahanan terhadap <i>Noise</i>	52
BAB VI. KESIMPULAN DAN SARAN	
6.1 Kesimpulan	65
6.2 Saran	66
DAFTAR PUSTAKA	67
LAMPIRAN	71

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Proses Penyisipan <i>Watermark</i> pada Citra <i>Digital</i> 11
Gambar 2.2	Proses Pendeteksian <i>Watermark</i> pada Citra <i>Digital</i> 11
Gambar 2.3	Dekomposisi <i>Wavelet</i> Diskrit 1 Dimensi..... 15
Gambar 2.4	Transformasi <i>Wavelet</i> Diskrit 2D Level 2 15
Gambar 2.5	Rekonstruksi <i>Wavelet</i> Diskrit 1 Dimensi..... 17
Gambar 3.1	Proses Penyisipan <i>Watermark</i> 20
Gambar 3.2	Proses Rekonstruksi <i>Watermark</i> 21
Gambar 3.3	Proses Pengujian 1 22
Gambar 3.4	Proses Pengujian 2 23
Gambar 4.1	Citra <i>Digital</i> (a) 26
Gambar 4.2	Dekomposisi DWT Level 1 (b) 26
Gambar 4.3	Dekomposisi DWT Level 2 (c) 26
Gambar 4.4	Proses Pengrekonstruksian <i>Watermark</i> 27
Gambar 4.5	Diagram Alir Proses <i>Watermark</i> dan Citra <i>Host</i> 28
Gambar 4.6	Diagram Alir Proses Rekonstruksi <i>Watermark</i> 29
Gambar 4.7	Diagram Alir Proses Uji <i>Noise</i> 30
Gambar 4.9	Antar Muka Halaman Awal 35
Gambar 4.10	Antar Muka Halaman Proses <i>Invisible Watermarking</i> 36
Gambar 4.11	Antar Muka Halaman Proses Uji terhadap <i>Noise</i> 37
Gambar 5.1	Grafik dari nilai PSNR terhadap 3 Level. 50
Gambar 5.2	Grafik Perbandingan Waktu dan PSNR dari 3 Level 51

DAFTAR TABEL

	Halaman
Tabel 4.1	Tabel Kebutuhan Sistem..... 28
Tabel 5.1	Tabel Citra <i>Host</i> yang Digunakan..... 43
Tabel 5.2	Tabel Citra <i>Watermark</i> yang Digunakan 44
Tabel 5.3	Tabel Data untuk Pengujian..... 45
Tabel 5.4	Tabel Hasil Pengujian..... 47
Tabel 5.5	Tabel Nilai PSNR Berdasarkan Levelnya..... 49
Tabel 5.6	Tabel Waktu Berdasarkan Levelnya 50
Tabel 5.7	Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap <i>Gaussian Noise</i> dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter <i>Noise</i> 0.005 52
Tabel 5.8	Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap <i>Gaussian Noise</i> dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter <i>Noise</i> 0.05 54
Tabel 5.9	Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap <i>Gaussian Noise</i> dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter <i>Noise</i> 0.5 56
Tabel 5.10	Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap <i>Poisson Noise</i> dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter <i>Noise</i> 0.005 58
Tabel 5.11	Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap <i>Poisson Noise</i> dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter <i>Noise</i> 0.05 60
Tabel 5.12	Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap <i>Poisson Noise</i> dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter <i>Noise</i> 0.5 62

DAFTAR LAMPIRAN

	Halaman
Lampiran A : Kode Program Proses Antarmuka Awal	71
Lampiran B : Kode Program Dekomposisi DWT2	73
Lampiran C : Kode Program Proses Penyisipan Citra	79
Lampiran D : Kode Program Rekonstruksi <i>Watermark</i>	83
Lampiran E : Kode Program Pengujian terhadap <i>Noise</i>	89

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini penggunaan internet dalam kehidupan sehari – hari menjadi hal yang umum. Banyak fasilitas yang ditawarkan di internet, salah satunya adalah unggah dan unduh *file* citra. Dengan adanya fasilitas unggah dan unduh *file* seseorang dapat mengunggah dan mengunduh *file* citra dengan mudah karena adanya jangkauan penggunaan internet yang sangat luas. Penggunaan internet yang sangat luas ternyata memiliki sisi positif dan negatif terutama bagi pemilik asli citra *digital*. Sisi positif dari kemudahan penyebaran tersebut antara lain dengan cepatnya pemilik citra menyebarkan *file* citra *digital* ke salah satu situs yang ada di internet. Sedangkan sisi negatifnya antara lain tidak adanya perlindungan kepemilikan citra yang disebarkan, citra *digital* ini akan sangat mudah diakui kepemilikannya oleh pihak lain.

Perlindungan hak cipta untuk data citra *digital* sudah menjadi suatu hal yang penting. Banyak kasus pelanggaran hak cipta yang beredar di dunia internet, salah satu contoh kasus, yaitu Prayoga adalah seorang desainer yang sedang berkuliah di Institut Teknologi Bandung Fakultas Desain Komunikasi Visual dan menjadi anggota Asosiasi Desain Grafis Indonesia (ADGI). Prayoga memasarkan karya-karya dan jasa membuat karya desain grafisnya melalui dunia maya (internet), salah satunya melalui <http://www.kreatifprofesional.com>. Pada tanggal 29 Agustus 2008 Prayoga mendapatkan laporan dari ADGI, bahwa karya desain grafisnya digunakan seseorang dalam blog di website <http://wordpress.com> dan diakui sebagai ciptaan dari seseorang warga negara India yang beridentitas Brahmana karya desain grafis tersebut didapatkan dengan cara didownload dari website <http://www.kreatifprofesional.com> tanpa seizin Prayoga [2]. Berdasarkan kasus-kasus yang terjadi dalam pelanggaran hak cipta, khususnya dalam kasus pengakuan karya citra *digital* orang

lain, pemberian *watermarking* menjadi hal yang sangat diperlukan. Pemberian *watermarking* bertujuan untuk melindungi data citra *digital* dari pengakuan orang lain.

Watermarking merupakan aplikasi dari steganografi, namun ada perbedaan antara keduanya. Jika pada steganografi informasi rahasia disembunyikan di dalam media *digital* di mana media penampung tidak berarti apa-apa, maka pada *watermarking* justru media *digital* tersebut yang akan dilindungi kepemilikannya dengan pemberian label hak cipta (*watermark*). Untuk *Watermarking* dibagi dalam dua kategori, yaitu *visible watermarking* dan *invisible watermarking*. Dalam penelitian ini yang penulis gunakan adalah *invisible watermarking*.

Dengan berkembangnya *watermarking* pada saat ini, berkembang pula metode – metode yang di terapkan pada proses *digital watermarking*. Sebagai contoh, metode transformasi DWT (*Discrete Wavelet Transform*), FFT (*Fast Fourier Transform*), DCT (*Discrete Cosine Transform*), dan sebagainya. Metode yang digunakan penulis dalam penelitian ini adalah metode transformasi DWT (*Discrete Wavelet Transform*). Adapun beberapa alasan dipilihnya domain *Discrete Wavelet Transform* (DWT) karena memiliki kelebihan dalam mengidentifikasi bagian-bagian di *cover image*, dimana *watermark* dapat disisipkan secara efektif. Distorsi yang disebabkan oleh *wavelet domain* dalam perbandingan kompresi tinggi tidak terlalu mengganggu dibandingkan *domain* lain dalam *bit rate* yang sama. *Bit-error rate* yang rendah [1].

DWT (*Discrete Wavelet Transform*) merupakan salah satu teknik pemrosesan sinyal *digital*. Teknik ini merupakan salah satu yang sedang berkembang di masyarakat dikarenakan teknik ini lebih mudah diaplikasikan dan hasilnya lebih bagus dibandingkan sebelumnya yaitu transformasi Fourier. DWT membagi sebuah dimensi sinyal menjadi dua bagian, yaitu frekuensi tinggi (*highpass filter*) dan frekuensi rendah (*lowpass filter*).

Penelitian yang menggunakan teknik *watermarking* antara lain penelitian oleh Moh Khavid Reza dengan judul Perancangan Sistem *Watermarking* Pada Citra *Digital* Menggunakan Metode DCT dan LSB [12], serta penelitian lain oleh B. Yudi Dwiandiyanta dalam penelitiannya yang berjudul Perbandingan *Watermarking* Citra dengan Alihagam *Wavelet* dan *Discrete Cosine Transform* [3]. Berdasarkan latar belakang tersebut maka penulis mengangkat penelitian dengan judul Teknik *Invisible Watermarking Digital* Menggunakan Metode DWT (*Discrete Wavelet Transform*).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, masalah yang akan diselesaikan sebagai berikut:

1. Bagaimana melakukan penyisipan *watermark* pada sebuah citra *digital* menggunakan DWT.
2. Bagaimana proses perekonstruksian *watermark* yang telah disisipkan ke dalam suatu citra (citra ber-*watermark*).
3. Bagaimana mengetahui tingkat kualitas citra ber-*watermark* dan kehandalan metode DWT terhadap *noise*, jika kemungkinan terjadi pengiriman dll.

1.3 Batasan Masalah

Dari permasalahan di atas dapat dirumuskan bagaimana dapat mengimplementasikan masalah - masalah tersebut ke dalam aplikasi suatu program. Adapun batasan masalahnya adalah sebagai berikut:

1. Citra *host* merupakan citra berwarna (RGB).
2. Citra *watermark* merupakan citra berwarna (RGB), dengan ukuran = citra *host*.
3. DWT yang digunakan adalah berbasis *wavelet Haar*.
4. Proses pengujian menggunakan PSNR.
5. Proses mengukur tingkat kualitas dan kehandalan citra ber-*watermark* dengan merekonstruksi citra *watermark* yang telah diberi *noise*.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah :

1. Melakukan *watermarking digital* pada citra dengan menggunakan transformasi DWT.
2. Melakukan perekonstruksian citra yang telah disisipi oleh *watermark*.
3. Mengetahui tingkat kualitas citra ber-*watermark* dan kehandalan metode DWT terhadap *noise*.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah :

1. Membantu pihak-pihak dalam memproteksi hak cipta pada citra *digital*.
2. Membantu pihak-pihak dalam mengirimkan pesan / informasi rahasia berupa citra *digital* ke dalam citra *digital*.
3. Sebagai referensi dan informasi tentang teknik *invisible watermarking* dengan menggunakan metode *Discrete Wavelet Transform* (DWT).

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan Tugas Akhir ini disusun berdasarkan beberapa bab sebagai berikut:

1. BAB I Pendahuluan
Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat dari penelitian yang dilakukan, serta sistematika penulisan Tugas Akhir.
2. BAB II Kajian Teori
Bab ini menjelaskan tentang studi penelitian sebelumnya dan beberapa teori pendukung yang berkaitan dengan Tugas Akhir ini.
3. BAB III Metodologi Penelitian
Bab ini menjelaskan tentang langkah-langkah yang digunakan untuk menyelesaikan Tugas Akhir ini.

4. BAB IV Perancangan dan Implementasi Sistem
Bab ini menjelaskan tentang perancangan dan implementasi sistem, serta proses pembuatan sistem secara keseluruhan.
5. BAB V Uji Coba dan Pembahasan
Bab ini menjelaskan tentang hasil uji coba sistem. Kemudian dicatat sebagai bahan untuk merumuskan beberapa kesimpulan dari Tugas Akhir ini.
6. BAB VI Kesimpulan dan Saran
Bab ini berisi tentang kesimpulan dari hasil penelitian dan saran untuk pengembangan dari Tugas Akhir ini.

BAB II

KAJIAN TEORI

2.1 Penelitian Sebelumnya

Beberapa penelitian terdahulu yang berkaitan dengan Tugas Akhir ini :

1. Penelitian yang dilakukan oleh Moh Khavid Reza dengan judul Perancangan Sistem *Watermarking* pada Citra *Digital* Menggunakan Metode DCT dan LSB [12]. Setelah melakukan penelitian ini, Khavid menarik kesimpulan bahwa proses penyisipan *watermarking* pada citra *digital* dengan menggunakan metode DCT dan LSB menunjukkan kinerja yang baik, yaitu dengan tidak adanya perbedaan antara citra ber-*watermark* dengan citra *host*. Namun dalam penelitian ini, penyisipan *watermark* pada citra *host* hanya berupa teks dan *file txt*, sehingga perubahan yang terjadi hanya pada 1 bit teks akan mengubah informasi *watermark* secara signifikan.
2. B. Yudi Dwiandiyanta dalam penelitiannya yang berjudul Perbandingan *Watermarking* Citra dengan Alihragam *Wavelet* dan *Discrete Cosine Transform* [3]. Setelah melakukan penelitian ini, Yudi menarik kesimpulan dari penelitiannya bahwa operasi-operasi pengolahan citra (penapisan, perubahan *brightness* dan penyamaan histogram) akan menyebabkan turunnya nilai korelasi *watermark* yang diurai. *Watermarking* citra yang dilakukan dengan alihragam *wavelet* dan DCT mempunyai keunggulan lebih kebal terhadap *noise*. Namun dalam penelitiannya, citra *watermark* yang digunakan hanya berupa citra *biner*.

2.2 Citra Digital

Citra didefinisikan sebagai fungsi $f(x,y)$ dimana x dan y adalah sebuah koordinat pada bidang dan amplitud dari f pada pasangan koordinat adalah intensitas atau sebuah tingkatan

keabu-abuan dari suatu citra pada titik tersebut. Jika x , y , dan nilai intensitas dari f tersebut bernilai diskrit, berhingga, citra tersebut dinamakan citra digital [4].

Citra digital dapat diwakili oleh sebuah matriks yang terdiri dari baris (N) dan kolom (M) dimana perpotongan antara kolom dan baris disebut *pixel*, yaitu elemen terkecil dari sebuah citra. *Pixel* mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai fungsi $f(x,y)$ pada koordinat citra (x,y) merupakan besar dari intensitas atau warna dari *pixel* pada titik itu [14].

$$\begin{array}{lll} N & = \text{jumlah baris} & 0 \leq x \leq N - 1. \\ M & = \text{jumlah kolom} & 0 \leq y \leq M - 1 \\ L & = \text{maksimal warna intensitas} & 0 \leq f(x,y) \leq L - 1 \end{array}$$

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

2.3 Watermark

2.3.1 Watermarking

Watermarking merupakan salah satu bentuk dari *steganography*, yang dapat diartikan sebagai suatu teknik penyembunyian data atau informasi ke dalam suatu data lainnya. Orang lain tidak menyadari kehadirannya. Jadi seolah-olah tidak ada perbedaan signifikan antara citra *host* dan citra *watermark*.

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* ke dalam media yang ingin dilindungi hak ciptanya [10].

Adapun manfaat utama secara umum dari *watermarking*, antara lain [14]:

- a. **Proteksi Hak Cipta**
Tujuan *watermark* dalam perlindungan hak cipta adalah sebagai bukti otentik atas hak kepemilikan pencipta atas konten yang dibuat atau diproduksi.
- b. ***Fingerprinting***
Tujuan *watermark* adalah mengidentifikasi setiap penggunaan dan distribusi suatu content.
- c. **Proteksi terhadap penggandaan (*copy protection*)**
Watermark bermanfaat untuk melindungi konten dari duplikasi dan pembajakan.
- d. **Autentikasi citra**
Watermark bermanfaat dalam proses autentikasi, sehingga modifikasi dari suatu citra dapat terdeteksi.

Jenis *watermark* berdasarkan persepsi manusia dibagi menjadi dua jenis [14]:

1. ***Visible Watermarking***
Merupakan jenis *watermark* dapat dilihat oleh panca indra manusia (mata telanjang). Sifat *watermark* ini sangat kuat bahkan sangat sulit dihapus keberadaannya walaupun tidak menjadi bagian dari image. Sebagai contohnya adalah logo transparan yang biasa diletakkan di pojok-pojok pada hasil foto.
2. ***Invisible Watermarking***
Watermark jenis ini tidak dapat dilihat oleh panca indera, yang bertujuan memberikan informasi yang bersifat rahasia dan untuk melindungi hak cipta orang lain dari orang yang tidak bertanggung jawab. Jenis *watermark* ini dapat dilihat melalui proses komputasi yaitu dengan cara mengekstrak gambar yang ter-*watermark*.

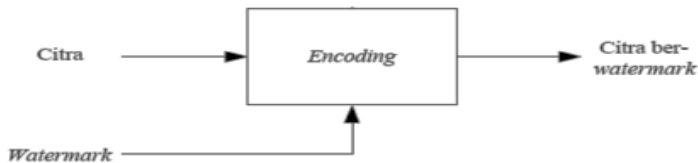
Klasifikasi teknik *watermarking* digital pada saat ini cukup banyak. Ada beberapa teknik *watermarking* berdasarkan domain kerjanya, yaitu [16]:

1. Teknik *watermarking* yang bekerja pada domain spasial (*spatial domain watermarking*).
Teknik ini bekerja dengan cara menyisipkan *watermark* secara langsung kedalam domain spasial dari suatu citra. Istilah domain spasial sendiri mengacu pada piksel-piksel penyusun sebuah citra. Teknik *watermarking* jenis ini beroperasi secara langsung pada piksel-piksel tersebut. Beberapa contoh teknik yang bekerja pada domain spasial adalah teknik penyisipan pada *Least Significant Bit* (LSB).
2. Teknik *watermarking* yang bekerja pada domain *transform* (*transform domain watermarking*).
Pada *transform domain watermarking* (sering juga disebut dengan *frequency domain watermarking*) ini penyisipan *watermark* dilakukan pada koefisien frekuensi hasil transformasi citra asalnya. Ada beberapa transformasi yang umum digunakan oleh para peneliti, yaitu: *Discrete Cosine Transform* (DCT), *Discrete Fourier Transform* (DFT), *Discrete Wavelet Transform* (DWT) maupun *Discrete Laguerre Transform* (DLT).
3. Teknik *watermarking* yang bekerja pada kedua domain diatas (*hybrid techniques watermarking*).
Teknik *watermarking* jenis ini bekerja dengan menggabungkan kedua teknik diatas. Pada teknik ini biasanya penanaman *watermark* dilakukan pada domain frekuensi beberapa bagian citra yang dipilih berdasarkan karakteristik spasial citra tersebut.

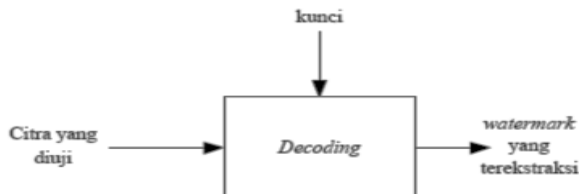
2.3.2 *Framework Watermarking*

Jika *watermark* merupakan sesuatu yang disisipkan, maka *watermarking* merupakan proses penyisipan *watermark* tersebut. Secara umum *framework* sebuah algoritma *watermarking* tersusun atas dua bagian, yaitu algoritma penyisipan *watermark*

(*encoder*) dan algoritma pendeteksian *watermark* (*decoder*). Algoritma penyisipan *watermark* (*encoder*) adalah algoritma yang menangani bagaimana sebuah *watermark* ditanamkan pada media induknya, sedangkan algoritma pendeteksian *watermark* adalah algoritma yang menentukan apakah di dalam sebuah media *digital* terdeteksi *watermark* yang sesuai atau tidak [17].



Gambar 2.1 Proses penyisipan *watermark* pada citra *digital*



Gambar 2.2 Proses pendeteksian *watermark* pada citra *digital*

Label *watermark* adalah suatu data/informasi yang akan dimasukkan ke dalam data *digital* yang ingin di-*watermark*. Ada 2 jenis label yang dapat digunakan, yakni teks biasa, logo atau citra atau suara. Kelemahan dari label teks adalah kesalahan pada satu bit saja dapat menghasilkan hasil yang berbeda dengan teks asli. Sedangkan logo atau citra atau suara, yakni kesalahan pada beberapa bit masih dapat memberikan persepsi yang sama dengan aslinya oleh pendengaran maupun penglihatan kita, tetapi kerugiannya adalah jumlah data yang cukup besar [17].

2.4 Transformasi Citra

Transformasi citra dapat diartikan sebagai perubahan bentuk suatu citra. Perubahan bentuk tersebut dapat berupa perubahan geometri piksel seperti perputaran (rotasi), pergeseran (translasi), penskalaan, dan lain sebagainya atau dapat juga berupa perubahan ruang (domain) citra ke domain lainnya.

Transformasi citra merupakan pokok bahasan yang sangat penting dalam pengolahan citra. Citra hasil proses transformasi dapat dianalisis kembali, diinterpretasikan, dan dijadikan acuan untuk melakukan pemrosesan selanjutnya. Tujuan diterapkannya transformasi citra adalah untuk memperoleh informasi (*feature extraction*) yang lebih jelas yang terkandung dalam suatu citra.

Melalui proses transformasi, suatu citra dapat dinyatakan sebagai kombinasi linear dari sinyal dasar (*basic signals*) yang sering disebut dengan fungsi basis (*basis fuction*). Suatu citra yang telah mengalami transformasi dapat diperoleh kembali dengan menggunakan transformasi balik (*invers transformation*). Pada transformasi citra terdapat berbagai macam transformasi diantaranya, *Fast Fourier Transform* (FFT), *Discrete Cosine Transform* (DCT), *Discrete Wavelet Transform* (DWT), dan lain-lain [11].

2.5 Discrete Wavelet Transform (DWT)

2.5.1 Wavelet

Gelombang (*wave*) adalah sebuah fungsi yang bergerak naik turun ruang dan waktu secara periodik. Sedangkan *wavelet* merupakan gelombang yang dibatasi atau dapat dikatakan sebagai gelombang pendek. *Wavelet* mengkonsentrasikan energinya dalam ruang dan waktu sehingga cocok untuk menganalisis sinyal yang sifatnya sementara saja.

Wavelet pertama kali digunakan dalam analisis dan pemrosesan digital dari sinyal gempa bumi, yang tercantum dalam literatur oleh A. Grossman dan J. Morlet. Penggunaan *wavelet* pada saat ini sudah semakin berkembang dengan munculnya area sains terpisah yang berhubungan dengan analisis

wavelet dan teori transformasi *wavelet*. Dengan munculnya area sains ini *wavelet* mulai digunakan secara luas dalam filtrasi dan pemrosesan data, pengenalan citra, sintesis dan pemrosesan berbagai variasi sinyal, kompresi dan pemrosesan citra, dan lain-lain [16].

2.5.2 Transformasi *Wavelet*

Transformasi *wavelet* adalah sebuah transformasi matematika yang digunakan untuk menganalisis sinyal bergerak. Sinyal bergerak ini dianalisis untuk didapatkan informasi spektrum frekuensi dan waktunya secara bersamaan. Salah satu seri pengembangan transformasi *wavelet* adalah *Discrete Wavelet transform* (DWT) [1].

Transformasi sinyal merupakan bentuk lain dari penggambaran sinyal yang tidak mengubah isi informasi dalam sinyal tersebut. Transformasi *wavelet* (*wavelet transform*) menyediakan penggambaran frekuensi waktu dari sinyal. Pada awalnya, transformasi *wavelet* digunakan untuk menganalisis sinyal bergerak (*non-stationary 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 frekuensi yang berbeda dianalisis menggunakan resolusi yang berbeda. Resolusi dari sinyal merupakan ukuran jumlah informasi di dalam sinyal yang dapat berubah melalui operasi filterisasi [1].

Transformasi *wavelet* selain mampu memberikan informasi frekuensi yang muncul, juga dapat memberikan informasi tentang skala atau waktu. *Wavelet* dapat digunakan untuk menganalisa suatu bentuk gelombang (sinyal) sebagai kombinasi dari waktu (skala) dan frekuensi. Selain itu perubahan sinyal pada suatu posisi tertentu tidak akan berdampak banyak terhadap sinyal pada posisi-posisi yang lainnya. Dengan *wavelet* suatu sinyal dapat disimpan lebih efisien dibandingkan dengan fourier dan lebih baik

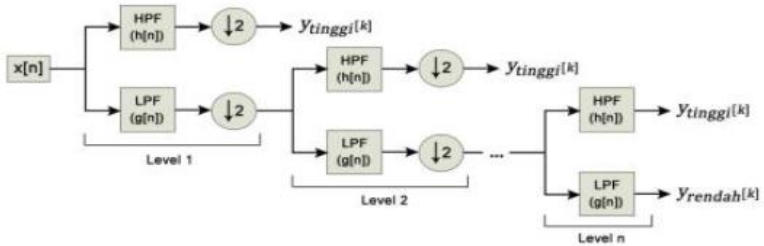
dalam hal melakukan suatu pendekatan untuk memperoleh nilai yang sedekat mungkin terhadap real-word signal. Transformasi *wavelet* dibagi menjadi dua bagian besar, yaitu *Continuous Wavelet Transform* (CWT) dan *Discrete Wavelet Transform* (DWT) [9].

2.5.3 *Discrete Wavelet Transform* (DWT)

Transformasi *wavelet* diskrit atau *Discrete Wavelet Transform* (DWT) secara umum merupakan dekomposisi citra pada frekuensi *subband* citra tersebut. Komponen *subband* transformasi *wavelet* dihasilkan dengan cara penurunan level dekomposisi. Dalam transformasi *wavelet* diskrit, penggambaran sebuah skala waktu sinyal digital didapatkan dengan menggunakan teknik filterisasi *digital*. Secara garis besar proses dalam teknik ini adalah dengan melewati sinyal yang akan dianalisis pada filter dengan frekuensi dan skala yang berbeda [1].

Implementasi transformasi *wavelet* diskrit dapat dilakukan dengan cara melewati sinyal ke dalam dua filterisasi DWT yaitu *highpass filter* (HPF) dan *lowpass filter* (LPF) agar frekuensi dari sinyal tersebut dapat dianalisis kemudian melakukan *downsampling* pada keluaran masing-masing filter.

Analisis sinyal dilakukan terhadap hasil filterisasi *highpass filter* (HPF) dan *lowpass filter* (LPF) di mana HPF digunakan untuk menganalisis frekuensi tinggi dan LPF digunakan untuk menganalisis frekuensi rendah. Analisis terhadap frekuensi dilakukan dengan cara menggunakan resolusi yang dihasilkan setelah sinyal melewati filterisasi. Analisis frekuensi yang berbeda dengan menggunakan resolusi yang berbeda inilah yang disebut dengan *multi-resolution analysis* [5].



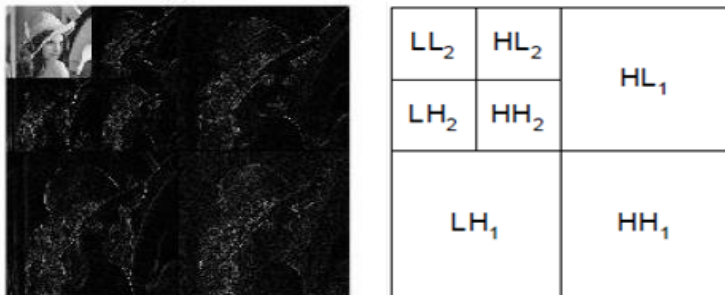
Gambar 2.3. Dekomposisi Wavelet Diskrit 1Dimensi [5]

Proses dekomposisi ini dapat melalui satu atau lebih tingkatan. Dekomposisi satu tingkat ditulis dengan ekspresi matematika berikut [1]:

$$y_{tinggi}[k] = \sum_n x[n]h[2k - n] \quad (2.1)$$

$$y_{rendah}[k] = \sum_n x[n]g[2k - n] \quad (2.2)$$

$y_{tinggi}[k]$ dan $y_{rendah}[k]$ yang merupakan hasil dari *highpass filter* dan *lowpassfilter*, $x[n]$ merupakan sinyal asal, $h[n]$ adalah *highpass filter*, dan $g[n]$ adalah *lowpass filter*. Untuk dekomposisi lebih dari satu tingkat, prosedur pada persamaan (2.1) dan (2.2) dapat digunakan pada masing-masing tingkatan [1].



Gambar 2.4 Transformasi Wavelet Diskrit 2D Level 2

Untuk transformasi citra dua dimensi pada Gambar 2.4, prosedur dekomposisi level pertama terdiri dari citra satu dimensi yang di-*filter* pada arah mendatar (baris) kemudian diikuti oleh citra satu dimensi yang di-*filter* pada arah tegak (kolom) yang diutilisasi dengan menggunakan *filter* tapis rendah (*lowpass filter*) dan *filter* tapis tinggi (*highpass filter*).

2.5.4 Inverse Discrete Wavelet Transform (IDWT)

Secara umum teknik *watermarking* dapat dibedakan menjadi dua, yaitu *non blind watermarking* dan *blind watermarking*. *Non blind watermarking* merupakan teknik *watermarking* yang memerlukan citra *host* dan citra *watermark* untuk mengekstrak *watermark*. Sedangkan *blind watermarking* merupakan teknik *watermarking* yang tidak memerlukan citra *host* untuk melakukan ekstraksi. Dalam penelitian ini menggunakan teknik *non blind watermarking*.

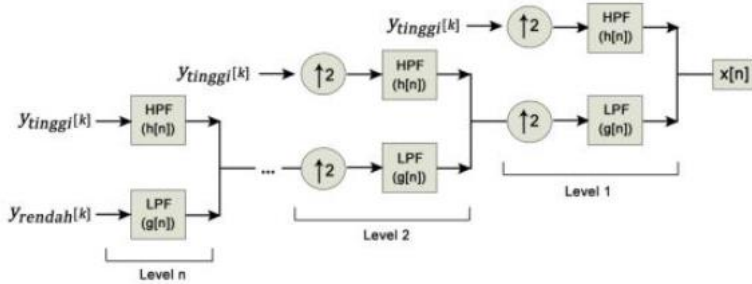
Dengan menggunakan koefisien DWT ini maka dapat dilakukan proses *Inverse Discrete Wavelet transform* (IDWT) untuk merekonstruksi menjadi sinyal asal melalui persamaan berikut [1]:

$$x[n] = \sum_k (y_{tinggi}[k]h[-n + 2k] + y_{rendah}[k]g[-n + 2k]) \quad (2.3)$$

Proses rekontruksi merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi. DWT menganalisis sinyal pada frekuensi berbeda dengan resolusi yang berbeda melalui dekomposisi sinyal sehingga menjadi detail informasi dan taksiran kasar. DWT bekerja pada dua kumpulan fungsi yang disebut fungsi penskalaan dan fungsi *wavelet* yang masing- masing berhubungan dengan *lowpass filter* dan *highpass filter*.

Proses rekonstruksi ini sepenuhnya merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi, yang mana untuk setiap rekonstruksi pada masing-masing tingkatan dapat menggunakan persamaan matematika.

Proses rekonstruksi wavelet untuk mendapatkan sinyal asal ditunjukkan pada gambar berikut:



Gambar 2.5 Rekonstruksi Wavelet Diskrit 1 Dimensi [5]

2.6 PSNR

Peak Signal-to-Noise Ratio (PSNR) adalah salah satu metode yang cukup populer digunakan dalam pengukuran kualitas video maupun citra secara objektif. Metode ini menggunakan sinyal citra sebagai parameter objektif. Metode ini membandingkan antara kualitas citra yang telah disisipi *watermark* (citra ber-*watermark*) dengan citra asal (citra *host*). Menurut Syahbana, dkk, PSNR dirumuskan sebagai berikut [15] :

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{2^M - 1}{\text{MSE}} \right)^2 \quad (2.4)$$

Dimana, $2^M - 1$ adalah maksimum nilai piksel untuk M-bit citra. MSE merupakan kepanjangan dari *Mean Square Error*. Sebagai contoh, untuk citra dengan resolusi (x,y) piksel, maka MSE didefinisikan dalam persamaan berikut [15] :

$$\text{MSE} = \frac{1}{XYT} \sum_{t=1}^T \sum_{y=1}^Y \sum_{x=1}^X [i(x, y, t) - i'(x, y, t)]^2 \quad (2.5)$$

dengan:

- X = panjang/baris;
- Y = lebar/kolom;
- T = banyaknya layer citra *host* (RGB) ;
- i = matriks citra awal / citra *host*;
- i' = matriks citra hasil / citra ber-*watermark*

Interpretasi nilai PSNR terhadap kualitas citra dapat dikatakan baik sekali jika nilai PSNR lebih besar dari 33 dB dan dikatakan buruk jika nilai PSNR kurang dari 30 dB [15].

BAB III

METODOLOGI PENELITIAN

Tahapan-tahapan yang akan dilalui dalam pelaksanaan Tugas akhir ini adalah:

3.1 Studi Literatur

Pada tahap ini dilakukan pengumpulan bahan-bahan referensi yang menunjang proses penelitian, yaitu yang berhubungan dengan *Invisible watermarking*.

3.2 Analisis

Analisis yang dibahas meliputi penyisipan *watermark* dan rekonstruksi *watermark*.

3.2.1 Proses Penyisipan *Watermark* pada Citra *Digital*

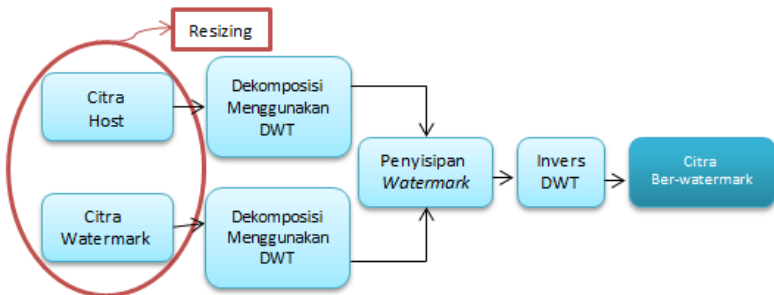
Langkah-langkah yang dilakukan dalam proses ini, sebagaimana yang terlihat pada Gambar 3.1.

- a. Pemilihan citra *host* dan citra *watermark*, jenis citra yang dapat digunakan adalah citra RGB dengan ukuran citra *watermark* $\leq$ citra *host*.
- b. Proses dekomposisi citra *host* menggunakan DWT. Proses dekomposisi ini dilakukan dengan DWT keluarga haar. Dekomposisi citra *digital* menghasilkan LL, LH, HL, dan HH. Dekomposisi Citra *Watermark* menggunakan DWT. Sama halnya dengan citra *host*, citra *watermark* ini didekomposisi menggunakan DWT keluarga haar.
- c. Setelah citra *host* dan citra *watermark* didekomposisi dalam DWT, citra *watermark* disisipkan ke dalam rentang frekuensi LL.

- d. Menyisipkan w ke dalam rentang frekuensi LL :

$$f'_{LL}(m, n) = f_{LL}(m, n) + \alpha \cdot w_{LL}(m, n) \quad (3.1)$$
 dengan $m, n = 1, \dots, L$
 Dimana $f_{LL}(m, n)$ merupakan koefisien yang dipilih, α merupakan kekuatan penyisipan *watermark* atau dapat dikatakan sebagai faktor skala persentase dari citra *host* dan citra *watermark* pada citra ber-*watermark* yang dibentuk.
- e. Menjalankan *Inverse Discrete Wavelet Transform* (IDWT) untuk membentuk citra ber-*watermark*.

Proses Penyisipan Watermark



Gambar 3.1 Proses Penyisipan *Watermark* [1]

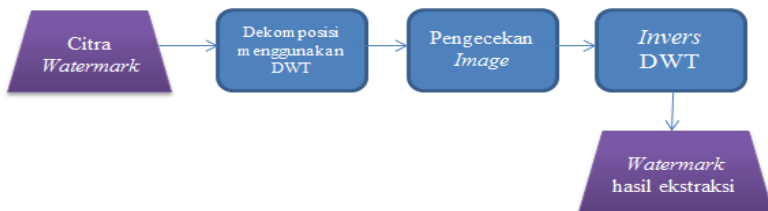
3.2.2 Proses Rekonstruksi *Watermark* pada Citra *Digital*

Langkah - langkah rekonstruksi *watermark* adalah :

- Citra ber-*watermark* didekomposisi dalam DWT.
- Memilih frekuensi LL dari hasil DWT.
- Melakukan pengurangan frekuensi LL citra ber-*watermark* dengan frekuensi LL citra *host* untuk menghasilkan frekuensi LL citra *watermark*.
- Menjalankan *Inverse Discrete Wavelet Transform* (IDWT) untuk membentuk citra *watermark*.

Secara umum proses rekonstruksi *watermark* ini merupakan kebalikan dari proses penyisipan *watermark*.

Proses Rekonstruksi



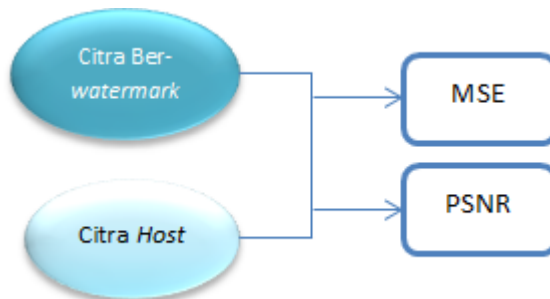
Gambar 3.2 Proses Rekonstruksi *Watermark* [1]

3.3 Implementasi

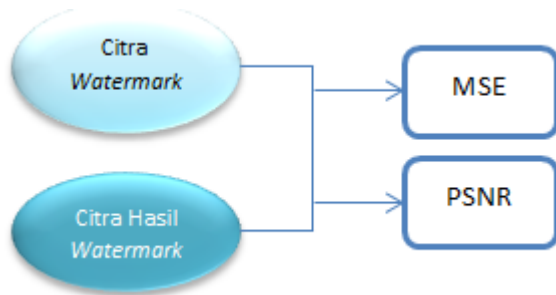
Pada tahap ini dilakukan implementasi (pengkodean) berdasarkan rancangan yang dilakukan pada tahap sebelumnya. Dibangun suatu aplikasi *watermarking* menggunakan metode *Invisible Watermarking* pada domain DWT. Modul yang dibangun meliputi proses penyisipan dan rekonstruksi. Bahasa pemrograman yang digunakan pada tahap implementasi ini yaitu bahasa pemrograman Matlab R2013a.

3.4 Pengujian

- 1) Pada tahap ini akan dilakukan pengujian terhadap metode yang digunakan, untuk mengetahui kualitas dari metode ini akan dilakukan pengujian PSNR. Pengujian PSNR ini dilakukan pada citra yang telah disisipi *watermark* (citra ber-*watermark*), yaitu dengan membandingkan citra ber-*watermark* dengan citra *host*.
- 2) Pengujian kedua dilakukan untuk mengetahui ketahanan metode ini terhadap *noise*. Untuk mengetahui ketahanan metode terhadap *noise*, maka akan diberi *noise* pada citra ber-*watermark*, selanjutnya akan direkonstruksi.



Gambar 3.3 Proses Pengujian



Gambar 3.4 Proses Pengujian

3.5 Penarikan Kesimpulan

Pada tahap ini adalah penarikan kesimpulan berdasarkan tabel dari hasil pengujian dan pemberian saran terkait kekurangan hasil penelitian untuk pengembangan berikutnya.

3.6 Penulisan Tugas Akhir

Pada tahap terakhir ini akan dibuat laporan dari hasil penelitian yang telah dilakukan pada tahap-tahap sebelumnya.

BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dibahas mengenai perancangan dan implementasi sistem dimulai dari pembahasan proses pengambilan citra *host* dan citra *watermark*, kemudian citra *host* dan citra *watermark* diubah ukurannya agar sama. Proses penyisipan *watermark* ini meliputi proses dekomposisi citra *host* dan citra *watermark* menggunakan metode *Discrete Wavelet Transform* (DWT), serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1 Analisis Sistem

Dalam sub bab ini akan dijelaskan proses analisis dalam membangun perangkat lunak *invisible watermark*, mulai dari proses pengambilan citra *host* dan citra *watermark*, *resize* citra *host* dan citra *watermark*, proses dekomposisi citra *host* dan citra *watermark* menggunakan metode *Discrete Wavelet Transform* (DWT), dan proses rekonstruksi, serta pengujian hasil dari sistem.

4.1.1 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang akan dibangun mempunyai beberapa kebutuhan fungsional, yaitu :

1. Dapat digunakan untuk memilih citra *host* dan citra *watermark*.
2. Dapat *resizing* citra *host* dan citra *watermark*.
3. Dapat melakukan proses *invisible watermarking* dengan *wavelet*.
4. Dapat melakukan proses rekonstruksi dari citra ber-*watermark* menjadi citra hasil rekonstruksi.

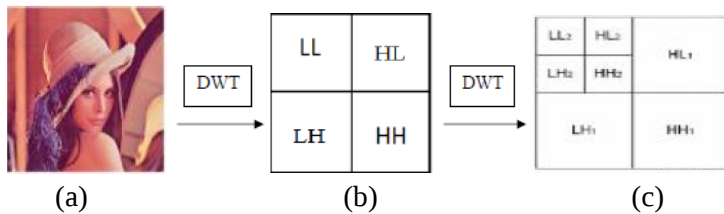
Untuk memenuhi beberapa kebutuhan fungsional tersebut, perangkat lunak yang dibangun mempunyai kemampuan sebagai berikut :

a. Akuisisi citra

Pada penelitian ini, proses akuisisi citra digital untuk citra *host* dan citra *watermark* berupa citra RGB dengan format PNG, BPM, dan JPG. Citra *host* dan citra *watermark* yang telah dipilih akan mengalami proses penyesuaian resolusi (*resize*).

b. Dekomposisi DWT

Pada penelitian ini setelah proses akuisisi citra untuk citra *host* dan citra *watermark*, maka tahap berikutnya adalah mendekomposisi citra *host* dan citra *watermark* menggunakan DWT dan untuk filter keluarga *wavelet* yang digunakan adalah haar. Pada proses ini citra yang digunakan adalah citra 2 dimensi. Proses dekomposisi terhadap citra digital dilakukan dalam dua tingkat.



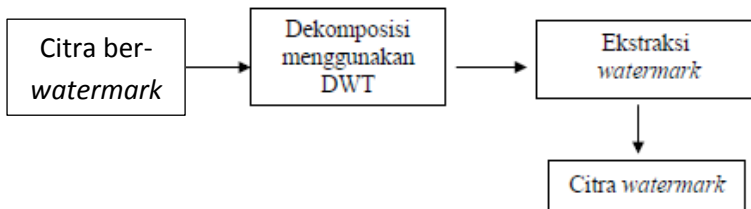
Gambar 4.3. (a) Citra digital (b) Dekomposisi DWT level 1 (c) Dekomposisi DWT level 2

c. Penyisipan citra

Setelah citra *host* dan citra *watermark* di dekomposisikan dengan DWT, maka tahap selanjutnya adalah menyisipkan citra *watermark* ke dalam citra *host*. Pemilihan koefisien aproksimasi dari citra *host* dan citra *watermark* berfungsi untuk digunakan dalam proses penyisipan. Pada proses penyisipan, koefisien penggabungan pada koefisien aproksimasi citra *host* ditambahkan dengan perkalian parameter dengan koefisien aproksimasi citra *watermark*. Proses ini mendapatkan koefisien aproksimasi baru dari citra ber-*watermark*.

d. Proses rekonstruksi *watermark*

Setelah mendapatkan citra ber-*watermark*, proses selanjutnya adalah proses pengekstrakan *watermark*. Proses ini berfungsi untuk mendapatkan kembali citra *watermark* dari citra yang telah disisipi (citra ber-*watermark*).



Gambar 4.4. Proses pengrekonstruksian *watermark*

4.1.2 Analisis Kebutuhan Sistem

Kebutuhan perancangan dari sistem yang akan dibangun meliputi perangkat keras dan perangkat lunak komputer. Perangkat lunak ini dibangun menggunakan software MATLAB R2013a baik dari desain interface, sistem *toolbox computer vision* dan pengolahan citra digital.

Tabel 4.1. Tabel kebutuhan sistem

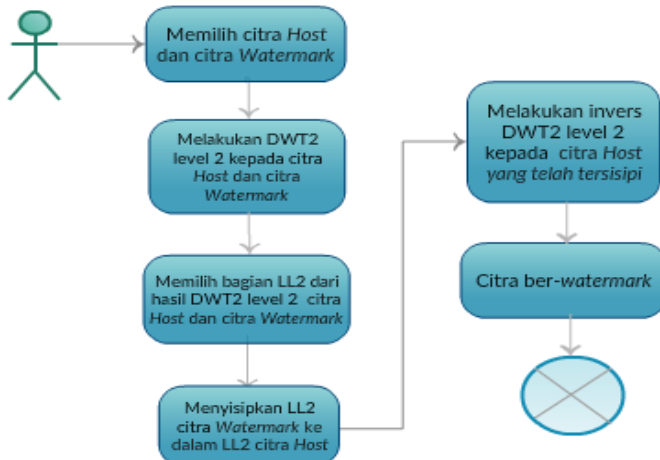
Perangkat Keras	Prosesor : Intel® Core™ i3-3110M CPU @2.40GHz ~2.40GHz
	RAM Memory : 2 GB
Perangkat Lunak	Sistem Operasi : Windows 8.1 Pro 64-bit
	Tools : MATLAB R2013a

4.2 Perancangan Sistem

Setelah analisis sistem, kemudian dilanjutkan dengan perancangan sistem. Perancangan sistem tersebut meliputi perancangan data sistem, perancangan *class* sistem, perancangan proses algoritma sistem, dan perancangan antarmuka sistem.

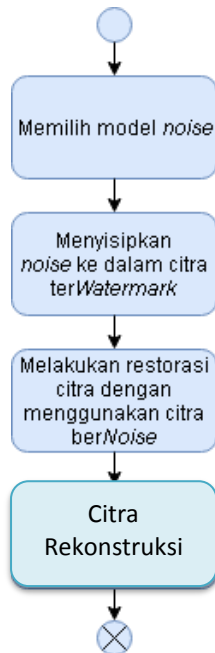
4.2.1 Gambaran Sistem

Pada subbab ini akan disajikan diagram alir sistem secara umum berupa proses berjalannya sistem secara umum yang terbagi 3 tahap, yaitu tahap penyisipan *watermark*, tahap restorasi *watermark*, dan yang terakhir adalah tahap pengujian. Berikut adalah diagram alir sistem yang disajikan pada gambar 4.5, gambar 4.6, dan gambar 4.7

**Gambar 4.5.** Diagram alir proses *watermark* citra *host*



Gambar 4.6. Diagram alir proses rekonstruksi *watermark*



Gambar 4.7. Diagram alir proses uji dengan *noise*

4.2.2 Penjelasan Umum Sistem

Proses algoritma ini dimulai dengan pemilihan citra *Host* dan citra *Watermark*, kemudian kedua citra akan diproses untuk dilakukan penyisipan *watermark* ke dalam citra *host* menggunakan DWT2 dengan keluarga Haar. Hasil proses tersebut adalah citra ber-*watermark*, yang kemudian akan dilakukan proses rekonstruksi *watermark* dan pengujian, menggunakan MSE, PSNR, dan juga pengujian terhadap *noise*.

Pada sub bab selanjutnya akan dijelaskan algoritma tiap proses dalam metode yang digunakan.

4.2.3 Perancangan Proses Algoritma

Perancangan proses algoritma ini secara garis besar di bagi dalam tiga tahapan, yaitu proses *watermarking*, proses rekonstruksi *watermark*, dan pengujian citra. Pada awalnya citra *host* dan citra *watermark* dipilih secara manual. Selanjutnya dari citra tersebut akan disimpan untuk proses *watermarking*.

4.2.3.1 Perancangan Proses Watermarking

Pada proses ini, citra *host* dan citra *watermark* yang terpilih terlebih dahulu akan dilakukan dekomposisi DWT2. Metode yang digunakan dalam DWT2 adalah metode Haar. Dalam keluarga Haar, proses DWT akan menggunakan LPF (*low-pass filter*) berupa $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ dan HPF (*high-pass filter*) berupa $\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$. Hal pertama yang dilakukan adalah membangun matriks H, dimana matriks H adalah gabungan dari matriks LPF dan HPF. Dapat ditulis bahwa [13]:

$$H = \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}$$

dimana:

H : Matriks Haar
 LPF : Matriks LPF
 HPF : Matriks HPF

Proses DWT2, dapat didefinisikan sebagai berikut[13]:

$$A = H B H^T$$

$$A = \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix} B \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}^T$$

$$A = \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix} B \begin{bmatrix} [LPF]^T & [HPF]^T \end{bmatrix}$$

$$A = \begin{bmatrix} [LPF] B [LPF]^T & [LPF] B [HPF]^T \\ [HPF] B [LPF]^T & [HPF] B [HPF]^T \end{bmatrix} \quad (4.1)$$

dimana:

- A : Matriks hasil DWT2
 B : Citra yang akan di-DWT2

Dapat dilihat bahwa dalam persamaan (4.1), Matriks hasil DWT2 dapat dibagi menjadi 4 bagian, yaitu LL yang merupakan hasil dari $[LPF]B[LPF]^T$, LH hasil dari $[LPF]B[HPF]^T$, HL hasil dari $[HPF]B[LPF]^T$, dan HH hasil dari $[HPF]B[HPF]^T$.

Dalam proses *watermarking*, citra yang akan disisipkan adalah citra hasil DWT2 dan citra *watermark*, pada bagian LL, begitu juga dengan citra *host* yang akan digunakan sebagai penampung adalah bagian LL. Dengan menggunakan parameter sebagai pengali, persamaan *watermarking* dapat ditulis menjadi:

$$LL_{hasil} = LL_{host} + (parameter \times LL_{watermark}) \quad (4.2)$$

Setelah didapatkan *LLhasil*, dalam hasil DWT2 citra *host*, *LLhost* akan diganti dengan *LLhasil*. Bagian DWT2 milik *watermark* dan *LLhost* disimpan sebagai *key* dalam rekonstruksi. Hal selanjutnya adalah *invers* DWT2. Secara garis besar, fungsi *invers* DWT2 adalah membalik urutan pengali, sebab dalam kasus ini, H^T memiliki nilai yang sama dengan H^{-1} sehingga akan menghasilkan [13]:

$$B = H^T B H$$

$$B = \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}^T \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix} B \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}^T \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}$$

$$B = \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}^{-1} \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix} B \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}^{-1} \begin{bmatrix} [LPF] \\ [HPF] \end{bmatrix}$$

$$B = 1 \quad B \quad 1$$

$$B = B$$

Dengan melakukan *invers* DWT2 kepada citra DWT2 yang telah tersisip *watermark*, citra ber-*watermark* berhasil didapatkan.

4.2.3.2 Perancangan Proses Rekonstruksi

Pada proses ini, citra *watermark* akan direkonstruksi dari citra ber-*watermark*. Hal pertama yang dilakukan adalah melakukan DWT2 kepada citra ber-*watermark*, setelah itu ambil bagian *LL* untuk dilakukan proses rekonstruksi.

Proses rekonstruksi dilakukan dengan parameter yang sama pada persamaan 4.2, dan juga *key* yang telah disimpan. *Key* tersebut berisi *LLhost* dan bagian selain *LLwatermark* dari DWT2 citra *watermark*. Langkah pertama adalah menghitung *LLrekonstruksi* dengan cara :

$$LLrekonstruksi = \frac{LLhasil - LL2host}{parameter}$$

Parameter < 1 (semakin kecil semakin baik)

Setelah didapatkan *LLrekonstruksi*, *LLrekonstruksi* akan dimasukkan ke dalam *key* yang berupa hasil DWT2 citra *watermark*. Setelah itu akan dilakukan proses *invers*, dan citra *watermark* hasil rekontruksi didapatkan.

4.2.3.3 Pengujian Citra dengan MSE

Pada proses ini citra ber-*watermark* akan dibandingkan dengan citra *host* berdasarkan MSE (*Mean Squared Error*). MSE dihitung dari rata – rata setiap perbedaan pixel yang bersesuaian. Rumus MSE dapat ditulis sebagai berikut [15]:

$$MSE = \frac{1}{XYT} \sum_{t=1}^T \sum_{y=1}^Y \sum_{x=1}^X [i(x, y, t) - i'(x, y, t)]^2$$

dimana:

- X = panjang/baris;
- Y = lebar/kolom;
- T = banyaknya layer citra *host* (RGB) ;
- i = matriks citra awal / citra *host*;
- i' = matriks citra hasil / citra ber-*watermark*

4.2.3.4 Pengujian Citra dengan PSNR

Pada proses ini citra ber-*watermark* akan dilihat nilai PSNR (*Peak Signal to Noise Ratio*) nya, semakin besar nilai PSNR maka kemiripan citra akan semakin baik. Secara umum, PSNR didefinisikan sebagai fungsi:

$$PSNR = 10 \log_{10} \left(\frac{\max_{pixel}^2}{MSE} \right)$$

4.2.3.5 Pengujian Citra Terhadap Noise

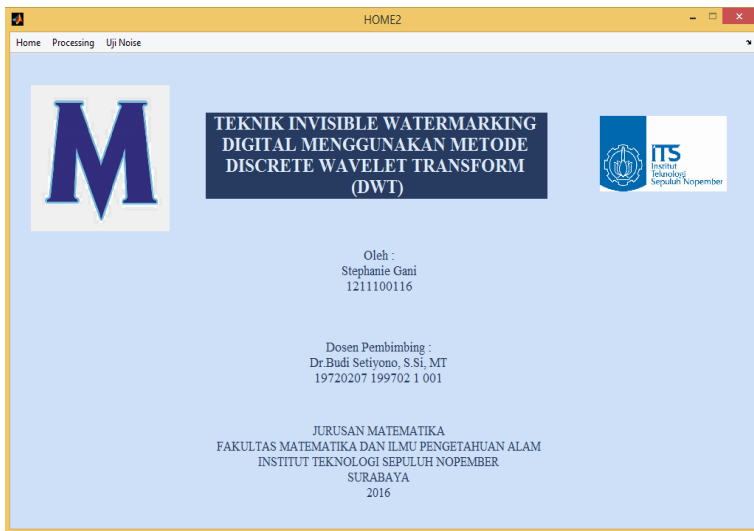
Pada proses ini, citra ber-*watermark* akan diberikan suatu *noise* tertentu. Setelah *noise* diberikan, citra *watermark* akan direkonstruksi dari citra ber-*noise* tersebut. Hasil rekonstruksi akan dilihat, apakah citra *watermark* masih dapat dikenali atau tidak.

4.2.4 Perancangan Antar Muka Sistem

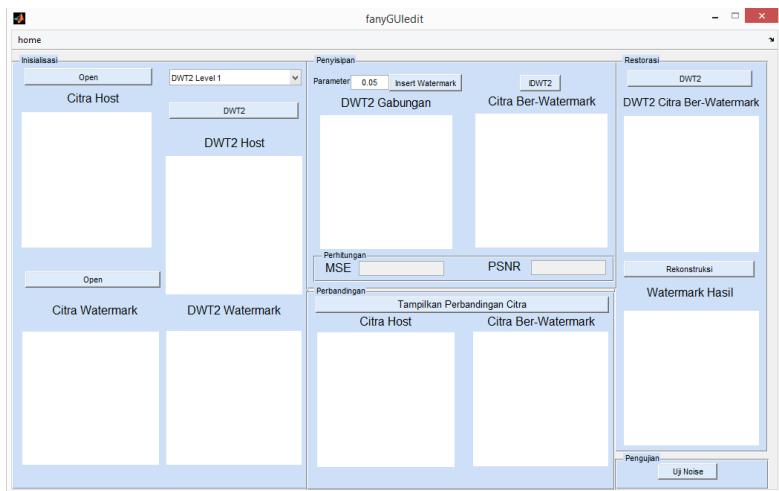
Desain antar muka sistem dibutuhkan agar pengguna dengan mudah mengoperasikan perangkat lunak yang dibangun.

4.2.4.1 Perancangan Halaman Utama

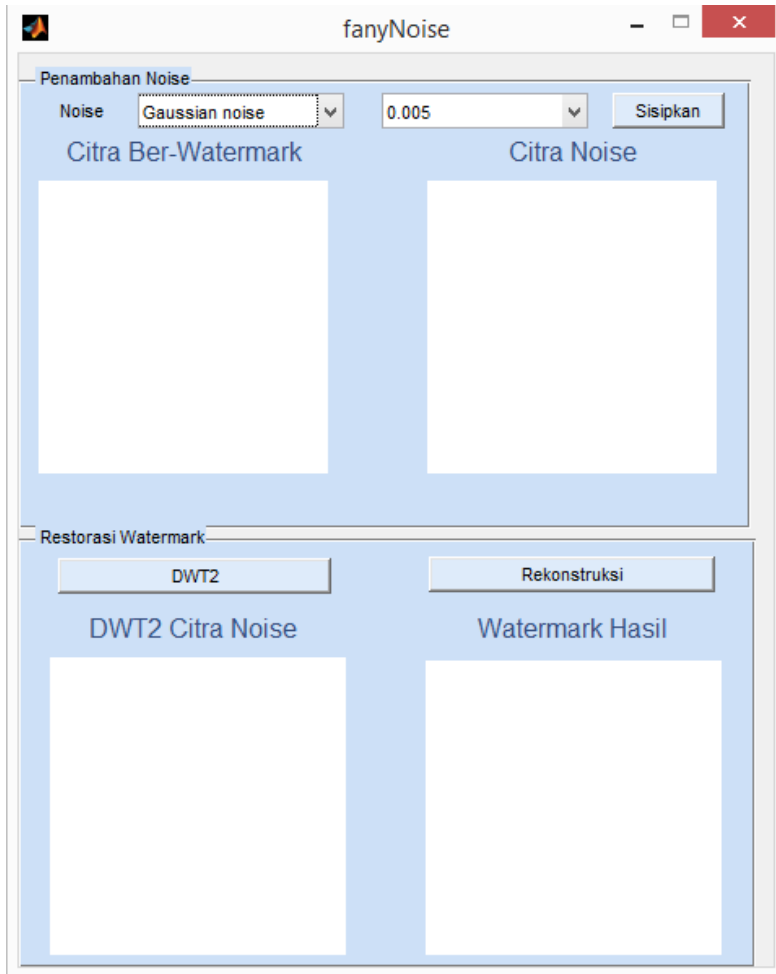
Halaman utama ini memiliki desain antar muka yang ditunjukkan pada Gambar 4.10. Pada halaman ini dilakukan proses pemilihan citra *host* dan citra *watermark*, menampilkan dekomposisi citra *host* dan citra *watermark*, penyisipan citra, menampilkan nilai MSE dan PSNR, rekonstruksi citra *watermark*, dan uji terhadap *noise*.



Gambar 4.9. Antar muka halaman awal.



Gambar 4.10. Antar muka halaman proses *invisible watermarking*.



Gambar 4.11. Antar muka halaman proses uji terhadap *noise*.

4.3 Implementasi Sistem

4.3.1 Implementasi *Input* Citra

Proses *input* citra digunakan untuk mendapatkan citra digital yang sebelumnya telah tersimpan. Proses *input* citra ini dilakukan dua kali yaitu untuk citra *host* dan citra *watermark*. Proses ini tidak dilakukan secara otomatis. Namun, membutuhkan interaksi pengguna untuk mencari dan mengambil citra yang dibutuhkan di dalam media penyimpanan.

Proses baca citra diimplementasikan menjadi sebuah program ke dalam fungsi berikut ini :

```
[filename pathname] =
uigetfile({'*.jpg'; '*.bpm'; '*.png';}, 'File
Selector');
alamat = strcat(pathname, filename);
asli = imread(alamat);
asli = imresize(asli, [256 256])
```

Fungsi diatas digunakan untuk meng-*input* dan me-*resize* citra *host* dijalankan adalah ketika tombol “Citra *Host*” dipilih. Kode program selengkapnya disajikan pada Lampiran A.

```
[filename pathname] =
uigetfile({'*.jpg'; '*.bpm'; '*.png';}, 'File
Selector');
alamat = strcat(pathname, filename);
watermark = imread(alamat);
watermark = imresize(watermark, [256 256])
```

Fungsi diatas digunakan untuk meng-*input* dan me-*resize* citra *watermark* dijalankan adalah ketika tombol “Citra *Watermark*” dipilih. Kode program selengkapnya disajikan pada Lampiran A.

4.3.2 Proses Dekomposisi DWT2 pada Citra

Setelah proses *input* citra untuk citra *host* dan citra *watermark*, maka tahap berikutnya adalah mendekomposisi citra *host* dan citra *watermark* menggunakan DWT dan untuk filter keluarga *wavelet* yang digunakan adalah haar. Pada proses ini citra yang digunakan adalah citra 2 dimensi.

```
%dwt merah
[cAr,cHr,cVr,cDr] = dwt2(asli(:, :, 1), 'haar');
%dwt level 2 merah
[cAr2,cHr2,cVr2,cDr2] = dwt2(cAr, 'haar');
%dwt hijau
[cAg,cHg,cVg,cDg] = dwt2(asli(:, :, 2), 'haar');
%dwt level 2 hijau
[cAg2,cHg2,cVg2,cDg2] = dwt2(cAg, 'haar');
%dwt biru
[cAb,cHb,cVb,cDb] = dwt2(asli(:, :, 3), 'haar');
%dwt level 2 biru
[cAb2,cHb2,cVb2,cDb2] = dwt2(cAb, 'haar');
cA = [cA2 cH2; cV2 cD2];
```

```
%dwt merah
[wAr,wHr,wVr,wDr] =
dwt2(watermark(:, :, 1), 'haar');
%dwt merah level 2
[wAr2,wHr2,wVr2,wDr2] = dwt2(wAr, 'haar');
%dwt hijau
[wAg,wHg,wVg,wDg] =
dwt2(watermark(:, :, 2), 'haar');
%dwt hijau level 2
[wAg2,wHg2,wVg2,wDg2] = dwt2(wAg, 'haar');
%dwt biru
[wAb,wHb,wVb,wDb] =
dwt2(watermark(:, :, 3), 'haar');
%dwt biru level 2
[wAb2,wHb2,wVb2,wDb2] = dwt2(wAb, 'haar');
wA = [wA2 wH2; wV2 wD2];
```

Fungsi diatas adalah fungsi dekomposisi DWT2 pada citra *host* dan citra *watermark* dengan format RGB. Fungsi akan dijalankan ketika tombol “DWT2” dipilih. Kode program selengkapnya disajikan pada Lampiran B.

4.3.3 Proses Penyisipan Citra

Setelah dekomposisi DWT2 pada citra *host* dan citra *watermark*, maka tahap selanjutnya adalah menyisipkan citra *watermark* ke dalam citra *host*.

```

cWA2 = cAr2 + wAr2 * parameter;
cWA2 = cAg2 + wAg2 * parameter;
cWA2 = cAb2 + wAb2 * parameter;
cWA2(:, :, 1) = cWA2;
cWA2(:, :, 2) = cWA2;
cWA2(:, :, 3) = cWA2;
cWA = [cWA2 cH2; cV2 cD2];

```

Kode program selengkapnya disajikan pada Lampiran C.

4.3.4 Proses Rekonstruksi Watermark

Setelah mendapatkan citra ber-*watermark*, proses selanjutnya adalah proses pengrekonstruksian *watermark*. Proses ini berfungsi untuk mendapatkan kembali citra *watermark* dari citra yang telah disisipi (citra ber-*watermark*).

```

wAr2 = (wtAr2 - cAr2) / parameter;
wAg2 = (wtAg2 - cAg2) / parameter;
wAb2 = (wtAb2 - cAb2) / parameter;
wAr = idwt2(wAr2, wHr2, wVr2, wDr2, 'haar');
wAg = idwt2(wAg2, wHg2, wVg2, wDg2, 'haar');
wAb = idwt2(wAb2, wHb2, wVb2, wDb2, 'haar');

```

Kode program selengkapnya disajikan pada Lampiran D.

4.3.5 Proses Pengujian Terhadap Noise

Setelah mendapatkan citra ber-*watermark*, dan setelah proses pengrekonstruksian *watermark*, selanjutnya adalah proses pengujian terhadap *noise*. Proses ini berfungsi untuk mengetahui tingkat kehandalan citra ber-*watermark* terhadap *noise*.

```

if(strcmp(noise, 'Gaussian noise')==1)
    noised =
imnoise(uint8(watermarked), 'gaussian', 0, parameter);
elseif(strcmp(noise, 'Poisson noise')==1)
    noised =
imnoise(uint8(watermarked), 'poisson');
elseif(strcmp(noise, 'Salt & pepper
noise')==1)
    noised = imnoise(uint8(watermarked), 'salt
& pepper', parameter);
else
    noised =
imnoise(uint8(watermarked), 'Speckle', parameter);
end

```

Kode program selengkapnya disajikan pada Lampiran E.

BAB V

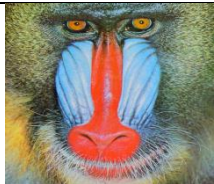
PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dilakukan pengujian serta pembahasan terhadap program yang telah dibuat. Pengujian sistem ini merupakan pengujian *invisible watermarking* pada citra berwarna RGB dalam format file BMP, PNG atau JPG. Pada bagian awal adalah hasil uji coba terhadap citra yang belum dikenai serangan, kemudian dilanjutkan terhadap citra yang dikenakan serangan. Pengujian dilakukan untuk mendapatkan nilai MSE, PSNR dan rekonstruksi *watermark*.

5.1 Data Uji Coba

Uji coba pada program dalam penelitian tugas akhir ini dilakukan pada citra *host* dan citra *watermark* dengan varian citra yaitu, citra dengan warna yang ekstrim beda, citra dengan warna yang hampir sama, dan citra dengan warna yang biasa-biasa. Hal ini didapat dengan menggunakan standar deviasi, jadi semakin besar standar deviasinya semakin berbeda tiap citranya. Jadi ada sembilan percobaan.

Tabel 5.1 Tabel Citra *Host* yang Digunakan.

No	Nama File	Ukuran citra	Varian citra	Citra
1	Baboon.png	512x512 pixel	Standar deviasi tinggi	

2	Lena. jpg	512x512 pixel	Standar deviasi rata- rata/ menengah	
3	Mawar. jpg	2560x1600 pixel	Standar deviasi rendah	

Tabel 5.2 Tabel Citra *Watermark* yang Digunakan.

No	Nama File	Ukuran citra	Varian citra	Citra
1	Orange. jpg	472x640 pixel	Standar deviasi tinggi	
2	World wide.jpg	809x360 pixel	Standar deviasi rata- rata/ menengah	
3	Ubur2. jpg	1140x550 pixel	Standar deviasi rendah	

5.2 Pengujian Hasil Watermark

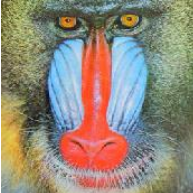
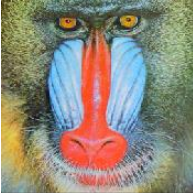
Untuk proses penyisipan *watermark* akan *resize* terlebih dahulu citra yang digunakan untuk penyisipan *watermark* (citra *host*) yang ditampilkan pada Tabel 5.1 dan citra untuk *watermark* (citra *watermark*) yang ditampilkan pada Tabel 5.2. Selanjutnya citra hasil penyisipan (citra ber-*watermark*) akan dibandingkan dengan citra *host* untuk mendapatkan nilai MSE dan PSNR. Setelah itu, akan dilakukan proses rekonstruksi untuk mendapatkan *watermark* hasil ekstraksi.

Tabel 5.3 Tabel Data untuk Pengujian

No	Citra <i>host</i>	Citra <i>watermark</i>	Nama File
1			Uji 1
2			Uji 2
3			Uji 3
4			Uji 4

5			Uji 5
6			Uji 6
7			Uji 7
8			Uji 8
9			Uji 9

Tabel 5.4 Tabel Hasil Pengujian.

No	Nama File	Hasil penyisipan	Watermark hasil ekstraksi
1	Uji 1		
2	Uji 2		
3	Uji 3		
4	Uji 4		
5	Uji 5		

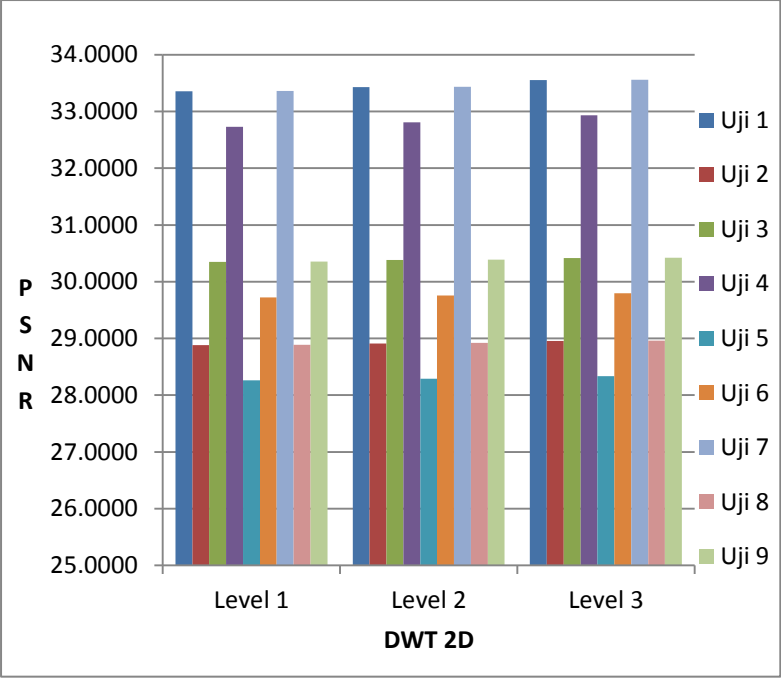
6	Uji 6		
7	Uji 7		
8	Uji 8		
9	Uji 9		

5.2.1 Pengujian PSNR (*Peak Signal to Noise Ratio*)

Pengujian PSNR ini dilakukan pada citra yang telah disisipi *watermark* (citra ber-*watermark*), yaitu dengan membandingkan citra ber-*watermark* dengan citra *host*. Untuk mendapatkan nilai PSNR, kita harus menghitung terlebih dahulu nilai *Mean Squared Error* (MSE) dari suatu citra ber-*watermark*.

Tabel 5.5 Tabel Nilai PSNR berdasarkan Levelnya.

No	Nama File	PSNR		
		Level 1	Level 2	Level 3
1	Uji 1	33.3545	33.4293	33.5541
2	Uji 2	28.8816	28.9113	28.9546
3	Uji 3	30.3474	30.381	30.4152
4	Uji 4	32.7328	32.8076	32.9324
5	Uji 5	28.2598	28.2895	28.3329
6	Uji 6	29.7257	29.7593	29.7935
7	Uji 7	33.3624	33.4372	33.562
8	Uji 8	28.8895	28.9192	28.9625
9	Uji 9	30.3553	30.3889	30.4231
	Rata-rata	29.9107	29.9464	29.9912

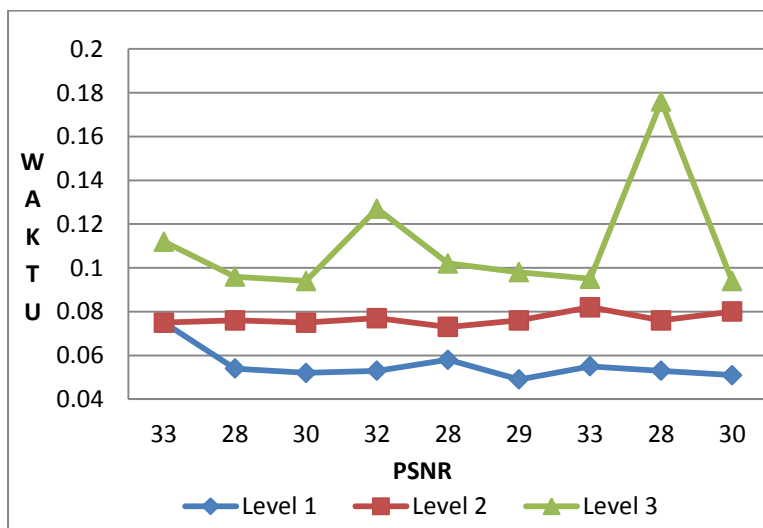


Gambar 5.1. Grafik dari nilai PSNR terhadap 3 Level.

Tabel 5.6 Tabel Waktu berdasarkan Levelnya.

No	Nama File	Waktu		
		Level 1	Level 2	Level 3
1	Uji 1	0.0753	0.0757	0.1129
2	Uji 2	0.0535	0.0761	0.0964
3	Uji 3	0.0518	0.0754	0.0938

4	Uji 4	0.0529	0.0773	0.1272
5	Uji 5	0.0578	0.0727	0.1015
6	Uji 6	0.0489	0.0759	0.0979
7	Uji 7	0.0553	0.0824	0.0954
8	Uji 8	0.053	0.0759	0.1756
9	Uji 9	0.0511	0.0798	0.0939
Rata-rata		0.0476	0.0792	0.1218

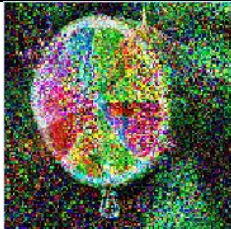
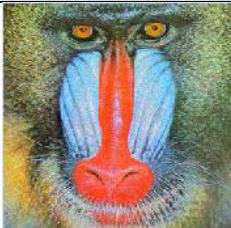


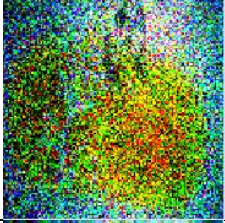
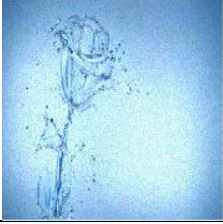
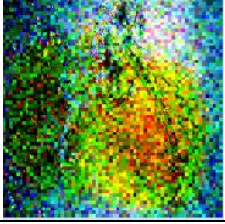
Gambar 5.2. Grafik Perbandingan Waktu dan PSNR dari 3 Level

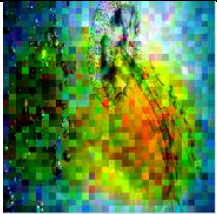
5.2.2 Pengujian Ketahanan terhadap Noise.

Pengujian terhadap kehandalan / ketahanan dari citra ber-watermark dapat dilakukan dengan memberi serangan (*noise*). Ada empat macam *noise* yang digunakan dalam penelitian ini, yaitu *noise Poisson*, *noise Gaussian*, *noise Salt & Pepper*, dan *noise Speckle*. Urutan pengujian citra sesuai dengan Tabel 5.3.

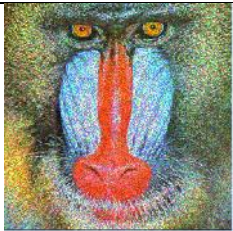
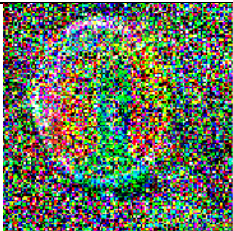
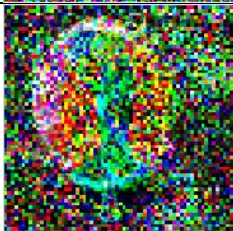
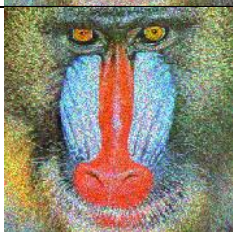
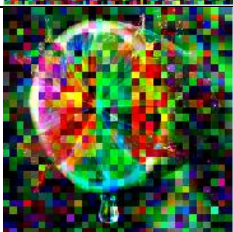
Tabel 5.7 Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap *Gaussian Noise* dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter Noise 0.005

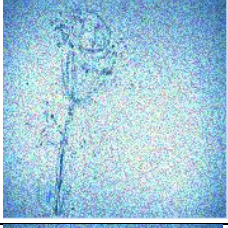
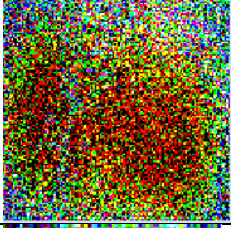
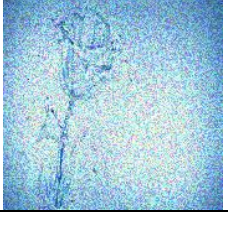
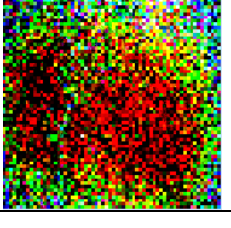
No	Uji dan Level	Citra Ber-noise	Watermark hasil ekstraksi
1	Uji 1 dan Level 1		
2	Uji 1 dan Level 2		
3	Uji 1 dan Level 3		

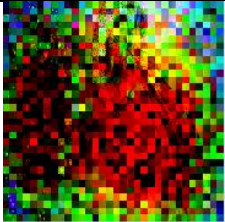
4	Uji 5 dan Level 1		
5	Uji 5 dan Level 2		
6	Uji 5 dan Level 3		
7	Uji 9 dan Level 1		
8	Uji 9 dan Level 2		

9	Uji 9 dan Level 3		
---	-------------------	---	---

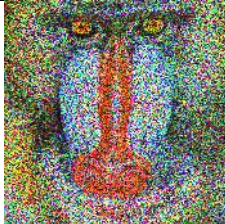
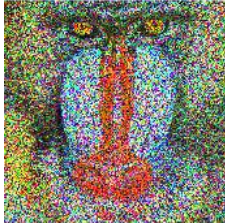
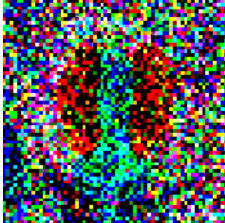
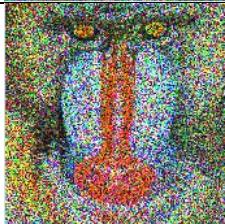
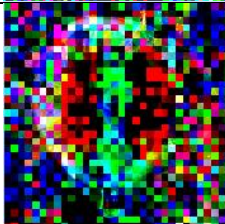
Tabel 5.8 Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap *Gaussian Noise* dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter *Noise* 0.05

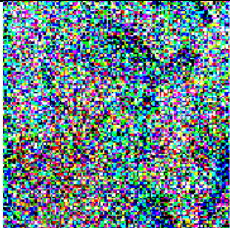
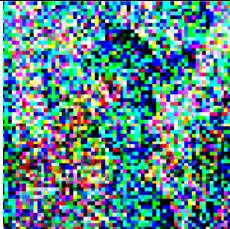
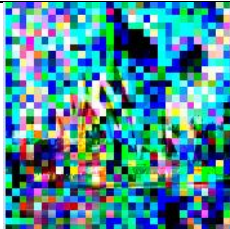
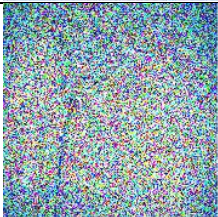
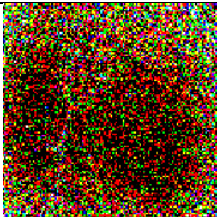
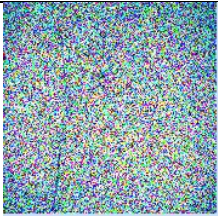
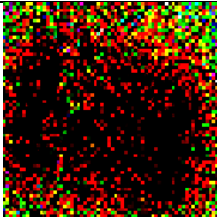
No	Uji dan Level	Citra Ber-noise	Watermark hasil ekstraksi
1	Uji 1 dan Level 1		
2	Uji 1 dan Level 2		
3	Uji 1 dan Level 3		

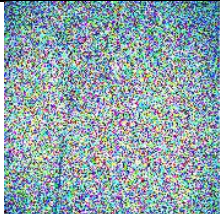
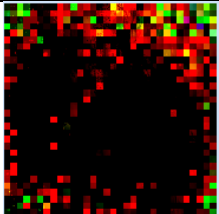
4	Uji 5 dan Level 1		
5	Uji 5 dan Level 2		
6	Uji 5 dan Level 3		
7	Uji 9 dan Level 1		
8	Uji 9 dan Level 2		

9	Uji 9 dan Level 3		
---	-------------------	---	---

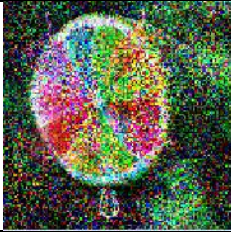
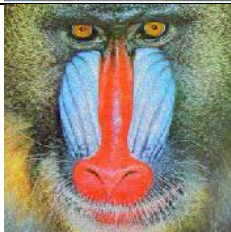
Tabel 5.9 Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap *Gaussian Noise* dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter *Noise* 0.5

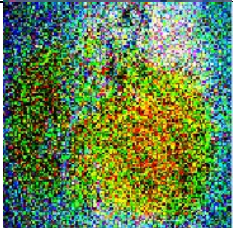
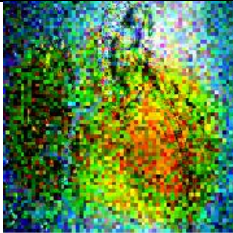
No	Uji dan Level	Citra Ber-noise	Watermark hasil ekstraksi
1	Uji 1 dan Level 1		
2	Uji 1 dan Level 2		
3	Uji 1 dan Level 3		

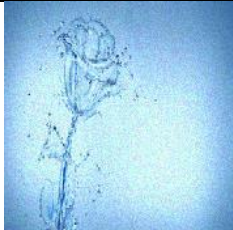
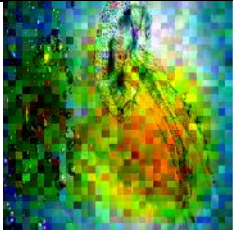
4	Uji 5 dan Level 1		
5	Uji 5 dan Level 2		
6	Uji 5 dan Level 3		
7	Uji 9 dan Level 1		
8	Uji 9 dan Level 2		

9	Uji 9 dan Level 3		
---	-------------------	---	---

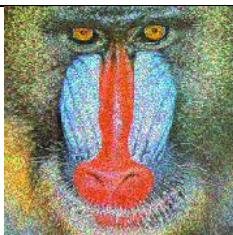
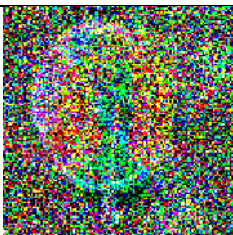
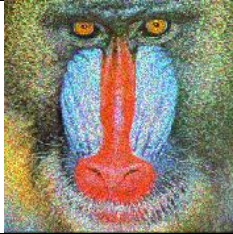
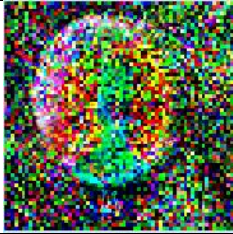
Tabel 5.10 Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap *Poisson Noise* dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter *Noise* 0.005

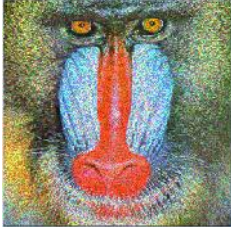
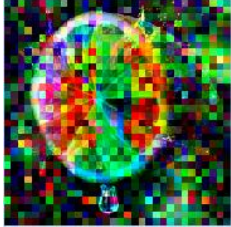
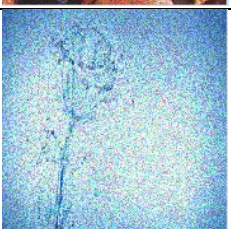
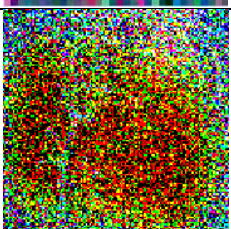
No	Uji dan Level	Citra Ber-noise	Watermark hasil ekstraksi
1	Uji 1 dan Level 1		
2	Uji 1 dan Level 2		
3	Uji 1 dan Level 3		

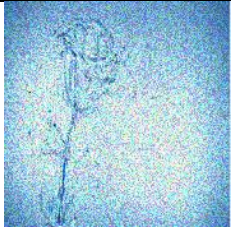
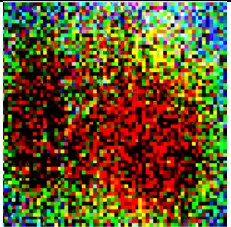
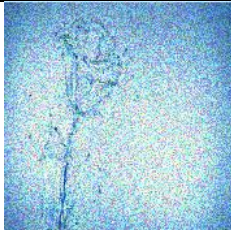
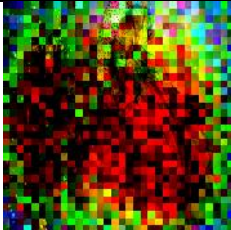
4	Uji 5 dan Level 1		
5	Uji 5 dan Level 2		
6	Uji 5 dan Level 3		
7	Uji 9 dan Level 1		
8	Uji 9 dan Level 2		

9	Uji 9 dan Level 3		
---	-------------------	---	---

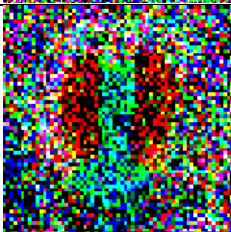
Tabel 5.11 Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap *Poisson Noise* dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter *Noise* 0.05

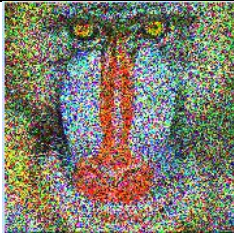
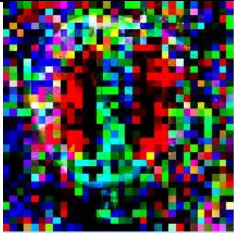
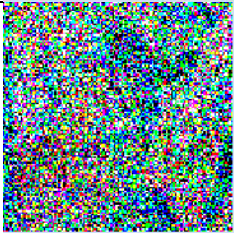
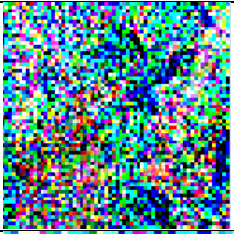
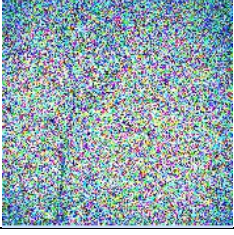
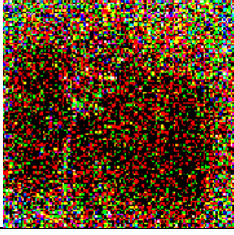
No	Uji dan Level	Citra Ber-noise	Watermark hasil ekstraksi
1	Uji 1 dan Level 1		
2	Uji 1 dan Level 2		

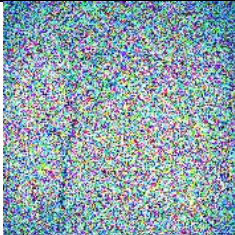
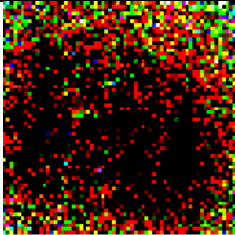
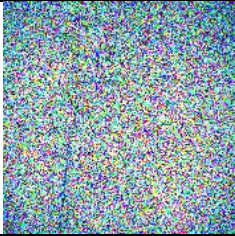
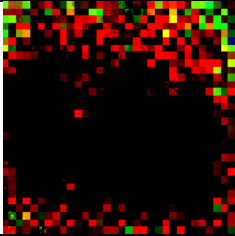
3	Uji 1 dan Level 3		
4	Uji 5 dan Level 1		
5	Uji 5 dan Level 2		
6	Uji 5 dan Level 3		
7	Uji 9 dan Level 1		

8	Uji 9 dan Level 2		
9	Uji 9 dan Level 3		

Tabel 5.12 Tabel Hasil Pengujian Ketahanan Uji 1, Uji 5, dan Uji 9 terhadap *Poisson Noise* dengan DWT2 Level 1, Level 2, dan Level 3 dengan Parameter *Noise* 0.5

No	Uji dan Level	Citra Ber-noise	Watermark hasil ekstraksi
1	Uji 1 dan Level 1		
2	Uji 1 dan Level 2		

3	Uji 1 dan Level 3		
4	Uji 5 dan Level 1		
5	Uji 5 dan Level 2		
6	Uji 5 dan Level 3		
7	Uji 9 dan Level 1		

8	Uji 9 dan Level 2		
9	Uji 9 dan Level 3		

Dari percobaan yang telah dilakukan yaitu dari tabel 5.7 sampai 5.12 dapat ditarik beberapa kesimpulan. Kesimpulan yang pertama bahwa semakin kecil nilai parameter *noise*, maka semakin bagus kualitas citra atau semakin tinggi ketahanan citra terhadap *noise*. Kedua, semakin tingginya standar deviasi citra *host* dan citra *watermark* maka semakin tahan citra terhadap *noise*. Ketiga, semakin tingginya level dekomposisi, maka semakin tinggi juga ketahanan citra terhadap *noise*.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan pada teknik *invisible watermarking* digital menggunakan metode *descrete wavelet transform*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Dekomposisi citra *digital* menggunakan *Discrete Wavelet Transform (DWT)* dalam tugas akhir ini telah berhasil dilakukan dengan cara mengambil koefisien *wavelet* dari citra tersebut, koefisien *wavelet* juga yang digunakan untuk dapat merekonstruksi citra kembali menggunakan *Inverse Discrete Wavelet Transform (IDWT)*.
2. Aplikasi *watermarking* yang telah dibuat memiliki tingkat *invisibility* yang bagus, artinya *watermark* yang disisipkan tidak terlihat, hal ini dapat dibuktikan dengan nilai PSNR yang cukup tinggi.
3. Aplikasi *watermarking* yang telah dibuat juga memiliki tingkat *readability*, artinya hal ini dapat dilihat dari perbandingan citra *watermark* dengan citra *watermark* hasil rekonstruksi, dan citra ber-*watermark* dengan citra *host*.
4. Karena pada proses penyisipan citra *host* dan citra *watermark* serta proses uji *noise* yang menggunakan parameter, maka dapat ditarik kesimpulan terhadap nilai parameter. Semakin kecil nilai parameter, maka semakin bagus kualitas citra *digital* pada proses penyisipan atau semakin tinggi ketahanan citra *digital* terhadap *noise*.

5. Dekomposisi citra *digital* yang menggunakan DWT2 dalam tugas akhir ini adalah pada level 1, level 2, dan level 3. Semakin tingginya level dekomposisi, maka semakin tinggi juga kualitas citra *digital* dan ketahanan citra *digital* terhadap *noise*.

6.2 Saran

Berdasarkan hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. *Discrete Wavelet Transform (DWT)* yang digunakan adalah Haar *Wavelet*, untuk pengembangan selanjutnya dapat digunakan jenis *DWT* lain seperti Daubechies, Coiflet, Symlet, Meyer, Morlet, dan Mexican Hat. Teknik-teknik ini merupakan teknik yang lebih tahan terhadap serangan dibandingkan Haar *wavelet*.
2. Teknik *watermarking* dapat dilakukan pada media *digital* lainnya seperti audio maupun video, tetapi kedua media tersebut sangat sukar karena *sensitive* terhadap perubahan data yang kecil sekalipun, sehingga diperlukan pengembangan perangkat lunak lebih lanjut sehingga memungkinkan untuk menyimpan pesan rahasia ke dalam dua media digital tersebut.
3. Dalam penelitian ini menggunakan teknik *non blind watermarking*, artinya teknik *watermarking* yang memerlukan citra *host* dan citra ber-*watermark* untuk merekonstruksi *watermark*. Sedangkan untuk pengembangan selanjutnya dapat digunakan teknik *blind watermarking* merupakan teknik *watermarking* yang tidak memerlukan citra *host* untuk melakukan rekonstruksi.

DAFTAR PUSTAKA

- [1] Alfatwa, Dean Fathony. (2009). "Watermarking Pada Citra Digital Menggunakan Discrete Wavelet Transform". Bandung: Institute Teknologi Bandung.
- [2] Asiyah, Nur. (2014). "Contoh Pelanggaran Hak Cipta Dalam Dunia Internet". Update: 25 februari 2014 .
- [3] Dwiandiyanta, B. Yudi. "Perbandingan Watermarking Citra dengan Alihagam Wavelet dan Discrete Cosine Transform". Yogyakarta : Universitas Atma Jaya Yogyakarta.
- [4] Gonzalez, R.C., E.W., Rafael. (2008). "Digital Image Processing". Prentice-Hall. Inc.. United State, America.
- [5] Hakim, Arif Rahman. (2012). "Analisa Perbandingan Watermarking Image Menggunakan Discrete Wavelet Transform". Jakarta: Universitas Indonesia.
- [6] Hidayat, Erwin Yudi., & Udayanti, Erika Devi. (2011). "Hybrid Watermarking Citra Digital Menggunakan Teknik DWT-DCT SVD," in Seminar Nasional Teknologi Informasi & Komunikasi Terapan, Semarang.
- [7] Ischam, Ali. (2010). "Watermarking Citra Digital Menggunakan Transformasi Hybrid DWT dan DCT". Universitas Diponegoro.
- [8] Kakade, Kiran, dkk. (2015). "Improving Image Steganography Using Invisible Watermarking". International Journal of Computer Science and Information Technology Research, 3(2):190-194.
- [9] Kutter, Martin; Fabien A. P. Petitcolas. 1999. A Fair Benchmark for Image Watermarking Systems. The International Society for Optical Engineering.
- [10] Munir, Rinaldi. (2006). "Sekilas Image Watermarking untuk Memproteksi Citra Digital dan Aplikasinya pada Citra Medis". Bandung : Institut Teknologi Bandung.

- [11] Putra, Darma. 2010. Pengolahan Citra Digital. Andi : Yogyakarta. Hal: 39-44, 63, 95-116.
- [12] Reza, Moh Khavid. (2014). “Perancangan Sistem Watermarking Pada Citra Digital Menggunakan Metode DCT dan LSB”. Surakarta : Universitas Muhammadiyah.
- [13] Sundararajan, Duraisamy (2011). “Fundamentals of the Discrete Haar Wavelet Transform”.
- [14] Sutoyo T, Mulyanto E, Suhartono V, Nurhayati O. D, Wijanarko. 2009. Teori Pengolahan Citra Digital. Andi: Yogyakarta. Hal: 20-23, 249-252.
- [15] Syahbana, Yoanda Alim, dkk. (2015). “Algoritma Penyisipan Frame untuk Peningkatan Akurasi Metode Aligned Peak Signal-to-Noise Ratio dalam Pengukuran Kualitas Video”. Jurnal Komputer Terapan. Vol. 01, no 02.
- [16] Tuakia, N.M., Suprpto, & Yulistira, N. “Implementasi Watermarking Pada Citra Medis Menggunakan Metode Discrete Wavelet Transform (DWT)”. Malang : Universitas Brawijaya.
- [17] Wong, Ny poi., & Cahyadi, Willy. (2012). “Rancang Bangun Aplikasi Watermarking Pada Gambar Dengan Algoritma Digital Semipublic”. ISSN. 1412-0100. Vol. 13, no 02.

LAMPIRAN A

Kode Program Proses Antarmuka Awal

```

function varargout = fanyGUIedit(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',
gui_Singleton, ...
                  'gui_OpeningFcn',
@fanyGUIedit_OpeningFcn, ...
                  'gui_OutputFcn',
@fanyGUIedit_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function fanyGUIedit_OpeningFcn(hObject,
eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = fanyGUIedit_OutputFcn
(hObject, eventdata, handles)
varargout{1} = handles.output;

```

LAMPIRAN A (LANJUTAN)

```
function pushbutton1_Callback(hObject,
eventdata, handles)
global asli;
[filename pathname] =
uigetfile({'*.jpg'; '*.bpm'; '*.png';}, 'File
Selector');
alamat = strcat(pathname, filename);
asli = imread(alamat);
asli = imresize(asli, [256 256])
axes(handles.axes1)
imshow(asli)
```

```
function pushbutton2_Callback(hObject,
eventdata, handles)
global watermark;
[filename pathname] =
uigetfile({'*.jpg'; '*.bpm'; '*.png';}, 'File
Selector');
alamat = strcat(pathname, filename);
watermark = imread(alamat);
watermark = imresize(watermark, [256 256])
axes(handles.axes2)
imshow(watermark)
```

LAMPIRAN B

Kode Program Proses Dekomposisi DWT2

```
function pushbutton3_Callback(hObject,
eventdata, handles)
global asli; global cA; global cH; global cV;
global cD; global cA2; global cH2; global cV2;
global cD2; global cA3; global cH3; global cV3;
global cD3; global watermark; global wA; global
wH; global wV; global wD; global wA2; global
wH2; global wV2; global wD2; global wA3; global
wH3; global wV3; global wD3;
global level;

contents = get(handles.popupmenu3, 'String');
level =
contents{get(handles.popupmenu3, 'Value')};

if(strcmp(level, 'DWT2 Level 1')==1)
    tic
    cA(:, :, 1) = (cAr);
    cA(:, :, 2) = (cAg);
    cA(:, :, 3) = (cAb);
    cH(:, :, 1) = (cHr);
    cH(:, :, 2) = (cHg);
    cH(:, :, 3) = (cHb);
    cV(:, :, 1) = (cVr);
    cV(:, :, 2) = (cVg);
    cV(:, :, 3) = (cVb);
    cD(:, :, 1) = (cDr);
    cD(:, :, 2) = (cDg);
    cD(:, :, 3) = (cDb);
    wA(:, :, 1) = (wAr);
    wA(:, :, 2) = (wAg);
    wA(:, :, 3) = (wAb);
    wH(:, :, 1) = (wHr);
    wH(:, :, 2) = (wHg);
    wH(:, :, 3) = (wHb);
```

LAMPIRAN B (LANJUTAN)

```

wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wV(:, :, 3) = (wVb);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);
wD(:, :, 3) = (wDb);
disp('Waktu untuk DWT2 Level 1 adalah');
toc
elseif(strcmp(level,'DWT2 Level 2')==1)

tic

cA2(:, :, 1) = (cAr2);
cA2(:, :, 2) = (cAg2);
cA2(:, :, 3) = (cAb2);
cH2(:, :, 1) = (cHr2);
cH2(:, :, 2) = (cHg2);
cH2(:, :, 3) = (cHb2);
cV2(:, :, 1) = (cVr2);
cV2(:, :, 2) = (cVg2);
cV2(:, :, 3) = (cVb2);
cD2(:, :, 1) = (cDr2);
cD2(:, :, 2) = (cDg2);
cD2(:, :, 3) = (cDb2);
cH(:, :, 1) = (cHr);
cH(:, :, 2) = (cHg);
cH(:, :, 3) = (cHb);
cV(:, :, 1) = (cVr);
cV(:, :, 2) = (cVg);
cV(:, :, 3) = (cVb);
cD(:, :, 1) = (cDr);
cD(:, :, 2) = (cDg);
cD(:, :, 3) = (cDb);

```

LAMPIRAN B (LANJUTAN)

```

wA2(:, :, 1) = (wAr2);
wA2(:, :, 2) = (wAg2);
wA2(:, :, 3) = (wAb2);
wH2(:, :, 1) = (wHr2);
wH2(:, :, 2) = (wHg2);
wH2(:, :, 3) = (wHb2);
wV2(:, :, 1) = (wVr2);
wV2(:, :, 2) = (wVg2);
wV2(:, :, 3) = (wVb2);
wD2(:, :, 1) = (wDr2);
wD2(:, :, 2) = (wDg2);
wD2(:, :, 3) = (wDb2);
wH(:, :, 1) = (wHr);
wH(:, :, 2) = (wHg);
wH(:, :, 3) = (wHb);
wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wV(:, :, 3) = (wVb);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);
wD(:, :, 3) = (wDb);
disp('Waktu untuk DWT2 Level 2 adalah');
toc
else(strcmp(level, 'DWT2 Level 3')==1);

tic
cA3(:, :, 1) = (cAr3);
cA3(:, :, 2) = (cAg3);
cA3(:, :, 3) = (cAb3);
cH3(:, :, 1) = (cHr3);
cH3(:, :, 2) = (cHg3);
cH3(:, :, 3) = (cHb3);
cV3(:, :, 1) = (cVr3);
cV3(:, :, 2) = (cVg3);
cV3(:, :, 3) = (cVb3);
cD3(:, :, 1) = (cDr3);
cD3(:, :, 2) = (cDg3);
cD3(:, :, 3) = (cDb3);
cH2(:, :, 1) = (cHr2);

```

LAMPIRAN B (LANJUTAN)

```

cH2 (:, :, 2) = (cHg2);
cH2 (:, :, 3) = (cHb2);
cV2 (:, :, 1) = (cVr2);
cV2 (:, :, 2) = (cVg2);
cV2 (:, :, 3) = (cVb2);
cD2 (:, :, 1) = (cDr2);
cD2 (:, :, 2) = (cDg2);
cD2 (:, :, 3) = (cDb2);
cH (:, :, 1) = (cHr);
cH (:, :, 2) = (cHg);
cH (:, :, 3) = (cHb);
cV (:, :, 1) = (cVr);
cV (:, :, 2) = (cVg);
cV (:, :, 3) = (cVb);
cD (:, :, 1) = (cDr);
cD (:, :, 2) = (cDg);
cD (:, :, 3) = (cDb);

```

```

wA3 (:, :, 1) = (wAr3);
wA3 (:, :, 2) = (wAg3);
wA3 (:, :, 3) = (wAb3);
wH3 (:, :, 1) = (wHr3);
wH3 (:, :, 2) = (wHg3);
wH3 (:, :, 3) = (wHb3);
wV3 (:, :, 1) = (wVr3);
wV3 (:, :, 2) = (wVg3);
wV3 (:, :, 3) = (wVb3);
wD3 (:, :, 1) = (wDr3);
wD3 (:, :, 2) = (wDg3);
wD3 (:, :, 3) = (wDb3);
wH2 (:, :, 1) = (wHr2);
wH2 (:, :, 2) = (wHg2);
wH2 (:, :, 3) = (wHb2);
wV2 (:, :, 1) = (wVr2);
wV2 (:, :, 2) = (wVg2);
wV2 (:, :, 3) = (wVb2);
wD2 (:, :, 1) = (wDr2);

```

LAMPIRAN B (LANJUTAN)

```

wD2(:, :, 2) = (wDg2);
wD2(:, :, 3) = (wDb2);
wH(:, :, 1) = (wHr);
wH(:, :, 2) = (wHg);
wH(:, :, 3) = (wHb);
wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wV(:, :, 3) = (wVb);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);
wD(:, :, 3) = (wDb);
disp('Waktu untuk DWT2 Level 3 adalah');
toc;
end

axes(handles.axes3)
imshow(uint8([cA,cH;cV,cD]))

axes(handles.axes4)
imshow(uint8([wA,wH;wV,wD]))

```

LAMPIRAN C

Kode Program Proses Penyisipan Citra

```
function edit2_Callback(hObject, eventdata,
handles)

global parameter;
parameter = str2double(get(hObject, 'String'));

function edit2_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function pushbutton5_Callback(hObject,
eventdata, handles)

global parameter; global cA; global cH; global
cV; global cD; global cA2; global cH2; global
cV2; global cD2; global cA3; global cH3; global
cV3; global cD3; global wA; global wA2; global
wA3; global cWA; global cWA2; global cWA3;
global level;

parameter =
str2double(get(handles.edit2, 'String'));
if(strcmp(level, 'DWT2 Level 1')==1)
    cWA = cA + wA * parameter;
elseif(strcmp(level, 'DWT2 Level 2')==1)
    cWA2 = cA2 + wA2 * parameter;
else
    cWA3 = cA3 + wA3 * parameter;
```

LAMPIRAN C (LANJUTAN)

```

end
axes(handles.axes5)
imshow(uint8([cWA,cH;cV,cD]))

function pushbutton6_Callback(hObject,
 eventdata, handles)

global asli; global cH; global cV; global cD;
global cH2; global cV2; global cD2; global cH3;
global cV3; global cD3; global cWA; global cWA2;
global cWA3; global level; global watermarked;

if(strcmp(level,'DWT2 Level 1')==1)
    watermarked(:,:,1) = idwt2(cWA(:,:,1),
cH(:,:,1), cV(:,:,1), cD(:,:,1), 'haar');
else
if(strcmp(level,'DWT2 Level 2')==1)
cWA(:,:,1) = idwt2(cWA2(:,:,1), cH2(:,:,1),
cV2(:,:,1), cD2(:,:,1), 'haar');
    watermarked(:,:,3) = idwt2(cWA(:,:,3),
cH(:,:,3), cV(:,:,3), cD(:,:,3), 'haar');
else

    cWA2(:,:,1) = idwt2(cWA3(:,:,1), cH3(:,:,1),
cV3(:,:,1), cD3(:,:,1), 'haar');
    cWA2(:,:,2) = idwt2(cWA3(:,:,2), cH3(:,:,2),
cV3(:,:,2), cD3(:,:,2), 'haar');
    cWA2(:,:,3) = idwt2(cWA3(:,:,3), cH3(:,:,3),
cV3(:,:,3), cD3(:,:,3), 'haar');
    cWA(:,:,1) = idwt2(cWA2(:,:,1), cH2(:,:,1),

```

LAMPIRAN C (LANJUTAN)

```

        watermarked(:,:,1) = idwt2(cwA(:,:,1),
cH(:,:,1), cV(:,:,1), cD(:,:,1), 'haar');
        watermarked(:,:,2) = idwt2(cwA(:,:,2),
cH(:,:,2), cV(:,:,2), cD(:,:,2), 'haar');
        watermarked(:,:,3) = idwt2(cwA(:,:,3),
cH(:,:,3), cV(:,:,3), cD(:,:,3), 'haar');
    end

    axes(handles.axes6)
    imshow(uint8(watermarked))

```

LAMPIRAN D

Kode Program Rekonstruksi *Watermark*

```
function pushbutton7_Callback(hObject,
eventdata, handles)

global parameter; global wtA; global wtA2;
global wtA3; global wH; global wH2;
global wH3; global wV; global wV2; global wV3;
global wD; global wD2; global wD3;
global level; global cA; global cA2; global cA3;

if(strcmp(level,'DWT2 Level 1')==1)
    waA = (wtA - cA) / parameter;
    restorasi(:,:,1),
wH(:,:,1), wV(:,:,1), wD(:,:,1), 'haar');
    restorasi(:,:,2),
wH(:,:,2), wV(:,:,2), wD(:,:,2), 'haar');
    restorasi(:,:,3),
wH(:,:,3), wV(:,:,3), wD(:,:,3), 'haar');
    restorasi = uint8(restorasi);

elseif(strcmp(level,'DWT2 Level 2')==1)
    waA2 = (wtA2 - cA2) / parameter;
    waA(:,:,1),
wH2(:,:,1), wV2(:,:,1), wD2(:,:,1), 'haar');
    waA(:,:,2),
wH2(:,:,2), wV2(:,:,2), wD2(:,:,2), 'haar');
    waA(:,:,3),
wH2(:,:,3), wV2(:,:,3), wD2(:,:,3), 'haar');
    restorasi(:,:,1),
wH(:,:,1), wV(:,:,1), wD(:,:,1), 'haar');
    restorasi(:,:,2),
wH(:,:,2), wV(:,:,2), wD(:,:,2), 'haar');
    restorasi(:,:,3),
wH(:,:,3), wV(:,:,3), wD(:,:,3), 'haar');
else
```

LAMPIRAN D (LANJUTAN)

```

        waA3 = (wtA3 - cA3) / parameter;
        waA2(:, :, 1) ,
    wH3(:, :, 1), wV3(:, :, 1), wD3(:, :, 1), 'haar');
        waA2(:, :, 2) ,
    wH3(:, :, 2), wV3(:, :, 2), wD3(:, :, 2), 'haar');
        waA2(:, :, 3) ,
    wH3(:, :, 3), wV3(:, :, 3), wD3(:, :, 3), 'haar');
        waA(:, :, 1) ,
    wH2(:, :, 1), wV2(:, :, 1), wD2(:, :, 1), 'haar');
        waA(:, :, 2) ,
    wH2(:, :, 2), wV2(:, :, 2), wD2(:, :, 2), 'haar');
        waA(:, :, 3) ,
    wH2(:, :, 3), wV2(:, :, 3), wD2(:, :, 3), 'haar');
        restorasi(:, :, 1) ,
    wH(:, :, 1), wV(:, :, 1), wD(:, :, 1), 'haar');
        restorasi(:, :, 2) ,
    wH(:, :, 2), wV(:, :, 2), wD(:, :, 2), 'haar');
        restorasi(:, :, 3) ,
    wH(:, :, 3), wV(:, :, 3), wD(:, :, 3), 'haar');
end

axes(handles.axes7)
imshow(uint8(restorasi))

```

LAMPIRAN D (LANJUTAN)

```

function pushbutton8_Callback(hObject,
eventdata, handles)
global watermarked; global wtA; global wtA2;
global wtA3; global level;

if(strcmp(level, 'DWT2 Level 1')==1)
[wAr, wHr, wVr, wDr] = (watermarked(:, :, 1), 'haar');
[wAg, wHg, wVg, wDg] = (watermarked(:, :, 2), 'haar');
[wAb, wHb, wVb, wDb] = (watermarked(:, :, 3), 'haar');

    wtA(:, :, 1) = (wAr);
    wtA(:, :, 2) = (wAg);
    wtA(:, :, 3) = (wAb);
    wH(:, :, 1) = (wHr);
    wH(:, :, 2) = (wHg);
    wH(:, :, 3) = (wHb);
    wV(:, :, 1) = (wVr);
    wV(:, :, 2) = (wVg);
    wV(:, :, 3) = (wVb);
    wD(:, :, 1) = (wDr);
    wD(:, :, 2) = (wDg);
    wD(:, :, 3) = (wDb);

elseif(strcmp(level, 'DWT2 Level 2')==1)
[wAr, wHr, wVr, wDr] = (watermarked(:, :, 1), 'haar');
[wAg, wHg, wVg, wDg] = (watermarked(:, :, 2), 'haar');
[wAb, wHb, wVb, wDb] = (watermarked(:, :, 3), 'haar');

    wtA2(:, :, 1) = (wAr2);
    wtA2(:, :, 2) = (wAg2);
    wtA2(:, :, 3) = (wAb2);
    wH2(:, :, 1) = (wHr2);
    wH2(:, :, 2) = (wHg2);
    wH2(:, :, 3) = (wHb2);
    wV2(:, :, 1) = (wVr2);
    wV2(:, :, 2) = (wVg2);
    wV2(:, :, 3) = (wVb2);

```

LAMPIRAN D (LANJUTAN)

```

wD2 (:, :, 1) = (wDr2);
wD2 (:, :, 2) = (wDg2);
wD2 (:, :, 3) = (wDb2);
wH (:, :, 1) = (wHr);
wH (:, :, 2) = (wHg);
wH (:, :, 3) = (wHb);
wV (:, :, 1) = (wVr);
wV (:, :, 2) = (wVg);
wV (:, :, 3) = (wVb);
wD (:, :, 1) = (wDr);
wD (:, :, 2) = (wDg);
wD (:, :, 3) = (wDb);

else (strcmp (level, 'DWT2 Level 3')==1);
[wAr, wHr, wVr, wDr] = (watermarked (:, :, 1), 'haar');
[wAg, wHg, wVg, wDg] = (watermarked (:, :, 2), 'haar');
[wAb, wHb, wVb, wDb] = (watermarked (:, :, 3), 'haar');

wtA3 (:, :, 1) = (wAr3);
wtA3 (:, :, 2) = (wAg3);
wtA3 (:, :, 3) = (wAb3);
wH3 (:, :, 1) = (wHr3);
wH3 (:, :, 2) = (wHg3);
wH3 (:, :, 3) = (wHb3);
wV3 (:, :, 1) = (wVr3);
wV3 (:, :, 2) = (wVg3);
wV3 (:, :, 3) = (wVb3);
wD3 (:, :, 1) = (wDr3);
wD3 (:, :, 2) = (wDg3);
wD3 (:, :, 3) = (wDb3);
wH2 (:, :, 1) = (wHr2);
wH2 (:, :, 2) = (wHg2);
wH2 (:, :, 3) = (wHb2);
wV2 (:, :, 1) = (wVr2);
wV2 (:, :, 2) = (wVg2);
wV2 (:, :, 3) = (wVb2);
wD2 (:, :, 1) = (wDr2);

```

LAMPIRAN D (LANJUTAN)

```

wD2(:, :, 2) = (wDg2);
wD2(:, :, 3) = (wDb2);
wH(:, :, 1) = (wHr);
wH(:, :, 2) = (wHg);
wH(:, :, 3) = (wHb);
wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wV(:, :, 3) = (wVb);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);
wD(:, :, 3) = (wDb);
end

axes(handles.axes8)
imshow(uint8([wtA, wH; wV, wD]))

```

LAMPIRAN E

Kode Program Pengujian terhadap *Noise*

```

function varargout = fanyNoise(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',
gui_Singleton, ...
                  'gui_OpeningFcn',
@fanyNoise_OpeningFcn, ...
                  'gui_OutputFcn',
@fanyNoise_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function fanyNoise_OpeningFcn(hObject,
eventdata, handles, varargin)
handles.output = hObject;

global parameter; global cA; global cA2; global
cA3; global wH; global wH2; global wH3; global
wV; global wV2; global wV3; global wD; global
wD2; global wD3;
global watermarked; global watermark; global
level;

```

LAMPIRAN E (LANJUTAN)

```

level = getappdata(0, 'level');
cA3 = getappdata(0, 'cA3');
cA2 = getappdata(0, 'cA2');
cA = getappdata(0, 'cA');
wH3 = getappdata(0, 'wH3');
wH2 = getappdata(0, 'wH2');
wH = getappdata(0, 'wHb');
wV3 = getappdata(0, 'wV3');
wV2 = getappdata(0, 'wV2');
wV = getappdata(0, 'wV');
wD3 = getappdata(0, 'wD3');
wD2 = getappdata(0, 'wD2');
wD = getappdata(0, 'wD');

axes(handles.axes1)
imshow(uint8(watermarked))

guidata(hObject, handles);

function varargout =
fanyNoise_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject,
eventdata, handles)

global watermarked; global noised;
contents = get(handles.popupmenu1, 'String');
noise = contents{get(handles.popupmenu1, 'Value')};

```

LAMPIRAN E (LANJUTAN)

```

contents = get(handles.popupmenu2, 'String');
parameter =
str2double(contents{get(handles.popupmenu2,
'Value')});

if(strcmp(noise, 'Gaussian noise')==1)
    noised =
imnoise(uint8(watermarked), 'gaussian', 0,
parameter);

elseif(strcmp(noise, 'Poisson noise')==1)
    noised =
imnoise(uint8(watermarked), 'poisson');

elseif(strcmp(noise, 'Salt & pepper noise')==1)
    noised = imnoise(uint8(watermarked), 'salt &
pepper', parameter);

else
    noised =
imnoise(uint8(watermarked), 'Speckle', parameter);
end

axes(handles.axes2)
imshow(noised)

function pushbutton3_Callback(hObject,
eventdata, handles)

global noised; global level; global wtA; global
wtA2; global wtA3;

if(strcmp(level, 'DWT2 Level 1')==1)
    [wAr, wHr, wVr, wDr] = (noised(:, :, 1), 'haar');
    [wAg, wHg, wVg, wDg] = (noised(:, :, 2), 'haar');

```

LAMPIRAN E (LANJUTAN)

```

wtA(:, :, 1) = (wAr);
wtA(:, :, 2) = (wAg);
wH(:, :, 1) = (wHr);
wH(:, :, 2) = (wHg);
wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);

```

```
elseif(strcmp(level, 'DWT2 Level 2')==1)
```

```

[wAr, wHr, wVr, wDr] = (noised(:, :, 1), 'haar');
[wAg, wHg, wVg, wDg] = (noised(:, :, 2), 'haar');
[wAb, wHb, wVb, wDb] = (noised(:, :, 3), 'haar');

```

```

wtA2(:, :, 1) = (wAr2);
wtA2(:, :, 2) = (wAg2);
wH2(:, :, 1) = (wHr2);
wH2(:, :, 2) = (wHg2);
wV2(:, :, 1) = (wVr2);
wV2(:, :, 2) = (wVg2);
wD2(:, :, 1) = (wDr2);
wD2(:, :, 2) = (wDg2);
wH(:, :, 1) = (wHr);
wH(:, :, 2) = (wHg);
wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);

```

```
else(strcmp(level, 'DWT2 Level 3')==1);
```

```

[wAr, wHr, wVr, wDr] = (noised(:, :, 1), 'haar');
[wAg, wHg, wVg, wDg] = (noised(:, :, 2), 'haar');
[wAb, wHb, wVb, wDb] = (noised(:, :, 3), 'haar');

```

LAMPIRAN E (LANJUTAN)

```

wtA3(:, :, 1) = (wAr3);
wtA3(:, :, 2) = (wAg3);
wtA3(:, :, 3) = (wAb3);
wH3(:, :, 1) = (wHr3);
wH3(:, :, 2) = (wHg3);
wH3(:, :, 3) = (wHb3);
wV3(:, :, 1) = (wVr3);
wV3(:, :, 2) = (wVg3);
wV3(:, :, 3) = (wVb3);
wD3(:, :, 1) = (wDr3);
wD3(:, :, 2) = (wDg3);
wD3(:, :, 3) = (wDb3);
wH2(:, :, 1) = (wHr2);
wH2(:, :, 2) = (wHg2);
wH2(:, :, 3) = (wHb2);
wV2(:, :, 1) = (wVr2);
wV2(:, :, 2) = (wVg2);
wV2(:, :, 3) = (wVb2);
wD2(:, :, 1) = (wDr2);
wD2(:, :, 2) = (wDg2);
wD2(:, :, 3) = (wDb2);
wH(:, :, 1) = (wHr);
wH(:, :, 2) = (wHg);
wH(:, :, 3) = (wHb);
wV(:, :, 1) = (wVr);
wV(:, :, 2) = (wVg);
wV(:, :, 3) = (wVb);
wD(:, :, 1) = (wDr);
wD(:, :, 2) = (wDg);
wD(:, :, 3) = (wDb);
end

axes(handles.axes4)
imshow(uint8([wtA,wH,wV,wD]))

```

LAMPIRAN E (LANJUTAN)

```
function pushbutton2_Callback(hObject,
 eventdata, handles)

global parameter; global wtA; global wtA2;
global wtA3; global wH; global wH2;
global wH3; global wV; global wV2; global wV3;
global wD; global wD2; global wD3;
global level; global cA; global cA2; global cA3;

if(strcmp(level,'DWT2 Level 1')==1)
    waA = (wtA - cA) / parameter;
    restorasi(:,:,1) ,
wH(:,:,1), wV(:,:,1), wD(:,:,1), 'haar');
    restorasi(:,:,2) ,
wH(:,:,2), wV(:,:,2), wD(:,:,2), 'haar');
    restorasi(:,:,3) ,
wH(:,:,3), wV(:,:,3), wD(:,:,3), 'haar');

    restorasi = uint8(restorasi);

elseif(strcmp(level,'DWT2 Level 2')==1)
    waA2 = (wtA2 - cA2) / parameter;
    waA(:,:,1),
wH2(:,:,1), wV2(:,:,1), wD2(:,:,1), 'haar');
    waA(:,:,2),
wH2(:,:,2), wV2(:,:,2), wD2(:,:,2), 'haar');
    waA(:,:,3),
wH2(:,:,3), wV2(:,:,3), wD2(:,:,3), 'haar');

    restorasi(:,:,1) ,
wH(:,:,1), wV(:,:,1), wD(:,:,1), 'haar');
    restorasi(:,:,2) ,
wH(:,:,2), wV(:,:,2), wD(:,:,2), 'haar');
    restorasi(:,:,3) ,
wH(:,:,3), wV(:,:,3), wD(:,:,3), 'haar');
```

LAMPIRAN E (LANJUTAN)

```

else
waA3 = (wtA3 - cA3) / parameter;

    waA2(:, :, 1) ,
wH3(:, :, 1), wV3(:, :, 1), wD3(:, :, 1), 'haar');
    waA2(:, :, 2),
wH3(:, :, 2), wV3(:, :, 2), wD3(:, :, 2), 'haar');
    waA2(:, :, 3) ,
wH3(:, :, 3), wV3(:, :, 3), wD3(:, :, 3), 'haar');
    waA(:, :, 1) ,
wH2(:, :, 1), wV2(:, :, 1), wD2(:, :, 1), 'haar');
    waA(:, :, 2) ,
wH2(:, :, 2), wV2(:, :, 2), wD2(:, :, 2), 'haar');
    waA(:, :, 3) ,
wH2(:, :, 3), wV2(:, :, 3), wD2(:, :, 3), 'haar');
    restorasi(:, :, 1) ,
wH(:, :, 1), wV(:, :, 1), wD(:, :, 1), 'haar');
    restorasi(:, :, 2) ,
wH(:, :, 2), wV(:, :, 2), wD(:, :, 2), 'haar');
    restorasi(:, :, 3) ,
wH(:, :, 3), wV(:, :, 3), wD(:, :, 3), 'haar');
end

axes(handles.axes3)
imshow(uint8(restorasi))

```

BIODATA PENULIS



Penulis bernama lengkap Stephanie Gani, yang biasa dipanggil Fanny. Penulis dilahirkan di Surabaya pada tanggal 4 November 1992. Penulis lulus dari SD Kristen Petra 7 Surabaya, SMP Kristen Petra 3 Surabaya, SMA Kristen Petra 2 Surabaya. Pada tahun 2011 penulis mengikuti tes SNMPTN dan penulis diterima di perguruan tinggi negeri ITS Surabaya, Jurusan Matematika..

Semenjak kuliah penulis senang mempelajari bahasa pemrograman. Bahasa Pemrograman yang pernah

penulis pelajari adalah C, C++, dan Java. Oleh karena itu, penulis memilih bidang minat Ilmu Komputer.

Semasa menempuh jenjang pendidikan S-1, penulis juga aktif dalam kegiatan no-akademis diantaranya pernah menjadi Pembicara di Seminar dengan Tema “***Stay Classy and Act Like a Boss***” di Universitas Ciputra pada tahun 2016. Untuk mendapatkan informasi yang berhubungan dengan Tugas Akhir ini dapat ditujukan ke alamat email : stephaniegani11@gmail.com