



TUGAS AKHIR -KI091391

IMPLEMENTASI WATERMARK CITRA DENGAN MENGGUNAKAN REGRESI LINIER DAN METODE WAVELET

**RINA KHARISMA JUWITASARI
NRP 5109 100 077**

**Dosen Pembimbing I
Diana Purwitasari, S.Kom., M.Sc.**

**Dosen Pembimbing II
Rully Soelaiman, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2014**



UNDERGRADUATE THESES - KI091391

IMPLEMENTATION IMAGE WATERMARKING USING LINEAR REGRESSION AND WAVELET METHOD

**RINA KHARISMA JUWITASARI
NRP 5109 100 077**

**First Supervisor
Diana Purwitasari, S.Kom., M.Sc.**

**Second Supervisor
Rully Soelaiman, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2014**

IMPLEMENTASI *WATERMARK* CITRA DENGAN MENGUNAKAN REGRESI LINIER DAN METODE WAVELET

Nama Mahasiswa : Rina Kharisma Juwitasari
NRP : 5109100077
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Diana Purwitasari, S.Kom, M.Sc.
Dosen Pembimbing II : Rully Soelaiman, S.Kom, M.Kom.

Abstrak

Perkembangan teknologi informasi yang semakin canggih mengakibatkan mudahnya suatu citra untuk disebarluaskan dan dimanipulasi. Oleh karena itu diperlukan suatu sistem untuk mengamankan suatu karya citra digital agar tidak dapat dengan mudah dimanipulasi oleh pihak yang tidak bertanggungjawab.

Proses pemilihan bagian yang mau disisipkan yaitu dengan menggunakan Lifting Wavelet Transform dan Regresi Linier. Pada proses penyisipan akan dilakukan pemilihan bagian sub-blok mana yang akan dipilih untuk disisipi citra logo watermark, sehingga tidak semua bagian dari sub-blok yang ada akan disisipi citra logo watermark. Pemilihan sub-blok ini menggunakan metode regresi linier untuk melakukan perhitungan nilai slope. Integer Lifting Wavelet Transform (LWT) merupakan suatu metode yang digunakan dalam melakukan transformasi sinyal ke dalam beberapa koefisien sinyal. Pada LWT, penggambaran sebuah skala waktu sinyal digital didapatkan dengan menggunakan teknik filterisasi digital. Pada filterisasi ini, hasil dari highpass filter disebut sebagai sinyal detil (detail signal) dan hasil dari lowpass filter disebut sebagai sinyal aproksimasi (approximation signal).

Sistem kemudian membandingkan kemiripan antara citra logo watermark original dengan citra logo watermark hasil output,

hasilnya sama atau tidak. Sebelum disisipkan, citra logo (watermark) diacak terlebih dahulu menggunakan metode Arnold Cat Map sehingga logo tersebut tidak dapat dengan mudah dideteksi dan hanya berupa gambar abstrak.

Kata kunci: Arnold Cat Map, Lifting Wavelet Transform, Regresi Linier, watermark.

IMPLEMENTATION IMAGE WATERMARKING USING LINEAR REGRESSION AND WAVELET METHOD

Name : Rina Kharisma Juwitasari
NRP : 5109100077
Department : Teknik Informatika – FTIf ITS
Supervisor I : Diana Purwitasari, S.Kom., M.Sc.
Supervisor II : Rully Soelaiman, S.Kom., M.Kom.

Abstract

The improvement of information and technology that much better make a digital image so easy to be spreaded an manipulated. So it's needed a system to secure a creation of digital image to make it hard to access by another people especially people who has no responsibility.

The selection process of part to embed which will be selected to embeded watermark is using Lifting Wavelet Transform and Linear Regression. In embedding process, choose which sub-block will be embedded by watermark image. To do sub-block selection using Linear Regression method in calculate value of slope. Integer Lifting Wavelet Transform (LWT) is a method which is used in signal transformation into several kind of signals. In LWT process, to get these signals using digital filtering method. In this filtering, the value of highpass filter called detail signal and the value of lowpass filter called approximation signal.

Next, this system will compare the similarity of original watermark logo image with watermark logo image from the output of the process, the result of watermark logo image same or not. Before doing embedding process, the watermark logo image is disordered using Arnold Cat Map method. So this watermark logo image can't be easily to detected and only being an abstract shape.

Keywords: Arnold Cat Map, Lifting Wavelet Transform, Linear Regression, watermark.

LEMBAR PENGESAHAN

IMPLEMENTASI WATERMARK CITRA DENGAN MENGUNAKAN WAVELET DAN REGRESI LINIER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Komputasi Cerdas dan Visualisasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

RINA KHARISMA JUWITASARI
NRP. 5109 100 077

Disetujui oleh Pembimbing Tugas Akhir:

1. **Diana Purwitasari, S.Kom., M.Sc.**
NIP. 19780410 200312 2 001
(pembimbing 1)
2. **Rully Soelaiman, S.Kom., M.Kom.**
NIP. 19700213 199402 1 001
(pembimbing 2)



SURABAYA
JULI, 2014

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT. yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis akhirnya dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi Watermark Citra dengan Menggunakan Regresi Linier dan Metode Wavelet”.

Harapan dari penulis, semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata bagi kampus Teknik Informatika, ITS, dan bangsa Indonesia.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak. Tanpa mengurangi rasa hormat, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Rully Soelaiman, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah meluangkan banyak waktu serta memberikan banyak arahan, bantuan, nasihat dan semangat sehingga penulis mampu menyelesaikan Tugas Akhir ini.
2. Kedua orang tua penulis yang senantiasa memberi nasehat dan mendoakan setiap waktu agar penulis dapat segera menyelesaikan Tugas Akhir ini tanpa putus asa.
3. Bu Diana Purwitasari, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberikan semangat serta dukungan sehingga penulis akhirnya mampu menyelesaikan Tugas Akhir ini.
4. Ibu Dr. Nanik Suciati selaku ketua jurusan Teknik Informatika ITS, serta segenap dosen Teknik Informatika yang telah mengajarkan banyak ilmu kepada penulis.
5. Nyoman Juniarta selaku teman seangkatan penulis yang telah rela membantu dan meluangkan banyak waktu untuk penulis pada saat pengerjaan Tugas Akhir ini.

6. Teman-teman seperjuangan di laboratorium Simulasi dan Optimasi: Tora, Ines, Wayan, Nana, Kiki, Valen, Fahry, Wilda yang menemani, saling membantu, dan menyemangati penulis dalam pengerjaan Tugas Akhir ini.
7. Teman-teman angkatan 2009 yang telah memberikan dukungan dan bantuan selama ini.
8. Segenap pihak dan teman dekat penulis yang tidak dapat disebutkan satu persatu. Terima kasih banyak atas perhatian dan semangat yang kalian berikan, sehingga penulis akhirnya mampu menyelesaikan Tugas Akhir ini.

Penulis menyadari masih banyak kekurangan dalam Tugas Akhir ini. Untuk itu penulis mengharapkan adanya kritik dan saran yang membangun guna menyempurnakan Tugas Akhir ini.

Surabaya, Juli 2014

Rina Kharisma Juwitasari

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi.....	3
1.7 Sistematika Penulisan	5
BAB II DASAR TEORI.....	7
2.1 Citra Digital	7
2.1.1 Citra Grayscale.....	7
2.1.2 Citra Biner.....	8
2.2 <i>Watermark</i>	8
2.3 Penyisipan dan Ekstraksi Citra Logo <i>Watermark</i>	9
2.3.1 Penyisipan Citra logo <i>Watermark</i>	9
2.3.2 Ekstraksi Citra Logo <i>Watermark</i>	11
2.4 Enkripsi Citra.....	11
2.4.1 Arnold Cat Map.....	12
2.4.2 Citra Terenkripsi	14
2.5 Integer Lifting Wavelet Transform.....	14

2.6	Regresi Linier.....	19
2.7	Penilaian Kualitas Citra	21
2.7.1	Mean Signal Error	21
2.7.2	Peak Signal to Noise Ratio.....	22
BAB III PERANCANGAN PERANGKAT LUNAK.....		23
3.1	Desain Metode Secara Umum.....	23
3.2	Perancangan Data.....	23
3.2.1	Data Masukan.....	25
3.2.2	Data Keluaran.....	25
3.3	Algoritma dan Diagram Alir	27
3.3.1	Proses Penyisipan.....	27
3.3.2	Proses Ekstraksi.....	46
BAB IV IMPLEMENTASI.....		57
4.1	Lingkungan Implementasi.....	57
4.2	Implementasi.....	57
4.2.1	Implementasi Program Utama Proses Penyisipan	58
4.2.2	Implementasi Program Utama Proses Ekstraksi...	67
BAB V HASIL UJI COBA DAN EVALUASI.....		71
5.1	Lingkungan Uji Coba.....	71
5.2	Data Uji Coba.....	71
5.3	Skenario Uji Coba.....	72
5.3.1	Uji Kebenaran.....	72
5.3.2	Uji Kinerja.....	73
BAB VI KESIMPULAN DAN SARAN.....		91
6.1	Kesimpulan	91
6.2	Saran	92
DAFTAR PUSTAKA.....		93
LAMPIRAN A		95

BIODATA PENULIS..... 135

DAFTAR GAMBAR

Gambar 2.1 (a) Citra Logo RGB (b) Citra Logo RGB yang Telah Diubah Menjadi Biner	14
Gambar 2.2 Citra Terenkripsi.....	14
Gambar 2.3 Contoh Hasil Keluaran Proses LWT Berupa Matriks cAint.....	18
Gambar 3.1 Diagram Alir Proses Secara Umum (a) Proses Penyisipan (b) Proses Ekstraksi	24
Gambar 3.2 Citra Lena <i>Grayscale</i> Berukuran 512×512	25
Gambar 3.3 Hasil Keluaran Berupa Citra Ter- <i>Watermark</i>	26
Gambar 3.4 Hasil Keluaran Berupa Data Manifest (manifest_logo2.txt)	26
Gambar 3.5 Hasil Keluaran Citra Logo Hasil Ekstraksi	27
Gambar 3.6 Diagram Alir Proses Penyisipan Secara Keseluruhan	28
Gambar 3.7 <i>Pseudocode</i> Enkripsi Arnold Cat Map	29
Gambar 3.8 Pembagian Sub-Blok 8×8	30
Gambar 3.9 Diagram Alir Pembagian Sub-Blok Citra <i>Cover</i>	31
Gambar 3.10 <i>Pseudocode</i> Pembagian Sub-Blok Citra <i>Cover</i>	31
Gambar 3.11 Diagram Alir Penerapan Integer LWT	33
Gambar 3.12 <i>Pseudocode</i> Penerapan Integer LWT	33
Gambar 3.13 Diagram Alir Proses Mod Q.....	35
Gambar 3.14 <i>Pseudocode</i> Proses Mod Q.....	35
Gambar 3.15 Diagram Alir Proses Penentuan Nilai x dan y	37
Gambar 3.16 <i>Pseudocode</i> Proses Penentuan Nilai x dan y	37
Gambar 3.17 Diagram Alir Komputasi Slope	38
Gambar 3.18 <i>Pseudocode</i> Komputasi Nilai Slope	39
Gambar 3.19 <i>Pseudocode</i> Pemilihan Sub-Blok Disisipi (bagian 1)	40
Gambar 3.20 <i>Pseudocode</i> Pemilihan Sub-Blok Disisipi (bagian 2)	41
Gambar 3.21 <i>Pseudocode</i> Proses Pembuatan Manifest.....	42
Gambar 3.22 <i>Pseudocode</i> Penerapan Inverse LWT (bagian 1)...	43
Gambar 3.23 <i>Pseudocode</i> Penerapan Inverse LWT (bagian 2)...	44

Gambar 3.24 <i>Pseudocode</i> Proses Komputasi PSNR	45
Gambar 3.25 Diagram Alir Proses Ekstraksi Secara Keseluruhan	47
Gambar 3.26 Diagram Alir Pembagian Sub-Blok Citra Ter- <i>Watermark</i>	48
Gambar 3.27 <i>Pseudocode</i> Proses Pembagian Sub-Blok Citra Ter- <i>Watermark</i> (bagian 1).....	48
Gambar 3.28 <i>Pseudocode</i> Proses Pembagian Sub-Blok Citra Ter- <i>Watermark</i> (bagian 2).....	49
Gambar 3.29 Diagram Alir Penerapan Integer LWT Pada Citra Ter- <i>Watermark</i>	50
Gambar 3.30 <i>Pseudocode</i> Penerapan Wavelet Pada Citra Ter- <i>Watermark</i>	50
Gambar 3.31 Diagram Alir proses Mod Q_w	52
Gambar 3.32 <i>Pseudocode</i> Proses Mod Q	52
Gambar 3.33 <i>Pseudocode</i> Proses Ekstraksi Citra Logo <i>Watermark</i> (bagian 1).....	53
Gambar 3.34 <i>Pseudocode</i> Proses Ekstraksi Citra Logo <i>Watermark</i> (bagian 2).....	54
Gambar 3.35 <i>Pseudocode Inverse</i> Enkripsi Citra Logo <i>Watermark</i>	55
Gambar 5.1 Pengujian Hasil Ekstraksi (a) Citra Logo <i>Watermark</i> Original (b) Citra Logo <i>Watermark</i> Hasil Ekstraksi.....	72
Gambar 5.2 Citra Hasil Keluaran Skenario 1	73
Gambar 5.3 Grafik Skenario 1 Berdasarkan <i>Running Time</i>	74
Gambar 5.4 Grafik Skenario 1 Berdasarkan PSNR.....	74
Gambar 5.5 Citra Hasil Keluaran Skenario 2	75
Gambar 5.6 Grafik Skenario 2 Berdasarkan <i>Running Time</i>	76
Gambar 5.7 Grafik Skenario 2 Berdasarkan PSNR.....	76
Gambar 5.8 Citra Hasil Keluaran Skenario 3	77
Gambar 5.9 Grafik Skenario 3 Berdasarkan <i>Running Time</i>	78
Gambar 5.10 Grafik Skenario 3 Berdasarkan PSNR.....	78
Gambar 5.11 Citra Hasil Keluaran Skenario 4	79
Gambar 5.12 Grafik Skenario 4 Berdasarkan <i>Running Time</i>	80
Gambar 5.13 Grafik Skenario 4 Berdasarkan PSNR.....	80

Gambar 5.14 Citra Hasil Keluaran Skenario 5	81
Gambar 5.15 Grafik Skenario 5 Berdasarkan <i>Running Time</i>	82
Gambar 5.16 Grafik Skenario 5 Berdasarkan PSNR.....	82
Gambar 5.17 Citra Hasil Keluaran Skenario 6	83
Gambar 5.18 Grafik Skenario 6 Berdasarkan <i>Running Time</i>	84
Gambar 5.19 Grafik Skenario 6 Berdasarkan PSNR.....	84
Gambar 5.20 Citra Hasil Keluaran Skenario 7	85
Gambar 5.21 Grafik Skenario 7 Berdasarkan <i>Running Time</i>	86
Gambar 5.22 Grafik Skenario 7 Berdasarkan PSNR.....	86
Gambar 5.23 Citra Hasil Keluaran Skenario 8	87
Gambar 5.24 Grafik Skenario 8 Berdasarkan <i>Running Time</i>	88
Gambar 5.25 Grafik Skenario 8 Berdasarkan PSNR.....	88

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode untuk Memilih dan Membaca Citra Logo <i>Watermark</i>	58
Kode Sumber 4.2 Kode untuk Memilih dan Membaca Citra <i>Cover</i>	58
Kode Sumber 4.3 Kode untuk Memanggil Fungsi Arnold Cat Map	59
Kode Sumber 4.4 Kode untuk Memanggil Fungsi PSNR	59
Kode Sumber 4.5 Kode Implementasi Fungsi Arnold Cat Map (bagian 1).....	59
Kode Sumber 4.6 Kode Implementasi Fungsi Arnold Cat Map (bagian 2).....	60
Kode Sumber 4.7 Kode untuk Membagi Sub-Blok.....	60
Kode Sumber 4.8 Kode untuk Menerapkan Integer Lifting Wavelet Transform-2D Level 1	61
Kode Sumber 4.9 Kode untuk Melakukan Mod Q.....	62
Kode Sumber 4.10 Kode untuk Menentukan Nilai x dan y.....	62
Kode Sumber 4.11 Kode untuk Menghitung Nilai Slope.....	63
Kode Sumber 4.12 Kode untuk Melakukan Pemilihan Sub-Blok yang Disisipi Citra Logo <i>Watermark</i>	64
Kode Sumber 4.13 Kode untuk Membuat Manifest.....	65
Kode Sumber 4.14 Kode untuk Melakukan Inverse LWT	65
Kode Sumber 4.15 Kode untuk Menghitung Nilai PSNR.....	66
Kode Sumber 4.16 Kode Pembagian Sub-Blok Pada Citra Ter- <i>Watermark</i>	67
Kode Sumber 4.17 Kode untuk Menerapkan Integer Lifting Wavelet Transform-2D Level 1 Pada Citra Ter- <i>Watermark</i>	68
Kode Sumber 4.18 Kode untuk Melakukan Mod Q.....	69
Kode Sumber 4.19 Kode untuk Melakukan Proses Ekstraksi Citra Logo <i>Watermark</i> (bagian 1)	69
Kode Sumber 4.20 Kode untuk Melakukan Proses Ekstraksi Citra Logo <i>Watermark</i> (bagian 2)	70
Kode Sumber 4.21 Kode untuk Melakukan <i>Inverse</i> Arnold Cat Map.....	70

BAB I

PENDAHULUAN

Pada bagian ini akan dijelaskan beberapa hal dasar mengenai Tugas Akhir ini meliputi: latar belakang, rumusan permasalahan, batasan permasalahan, tujuan, manfaat dari Tugas Akhir, metodologi serta sistematika penulisan Tugas Akhir. Penjelasan tentang hal-hal tersebut diharapkan dapat memberikan gambaran umum mengenai permasalahan sehingga penyelesaian masalah dapat dipahami dengan baik.

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat dan canggih pada saat ini semakin menuntut peningkatan keamanan, terutama dalam bidang pengolahan citra digital. Semakin canggih teknologi yang ada, maka semakin memungkinkan banyak terjadi manipulasi dan pelanggaran yang dilakukan oleh banyak pihak yang tidak bertanggung jawab. Pelanggaran yang biasanya terjadi misalkan saja terjadi penduplikasian dan pembajakan karya citra digital milik orang lain. Hal ini terjadi salah satunya karena minimnya pengamanan yang dilakukan terhadap suatu citra digital, sehingga dapat dengan mudah disalah gunakan oleh pihak yang tidak diinginkan.

Berbagai upaya dilakukan untuk mengamankan citra digital, salah satunya yaitu dengan menggunakan *watermark* digital guna menandai karya citra digital dengan harapan citra digital tidak dapat dengan mudah diduplikasi dan disalah gunakan. *Watermark* digital [1] merupakan suatu metode untuk menyisipkan suatu penanda kepemilikan seperti citra logo atau citra tanda tangan ke dalam citra digital yang ingin dilakukan pengamanan. Penyisipan yang dilakukan bersifat *invisible* [2], artinya citra *watermark* yang disisipkan ke dalam citra digital secara kasat mata manusia tidak dapat dengan mudah dilihat dan diketahui. Sehingga

citra *watermark* yang disisipkan tidak dapat dengan mudah dimanipulasi oleh pihak yang tidak diinginkan.

Pada proses pembuatan sistem perangkat lunak ini menggunakan metode Integer LWT dan Regresi Linier. Penggunaan Integer LWT dilakukan karena hasil transformasi sinyal yang dihasilkan berupa bilangan *integer*, sehingga dapat menyederhanakan tingkat kompleksitasnya. Ketika suatu citra ditransformasikan ke dalam sebuah domain *wavelet* menggunakan transformasi Wavelet konvensional, nilai-nilai dari koefisien-koefisien Wavelet akan berupa *floating point*. Jika koefisien-koefisien ini diubah pada waktu penyisipan *watermark*, blok citra ber-*watermark* yang berkorespondensi terhadap *watermark* mempunyai nilai yang tidak akurat [2]. Sedangkan penggunaan Regresi Linier dilakukan untuk mengetahui korelasi antara variabel x dan variabel y yang digunakan pada perhitungan slope [3].

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana menyisipkan citra logo *watermark* ke dalam citra *cover*, sehingga tidak dapat dengan mudah diketahui oleh pihak lain?
2. Bagaimana melakukan ekstraksi logo *watermark* yang telah disisipkan ke dalam citra *cover*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Citra *cover* yang digunakan bertipe *grayscale* dan berukuran 512×512 .
2. Citra logo *watermark* yang digunakan berukuran 32×32 dan bertipe biner.
3. Implementasi dilakukan dengan menggunakan Matlab 7.6.0 (R2008a).

4. Citra logo *watermark* menggunakan citra berformat .bmp.
5. Citra *cover* menggunakan citra berformat .bmp, .jpg, dan .png.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Melakukan pengamanan citra digital pada proses penyisipan dan ekstraksi citra logo *watermark* dengan menggunakan Arnold Cat Map.
2. Mengimplementasikan Lifting Wavelet Transform yang digunakan dalam proses penyisipan dan ekstraksi citra logo *watermark*.
3. Mengevaluasi hasil keluaran citra logo *watermark* pada proses ekstraksi.

1.5 Manfaat

Tugas Akhir ini dikerjakan dengan harapan dapat memberikan manfaat pada bidang teknologi informasi dalam melakukan pengamanan citra digital berupa penyisipan citra logo *watermark* dan ekstraksi citra logo *watermark*. Sehingga dapat meminimalkan terjadinya penduplikasian dan pemanipulasian citra digital yang dilakukan oleh pihak yang tidak diinginkan.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan untuk mengimplementasikan penyisipan *watermark* menggunakan metode Wavelet dan regresi linier. Setelah dilakukan penyisipan, selanjutnya akan

dilakukan proses ekstraksi untuk menampilkan hasil citra logo yang telah disisipkan.

2. Studi Literatur

Pada tahapan ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi serta literatur yang diperlukan untuk pembuatan implementasi penyisipan dan ekstraksi citra logo *watermark*.

Dasar informasi yang diperlukan pada pembuatan implementasi ini diantaranya mengenai gambaran citra secara umum, algoritma Arnold Cat Map, Wavelet dan Regresi Linier yang digunakan untuk menyeleksi bagian mana yang akan disisipi *watermark*.

3. Perancangan Perangkat Lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi Perangkat Lunak

Implementasi merupakan tahap membangun dan merealisasikan rancangan sistem perangkat lunak yang telah dibuat sebelumnya, sehingga rancangan tersebut dapat menjadi sebuah sistem perangkat lunak yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan Evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan

dengan melihat kesesuaian dengan perencanaan yang telah dibuat.

Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem perangkat lunak, mencari masalah yang mungkin timbul, membandingkan seberapa besar tingkat akurasi citra yang telah disisipi citra logo *watermark* dengan citra asli serta mengadakan perbaikan jika terdapat kesalahan pada sistem perangkat lunak.

6. Penyusunan Buku Tugas Akhir

Tahapan ini merupakan tahap penyusunan buku yang memuat dokumentasi mengenai pembuatan serta hasil yang diperoleh dari implementasi perancangan sistem perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan
Bab ini berisi penjelasan mengenai latar belakang masalah, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu rumusan permasalahan, batasan permasalahan, dan sistematika penulisan juga merupakan bagian dari bab ini.
2. Bab II. Dasar Teori
Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang untuk mendukung pembuatan Tugas Akhir ini.
3. Bab III. Perancangan Perangkat Lunak
Bab ini berisi penjelasan mengenai desain dan perancangan yang digunakan dalam Tugas Akhir yang disajikan dalam bentuk diagram alir.

4. Bab IV. Implementasi
Bab ini akan dilakukan implementasi aplikasi yang dibangun dengan MATLAB 7.6.0 (R2008a) sesuai dengan permasalahan dan batasannya yang telah dijabarkan pada bab pertama.
5. Bab V. Uji Coba dan Evaluasi
Bab ini berisi penjelasan mengenai data hasil percobaan atau pengukuran, dan pembahasan mengenai hasil percobaan yang telah dilakukan.
6. Bab VI. Kesimpulan dan Saran
Bab ini berupa hasil penelitian yang menjawab permasalahan atau yang berupa konsep, program, dan karya rancangan. Selain itu, pada bab ini berisi saran dan kesimpulan yang berisi hal-hal yang masih dapat dikerjakan dengan lebih baik dan dapat dikembangkan lebih lanjut, atau berisi masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir.

BAB II

DASAR TEORI

Bab ini membahas tentang teori dasar yang menunjang dalam penyusunan Tugas Akhir mengenai permasalahan penyisipan citra logo *watermark*, ekstraksi citra logo *watermark*, wavelet transform, dan Arnold Cat Map.

2.1 Citra Digital

Citra digital merupakan representasi citra dalam bidang digital yang biasanya diwakili oleh suatu matriks, terdiri dari M baris dan N kolom. Citra ini memiliki bentuk dua dimensi yang dapat ditampilkan pada layar komputer dalam piksel, yaitu perpotongan antara garis dan kolom dalam suatu citra. Beberapa jenis citra digital antara lain citra *grayscale* dan citra biner yang akan dijelaskan lebih lanjut pada bagian ini.

2.1.1 Citra Grayscale

Citra *grayscale* merupakan suatu citra digital yang hanya memiliki informasi kecerahan tanpa adanya unsur warna. Masing-masing nilai piksel pada citra *grayscale* berisi informasi mengenai banyaknya cahaya yang diterima pada titik tersebut. Citra yang ditampilkan dari citra *grayscale* terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang memiliki intensitas paling lemah sampai warna putih pada bagian yang memiliki intensitas paling kuat. Pada citra *grayscale* warna bervariasi antara hitam dan putih. Tetapi variasi warna di antaranya sangatlah banyak.

Nilai piksel pada citra *grayscale* pada umumnya memiliki kedalaman 8 bit untuk setiap pikselnya dan berada pada rentang antara 0 sampai 255. Semakin mendekati nilai 0, maka warna pada citra akan semakin gelap yang berarti semakin sedikit cahaya yang diterima. Sedangkan semakin mendekati nilai 255, maka warna

pada citra akan semakin cerah yang berarti semakin banyak cahaya yang diterima. Contoh citra *grayscale* dapat dilihat pada Gambar 3.2.

2.1.2 Citra Biner

Citra biner merupakan suatu citra yang hanya mempunyai dua nilai derajat keabuan, yaitu hitam dan putih. Citra biner direpresentasikan nilai tiap-tiap pikselnya dalam satu bit (satu nilai *binary*). Nilai piksel 0 menunjukkan warna hitam, sedangkan nilai piksel 1 menunjukkan warna putih [4].

Meski saat ini citra berwarna lebih banyak digunakan untuk merepresentasikan suatu citra yang memiliki banyak variasi warna, tetapi bukan berarti citra biner sudah tidak digunakan. Citra biner biasanya masih digunakan pada citra logo instansi, citra kode batang (*bar code*) yang biasanya digunakan untuk memberi label kode pada barang, dan lain sebagainya. Contoh citra biner dapat dilihat pada Gambar 2.1.

2.2 Watermark

Watermark merupakan suatu teknik yang digunakan untuk proses penandaan suatu citra digital dengan menyisipkan informasi tersembunyi ke dalam citra digital [5]. *Watermark* yang digunakan pada gambar digital dapat berupa gambar logo maupun teks. *Watermark* harus memenuhi beberapa syarat [5], antara lain:

1. Tidak tampak, artinya *watermark* haruslah tidak dapat dengan mudah diketahui oleh penglihatan mata normal (*invisible*).
2. Keunikan kunci, artinya yaitu *watermark* yang digunakan haruslah unik dan berbeda dengan *watermark* yang lain.
3. Tidak dapat dibalik, artinya proses *watermark* tersebut haruslah tidak dapat dengan mudah dibalik untuk menghilangkan *watermark* yang telah disisipkan.
4. Ketergantungan citra, artinya citra *watermark* yang dihasilkan hanya dapat menggunakan satu kunci saja.

5. Pendeteksiannya efektif, artinya *watermark* haruslah dapat dengan efisien dideteksi.
6. Ketahanan, artinya *watermark* harus tahan terhadap berbagai macam serangan yang mungkin terjadi.

2.3 Penyisipan dan Ekstraksi Citra Logo Watermark

Penyisipan dan ekstraksi merupakan suatu proses yang terpisah tetapi saling berhubungan. Penyisipan merupakan suatu proses untuk menyisipkan citra logo *watermark* ke dalam suatu citra yang dikehendaki. Proses ini bertujuan untuk mengamankan suatu citra digital sehingga tidak mudah diklaim dan dimanipulasi oleh orang lain. Penyisipan dilakukan secara *invisible*, agar citra logo *watermark* yang disisipkan tidak dapat dengan mudah diketahui secara ilegal sehingga tidak dapat dengan mudah juga untuk dihilangkan.

Sedangkan ekstraksi merupakan suatu proses untuk mengekstrak suatu citra logo *watermark* dari suatu citra *cover* yang sebelumnya telah disisipi citra logo *watermark* pada proses penyisipan. Hasil keluaran proses ekstraksi berupa citra logo *watermark* yang telah disisipkan di dalam citra *cover* tersebut untuk membuktikan bahwa di dalam citra *cover* tersebut memang benar terdapat citra logo *watermark* yang disisipkan.

2.3.1 Penyisipan Citra logo Watermark

Proses penyisipan citra logo *watermark* terdiri dari beberapa tahapan, yaitu:

1. Citra *cover* yang bertipe *grayscale* berukuran 512×512 dibagi menjadi sub-blok berukuran 8×8 , sehingga jumlah sub-blok yang ada yaitu 64×64 sub-blok. Misalkan saja digunakan citra *cover* Lena.
2. Dilakukan Integer Lifting Wavelet Transform 2D level 1 ke dalam tiap-tiap sub-blok tersebut, yang kemudian menghasilkan 4 macam frekuensi sinyal yang masing-

masing memiliki ukuran 4×4 . Beberapa sinyal tersebut yaitu: c_{Aint} , c_{Hint} , c_{Vint} dan c_{Dint} . Tetapi hanya sinyal rendah (c_{Aint}) yang diambil untuk proses selanjutnya.

3. Ekstrak frekuensi rendah (c_{Aint}) dan simpan sebagai variabel B_i yang berukuran 4×4 .
4. Lakukan Modulus Q terhadap B_i , yang kemudian menghasilkan C_i dan D_i . Masing-masing dari C_i dan D_i berukuran 4×4 . C_i merupakan hasil sisa, sedangkan D_i merupakan hasil pembagiannya.
5. Siapkan citra logo watermark dengan format biner berukuran 32×32 .
6. Metode penyisipan dilakukan secara berselang pada tiap-tiap sub-blok dengan menggunakan slope. Ketika slope berada di dalam rentang antara 0.1 sampai dengan 1.0, maka dipilih untuk disisipi citra logo *watermark*.
7. Ketika piksel citra logo watermark bernilai 0, maka C_i yang ada pada sub-blok tersebut akan diproses. Pilih 4 piksel bertetangga yang ada pada bagian tengah C_i , hitung nilai mean dari 4 piksel tersebut. Kemudian gunakan nilai mean tersebut untuk mengganti nilai 4 piksel bertetangga yang ada sebelumnya, dan disimpan pada variabel baru yaitu variabel C_{iaksen} berukuran 4×4 . Tetapi jika piksel citra logo watermark bernilai 1, maka tidak akan dimodifikasi.
8. Setelah itu jumlahkan matriks C_{iaksen} dengan D_i , simpan hasilnya pada variabel B_{iaksen} berukuran 4×4 .
9. Lakukan inverse LWT untuk menampilkan citra Lena yang telah disisipi watermark (citra terwatermark).
10. Untuk membandingkan kualitas citra Lena original dengan citra terwatermark yaitu dengan menggunakan PSNR. Semakin besar nilai PSNR, semakin mendekati citra aslinya. Sedangkan MSE adalah kebalikan dari PSNR, semakin kecil nilai MSE maka semakin bagus kualitas citra tersebut.

2.3.2 Ekstraksi Citra Logo *Watermark*

Proses ekstraksi citra logo *watermark* terdiri dari beberapa tahapan, yaitu:

1. Gunakan citra *cover* Lena yang telah disisipi watermark (citra ter-*watermark*) sebagai citra masukan.
2. Bagi citra ter-*watermark* berukuran 512×512 tersebut menjadi sub-blok yang berukuran 8×8 , sehingga total sub-blok yang ada yaitu 64×64 sub-blok.
3. Lakukan Integer LWT-2D level 1 pada masing-masing sub-blok tersebut, sehingga menghasilkan cA_{int_w} , $cHint_w$, cV_{int_w} dan cD_{int_w} .
4. Ambil cA_{int_w} lalu simpan sebagai variabel B_{i_w} berukuran 4×4 .
5. Lakukan MOD Q terhadap B_{i_w} , yang kemudian menghasilkan C_{i_w} berukuran 4×4 dan D_{i_w} berukuran 4×4 . C_{i_w} merupakan hasil sisa, sedangkan D_{i_w} merupakan hasil pembagiannya.
6. Ambil nilai C_{i_w} untuk dilakukan proses ekstraksi. Lakukan pengecekan tiap-tiap sub-blok dengan cara berselang, lalu cek apakah pada 4 matriks bertangga di bagian tengah C_{i_w} terdapat 4 angka yang sama. Jika ada, maka bit citra logo *watermark* yang ada pada sub-blok tersebut bernilai 0. Jika tidak ada, maka bernilai 1. Begitu seterusnya hingga membentuk citra logo *watermark* utuh berukuran 32×32 .

2.4 Enkripsi Citra

Citra digital yang akan ditransmisikan ke media umum seperti web maupun email, perlu untuk dilakukan pengamanan agar tidak diklaim oleh pihak lain yang tidak bertanggung jawab. Salah satu cara yang dapat dilakukan untuk melakukan pengamanan citra yaitu dengan menggunakan enkripsi citra.

Enkripsi citra biasanya digunakan untuk melindungi konten yang terkandung di dalam citra dari pengaksesan oleh pihak yang tidak diinginkan. Enkripsi citra mentransformasikan citra ke dalam bentuk lain yang kurang bisa dipahami secara visual.

Beberapa tahapan yang dapat dilakukan antara lain, yang pertama adalah dengan menggunakan fase enkripsi. Tujuan enkripsi yaitu mengubah pesan atau logo asli ke dalam bentuk yang tidak beraturan sehingga tidak dapat dengan mudah untuk dikenali. Tahapan yang kedua adalah menggunakan fase penyisipan. Pesan atau logo yang telah dienkripsi, disisipkan ke dalam bagian dari suatu gambar yang akan dikirimkan. Tahapan yang ketiga adalah dengan menggunakan fase penyembunyian, yaitu dengan cara menyembunyikan pesan maupun logo yang telah dienkripsi dan disisipkan menjadi tidak terlihat (*invisible*) [6].

Pada bagian ini akan dibahas mengenai metode enkripsi yang digunakan dalam Tugas Akhir ini. Metode yang digunakan yaitu transformasi menggunakan algoritma Arnold Cat Map yang akan dijelaskan lebih lanjut pada bagian ini.

2.4.1 Arnold Cat Map

Logo yang digunakan untuk disisipkan ke dalam citra *cover* terlebih dahulu dilakukan pengacakan dengan menggunakan metode Arnold Cat Map. Tujuan pengacakan yaitu agar logo yang disisipkan ke dalam citra *cover* tidak dapat dengan mudah diketahui oleh pihak lain yang tidak diinginkan. Metode ini mengacak citra dengan cara mentransformasikan koordinat (x, y) yang ada pada citra berukuran $N \times N$ ke koordinat baru (x', y') .

Pada setiap iterasi yang terjadi di dalam proses Arnold Cat Map akan dilakukan pergeseran ke arah sumbu y , kemudian dilakukan pergeseran ke arah sumbu x . Setelah itu semua hasilnya akan dimodulo dengan N agar hasil pengacakan akan tetap berada di dalam area citra [7]. Proses Arnold Cat map dilakukan seperti pada Persamaan 2.1.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} = A \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod F \quad (2.1)$$

Arnold Cat Map merupakan fungsi *chaos* dwimatra dan memiliki sifat *reversible*, artinya citra yang telah diacak dapat dikembalikan lagi ke bentuk citra semula [7]. Ketika dilakukan proses pengacakan, pada iterasi tertentu hasil pengacakan akan kembali ke citra semula tergantung pada ukuran citra yang diacak. Sehingga jumlah iterasi yang dilakukan untuk mengembalikan citra ke bentuk citra aslinya tidak selalu sama.

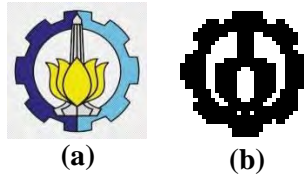
Algoritma [7] yang digunakan pada Arnold Cat Map adalah sebagai berikut. Y merupakan citra berukuran 32×32 , sedangkan p bernilai 32.

```

p = size(Y,1);
X = zeros(size(Y));
for i = 1 to p do
    for j = 1 to p do
        newi = mod(((i-1) + (j-1)),p) + 1;
        newj = mod(((i-1) + 2*(j-1)),p) + 1;
        X(newi,newj,:) = Y(i,j,:);
    end for
end for

```

Pada Tugas Akhir ini metode Arnold Cat Map digunakan dalam proses pengacakan citra logo *watermark* ketika sebelum disisipkan sehingga menghasilkan citra terenkripsi. Kemudian pada proses ekstraksi, citra logo *watermark* yang telah dihasilkan sebagai keluaran, diacak sebanyak iterasi tertentu sehingga citra logo *watermark* dapat kembali menjadi bentuk semula. Citra logo *watermark* yang digunakan berukuran 32×32 sehingga iterasi yang harus dilakukan untuk mengembalikan citra logo *watermark* yaitu sebanyak 23 kali. Citra logo *watermark* yang digunakan berukuran 32×32 , seperti pada Gambar 2.1. Contoh citra yang diacak menggunakan Arnold Cat Map dapat dilihat pada Gambar 2.2.



Gambar 2.1 (a) Citra Logo RGB (b) Citra Logo RGB yang Telah Diubah Menjadi Biner

2.4.2 Citra Terenkripsi

Citra terenkripsi merupakan citra yang disimpan setelah dilakukan suatu proses tertentu, sehingga citra tersebut menjadi bentuk lain yang tidak bermakna dengan tujuan agar tidak dapat dilihat secara visual oleh semua pihak. Hanya pihak yang memiliki wewenang saja yang mampu melihat isi citra yang sebenarnya.

Citra terenkripsi biasanya digunakan dalam menyembunyikan rahasia atau informasi yang tidak ingin diketahui secara ilegal oleh pihak lain. Biasanya citra terenkripsi direpresentasikan dalam derajat keabuan, sehingga warna yang terlihat hanya hitam dan putih. Contoh citra terenkripsi dapat dilihat pada Gambar 2.2.



Gambar 2.2 Citra Terenkripsi

2.5 Integer Lifting Wavelet Transform

Transformasi Wavelet (*Wavelet Transform*) merupakan suatu metode yang digunakan untuk menganalisa data dengan melakukan dekomposisi suatu sinyal ke dalam komponen yang

berbeda-beda. Biasanya sinyal dibedakan berdasarkan sinyal rendah (*low frequency*) dan sinyal tinggi (*high frequency*).

Integer Lifting Wavelet Transform (LWT) merupakan suatu metode yang digunakan dalam melakukan transformasi sinyal ke dalam beberapa koefisien sinyal. Integer LWT merupakan transformasi Wavelet generasi kedua. Transformasi Wavelet generasi pertama adalah DWT atau Transformasi Wavelet konvensional. Transformasi Wavelet konvensional tidak cocok untuk skema *watermarking* yang digunakan untuk pembuktian autentikasi suatu citra yang *reversible*. Ketika sebuah citra ditransformasikan ke dalam sebuah domain Wavelet menggunakan transformasi Wavelet konvensional, nilai-nilai dari koefisien-koefisien Wavelet akan berupa *floating point*. Apabila koefisien-koefisien ini diubah pada waktu penyisipan *watermark*, blok citra ter-*watermark* yang berkorespondensi terhadap *watermark* akan mempunyai nilai yang tidak akurat. Beberapa pemotongan nilai *floating point* dari piksel menyebabkan hilangnya informasi dan menyebabkan gagalnya sistem agar menjadi sistem *watermarking* autentikasi yang *reversible*. Untuk mengatasai persoalan ini, digunakan sebuah transformasi Wavelet *integer-to-integer* yang *invertible* [2].

Pada Transformasi Wavelet, penggambaran sebuah skala waktu sinyal digital didapatkan dengan menggunakan teknik filterisasi digital. Secara garis besar proses dalam teknik ini adalah dengan cara melewati sinyal yang akan dianalisis pada filter dengan frekuensi dan skala yang berbeda. Filterisasi sendiri merupakan sebuah fungsi yang digunakan dalam pemrosesan sinyal. Wavelet dapat direalisasikan menggunakan iterasi filter dengan penskalaan. Resolusi dari sinyal, yang merupakan rata-rata dari jumlah detail informasi dalam sinyal, ditentukan melalui filterisasi ini dan skalanya didapatkan dengan *upsampling* dan *downsampling*.

Sebuah sinyal harus dilewatkan dalam dua filterisasi yaitu *highpass filter* dan *lowpass filter* agar frekuensi dari sinyal tersebut dapat dianalisis. Analisis sinyal dilakukan terhadap hasil filterisasi

highpass filter dan *lowpass filter*. Untuk *highpass filter* digunakan untuk menganalisis sinyal frekuensi tinggi dan *lowpass filter* digunakan untuk menganalisis sinyal frekuensi rendah. Pembagian sinyal menjadi frekuensi rendah dan frekuensi tinggi ini disebut sebagai dekomposisi. Proses dekomposisi dimulai dengan melewati sinyal asal melalui *highpass filter* dan *lowpass filter*. Misalkan sinyal asal ini memiliki rentang frekuensi dari 0 sampai dengan $\pi \text{ rad/s}$. Pada saat melewati *highpass filter* dan *lowpass filter* ini, pada rentang frekuensi dilakukan *downsample* menjadi dua sehingga rentang tertinggi pada masing-masing *downsample* menjadi $\frac{\pi}{2} \text{ rad/s}$. Setelah filterisasi, setengah dari *sample* atau salah satu *downsample* dapat dieliminasi sehingga sinyal dapat selalu di-*downsample* oleh 2 ($\downarrow 2$) dengan cara mengabaikan setiap *sample* yang kedua. Pada filterisasi ini, hasil dari *highpass filter* disebut sebagai sinyal detail (*detail signal*) dan hasil dari *lowpass filter* disebut sebagai sinyal aproksimasi (*approximation signal*).

Dengan menggunakan 2 jenis sinyal ini kemudian dapat digunakan dalam pemrosesan *Inverse Integer LWT*, dengan tujuan merekonstruksi sinyal yang telah diproses menjadi sinyal asal. Proses rekonstruksi sinyal ini diawali dengan menggabungkan koefisien hasil akhir yang didapatkan dari proses dekomposisi dengan sebelumnya meng-*upsample* sinyal oleh 2 ($\uparrow 2$) melalui *highpass filter* dan *lowpass filter*. Contoh proses perhitungannya dapat dilihat pada bagian ini. Proses rekonstruksi ini merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi. Proses dekomposisi dilakukan sebanyak jumlah level yang telah ditentukan sebelumnya, yaitu menggunakan satu level dekomposisi. Untuk melakukan dekomposisi lebih dari satu kali, proses dekomposisi selanjutnya dilakukan dekomposisi pada koefisien aproksimasi (*cAint*), karena berisi sebagian besar informasi citra.

$$\text{Lowpass Filter} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\text{Highpass Filter} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Berikut ini contoh perhitungan *lowpass filter* [8] yang digunakan dalam proses dekomposisi:

- Langkah pertama adalah mengalikan filter *lowpass* dengan matriks yang akan diproses. Dalam hal ini, menggunakan contoh matriks M. Untuk memudahkan perkalian terhadap baris, sebaiknya dilakukan *transpose* pada matriks M sehingga diperoleh:

$$M^T = \begin{bmatrix} 1 & 5 & 2 & 7 \\ 2 & 8 & 5 & 4 \\ 3 & 2 & 8 & 7 \\ 2 & 6 & 9 & 4 \end{bmatrix}$$

- Selanjutnya dilakukan perkalian matriks M^T dengan filter *lowpass* yang menghasilkan D_1^T .

$$D_1^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \times \begin{bmatrix} 1 & 5 & 2 & 7 \\ 2 & 8 & 5 & 4 \\ 3 & 2 & 8 & 7 \\ 2 & 6 & 9 & 4 \end{bmatrix}$$

$$D_1^T = \begin{bmatrix} 2.1213 & 9.1923 & 4.9497 & 7.7781 \\ 3.5355 & 5.6568 & 12.0207 & 7.7781 \end{bmatrix}$$

- Untuk mengembalikan ke baris dan kolom sebenarnya, dilakukan proses *transpose* kembali pada matriks D_1^T . Keluaran ini merupakan hasil dari proses dekomposisi level 1.

$$D_1 = \begin{bmatrix} 2.1213 & 3.5355 \\ 9.1923 & 5.568 \\ 4.9497 & 12.0207 \\ 7.7781 & 7.7781 \end{bmatrix}$$

Perangkat lunak ini menggunakan fungsi Integer Lifting Wavelet Transform Level 1 yang ada di Matlab, dengan tujuan untuk membagi sinyal menjadi beberapa bagian, yaitu cAint, cDint, cVint dan cHint. cAint merupakan bagian yang mengandung sinyal frekuensi rendah atau biasa disebut sebagai koefisien aproksimasi (*approximation coefficients*). Sedangkan cDint (*diagonal detail coefficients*), cVint (*vertical detail coefficients*) dan cHint (*horizontal detail coefficients*) mengandung sinyal tinggi atau yang biasa disebut sebagai koefisien detail. Dalam program ini sinyal yang akan diambil dan diproses pada tahapan selanjutnya adalah sinyal frekuensi rendah (cAint).

Input yang digunakan pada proses ini yaitu berupa citra masukan berukuran 512×512 dengan format *grayscale*. Keluaran yang dihasilkan berupa matriks yang terbagi menjadi beberapa bagian, yaitu: cAint, cHint, cVint dan cDint. Contoh salah satu matriks hasil keluaran proses *filtering* dapat dilihat pada Gambar 2.3.

45	48	47	49
44	46	48	50
47	48	50	49
44	45	48	50

Gambar 2.3 Contoh Hasil Keluaran Proses LWT Berupa Matriks cAint

Tahapan pada Lifting Wavelet Transform yang ada pada fungsi Matlab `lwt2` yaitu:

1. Tahap Splitting
Tahapan ini membagi sinyal menjadi dua, yaitu sinyal rendah (L) pada proses *lowpass filter* dan sinyal tinggi (H) pada proses *highpass filter*.
2. Tahap Lifting
Terdapat dua jenis koefisien yang didapatkan dalam proses Lifting Scheme, yaitu proses *predict* atau biasa disebut dual (d) dengan tujuan untuk mendapatkan sinyal detail

yang diperoleh dari proses *highpass filter* dan proses *update* atau biasa disebut primal (p) untuk mendapatkan sinyal aproksimasi yang diperoleh dari proses *lowpass filter*.

3. Tahap Normalisasi

Setelah dilakukan proses *splitting* dan *lifting*, selanjutnya dilakukan proses normalisasi. Tahapan ini dilakukan dengan tujuan untuk menyeragamkan nilai amplitudo dari sinyal.

Lifting Scheme merupakan suatu metode berbentuk *cell array* $N \times 3$ yang digunakan dalam proses LWT agar proses yang dilakukan menjadi lebih efisien jika dibandingkan dengan Wavelet Transform generasi pertama, yaitu DWT (*Discrete Wavelet Transform*). Baris $N-1$ pertama pada *array Lifting Scheme* disebut "*elementary lifting steps*" (ELS). Baris terakhir pada *Lifting Scheme* digunakan untuk proses normalisasi.

2.6 Regresi Linier

Regresi linier merupakan suatu metode yang digunakan untuk menghitung korelasi yang ada antara variabel x dan y . Pada proses penyisipan akan dilakukan pemilihan bagian sub-blok mana yang akan dipilih untuk disisipi citra logo *watermark*. Sehingga tidak semua bagian dari sub-blok yang ada akan disisipi citra logo *watermark*. Pemilihan sub-blok ini menggunakan metode regresi linier untuk melakukan perhitungan nilai slope.

Tiap-tiap sub-blok memiliki slope yang berbeda tergantung pada hasil perhitungan yang diperoleh. Nilai slope kemudian digunakan untuk menentukan sub-blok mana yang akan disisipi informasi citra logo *watermark*. Ketika slope berada di dalam *range*, maka sub-blok yang bersangkutan akan dipilih untuk disisipi bagian dari informasi citra logo *watermark*. Ketika slope tidak berada di dalam rentang tersebut, maka sub-blok yang bersangkutan tidak disisipi informasi citra logo *watermark*. Pemilihan sub-blok untuk disisipi dengan cara seperti ini dilakukan

untuk melakukan penyebaran posisi penyisipan informasi citra logo *watermark*. Jika dalam proses penyisipan tidak dilakukan dengan cara menyisipkan pada posisi menyebar, maka citra *watermark* yang dihasilkan menjadi kurang tahan terhadap distorsi (serangan kerusakan) yang mungkin terjadi.

Untuk menentukan perhitungan nilai slope dibutuhkan variabel x dan y yang diambil dari hasil matriks C_i yang ada pada masing-masing sub-blok. Nilai x diperoleh dari (baris ke- m , kolom ke- n) yang ada pada matriks C_i , sedangkan nilai y diperoleh dari (baris ke- $(m+1)$, kolom ke- n) yang ada pada matriks C_i .

Setelah didapatkan nilai x dan y , maka selanjutnya dilakukan perhitungan untuk menentukan nilai slope dengan menggunakan rumus slope pada Persamaan 2.2. Pada persamaan tersebut, x_i merupakan variabel x dan y_i merupakan variabel y . Variabel \bar{x} merupakan nilai rata-rata x , sedangkan variabel \bar{y} merupakan nilai rata-rata y .

$$\alpha = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \quad (2.2)$$

Contoh proses perhitungan nilai slope adalah sebagai berikut:

$$x = \begin{bmatrix} 0 & 3 & 2 & 4 \\ 4 & 1 & 3 & 0 \\ 2 & 3 & 0 & 4 \end{bmatrix} y = \begin{bmatrix} 4 & 1 & 3 & 0 \\ 2 & 3 & 0 & 4 \\ 4 & 0 & 3 & 0 \end{bmatrix}$$

$$\alpha = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \rightarrow \alpha = \frac{a1}{a2}$$

Diketahui: $\bar{x} = 2.1667$, $\bar{y} = 2$

$$\begin{aligned} a1 = & ((0 - 2.1667) \times (4 - 2)) + ((3 - 2.1667) \times (1 - 2)) + \\ & ((2 - 2.1667) \times (3 - 2)) + ((4 - 2.1667) \times (0 - 2)) + \\ & ((4 - 2.1667) \times (2 - 2)) + ((1 - 2.1667) \times (3 - 2)) + \\ & ((3 - 2.1667) \times (0 - 2)) + ((0 - 2.1667) \times (4 - 2)) + \end{aligned}$$

$$\begin{aligned} & ((2 - 2.1667) \times (4 - 2)) + ((3 - 2.1667) \times (0 - 2)) + \\ & ((0 - 2.1667) \times (3 - 2)) + ((4 - 2.1667) \times (0 - 2)) \end{aligned}$$

$$\begin{aligned} a_2 = & ((0 - 2.1667)^2) + ((3 - 2.1667)^2) + ((2 - 2.1667)^2) + \\ & ((4 - 2.1667)^2) + ((4 - 2.1667)^2) + ((1 - 2.1667)^2) + \\ & ((3 - 2.1667)^2) + ((0 - 2.1667)^2) + ((2 - 2.1667)^2) + \\ & ((3 - 2.1667)^2) + ((0 - 2.1667)^2) + ((4 - 2.1667)^2) \end{aligned}$$

$$\alpha = \frac{a_1}{a_2} = -0.8675$$

∴ Kesimpulan yang didapatkan dari hasil perhitungan slope berdasarkan rumus slope yang ada pada Persamaan 2.2 untuk x dan y tersebut, didapatkan nilai slope -0.8675 . Sehingga slope tersebut tidak termasuk di dalam *range* sub-blok yang disisipi citra logo *watermark*.

2.7 Penilaian Kualitas Citra

Untuk mengetahui citra yang dihasilkan sudah mendekati citra aslinya atau tidak, dapat dilakukan beberapa metode penilaian dengan guna mengetahui kualitas citra tersebut. Penilaian kualitas citra hasil keluaran (*output*) dilakukan dengan menggunakan PSNR dan MSE yang akan dijelaskan lebih lanjut pada bagian ini.

2.7.1 Mean Signal Error

MSE (*Mean Signal Error*) merupakan metode yang digunakan untuk mengukur tingkat error suatu citra. Berkebalikan dengan perhitungan PSNR, semakin tinggi nilai MSE maka kualitas citra tersebut semakin jelek dan semakin tidak mendekati citra aslinya. MSE dapat dirumuskan dalam Persamaan (2.3).

$$MSE = \frac{1}{wh} \sum_{i=1} \sum_{j=1} (C_{ij} - S_{ij})^2 \quad (2.3)$$

Pada Persamaan 2.3, terdapat variabel w dan h yang merupakan lebar dan tinggi citra, dan C_{ij}, S_{ij} merupakan nilai dari piksel pada koordinat (i,j) pada citra *cover* dan citra *ter-watermark*.

2.7.2 Peak Signal to Noise Ratio

Pada sistem perangkat lunak ini, metode PSNR (*Peak Signal to Noise Ratio*) digunakan untuk mengukur perbandingan antara kualitas suatu citra yang telah dilakukan proses penyisipan citra logo *watermark* dengan citra asli yang masih belum diproses. PSNR didefinisikan melalui *signal-to-noise* ratio (SNR) yang digunakan untuk mengukur tingkat kualitas sinyal. Nilai ini dihitung berdasarkan perbandingan antar sinyal dengan nilai derau (*noise*). Dari perbandingan tersebut dapat dilihat bahwa semakin besar nilai PSNR suatu citra, maka kualitas citra tersebut semakin bagus dan semakin mendekati citra aslinya. Nilai PSNR umumnya berkisar antara 30 sampai 50 dalam skala desibel (dB). Perhitungan nilai PSNR dirumuskan dalam Persamaan 2.4.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (2.4)$$

Nilai 255 pada persamaan PSNR tersebut merupakan batas atas nilai piksel pada citra 8 bit (0-255). MSE merupakan kebalikan dari PSNR, sehingga MSE digunakan dalam perhitungan PSNR sebagai pembanding.

BAB III

PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan perancangan perangkat lunak tentang penyisipan dan ekstraksi citra logo *watermark*. Perancangan yang dilakukan akan terbagi menjadi dua kategori proses, yaitu proses untuk penyisipan citra logo *watermark* dan ekstraksi citra logo *watermark*. Untuk masing-masing proses utama akan dibagi menjadi beberapa proses kecil yang terlibat di dalamnya.

3.1 Desain Metode Secara Umum

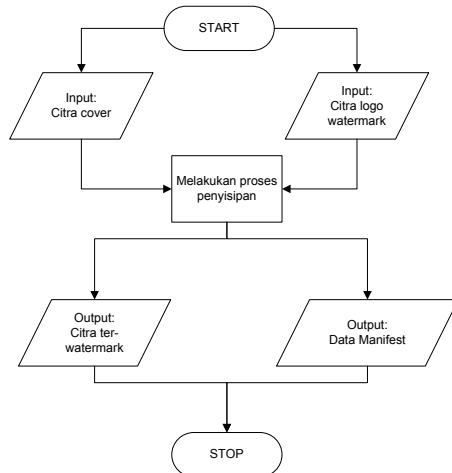
Sistem perangkat lunak penyisipan citra *grayscale* ini terdapat dua tahapan utama, yaitu proses penyisipan (*embedding*) dan proses ekstraksi (*extracting*).

Dalam proses penyisipan membutuhkan dua data masukan, yaitu citra *cover* dan citra logo *watermark* yang kemudian diproses dalam penyisipan sehingga menghasilkan keluaran citra ter-*watermark* dan hasil pengujian PSNR dan MSE. Selanjutnya data keluaran (*output*) berupa citra ter-*watermark* akan digunakan dalam proses ekstraksi sebagai data masukan (*input*). Sedangkan hasil perolehan PSNR dan MSE pada proses penyisipan akan digunakan sebagai pembandingan pada keluaran hasil pengujian di proses ekstraksi. Diagram alir dari proses secara umum ini dapat dilihat pada Gambar 3.1.

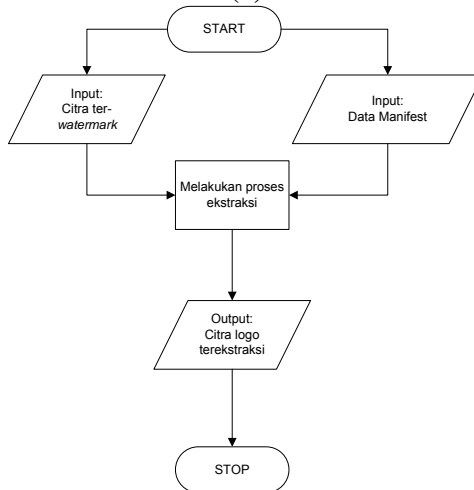
3.2 Perancangan Data

Perancangan data merupakan bagian penting dalam pengoperasian perangkat lunak. Apabila data masukan (*input*) yang digunakan ternyata salah atau kurang benar, maka hasil keluaran juga akan salah atau bahkan tidak akan dapat diproses. Sehingga dibutuhkan data masukan yang benar agar dapat menghasilkan data keluaran (*output*) yang benar pula.

Data yang diperlukan dalam pengoperasian perangkat lunak adalah data masukan (*input*), data proses dan data keluaran (*output*).



(a)



(b)

Gambar 3.1 Diagram Alir Proses Secara Umum (a) Proses Penyisipan (b) Proses Ekstraksi

3.2.1 Data Masukan

Data masukan merupakan data yang dimasukkan oleh pengguna untuk dapat memulai menjalankan sistem perangkat lunak. Data masukan ada dua, yaitu citra *cover grayscale* dan citra logo *watermark* dalam bentuk biner.

Untuk citra *cover*, data masukan berupa citra Lena yang berukuran 512×512 yang dapat dilihat pada Gambar 3.2. Sedangkan citra logo *watermark* berupa citra logo instansi berukuran 32×32 yang dapat dilihat pada Gambar 2.1.



Gambar 3.2 Citra Lena Grayscale Berukuran 512×512

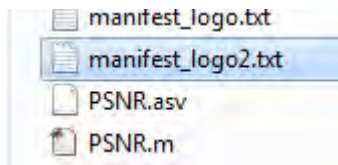
3.2.2 Data Keluaran

Pada sub bab ini dijelaskan mengenai hasil keluaran sistem perangkat lunak. Data keluaran terbagi menjadi dua, yaitu data keluaran pada proses penyisipan dan data keluaran pada proses ekstraksi.

Data keluaran untuk proses penyisipan berupa citra ter-*watermark* dan data manifest. Data manifest bertipe *text* (*manifest_logo2.txt*) yang mengandung informasi mengenai informasi slope citra logo ter-*watermark*. Contoh hasil keluaran dari proses penyisipan dapat dilihat pada Gambar 3.3 dan Gambar 3.4



Gambar 3.3 Hasil Keluaran Berupa Citra Ter-*Watermark*



**Gambar 3.4 Hasil Keluaran Berupa Data Manifest
(manifest_logo2.txt)**

Data keluaran pada proses ekstraksi berupa citra logo *watermark* hasil ekstraksi. Citra logo *watermark* hasil ekstraksi harus sama dengan citra logo *watermark* yang disisipkan pada proses penyisipan. Contoh citra logo *watermark* hasil ekstraksi dapat dilihat pada Gambar 3.5.



Gambar 3.5 Hasil Keluaran Citra Logo Hasil Ekstraksi

3.3 Algoritma dan Diagram Alir

Pada bagian ini dijelaskan mengenai algoritma sistem melalui diagram alir (*flowchart*). Dengan menggunakan diagram alir, dapat diketahui secara jelas alur sistem dari awal hingga akhir proses. Langkah-langkah algoritma sistem dijelaskan pada beberapa sub bab berikut ini.

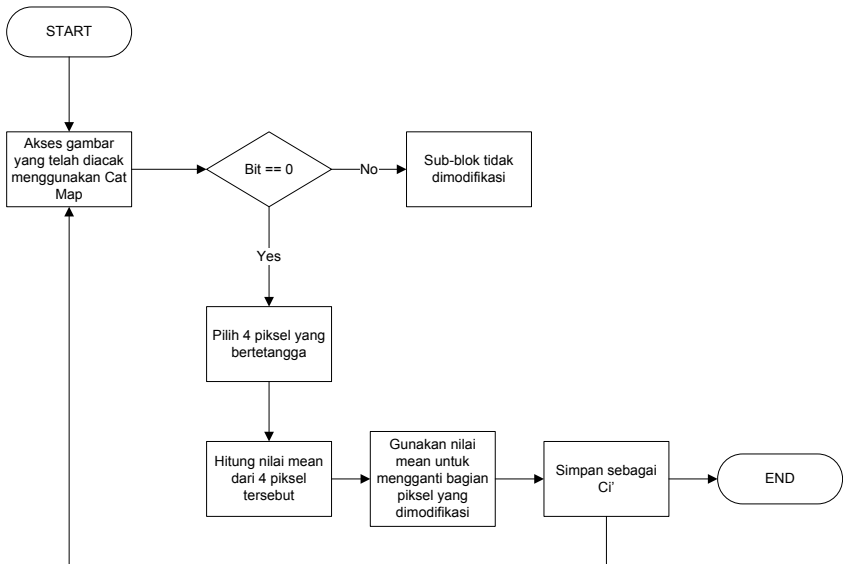
3.3.1 Proses Penyisipan

Proses penyisipan citra *watermark* terdapat 2 tahapan yang harus dilakukan, yaitu pra-proses dan proses inti. Langkah-langkah yang harus dilakukan antara lain:

1. Pra-proses merupakan suatu proses yang harus dilakukan sebelum proses inti. Tahapan pertama yang dilakukan yaitu citra logo *watermark* diubah menjadi citra biner (binerisasi). Pada proses inti, citra logo *watermark* yang telah diubah ke biner disebut sebagai logo biner *watermark*. Citra logo *watermark* yang digunakan berukuran 32×32 . Pada tahapan kedua, citra *cover* (citra yang akan disisipi watermark) diubah menjadi gambar *grayscale*. Pada proses inti, citra *cover* yang telah diubah menjadi *grayscale* disebut sebagai citra *cover grayscale*. Citra *cover* yang digunakan berukuran 512×512 .

2. Proses inti merupakan suatu proses penyisipan yang dilakukan pada citra *cover grayscale*. Tahapan pertama yaitu dilakukan pengacakan terhadap citra logo biner *watermark* sehingga citra logo *watermark* menjadi abstrak dan tidak beraturan. Tujuannya yaitu untuk mengamankan citra agar tidak dapat dengan mudah diketahui oleh pihak yang tidak diinginkan. Setelah itu citra *cover grayscale* dibagi menjadi sub-blok 8×8 yang kemudian akan dipilih sub-blok bagian mana yang akan disisipi citra logo biner *watermark*.

Diagram alir proses penyisipan secara keseluruhan dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Alir Proses Penyisipan Secara Keseluruhan

Pada diagram alir tersebut digunakan citra terenkripsi yang telah diacak menggunakan Arnold Cat Map. Kemudian dicek, ketika bitnya bernilai 1 maka tidak dimodifikasi, tetapi jika bernilai 0 maka dipilih 4 piksel bertetangga pada matriks C_i yang ada pada citra *cover*. Hitung nilai *mean* dari 4 piksel tersebut, lalu gunakan untuk meng-*cover* 4 piksel tersebut dan simpan sebagai variabel C_i' .

3.3.1.1 Tahap Enkripsi Citra Logo *Watermark*

Pada tahap ini dilakukan enkripsi terhadap masukan berupa citra logo *watermark*. Proses enkripsi dilakukan dengan cara melakukan pergeseran ke arah sumbu y , kemudian melakukan pergeseran ke arah sumbu x . Setelah itu semua hasilnya akan dimodulo dengan N agar hasil pengacakan akan tetap berada di dalam area citra. *Pseudocode* untuk proses ini ditunjukkan pada Gambar 3.7.

Masukan	Citra logo <i>watermark</i> berukuran 32×32
Keluaran	Citra terenkripsi
<pre> 1. p = size(Y,1); 2. X = zeros(size(Y)); 3. for i = 1 to p do 4. for j = 1 to p do 5. newi = mod(((i-1) + (j-1)),p) + 1; 6. newj = mod(((i-1) + 2*(j-1)),p) + 1; 7. X(newi,newj,:) = Y(i,j,:); 8. end for 9. end for </pre>	

Gambar 3.7 Pseudocode Enkripsi Arnold Cat Map

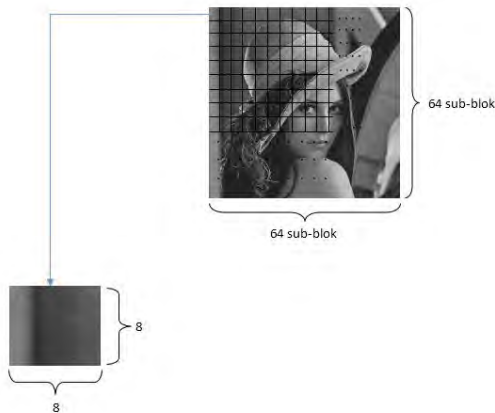
Daftar variabel yang digunakan dalam tahapan enkripsi citra logo *watermark* dapat dilihat pada Tabel 3.1.

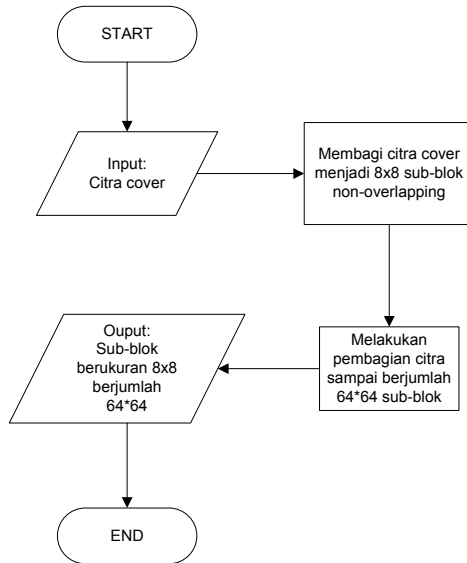
Tabel 3.1 Daftar Variabel Pada Proses Enkripsi Arnold Cat Map

No.	Nama Variabel	Type Data	Keterangan
1.	X	<i>double</i>	Berisi matriks berukuran 32×32 .
2.	Y	<i>double</i>	Berisi ukuran baris dari citra logo <i>watermark</i> . $Y = 32$
3.	<i>newi, newj</i>	<i>double</i>	Inisialisasi variabel untuk mengacak citra logo <i>watermark</i>

3.3.1.2 Tahap Pembagian Sub-Blok Citra Cover

Pada tahapan ini dilakukan pembagian sub-blok citra *cover* yang berukuran 512×512 menjadi berukuran 8×8 . Pembagian sub-blok dilakukan hingga mencapai total 64×64 sub-blok. Contoh pembagian sub-blok dapat dilihat pada Gambar 3.8. *Pseudocode* pada tahapan ini ditunjukkan pada Gambar 3.10, sedangkan diagram alir tahapan ini ditunjukkan pada Gambar 3.9.

**Gambar 3.8 Pembagian Sub-Blok 8×8**



Gambar 3.9 Diagram Alir Pembagian Sub-Blok Citra Cover

Masukan	Citra <i>cover</i> berukuran 512x512
Keluaran	Sub-blok berukuran 8x8 berjumlah 64x64
<ol style="list-style-type: none"> 1. $subM = \text{floor}(M/64)$; 2. $subN = \text{floor}(N/64)$; 3. Penentuan ukuran total sub-blok keseluruhan berdasarkan baris dan kolom 4. $batas1 = (i-1)*subM+1:i*subM$ 5. $batas2 = (j-1)*subN+1:j*subN$ 6. $cell_subImage\{i,j\} = im2(batas1,batas2)$ 7. Pengubahan tipe data <i>subImage</i> menjadi double 	

Gambar 3.10 Pseudocode Pembagian Sub-Blok Citra Cover

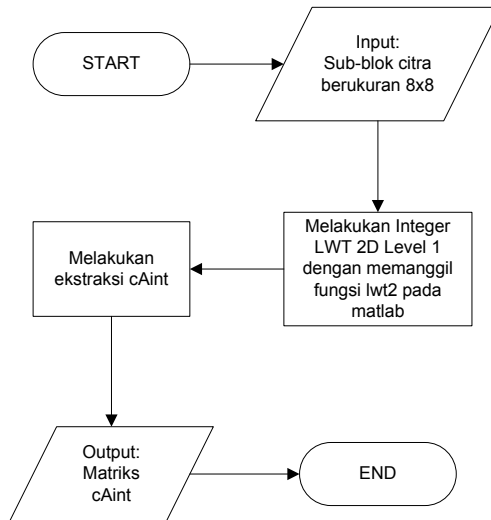
Daftar variabel yang digunakan dalam tahapan pembagian sub-blok citra *cover* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Daftar Variabel Pada Proses Pembagian Sub-Blok Citra Cover

No.	Nama Variabel	Tipe Data	Keterangan
1.	subM, subN	<i>double</i>	Ukuran baris dan kolom tiap sub-blok. subM = 8 (baris), subN = 8 (kolom)
2.	batas1, batas2	<i>double</i>	Berisi nilai batas awal dan akhir tiap sub-blok citra cover. batas1 = 1 sampai 8 (baris), batas2 = 1 sampai 8 (kolom)
3.	cell_subImage	<i>cell</i>	Cell berukuran 64×64 yang menyimpan nilai variabel batas1 dan batas2.

3.3.1.3 Tahap Penerapan Wavelet Pada Citra Cover

Pada tahapan ini dilakukan penerapan Integer LWT level 1 yang digunakan untuk mengolah sinyal menjadi beberapa macam sinyal frekuensi, yaitu berupa sinyal frekuensi rendah (*low frequency*) dan sinyal frekuensi tinggi (*high frequency*). Diagram alir tahapan ini ditunjukkan pada Gambar 3.11.



Gambar 3.11 Diagram Alir Penerapan Integer LWT

Selanjutnya dilakukan ekstraksi sinyal frekuensi rendah untuk diproses pada langkah selanjutnya. Sinyal yang diekstraksi hanya sinyal frekuensi rendah karena sinyal frekuensi rendah lebih mendekati citra asli dibandingkan dengan sinyal frekuensi tinggi, sehingga proses pengolahan sistem perangkat lunak ini dilakukan pada sinyal frekuensi rendah. *Pseudocode* tahapan ini menggunakan fungsi di Matlab yang ditunjukkan pada Gambar 3.12.

Masukan	Sub-blok citra berukuran 8×8
Keluaran	cAint (<i>low frequency</i>)
1. <code>lshaarInt = liftwave('haar','int2int');</code> 2. <code>[cAint,cHint,cVint,cDint] = lwt2(cell_A{i,j},lshaarInt);</code> 3. Penyimpanan cAint ke dalam variabel Bi	

Gambar 3.12 Pseudocode Penerapan Integer LWT

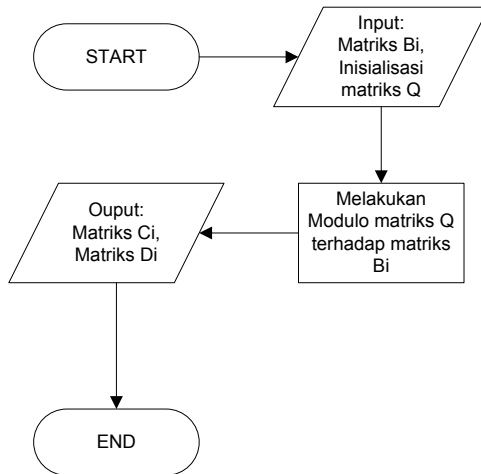
Daftar variabel yang digunakan dalam tahapan penerapan Wavelet pada citra *cover* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Daftar Variabel Pada Proses Penerapan Wavelet Pada Citra Cover

No.	Nama Variabel	Tipe Data	Keterangan
1.	<code>lshaarInt</code>	<i>cell</i>	Cell yang berisi: ‘d’ [-1] [0] ‘p’ [0.5000] [0] [1.4142] [0.7071] ‘I’
2.	<code>cAint</code>	<i>double</i>	Berisi nilai low frekuensi <i>approximation coefficients</i> dari proses LWT pada citra <i>cover</i> .
3.	<code>cHint</code>	<i>double</i>	Berisi nilai <i>horizontal detail coefficients</i> dari proses LWT pada citra <i>cover</i> .
4.	<code>cVint</code>	<i>double</i>	Berisi nilai <i>vertical detail coefficients</i> dari proses LWT pada citra <i>cover</i> .
5.	<code>cDint</code>	<i>double</i>	Berisi nilai <i>diagonal detail coefficients</i> dari proses LWT pada citra <i>cover</i> .

3.3.1.4 Tahap MOD Q

Pada tahap ini dilakukan Modulo matriks Q terhadap matriks Bi. Terdapat dua keluaran pada proses Modulo ini, yaitu hasil bagi dan hasil sisa. Hasil sisa disimpan sebagai variabel Ci, sedangkan hasil bagi disimpan sebagai variabel Di. Diagram alir tahapan ini ditunjukkan pada Gambar 3.13, sedangkan *pseudocode* tahap Mod Q ditunjukkan pada Gambar 3.14.



Gambar 3.13 Diagram Alir Proses Mod Q

Masukan	Matriks Bi
Keluaran	Matriks Ci, Matriks Di
1. Inisialisasi matriks Q 2. $C_i = \text{mod}(B_i, Q)$; 3. $\text{cell_}C_i\{i, j\} = C_i$; 4. $D_i = B_i - C_i$; 5. $\text{cell_}D_i\{i, j\} = D_i$;	

Gambar 3.14 Pseudocode Proses Mod Q

Daftar variabel yang digunakan dalam tahapan Mod Q dapat dilihat pada Tabel 3.4 dan Tabel 3.5.

Tabel 3.4 Daftar Variabel Pada Proses Mod Q (bagian 1)

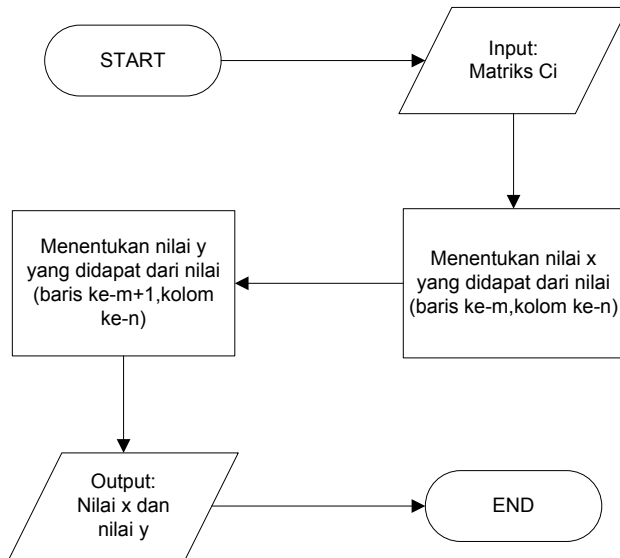
No.	Nama Variabel	Tipe Data	Keterangan
1.	C_i	<i>double</i>	Berisi nilai hasil sisa dari MOD Q terhadap B_i .

Tabel 3.5 Daftar Variabel Pada Proses Mod Q (bagian 2)

No.	Nama Variabel	Tipe Data	Keterangan
2.	cell_Ci	<i>cell</i>	Cell berukuran 64×64 yang menampung keseluruhan nilai dari matriks C_i .
3.	Di	<i>double</i>	Berisi nilai hasil bagi dari MOD Q terhadap B_i .
4.	Cell_Di	<i>cell</i>	Cell berukuran 64×64 yang menampung keseluruhan nilai dari matriks D_i .
5.	Q	<i>double</i>	Matriks yang digunakan untuk melakukan Modulus, nilainya konstan. $Q = [5\ 5\ 5\ 5; 5\ 5\ 5\ 5; 5\ 5\ 5\ 5; 5; 5\ 5\ 5\ 5]$
6.	Di	<i>double</i>	Berisi nilai hasil bagi dari MOD Q terhadap B_i .

3.3.1.5 Tahap Penentuan Nilai x dan y

Pada tahapan ini dilakukan penentuan nilai x dan y untuk proses perhitungan nilai slope, yang didapatkan dari matriks C_i tiap-tiap sub-blok, jadi nilai x dan y pada tiap sub-blok berbeda-beda. Perolehan nilai x didapatkan dari (baris ke- m , kolom ke- n) pada matriks C_i , sedangkan perolehan nilai y didapatkan dari (baris ke- $(m+1)$, kolom ke- n) pada matriks C_i . Diagram alir proses penentuan nilai x dan y ditunjukkan pada Gambar 3.15, sedangkan alur *pseudocode* tahapan ini ditunjukkan pada Gambar 3.16.



Gambar 3.15 Diagram Alir Proses Penentuan Nilai x dan y

Nilai x dan y digunakan untuk melakukan perhitungan nilai slope.

Masukan	Matriks C_i
Keluaran	Nilai x dan nilai y
1. Pengambilan ukuran C_i 2. for $a=1$ to $m-1$ do 3. for $b=1$ to n do 4. $x(a,b)=C_i(a,b)$; 5. $y(a,b)=C_i(a+1,b)$; 6. end for 7. end for	

Gambar 3.16 Pseudocode Proses Penentuan Nilai x dan y

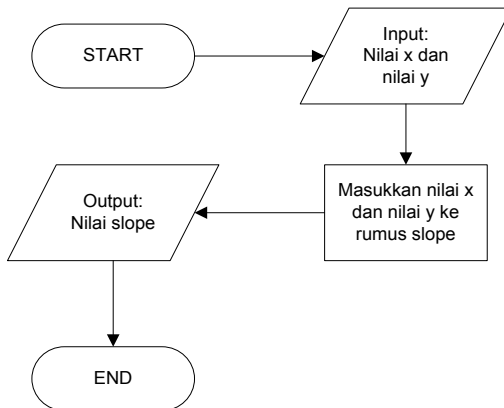
Daftar variabel yang digunakan dalam proses penentuan nilai x dan y dapat dilihat pada Tabel 3.6.

Tabel 3.6 Daftar Variabel Pada Proses Penentuan Nilai x dan y

No.	Nama Variabel	Tipe Data	Keterangan
1.	x, y	<i>double</i>	Nilai variabel x dan y yang digunakan dalam proses perhitungan nilai slope.
2.	C_i	<i>double</i>	Berisi nilai hasil sisa dari MOD Q terhadap B_i .

3.3.1.6 Tahap Komputasi Nilai Slope

Pada tahapan ini dilakukan perhitungan nilai slope dengan menggunakan rumus pada Persamaan 2.1. Perhitungan slope ini menggunakan data masukan nilai x dan nilai y yang didapatkan pada tahapan sebelumnya. Diagram alir perhitungan slope ditunjukkan pada Gambar 3.17. Proses perhitungan nilai slope ini juga dapat dilihat dengan menggunakan alur *pseudocode* yang ditunjukkan pada Gambar 3.18.

**Gambar 3.17 Diagram Alir Komputasi Slope**

Masukan	Nilai x dan nilai y
Keluaran	Nilai slope
1. Perhitungan nilai rata-rata x 2. Perhitungan nilai rata-rata y 3. $a1(i,j) = \text{sum}(\text{sum}((x-\text{ratarata_x}) \cdot (y-\text{ratarata_y})));$ 4. $a2(i,j) = \text{sum}(\text{sum}((x-\text{ratarata_x}) \cdot (x-\text{ratarata_x})));$ 5. $\text{slope_a}(i,j) = a1(i,j)/a2(i,j);$	

Gambar 3.18 Pseudocode Komputasi Nilai Slope

Baris ke-1 sampai dengan baris ke-5 pada *pseudocode* yang ada di Gambar 3.18 menerapkan rumus perhitungan slope sesuai dengan Persamaan 2.1 yang telah dijelaskan sebelumnya pada Bab II. Daftar variabel yang digunakan dalam proses penentuan nilai x dan y dapat dilihat pada Tabel 3.7.

Tabel 3.7 Daftar Variabel Pada Proses Komputasi Nilai Slope

No.	Nama Variabel	Tipe Data	Keterangan
1.	slope_a	double	Menyimpan nilai hasil perhitungan slope citra <i>cover</i> .
2.	a1, a2	double	Nilai pembilang dan penyebut yang digunakan pada perhitungan slope.
3.	ratarata_x, ratarata_y	double	Menyimpan nilai perhitungan rata-rata variabel x dan rata-rata variabel y.

3.3.1.7 Tahap Pemilihan Sub-Blok Disisipi

Pemilihan sub-blok dilakukan secara menyebar, sehingga citra logo *watermark* yang disisipkan tidak dapat dengan mudah rusak ketika terjadi distorsi pada citra ter-*watermark*. Tahapan selanjutnya dilakukan pemilihan 4 *adjacent* piksel yang ada pada variabel Ci apabila kondisi piksel pada citra logo *watermark* bernilai 0. Pemilihan 4 *adjacent* piksel ini diambil di bagian tengah pada tiap-tiap Ci, kemudian dihitung nilai rata-ratanya. Setelah itu hasil rata-rata (*mean*) tersebut digunakan untuk meng-*cover* nilai 4 *adjacent* piksel tersebut, sehingga 4 *adjacent* piksel tersebut memiliki nilai yang sama. Tiap-tiap Ci memiliki 4 *adjacent* piksel masing-masing dan nilai *mean* yang berbeda-beda. *Pseudocode* tahapan ini ditunjukkan pada Gambar 3.19 dan Gambar 3.20.

Masukan	Sub-blok berukuran 8x8 berjumlah 64x64
Keluaran	Sub-blok yang dipilih untuk disisipi
<pre> 1. for i=1 to tesM do 2. for j=1 to tesN do 3. if mod(i-1,2)=true && mod(j-1,2)=true then 4. if im(counter) equals 0 then 5. matriksTengah(i,j) = mean(mean(cell_Ci{i,j}(2:3,2:3))); 6. ab=ones(2); 7. cell_Ciaksen{i,j} = cell_Ci{i,j}; 8. cell_Ciaksen{i,j}(2:3,2:3) = ab*matriksTengah(i,j); 9. matriksCek(i,j)=1; 10. else 11. cell_Ciaksen{i,j}=cell_Ci{i,j}; 12. end if </pre>	

Gambar 3.19 Pseudocode Pemilihan Sub-Blok Disisipi (bagian 1)

```

13.         counter=counter+1;
14.         else
15.             cell_Ciaksen{i,j}=cell_Ci{i,j};
16.         end if
17.     end for
18. end for

```

Gambar 3.20 Pseudocode Pemilihan Sub-Blok Disisipi (bagian 2)

Daftar variabel yang digunakan dalam proses pemilihan sub-blok disisipi dapat dilihat pada Tabel 3.8 dan Tabel 3.9.

Tabel 3.8 Daftar Variabel Pada Proses Pemilihan Sub-Blok Disisipi (bagian 1)

No.	Nama Variabel	Tipe Data	Keterangan
1.	tesM, tesN	<i>double</i>	Total baris dan kolom sub-blok citra <i>cover</i> . tesM = 64 (baris) tesN = 64 (kolom)
2.	im	<i>double</i>	Citra <i>cover</i> berukuran 512×512.
3.	matriksTengah	<i>double</i>	Berisi nilai rata-rata 4 adjacent piksel yang berada di tengah pada tiap matriks <i>Ci</i> .
4.	cell_Ci	<i>cell</i>	Cell berukuran 64 × 64 yang menampung keseluruhan nilai dari matriks <i>Ci</i> .
5.	cell_Ciaksen	<i>cell</i>	Cell berukuran 64 × 64 berisi nilai <i>Ci</i> yang sudah dimodifikasi ketika kondisinya 0, jika 1 maka nilai <i>Ci</i> tetap.

Tabel 3.9 Daftar Variabel Pada Proses Pemilihan Sub-Blok Disisipi (bagian 2)

No.	Nama Variabel	Tipe Data	Keterangan
6.	matriksCek	<i>double</i>	Untuk melakukan pengecekan sub-blok mana saja yang disisipi citra logo <i>watermark</i> .
7.	counter	<i>double</i>	Sebagai counter untuk citra <i>cover</i> . counter = 1
8.	cell_Ci	<i>cell</i>	Cell berukuran 64 × 64 yang menampung keseluruhan nilai dari matriks Ci.

3.3.1.8 Tahap Manifest Citra Logo *Watermark*

Pada sub bab ini dilakukan pembuatan manifest dengan tujuan untuk mencatat bagian sub-blok mana saja yang disisipi citra logo *watermark*. *Pseudocode* tahapan ini ditunjukkan pada Gambar 3.21. Daftar variabel yang digunakan dalam proses pembuatan Manifest citra logo *watermark* dapat dilihat pada Tabel 3.10.

Masukan	matriksCek
Keluaran	manifest_logo.txt
<pre> 1. fid = fopen('manifest_logo.txt', 'w'); 2. for ix=1 to tesM*tesN do 3. if matriksCek(ix)==1 then 4. fprintf(fid,'%d\n',ix); 5. end if 6. end for 7. fclose(fid); </pre>	

Gambar 3.21 Pseudocode Proses Pembuatan Manifest

Tabel 3.10 Daftar Variabel Pada Proses Pembuatan Manifest Citra Logo Watermark

No.	Nama Variabel	Tipe Data	Keterangan
1.	fid	<i>double</i>	Variabel penyimpan data <code>manifest_logo.txt</code>
2.	tesM, tesN	<i>double</i>	Total baris dan kolom sub-blok citra <i>cover</i> . tesM = 64 (baris) tesN = 64 (kolom)
3.	matriksCek	<i>double</i>	Matriks yang menyimpan sub-blok mana saja yang disisipi citra logo <i>watermark</i>

3.3.1.9 Tahap Penerapan Inverse Wavelet

Pada tahapan ini dilakukan proses Inverse Integer LWT dengan tujuan untuk mengembalikan kembali sinyal yang telah diproses menjadi hasil keluaran citra ter-*watermark*. *Pseudocode* pada tahapan ini ditunjukkan pada Gambar 3.22.

Masukan	Biaksen, cHint, cvint, cDint
Keluaran	Hasil pengembalian dari matriks menjadi citra ter- <i>watermark</i>
<pre> 1. for ii=1 to tesM do 2. for jj=1 to tesN do 3. cell_Biaksen{ii,jj} = cell_Ciaksen{ii,jj}+cell_Di{ii,jj}; 4. cell_hasil_ilwt{ii,jj} = ilwt2(cell_Biaksen{ii,jj},cell_cHint{ i,jj},cell_cvint{ii,jj},cell_cDint{ii, jj},lshaarInt); </pre>	

Gambar 3.22 Pseudocode Penerapan Inverse LWT (bagian 1)

```

5. end for
6. end for
7. mat_hasil_ilwt=cell2mat(cell_hasil_ilwt);
8. image_watermarked=uint8(mat_hasil_ilwt);

```

Gambar 3.23 Pseudocode Penerapan Inverse LWT (bagian 2)

Daftar variabel yang digunakan dalam proses penerapan *Inverse* LWT dapat dilihat pada Tabel 3.11.

Tabel 3.11 Daftar Variabel Pada Proses Penerapan *Inverse* LWT

No.	Nama Variabel	Tipe Data	Keterangan
1.	tesM, tesN	<i>double</i>	Total baris dan kolom sub-blok citra <i>cover</i> . tesM = 64 (baris) tesN = 64 (kolom)
2.	cell_Biaksen	<i>cell</i>	Cell berukuran 64 × 64 berisi nilai penjumlahan Ciaksen dengan Di.
3.	cell_Ciaksen	<i>cell</i>	Cell berukuran berisi nilai Ci yang sudah dimodifikasi ketika kondisinya 0, jika 1 maka nilai Ci tetap.
4.	cell_hasil_ilwt	<i>cell</i>	Cell berukuran 64 × 64 berisi nilai hasil <i>inverse</i> LWT.
5.	cell_Biaksen	<i>cell</i>	Cell berukuran 64 × 64 berisi nilai penjumlahan Ciaksen dengan Di.

3.3.1.10 Tahap Komputasi Nilai PSNR

Pada tahapan ini akan dilakukan perhitungan terhadap nilai PSNR untuk mengetahui seberapa besar tingkat kualitas citra yang dihasilkan dalam proses penyisipan dan ekstraksi citra logo *watermark*. Apabila nilai PSNR yang didapatkan semakin mendekati nilai 100, maka kualitas citra tersebut semakin bagus dan semakin mendekati citra aslinya. Tetapi jika nilai PSNR yang didapatkan semakin mendekati nilai 0, maka kualitas citra yang dihasilkan semakin buruk dan semakin jauh kualitasnya jika dibandingkan dengan citra aslinya. Perhitungan PSNR berbanding terbalik dengan MSE, semakin kecil nilai MSE yang dihasilkan maka tingkat *error* yang dihasilkan pun semakin kecil sehingga citra yang dihasilkan memiliki kualitas citra yang mendekati citra aslinya. *Pseudocode* yang menjelaskan mengenai tahapan ini dapat dilihat pada Gambar 3.24.

Masukan	Citra ter- <i>watermark</i> , citra cover original
Keluaran	Nilai PSNR
<pre> 1. Pengubahan tipe data im2 dan image_watermarked menjadi double 2. if (im2==image_watermarked) then 3. psnr=100; 4. else 5. Pengambilan ukuran im2 6. Inisialisasi d=0 7. for i=1 to r do 8. for j=1 to c do 9. d=d+(im2(i,j)-image_watermarked(i,j))^2; 10. end for 11. end for 12. mse=d/(r*c) % hasil mse 13. maximum = max(im2(:)); 14. psnr=10*log10(maximum^2/mse); 15. end if </pre>	

Gambar 3.24 *Pseudocode* Proses Komputasi PSNR

Daftar variabel yang digunakan dalam proses komputasi nilai PSNR dapat dilihat pada Tabel 3.12.

Tabel 3.12 Daftar Variabel Pada Proses Komputasi Nilai PSNR

No.	Nama Variabel	Tipe Data	Keterangan
1.	<code>im2</code>	<i>uint8</i>	Citra logo <i>watermark</i>
2.	<code>image_watermarked</code>	<i>uint8</i>	Citra ter- <i>watermark</i>
3.	<code>mse</code>	<i>double</i>	Hasil perhitungan MSE
4.	<code>r, c</code>	<i>double</i>	Ukuran baris dan kolom <code>im2</code> . <code>r = 512</code> <code>c = 512</code>

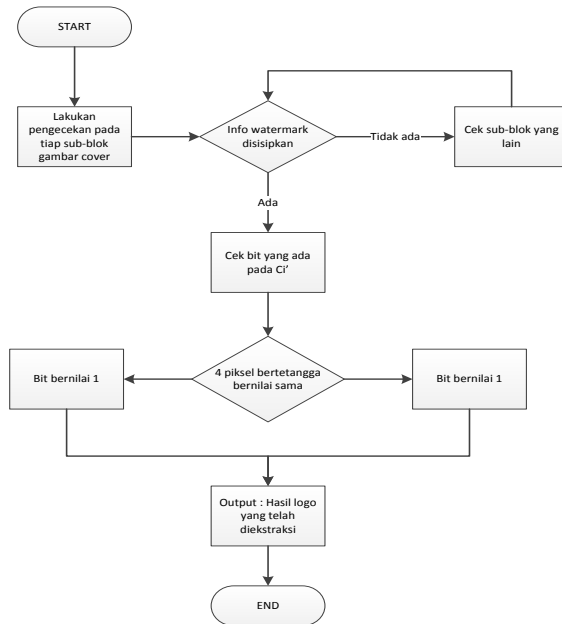
3.3.2 Proses Ekstraksi

Pada proses ekstraksi citra logo *watermark* dilakukan beberapa tahapan yang hampir sama dengan tahapan awal yang dilakukan pada proses penyisipan, antara lain :

1. Dilakukan pendeteksian pada sub-blok citra untuk mengetahui apakah terdapat informasi citra logo *watermark* yang disisipkan atau tidak.
2. Jika terdapat info citra logo *watermark* yang disisipkan, maka dilakukan proses untuk mengekstraksi bentuk citra logo biner *watermark* yang sebelumnya telah disisipkan.

Setelah citra logo biner *watermark* telah diketahui, maka dilakukan *inverse* pengacakan dengan menggunakan metode Arnold Cat Map. Hal ini dilakukan dengan tujuan untuk mengembalikan citra logo biner *watermark* yang sebelumnya berbentuk tidak beraturan dan abstrak, menjadi seperti citra logo *watermark* semula.

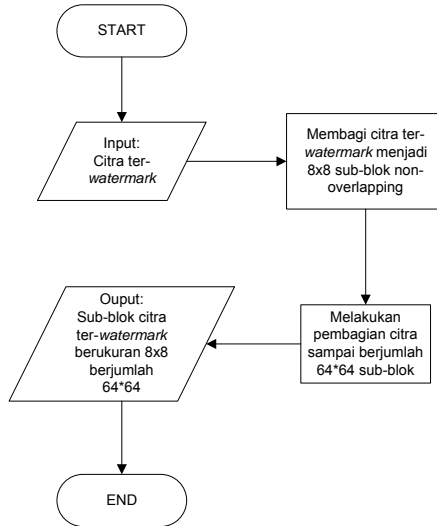
Diagram alir proses ekstraksi secara keseluruhan dapat dilihat pada Gambar 3.25.



Gambar 3.25 Diagram Alir Proses Ekstraksi Secara Keseluruhan

3.3.2.1 Tahap Pembagian Sub-Blok Citra Ter-Watermark

Pada tahapan ini dilakukan pembagian sub-blok citra ter-*watermark* yang berukuran 512×512 menjadi berukuran 8×8 . Sehingga batasan berdasarkan baris masing-masing sub-blok memiliki 8 baris, sedangkan batasan berdasarkan kolom masing-masing sub-blok memiliki 8 kolom.



Gambar 3.26 Diagram Alir Pembagian Sub-Blok Citra Ter-Watermark

Pembagian sub-blok dilakukan hingga mencapai total 64×64 sub-blok. Alur *pseudocode* proses pembagian sub-blok ini dapat dilihat pada Gambar 3.27 dan Gambar 3.28, sedangkan diagram alir tahapan ini ditunjukkan pada Gambar 3.26. Contoh pembagian sub-blok dapat dilihat pada Gambar 3.8.

Masukan	Citra ter-watermark
Keluaran	Sub-blok citra ter-watermark berukuran 8x8 berjumlah 64x64
<ol style="list-style-type: none"> 1. <code>subM_w = floor(m/64); %baris</code> 2. <code>subN_w = floor(n/64); %kolom</code> 3. Penentuan ukuran total sub-blok keseluruhan berdasarkan baris dan kolom 4. <code>batas w1 = (i-1)*subM w+1:i*subM w</code> 	

Gambar 3.27 Pseudocode Proses Pembagian Sub-Blok Citra Ter-Watermark (bagian 1)

```

5. batas_w2 = (j-1)*subN_w+1:j*subN_w
6. cell_subImage_w{i,j} =
   im_w(batas_w1,batas_w2)
7. Pengubahan tipe data subImage_w menjadi
   double

```

Gambar 3.28 Pseudocode Proses Pembagian Sub-Blok Citra Ter-Watermark (bagian 2)

Daftar variabel yang digunakan dalam proses pembagian sub-blok citra ter-*watermark* dapat dilihat pada Tabel 3.13.

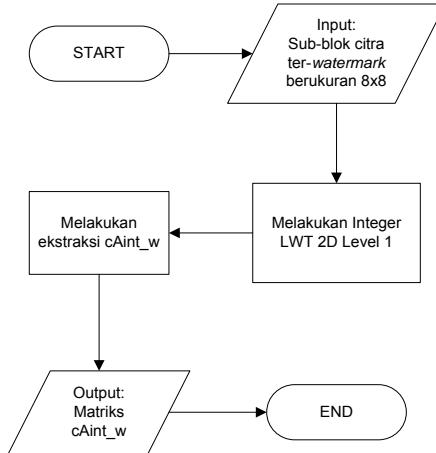
Tabel 3.13 Daftar Variabel Pada Proses Pembagian Sub-Blok Citra Ter-Watermark

No.	Nama Variabel	Tipe Data	Keterangan
1.	subM_w, subN_w	<i>double</i>	Ukuran baris dan kolom tiap sub-blok citra <i>cover</i> .
2.	batas_w1, batas_w2	<i>double</i>	Berisi nilai batas awal dan batas akhir sub-blok citra ter- <i>watermark</i> .

3.3.2.2 Tahap Penerapan Wavelet Pada Citra Ter-Watermark

Pada tahapan ini dilakukan penerapan Integer LWT level 1 yang digunakan untuk mengolah sinyal dari citra ter-*watermark* menjadi beberapa macam sinyal frekuensi. Hasil keluaran yang didapatkan berupa sinyal frekuensi rendah (*low frequency*) dan sinyal frekuensi tinggi (*high frequency*). Selanjutnya dilakukan ekstraksi sinyal frekuensi rendah untuk diproses pada langkah selanjutnya. Sinyal yang diekstraksi hanya sinyal frekuensi rendah karena sinyal frekuensi rendah lebih mendekati citra asli dibandingkan dengan sinyal frekuensi tinggi, sehingga proses pengolahan sistem perangkat lunak ini dilakukan pada sinyal

frekuensi rendah. Diagram alir proses ini ditunjukkan pada Gambar 3.29, sedangkan alur *pseudocode* dapat dilihat pada Gambar 3.30. Daftar variabel yang digunakan dalam proses penerapan Integer LWT pada citra ter-*watermark* dapat dilihat pada Tabel 3.14.



Gambar 3.29 Diagram Alir Penerapan Integer LWT Pada Citra Ter-Watermark

Masukan	Sub-blok citra ter- <i>watermark</i> berukuran 8x8
Keluaran	cAint_w (<i>low frequency</i>)
1. <code>lshaarInt = liftwave('haar','int2int');</code> 2. <code>[cAint_w,cHint_w,cVint_w,cDint_w] = lwt2(cell_W{i,j},lshaarInt);</code> 3. Penyimpanan cAint ke dalam variabel Bi_w	

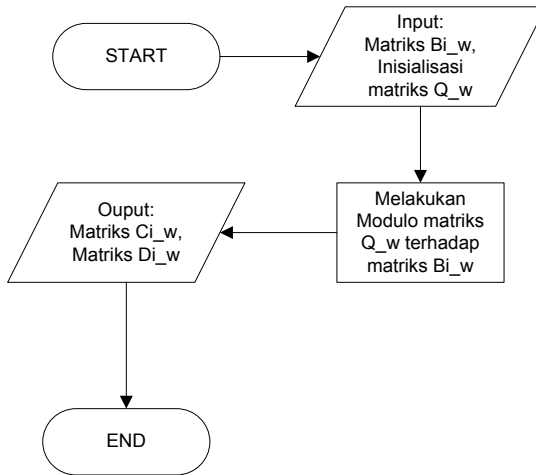
Gambar 3.30 Pseudocode Penerapan Wavelet Pada Citra Ter-Watermark

Tabel 3.14 Daftar Variabel Pada Proses Penerapan Wavelet Terhadap Citra Ter-Watermark

No.	Nama Variabel	Tipe Data	Keterangan
1.	lshaarInt	<i>cell</i>	Cell yang berisi: 'd' [-1] [0] 'p' [0.5000] [0] [1.4142] [0.7071] 'I'
2.	cAint	<i>double</i>	Berisi nilai low frekuensi <i>approximation coefficients</i> dari proses LWT pada citra <i>cover</i> .
3.	cHint	<i>double</i>	Berisi nilai <i>horizontal detail coefficients</i> dari proses LWT pada citra <i>cover</i> .
4.	cVint	<i>double</i>	Berisi nilai <i>vertical detail coefficients</i> dari proses LWT pada citra <i>cover</i> .
5.	cDint	<i>double</i>	Berisi nilai <i>diagonal detail coefficients</i> dari proses LWT pada citra <i>cover</i> .
6.	Bi_w	<i>double</i>	Inisialisasi yang berisi nilai dari cAint w

3.3.2.3 Tahap MOD Q_w

Pada tahapan ini dilakukan proses Modulo matriks Q_w terhadap matriks Bi_w. Terdapat dua keluaran pada proses Modulo ini, yaitu hasil bagi dan hasil sisa. Hasil sisa disimpan sebagai variabel Ci_w, sedangkan hasil bagi disimpan sebagai variabel Di_w. Diagram alir proses ini dapat dilihat pada Gambar 3.31 dan Gambar 3.32.



Gambar 3.31 Diagram Alir proses Mod Q_w

Masukan	Matriks Bi
Keluaran	Matriks Ci, matriks Di
1. Inisialisasi matriks Q _w 2. $Ci_w = \text{mod}(Bi_w, Q_w)$; 3. $cell_Ci_w\{i, j\} = Ci_w$; 4. $Di_w = Bi_w - Ci_w$; 5. $cell_Di_w\{i, j\} = Di_w$;	

Gambar 3.32 Pseudocode Proses Mod Q

Tabel 3.15 Daftar Variabel Pada Proses Mod Q (bagian 1)

No.	Nama Variabel	Tipe Data	Keterangan
1.	Q _w	double	Q _w = [5 5 5 5; 5 5 5 5; 5 5 5 5]
2.	Ci _w	double	Berisi nilai hasil sisa dari MOD Q terhadap Bi _w

Tabel 3.16 Daftar Variabel Pada Proses Mod Q (bagian 2)

No.	Nama Variabel	Tipe Data	Keterangan
3.	Bi_w	double	Berisi nilai hasil bagi dari MOD Q terhadap Bi_w
4.	Di_w	double	Berisi nilai hasil bagi dari MOD Q terhadap Bi_w
5.	cell_Ci_w	cell	Cell berukuran 64x64 berisi nilai hasil sisa dari MOD Q terhadap Bi_w
6.	cell_Di_w	cell	Cell berukuran 64x64 berisi nilai hasil bagi dari MOD Q terhadap Bi_w

3.3.2.4 Tahap Ekstraksi Citra Logo *Watermark*

Pada tahapan ini dilakukan proses ekstraksi citra logo *watermark* yang telah disisipkan pada citra ter-*watermark*. Tahapan pertama yang dilakukan yaitu membaca data manifest yang sebelumnya telah dibuat, berisi mengenai nomer sub-blok mana saja yang dipilih untuk disisipi citra logo *watermark*. Setelah itu kemudian dilakukan proses ekstraksi citra logo *watermark* hingga membentuk citra logo berukuran 32×32 sesuai dengan bentuk semula logo citra *watermark* saat sebelum disisipkan. *Pseudocode* mengenai tahapan ini ditunjukkan pada Gambar 3.33 dan Gambar 3.34.

Masukan	Sub-blok citra ter- <i>watermark</i>
Keluaran	Nomer sub-blok yang disisipi citra logo <i>watermark</i> , citra logo <i>watermark</i> yang disisipkan
1. Import data manifest 2. for i=1 to tesM_w do	

Gambar 3.33 *Pseudocode* Proses Ekstraksi Citra Logo *Watermark* (bagian 1)


```

3.   for j=1 to tesN_w do
4.   if mod(i-1,2)=true &&mod(j-1,2)=true
      then
5.     temp_Ciaksen = cell_Ci_w{i,j}(2:3,2:3);
6.     var_nomer=i+(j-1)*tesM_w;
7.     if ismember(var_nomer,manifest) then
8.       temp_logo(counter) = 0;
9.     end if
10.    counter=counter+1;
11.  end if
12. end for
13. end for

```

Gambar 3.34 Pseudocode Proses Ekstraksi Citra Logo Watermark (bagian 2)

Tabel 3.17 Daftar Variabel Pada Proses Ekstraksi Citra Logo Watermark (bagian 1)

No.	Nama Variabel	Tipe Data	Keterangan
1.	manifest	double	Berisi data nomer sub-blok yang disisipi citra logo watermark
2.	temp_Ciaksen	double	Variabel temporary untuk menyimpan nilai Ci_w pada citra ter-watermark
3.	cell_Ci_w	double	Cell berukuran 64×64 berisi nilai hasil sisa dari MOD Q terhadap Bi_w pada citra ter-watermark
4.	tesM_w	double	Total baris sub-blok citra ter-watermark

Tabel 3.18 Daftar Variabel Pada Proses Ekstraksi Citra Logo Watermark (bagian 2)

No.	Nama Variabel	Tipe Data	Keterangan
5.	var_nomer	double	Berisi urutan nomer sub-blok
6.	counter	double	Counter untuk citra ter-watermark. counter = 1

3.3.2.5 Tahap Inverse Enkripsi Citra Logo Watermark

Pada tahapan ini dilakukan *inverse* enkripsi citra logo *watermark* yang telah diekstraksi pada tahapan sebelumnya. Ukuran citra logo *watermark* berukuran 32×32 , sehingga proses *inverse* enkripsi dilakukan dengan melakukan iterasi Arnold Cat Map sebanyak 23 kali agar citra logo *watermark* dapat kembali menjadi seperti bentuk saat sebelum disisipkan ke dalam citra *cover*. *Pseudocode* yang menjelaskan mengenai alur tahap *inverse* enkripsi citra logo *watermark* ditunjukkan pada Gambar 3.35.

Masukan	Citra logo <i>watermark</i> terenkripsi yang telah diekstraksi
Keluaran	Citra logo <i>watermark</i> telah terenkripsi
<pre> 1. for ix to 1:23 do 2. temp_logo = catmap(temp_logo); 3. end </pre>	

Gambar 3.35 Pseudocode Inverse Enkripsi Citra Logo Watermark

Tabel 3.19 Daftar Variabel Pada Proses *Inverse* Enkripsi Citra Logo *Watermark*

No.	Nama Variabel	Tipe Data	Keterangan
1.	temp_logo	<i>double</i>	Berisi hasil ekstraksi citra logo <i>watermark</i>

BAB IV IMPLEMENTASI

Pada Bab III telah dijelaskan mengenai desain perancangan perangkat lunak. Sedangkan pada Bab IV ini dijelaskan mengenai tahapan implementasi perangkat lunak yang digunakan untuk melakukan proses penyisipan dan ekstraksi citra logo *watermark*. Tahapan yang dilakukan disajikan dalam bentuk *pseudocode* untuk memudahkan proses pemahaman kode program yang terdapat dalam perangkat lunak.

4.1 Lingkungan Implementasi

Implementasi perangkat lunak penyisipan dan ekstraksi citra logo *watermark* dilakukan dengan menggunakan perangkat keras dan perangkat lunak yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat Keras	Prosesor: Intel® Core™ i3-3240 CPU @ 3.40 GHz 3.40 GHz Memori: 4.00 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 7 Ultimate Perangkat Pengembang: Matlab 7.6.0 (R2008a)

4.2 Implementasi

Implementasi sistem perangkat lunak penyisipan dan ekstraksi citra logo *watermark* dilakukan sesuai dengan alur yang telah dijelaskan pada bab sebelumnya. Program utama pada sistem perangkat lunak ini terbagi menjadi dua, yaitu: implementasi program penyisipan citra logo *watermark* (*TA_Penyisipan.m*) dan implementasi program ekstraksi citra logo *watermark* (*TA_Ekstraksi.m*). Masing-masing program utama tersebut memiliki fungsi dan program pendukung yang dibutuhkan dalam

menjalankan sistem. Setiap bagiannya akan dijabarkan pada sub bab selanjutnya.

4.2.1 Implementasi Program Utama Proses Penyisipan

Program utama proses penyisipan dalam sistem perangkat lunak penyisipan dan ekstraksi citra logo *watermark* diimplementasikan pada *TA_Penyisipan.m*. Pada program ini terdapat beberapa tahapan, antara lain: input citra *cover*, input citra logo *watermark*, enkripsi citra logo *watermark*, pembagian sub-blok citra *cover*, penerapan wavelet, ekstraksi low frekuensi, Mod Q, penentuan nilai x dan y, komputasi nilai slope, pemilihan sub-blok yang disisipi, pemilihan 4 *adjacent* piksel, penerapan inverse wavelet, serta komputasi nilai PSNR.

Pada proses awal, *user* diminta untuk memilih citra logo *watermark* berukuran 32×32 yang akan disisipkan ke dalam citra *cover*. Setelah itu *user* diminta untuk memilih citra *cover* berukuran 512×512 yang akan digunakan. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.1 dan Kode Sumber 4.2.

1	<code>% memilih citra logo watermark</code>
2	<code>im = uigetfile({'*.bmp'; '*.tif'; '*.jpg'}, 'Pilih Citra Logo Watermark ');</code>
3	<code>im = imread(im);</code>
4	<code>% mengambil ukuran citra logo watermark</code>
5	<code>[P,R] = size(im);</code>

Kode Sumber 4.1 Kode untuk Memilih dan Membaca Citra Logo *Watermark*

1	<code>% memilih citra cover</code>
2	<code>im2 = uigetfile({'*.bmp'; '*.tif'; '*.jpg'}, 'Pilih Citra Cover');</code>
3	<code>im2 = imread(im2);</code>
4	<code>% mengambil ukuran citra cover</code>
5	<code>[M,N] = size(im2);</code>

Kode Sumber 4.2 Kode untuk Memilih dan Membaca Citra *Cover*

Selanjutnya citra logo *watermark* dienkripsi dengan memanggil fungsi Arnold Cat Map. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.3.

1	<code>% memanggil fungsi Arnold Cat Map</code>
2	<code>im = catmap(im);</code>

Kode Sumber 4.3 Kode untuk Memanggil Fungsi Arnold Cat Map

Untuk mengukur kualitas citra hasil keluaran, dilakukan dengan menggunakan fungsi PSNR. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.4.

1	<code>% memanggil fungsi PSNR</code>
2	<code>[psnr]=PSNR(im2,image_watermarked)</code>

Kode Sumber 4.4 Kode untuk Memanggil Fungsi PSNR

Penjelasan implementasi pada tahapan yang lainnya akan dijelaskan pada sub bab selanjutnya.

4.2.1.1 Implementasi Enkripsi Citra Logo *Watermark*

Pada sub bab ini dilakukan proses enkripsi citra logo *watermark* dengan menggunakan Arnold Cat Map (*catmap.m*). Hasil keluaran dari proses ini didapatkan citra logo watermark yang telah diacak, sehingga bentuknya menjadi abstrak dan tidak dapat dengan mudah dikenali. Implementasi enkripsi citra logo *watermark* ditunjukkan pada Kode Sumber 4.5 dan Kode Sumber 4.6.

1	<code>% applies Arnold's Cat Map to the image Y to produce image X</code>
2	<code>function X = catmap(Y)</code>
3	<code>% get the number of pixels on each side</code>
4	<code>p = size(Y,1);</code>
5	<code>% make space for X (all zeros to start)</code>

Kode Sumber 4.5 Kode Implementasi Fungsi Arnold Cat Map (bagian 1)

6	<code>X = zeros(size(Y));</code>
7	<code>% loop through all the pixels</code>
8	<code>for i = 1:p</code>
9	<code> for j = 1:p</code>
10	<code> % get new i coord (m+n) mod p</code>
11	<code> newi = mod(((i-1) + (j-1)),p) + 1;</code>
12	<code> % get new j coord (m+2n) mod p</code>
12	<code> newj = mod(((i-1) + 2*(j-1)),p) + 1;</code>
13	<code> X(newi,newj,:) = Y(i,j,:);</code>
14	<code> end</code>
15	<code>end</code>

Kode Sumber 4.6 Kode Implementasi Fungsi Arnold Cat Map (bagian 2)

4.2.1.2 Implementasi Pembagian Sub-blok Citra Cover

Pada sub bab ini akan dijelaskan mengenai pembagian sub-blok citra *cover*. Tahapan pertama, citra *cover* yang berukuran 512×512 akan dibagi menjadi 64×64 sub-blok, yang masing-masing sub-blok berukuran 8×8 . Hasil keluaran dari proses ini, diperoleh sub-blok yang berukuran 8×8 yang tiap-tiap baris dan kolomnya memiliki batasan sampai 1 sampai 8. Setelah itu dilakukan konversi tipe data menjadi double agar dapat diproses pada tahapan selanjutnya. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.7.

1	<code>% membagi ukuran tiap sub-blok menjadi berukuran 8x8</code>
2	<code>subM = floor(M/64); %baris</code>
3	<code>subN = floor(N/64); % kolom</code>
4	<code>% untuk menghitung total sub-blok</code>
5	<code>tesM = M/subM; % ukuran total sub-blok (baris)</code>
6	<code>tesN = N/subN; % ukuran total sub-blok (kolom)</code>
7	<code>% untuk menentukan batasan sub-blok</code>
8	<code>batas1 = (i-1)*subM+1:i*subM</code>
9	<code>batas2 = (j-1)*subN+1:j*subN</code>
10	<code>cell_subImage{i,j} = im2(batas1,batas2)</code>
11	<code>% mengubah tipe data menjadi double</code>
12	<code>cell_A{i,j} = double(cell_subImage{i,j});</code>

Kode Sumber 4.7 Kode untuk Membagi Sub-Blok

4.2.1.3 Implementasi Penerapan Wavelet Pada Citra Cover

Pada sub bab ini dijelaskan penerapan Integer Lifting Wavelet Transform level 1. Penerapan Integer LWT level 1 digunakan untuk mengolah sinyal menjadi beberapa macam sinyal frekuensi.

Hasil keluaran yang didapatkan berupa sinyal frekuensi rendah (*low frequency*), yaitu: cAint dan sinyal frekuensi tinggi (*high frequency*), yaitu: cHint, cDint, cVint. Masing-masing sinyal frekuensi tersebut memiliki ukuran 4×4 pada tiap-tiap sub-blok. Selanjutnya dilakukan ekstraksi sinyal frekuensi rendah (cAint), kemudian cAint diinisialisasikan menjadi variabel Bi. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.8.

1	<code>% penerapan Integer LWT-2D Level 1</code>
2	<code>lshaarInt = liftwave('haar', 'int2int');</code>
3	<code>[cAint, cHint, cVint, cDint] = lwt2(cell A{i,j}, lshaarInt);</code>
4	<code>% cAint disimpan dalam variabel Bi (ekstraksi low frekuensi)</code>
5	<code>Bi = cAint;</code>
6	<code>cell Bi{i,j} = cAint;</code>

Kode Sumber 4.8 Kode untuk Menerapkan Integer Lifting Wavelet Transform-2D Level 1

4.2.1.4 Implementasi MOD Q

Pada tahapan ini dilakukan proses Modulo matriks Q terhadap matriks Bi. Sebelumnya dilakukan inisialisasi matriks Q yang isi matriksnya telah ditetapkan sebelumnya, yaitu berupa matriks berukuran 4×4 . Terdapat dua keluaran pada proses Modulo ini, yaitu hasil bagi dan hasil sisa. Hasil sisa disimpan sebagai variabel Ci, sedangkan hasil bagi disimpan sebagai variabel Di. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.9.

1	<code>% inisialisasi matriks Q</code>
2	<code>Q = [5 5 5 5;5 5 5 5;5 5 5 5;5 5 5 5];</code>
3	<code>Ci = mod(Bi,Q); % remainder matriks</code>
4	<code>cell Ci{i,j}=Ci;</code>
5	<code>Di = Bi-Ci; % divisible matriks</code>
6	<code>cell Di{i,j}=Di;</code>

Kode Sumber 4.9 Kode untuk Melakukan Mod Q

4.2.1.5 Implementasi Penentuan Nilai x dan y

Pada tahapan ini dilakukan penentuan nilai x dan y yang akan digunakan pada perhitungan nilai slope di sub bab selanjutnya. Iterasi untuk mendapatkan nilai x dan y hanya sampai sebanyak $m-1$, karena jika dilakukan sebanyak m maka akan melebihi batas matriks yang ada. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.10.

1	<code>% mengambil ukuran Ci</code>
2	<code>[m,n] = size(Ci);</code>
3	<code>for a=1:m-1</code>
4	<code> for b=1:n</code>
5	<code> x(a,b)=Ci(a,b); % nilai x</code>
6	<code> y(a,b)=Ci(a+1,b); % nilai y</code>
7	<code> end</code>
8	<code>end</code>

Kode Sumber 4.10 Kode untuk Menentukan Nilai x dan y

4.2.1.6 Implementasi Komputasi Nilai Slope

Pada sub bab ini akan dilakukan komputasi nilai slope dengan menggunakan nilai x dan y yang telah didapatkan dari perhitungan pada proses sebelumnya. Rumus yang digunakan dalam perhitungan nilai slope menggunakan rumus yang ada pada Persamaan 2.2. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.11.

1	<code>% menghitung nilai rata-rata x</code>
2	<code>ratarata_x=mean(mean(x));</code>
3	<code>% menghitung nilai rata-rata y</code>
4	<code>ratarata_y=mean(mean(y));</code>
5	<code>a1(i,j) = sum(sum((x-ratarata_x).*(y-ratarata_y)));</code>
6	<code>a2(i,j) = sum(sum((x-ratarata_x).*(x-ratarata_x)));</code>
7	<code>% menghitung nilai slope</code>
8	<code>slope a(i,j) = a1(i,j)/a2(i,j);</code>

Kode Sumber 4.11 Kode untuk Menghitung Nilai Slope

4.2.1.7 Implementasi Pemilihan Sub-Blok Disisipi

Pada sub bab ini dilakukan pemilihan sub-blok yang akan disisipi informasi citra logo *watermark*. Pemilihan sub-blok dilakukan secara menyebar, sehingga citra logo *watermark* yang disisipkan tidak dapat dengan mudah rusak ketika terjadi distorsi pada citra ter-*watermark*.

Tahapan selanjutnya dilakukan pemilihan 4 *adjacent* piksel yang ada pada variabel C_i apabila kondisi piksel pada citra logo *watermark* bernilai 0. Pemilihan 4 *adjacent* piksel ini diambil di bagian tengah pada tiap-tiap C_i , kemudian dihitung nilai rata-ratanya. Setelah itu hasil rata-rata (*mean*) yang ada pada variabel *matriksTengah* tersebut digunakan untuk meng-*cover* nilai 4 *adjacent* piksel tersebut, sehingga 4 *adjacent* piksel tersebut memiliki nilai yang sama. Tiap-tiap C_i memiliki 4 *adjacent* piksel masing-masing dan nilai *mean* yang berbeda-beda.

Selanjutnya hasil yang telah didapatkan disimpan ke dalam variabel C_{iaksen} . Ketika piksel tersebut memiliki kondisi 0, maka C_i yang disimpan ke dalam variabel C_{iaksen} merupakan C_i hasil modifikasi yang memiliki 4 *adjacent* piksel bertetangga. Tetapi ketika piksel tersebut tidak memiliki 4 *adjacent* piksel yang bertetangga, maka nilai C_i yang disimpan ke dalam C_{iaksen} merupakan *matriks* C_i semula yang tidak dimodifikasi. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.12.

1	counter_slope=1;
2	c2=1;
3	slopeUbah = zeros(1024,2);
4	for i=1:size(im,1)*size(im,2)
5	cekTengah=cell_Ci{counter_slope}(2:3,2:3);
6	noBlock=temp_slope(counter_slope,2);
7	if im(i)==0
8	matriksTengah(noBlock)=mean(mean(cekTengah));
9	if matriksTengah(noBlock)<1
10	matriksTengah(noBlock)=1;
11	end
12	ab=ones(2);
13	cell_Ciaksen{noBlock}=cell_Ci{noBlock};
14	cell_Ciaksen{noBlock}(2:3,2:3)=ab*matriksTengah(noBlock);
15	matriksCek(noBlock)=1;
16	slopeUbah(c2,:)=temp_slope(counter_slope,:);
17	elseif im(i)==1;
18	matriksTengah(noBlock)=mean(mean(cekTengah));
19	cell_Ciaksen{noBlock}=cell_Ci{noBlock};
20	cell_Ciaksen{noBlock}(2:3,2:3) = 0;
21	matriksCek(noBlock)=1;
22	slopeUbah(c2,:)=temp_slope(counter_slope,:);
23	end
24	c2=c2+1;
25	counter_slope=counter_slope+1;
26	end

Kode Sumber 4.12 Kode untuk Melakukan Pemilihan Sub-Blok yang Disisipi Citra Logo *Watermark*

4.2.1.8 Implementasi Manifest Citra Logo *Watermark*

Pada sub bab ini dilakukan pembuatan manifest dengan tujuan untuk mencatat bagian sub-blok mana saja yang disisipi citra logo *watermark*. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.13.

1	<code>fid = fopen('manifest_logo.txt', 'w');</code>
2	<code>for ix=1:tesM*tesN</code>
3	<code> fprintf(fid, '%d\n', slope_a(ix));</code>
4	<code>end</code>
5	<code>fclose(fid);</code>

Kode Sumber 4.13 Kode untuk Membuat Manifest

4.2.1.9 Implementasi Penerapan Inverse Wavelet

Pada sub bab ini dilakukan proses Inverse Integer LWT dengan tujuan untuk mengembalikan kembali sinyal yang telah diproses menjadi hasil keluaran citra ter-*watermark*. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.14

1	<code>for ii=1:tesM</code>
2	<code> for jj=1:tesN</code>
3	<code> cell_Biaksen{ii,jj}=</code> <code> cell_Ciaksen{ii,jj}+cell_Di{ii,jj};</code>
4	<code> cell_hasil_ilwt{ii,jj} =</code> <code> ilwt2(cell_Biaksen{ii,jj},cell_cHint{ii,jj},ce</code> <code> ll_cVint{ii,jj},cell_cDint{ii,jj},lshaarInt);</code>
5	<code> end</code>
6	<code>end</code>
7	<code>% mengubah tipe cell ke bentuk matriks</code>
8	<code>mat_hasil_ilwt = cell2mat(cell_hasil_ilwt);</code>
9	<code>image_watermarked = uint8(mat_hasil_ilwt);</code>

Kode Sumber 4.14 Kode untuk Melakukan Inverse LWT

4.2.1.10 Implementasi Komputasi Nilai PSNR

Pada sub bab ini dilakukan komputasi nilai PSNR untuk mengukur kualitas citra ter-*watermark* yang didapatkan dari hasil keluaran. Apabila hasil keluaran nilai PSNR yang didapatkan semakin mendekati nilai 100, maka kualitas citra tersebut semakin bagus dan semakin mendekati citra aslinya. Tetapi jika nilai PSNR yang didapatkan semakin mendekati nilai 0, maka kualitas citra yang dihasilkan semakin buruk dan semakin jauh kualitasnya jika dibandingkan dengan citra aslinya.

Pada sistem perangkat lunak ini dilakukan pengukuran kualitas, ketika hasil keluaran $PSNR \leq 30$ maka kualitas citra tersebut buruk. Ketika hasil keluaran PSNR berada pada rentang antara 30 sampai 60, maka kualitas citra tersebut berada pada kategori sedang. Sedangkan apabila nilai hasil keluaran $PSNR \geq 60$, maka kualitas citra tersebut masuk ke dalam kategori bagus. Implementasi tahapan komputasi nilai PSNR ditunjukkan pada Kode Sumber 4.15.

1	<code>% fungsi untuk menghitung PSNR</code>
2	<code>function [psnr]=PSNR(im2,image watermarked)</code>
3	<code>im2=double(im2);</code>
4	<code>image watermarked=double(image watermarked);</code>
5	<code>if (im2==image watermarked)</code>
6	<code>psnr=100;</code>
7	<code>else</code>
8	<code>[r c]=size(im2);</code>
9	<code>d=0;</code>
10	<code>for i=1:r</code>
11	<code>for j=1:c</code>
12	<code>d=d+(im2(i,j)- image_watermarked(i,j))^2;</code>
13	<code>end</code>
14	<code>end</code>
15	<code>mse=d/(r*c) % hasil mse</code>
16	<code>maximum=max(im2(:));</code>
17	<code>psnr=10*log10(maximum^2/mse);</code>
18	<code>end</code>

Kode Sumber 4.15 Kode untuk Menghitung Nilai PSNR

4.2.2 Implementasi Program Utama Proses Ekstraksi

Program utama pada proses ekstraksi dalam sistem perangkat lunak penyisipan dan ekstraksi citra logo *watermark* diimplementasikan pada *TA_Ekstraksi.m*. Tahapan yang dilakukan pada proses ekstraksi ini hampir sama dengan proses awal yang dilakukan pada proses penyisipan. Tahapannya yaitu: membaca input citra ter-*watermark*, pembagian sub-blok citra ter-*watermark*, penerapan wavelet, ekstraksi low frekuensi, Mod Q, ekstraksi citra logo *watermark*, serta *inverse* enkripsi citra logo *watermark*.

4.2.2.1 Implementasi Pembagian Sub-Blok Citra Ter-Watermark

Pada sub bab ini dilakukan pembagian sub-blok pada citra ter-*watermark*. Tidak jauh berbeda dengan proses penyisipan, citra ter-*watermark* berukuran 512×512 dibagi menjadi 64×64 sub-blok yang masing-masing sub-bloknya berukuran 8×8 . Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.16.

1	<code>% membagi ukuran tiap sub-blok menjadi berukuran 8x8</code>
2	<code>subM_w = floor(m/64); %baris</code>
3	<code>subN_w = floor(n/64); % kolom</code>
4	<code>% untuk menghitung total sub-blok</code>
5	<code>tesM_w = m/subM_w; % ukuran total sub-blok (baris)</code>
6	<code>tesN_w = n/subN_w; % ukuran total sub-blok (kolom)</code>
7	<code>% untuk menentukan batasan sub-blok</code>
8	<code>batas w1 = (i-1)*subM_w+1:i*subM_w</code>
9	<code>batas w2 = (j-1)*subN_w+1:j*subN_w</code>
10	<code>cell subImage w{i,j} = im_w(batas_w1,batas_w2)</code>
11	<code>% mengubah tipe data menjadi double</code>
12	<code>cell W{i,j} = double(cell_subImage w{i,j});</code>

Kode Sumber 4.16 Kode Pembagian Sub-Blok Pada Citra Ter-Watermark

Selanjutnya dilakukan pembagian batasan awal dan batasan akhir piksel pada tiap-tiap sub-blok citra ter-*watermark* berdasarkan baris dan kolom, sehingga tiap-tiap baris dan kolomnya memiliki batasan sampai 1 sampai 8. Hasil keluaran dari proses ini didapatkan sub-blok yang mempunyai batasan 8 baris dan batasan 8 kolom. Kemudian dilakukan konversi tipe data menjadi double agar dapat diproses pada tahapan selanjutnya.

4.2.2.2 Implementasi Penerapan Wavelet Pada Citra Ter-*Watermark*

Pada sub bab ini dijelaskan mengenai penerapan Integer Lifting Wavelet Transform level 1 untuk citra ter-*watermark*. Tahapan yang dilakukan juga sama dengan tahapan penerapan wavelet yang dilakukan pada proses penyisipan. Penerapan Integer LWT level 1 digunakan untuk mengolah suatu sinyal menjadi beberapa macam sinyal frekuensi. Hasil keluaran yang didapatkan dari hasil pengolahan tersebut berupa sinyal frekuensi rendah (*low frequency*) dan sinyal frekuensi tinggi (*high frequency*).

Hasil keluaran yang didapatkan berupa sinyal frekuensi rendah (*low frequency*), yaitu: *cAint_w* dan sinyal frekuensi tinggi (*high frequency*), yaitu: *cHint_w*, *cDint_w*, *cVint_w*. Masing-masing sinyal frekuensi tersebut memiliki ukuran 4×4 pada tiap-tiap sub-blok. Selanjutnya dilakukan ekstraksi sinyal frekuensi rendah (*cAint_w*), kemudian *cAint_w* diinisialisasikan menjadi variabel *Bi*. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.17.

1	<code>% penerapan Integer LWT-2D Level 1</code>
2	<code>lshaarInt = liftwawe('haar', 'int2int');</code>
3	<code>[cAint_w, cHint_w, cVint_w, cDint_w]=lwt2(cell_W{i, j}, lshaarInt);</code>
4	<code>% cAint disimpan dalam variabel Bi w</code>
5	<code>Bi w = cAint w;</code>
6	<code>cell_Bi_w{i, j} = cAint_w;</code>

Kode Sumber 4.17 Kode untuk Menerapkan Integer Lifting Wavelet Transform-2D Level 1 Pada Citra Ter-*Watermark*

Tahapan selanjutnya kemudian dilakukan ekstraksi sinyal frekuensi rendah untuk diproses pada proses ekstraksi citra logo *watermark*.

4.2.2.3 Implementasi MOD Q_w

Pada sub bab ini dilakukan Modulo matriks Q_w terhadap variabel Bi_w . Terdapat dua keluaran pada proses Modulo ini, yaitu hasil bagi dan hasil sisa. Hasil sisa disimpan sebagai variabel Ci_w , sedangkan hasil bagi disimpan sebagai variabel Di_w . Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.18.

1	<code>% inisialisasi matriks Q_w</code>
2	<code>$Q_w = [5\ 5\ 5\ 5; 5\ 5\ 5\ 5; 5\ 5\ 5\ 5; 5\ 5\ 5\ 5];$</code>
3	<code>$Ci_w = \text{mod}(Bi_w, Q_w);$ % remainder matriks</code>
4	<code>cell $Ci_w\{i,j\}=Ci_w;$</code>
5	<code>$Di_w = Bi_w - Ci_w;$ % divisible matriks</code>
6	<code>cell $Di_w\{i,j\}=Di_w;$</code>

Kode Sumber 4.18 Kode untuk Melakukan Mod Q

4.2.2.4 Implementasi Ekstraksi Citra Logo *Watermark*

Pada tahapan ini dilakukan proses ekstraksi citra logo *watermark* yang telah disisipkan pada citra ter-*watermark*. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.19 dan Kode Sumber 4.20.

1	<code>% import data manifest</code>
2	<code>manifest=importdata('manifest logo2.txt');</code>
3	<code>% proses ekstraksi citra logo watermark</code>
4	<code>for i=1:tesM w</code>
5	<code> for j=1:tesN w</code>
6	<code> if mod(i-1,2)==0&&mod(j-1,2)==0</code>
7	<code> temp Ciaksen=cell Ci_w{i,j}(2:3,2:3);</code>
8	<code> var_nomer=i+(j-1)*tesM_w;</code>

Kode Sumber 4.19 Kode untuk Melakukan Proses Ekstraksi Citra Logo *Watermark* (bagian 1)

9	<code>if ismember(var nomer,manifest)</code>
10	<code>temp_logo(counter) = 0;</code>
11	<code>end</code>
12	<code>counter=counter+1;</code>
13	<code>end</code>
14	<code>end</code>
15	<code>end</code>

Kode Sumber 4.20 Kode untuk Melakukan Proses Ekstraksi Citra Logo *Watermark* (bagian 2)

4.2.2.5 Implementasi *Inverse* Enkripsi Citra Logo *Watermark*

Setelah citra logo *watermark* telah berhasil diekstraksi, maka akan dilakukan *inverse* enkripsi dengan menggunakan *inverse* Arnold Cat Map sebanyak 23 kali iterasi agar citra logo *watermark* dapat kembali menjadi bentuknya seperti semula saat sebelum disisipkan. Implementasi tahapan ini ditunjukkan pada Kode Sumber 4.21.

1	<code>% inverse Arnold Cat Map</code>
2	<code>for ix = 1:23</code>
3	<code>temp_logo = catmap(temp_logo);</code>
4	<code>end</code>

Kode Sumber 4.21 Kode untuk Melakukan *Inverse* Arnold Cat Map

BAB V

HASIL UJI COBA DAN EVALUASI

Bab V ini merupakan pembahasan mengenai hasil uji coba dan evaluasi terhadap sistem perangkat lunak penyisipan dan ekstraksi citra logo *watermark*. Uji coba ini dilakukan dengan menggunakan data masukan citra *cover* dan citra logo *watermark*. Kemudian masing-masing citra tersebut akan diolah sesuai dengan tahapannya masing-masing dalam proses penyisipan maupun dalam proses ekstraksi citra logo *watermark*. Tahapan pada bab ini antara lain meliputi lingkungan uji coba, data uji coba, hasil uji coba, dan evaluasi yang akan dijelaskan lebih lanjut pada sub bab selanjutnya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan mengenai lingkup lingkungan yang digunakan dalam proses uji coba implementasi sistem perangkat lunak penyisipan dan ekstraksi citra logo *watermark*. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Uji Coba

Perangkat Keras	Prosesor: Intel® Core™ i3-3240 CPU @ 3.40 GHz 3.40 GHz Memori: 4.00 GB
Perangkat Lunak	Sistem Operasi: Microsoft Windows 7 Ultimate Perangkat Pengembang: Matlab 7.6.0 (R2008a)

5.2 Data Uji Coba

Pada sistem perangkat lunak ini menggunakan data uji coba berupa citra *cover* dengan tipe *grayscale* yang memiliki ukuran 512×512 dan citra logo *watermark* bertipe biner yang memiliki ukuran 32×32 . Contoh data uji coba dan citra logo

watermark yang digunakan dapat dilihat pada Tabel A. 25 sampai Tabel A. 46.

5.3 Skenario Uji Coba

Uji coba dilakukan untuk melakukan pengujian terhadap fungsionalitas pada sistem perangkat lunak telah diimplementasikan dan berjalan dengan baik atau tidak. Skenario uji coba akan dilakukan pada beberapa percobaan untuk mengetahui kinerja sistem perangkat lunak.

Pengujian dilakukan berdasarkan dua hal, yaitu uji kebenaran sistem perangkat lunak dan uji kinerja sistem perangkat lunak yang akan dijelaskan pada sub bab selanjutnya.

5.3.1 Uji Kebenaran

Pada tahap uji kebenaran ini akan dilakukan pengujian mengenai kebenaran jalannya sistem perangkat lunak, sudah sesuai dengan rancangan yang sebelumnya telah dibuat atau belum.

Citra *cover* yang dimasukkan berupa citra *grayscale* dengan format .bmp, sedangkan citra logo *watermark* yang dimasukkan berupa citra biner dengan format .bmp. Setelah didapatkan citra masukan, kemudian dilakukan pengujian dengan melakukan proses penyisipan informasi citra logo *watermark* ke dalam citra *cover*. Keluaran hasil penyisipan citra logo *watermark* tidak terlihat banyak perbedaan secara signifikan. Contoh citra hasil ekstraksi dapat dilihat pada Gambar 5.1.



Gambar 5.1 Pengujian Hasil Ekstraksi (a) Citra Logo *Watermark* Original (b) Citra Logo *Watermark* Hasil Ekstraksi

5.3.2 Uji Kinerja

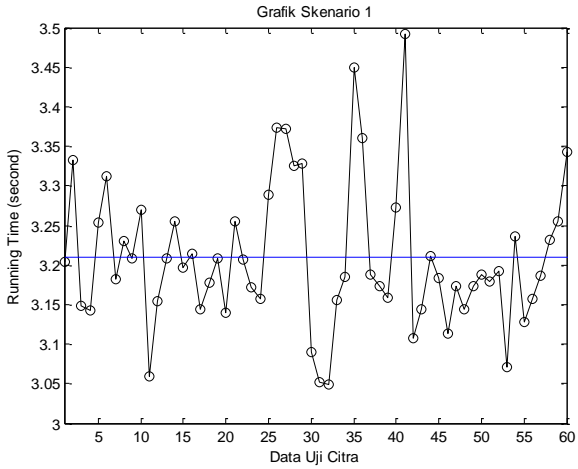
Pada tahapan uji kinerja dilakukan pengujian mengenai kinerja sistem perangkat lunak terhadap beberapa pengolahan citra digital terutama ketika terjadi distorsi, citra logo *watermark* hasil ekstraksi masih kualitasnya masih bagus atau tidak. Citra *cover* yang digunakan berupa citra *grayscale* berukuran 512×512 , sedangkan citra logo *watermark* yang digunakan berupa citra biner berukuran 32×32 . Skenario uji kinerja akan dilakukan berdasarkan PSNR dan *running time* program antara citra yang berbeda format, yaitu format .bmp, .jpg, dan .png. Penjelasan lebih lanjut mengenai beberapa macam uji kinerja akan dibahas pada sub bab selanjutnya.

5.3.2.1 Uji Kinerja dan Evaluasi Skenario 1: Uji Kinerja Citra Ter-Watermark dengan Format .bmp Berdasarkan *Running Time*

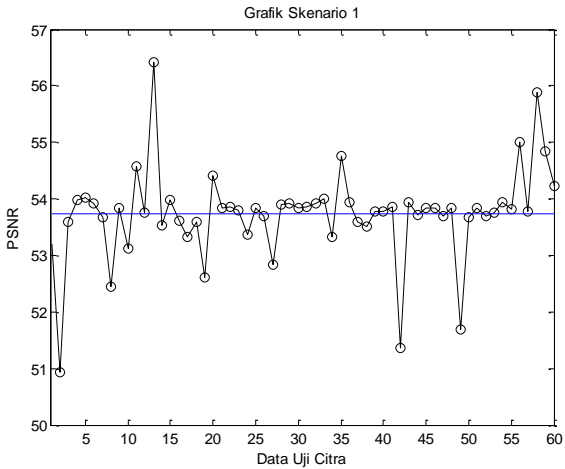


Gambar 5.2 Citra Hasil Keluaran Skenario 1

Skenario ini menggunakan citra ter-*watermark* dengan format .bmp berukuran 512×512 dengan perhitungan nilai PSNR. Hasil perhitungan nilai PSNR yang dihasilkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 1 dapat dilihat pada Gambar 5.3 dan Gambar 5.4.



Gambar 5.3 Grafik Skenario 1 Berdasarkan *Running Time*



Gambar 5.4 Grafik Skenario 1 Berdasarkan PSNR

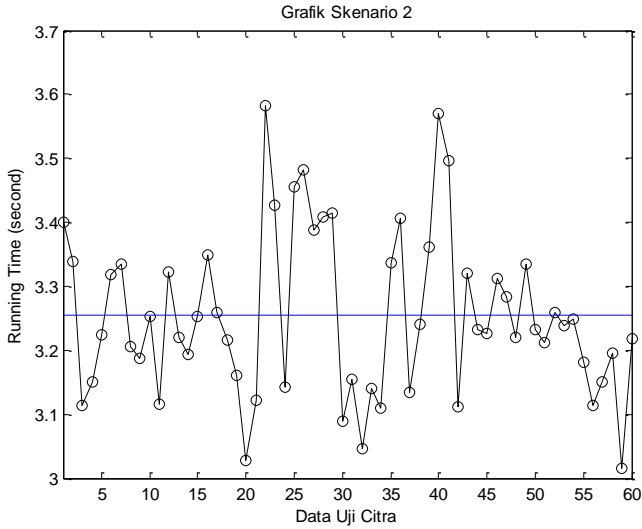
Dari hasil keluaran uji coba skenario 1 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .bmp. Pada Gambar 5.3 diperoleh nilai rata-rata *running time* sebesar 53.72857, sedangkan pada Gambar 5.4 diperoleh nilai rata-rata PSNR sebesar 3.209537. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.2 Uji Kinerja dan Evaluasi Skenario 2: Uji Kinerja Citra Ter-Watermark dengan Format .jpg Berdasarkan *Running Time*

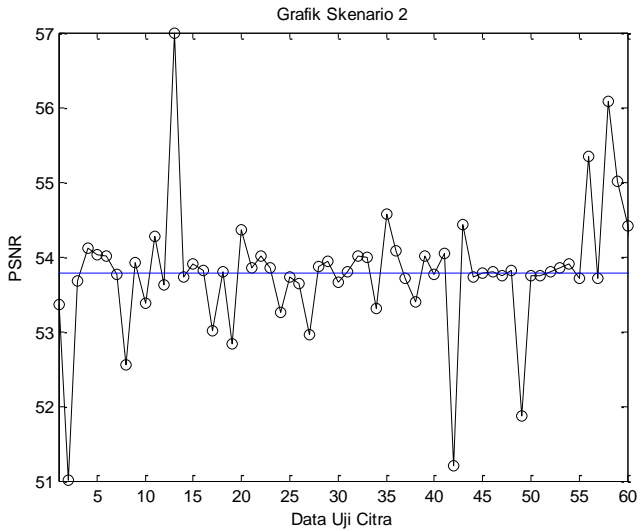


Gambar 5.5 Citra Hasil Keluaran Skenario 2

Skenario ini menggunakan citra ter-*watermark* dengan format .jpg berukuran 512×512 dengan perhitungan nilai PSNR. Hasil perhitungan nilai PSNR yang dihasilkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 2 dapat dilihat pada Gambar 5.6 dan Gambar 5.7.



Gambar 5.6 Grafik Skenario 2 Berdasarkan *Running Time*



Gambar 5.7 Grafik Skenario 2 Berdasarkan PSNR

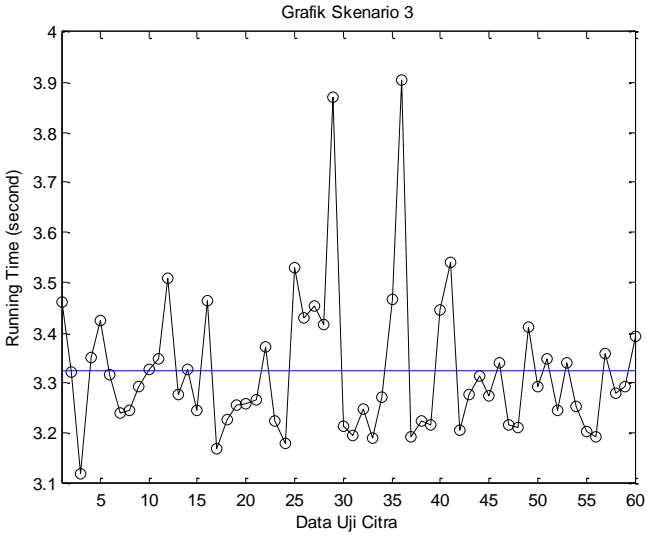
Dari hasil keluaran uji coba skenario 2 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .jpg. Pada Gambar 5.6 diperoleh nilai rata-rata *running time* sebesar 53.78064, sedangkan pada Gambar 5.7 diperoleh nilai rata-rata PSNR sebesar 3.255092. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.3 Uji Kinerja dan Evaluasi Skenario 3: Uji Kinerja Citra Ter-Watermark dengan Format .png Berdasarkan *Running Time*

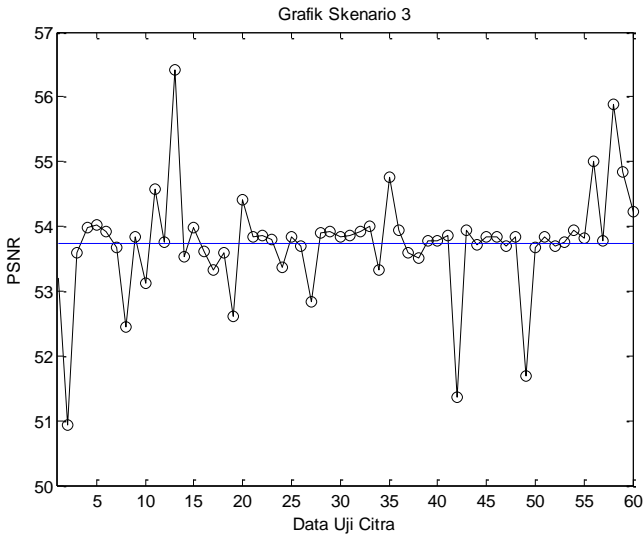


Gambar 5.8 Citra Hasil Keluaran Skenario 3

Skenario ini menggunakan citra ter-*watermark* dengan format .png berukuran 512×512 dengan perhitungan nilai PSNR. Hasil perhitungan nilai PSNR yang dihasilkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 3 dapat dilihat pada Gambar 5.9 dan Gambar 5.10.



Gambar 5.9 Grafik Skenario 3 Berdasarkan *Running Time*



Gambar 5.10 Grafik Skenario 3 Berdasarkan PSNR

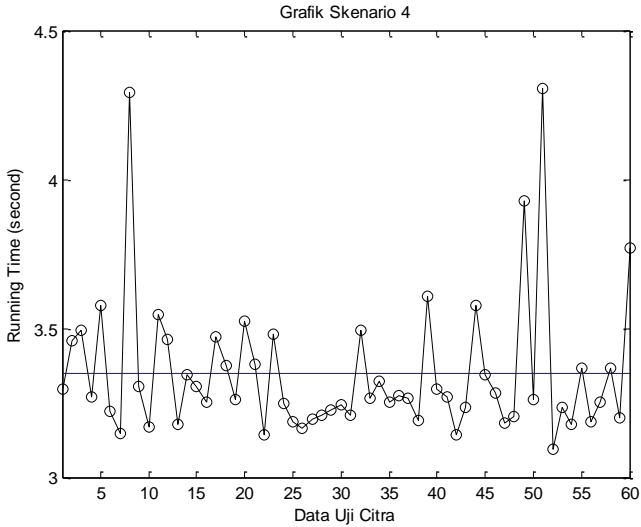
Dari hasil keluaran uji coba skenario 3 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .png. Pada Gambar 5.9 diperoleh nilai rata-rata *running time* sebesar 53.72857, sedangkan pada Gambar 5.10 diperoleh nilai rata-rata PSNR sebesar 3.323744. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.4 Uji Kinerja dan Evaluasi Skenario 4: Uji Kinerja Citra Ter-Watermark dengan Distorsi Derau Salt & Pepper (Intensitas Derau: 0.02)

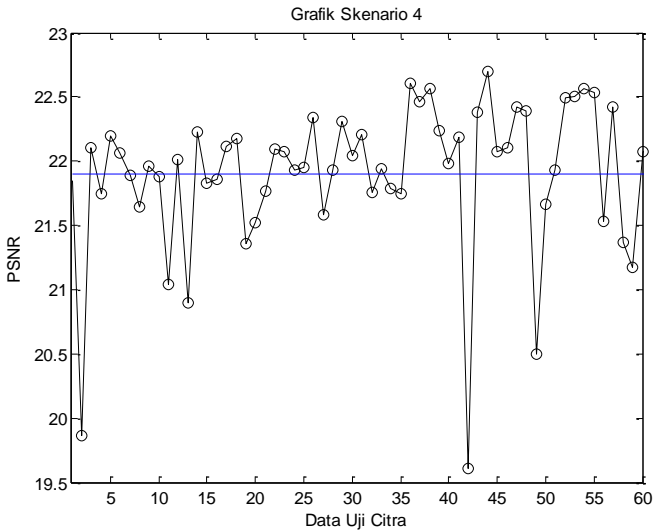


Gambar 5.11 Citra Hasil Keluaran Skenario 4

Skenario ini menggunakan citra ter-*watermark* dengan format .bmp berukuran 512×512 dengan pengujian berdasarkan pengujian terhadap distorsi derau salt & pepper dengan intensitas 0.02. Hasil PSNR dan *running time* yang didapatkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 4 dapat dilihat pada Gambar 5.12 dan Gambar 5.13.



Gambar 5.12 Grafik Skenario 4 Berdasarkan *Running Time*



Gambar 5.13 Grafik Skenario 4 Berdasarkan PSNR

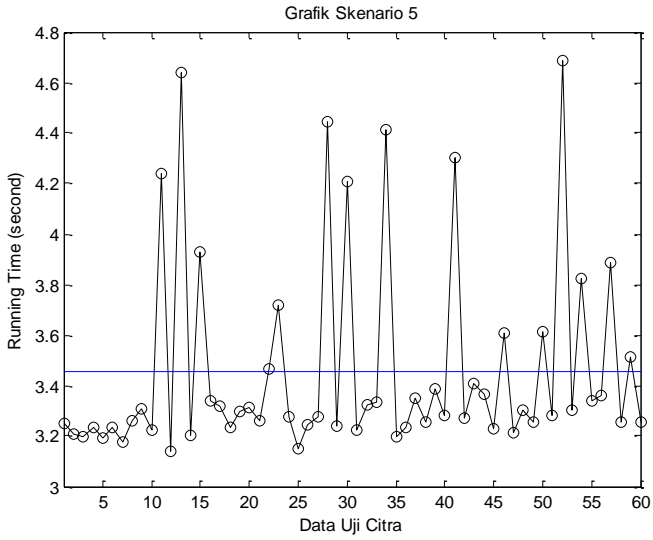
Dari hasil keluaran uji coba skenario 4 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .png. Pada Gambar 5.12 diperoleh nilai rata-rata *running time* sebesar 21.9049, sedangkan pada Gambar 5.13 diperoleh nilai rata-rata PSNR sebesar 3.346948. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.5 Uji Kinerja dan Evaluasi Skenario 5: Uji Kinerja Citra Ter-Watermark dengan Distorsi Derau Salt & Pepper (Intensitas Derau: 0.04)

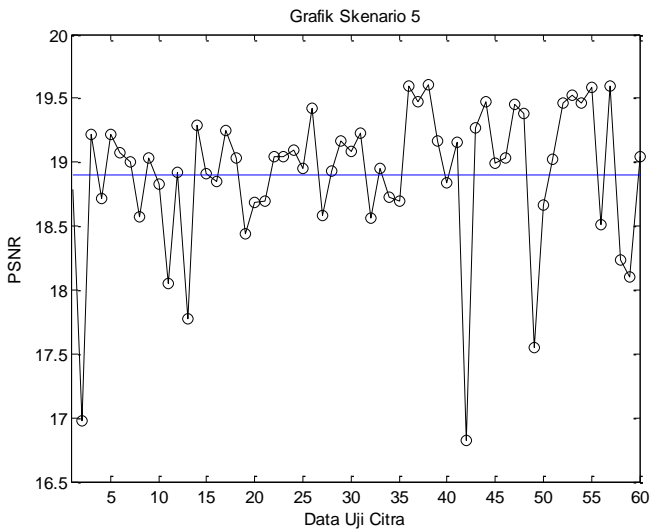


Gambar 5.14 Citra Hasil Keluaran Skenario 5

Skenario ini menggunakan citra ter-*watermark* dengan format .bmp berukuran 512×512 dengan pengujian berdasarkan pengujian terhadap distorsi derau salt & pepper dengan intensitas 0.04. Hasil PSNR dan *running time* yang didapatkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 5 dapat dilihat pada Gambar 5.15 dan Gambar 5.16.



Gambar 5.15 Grafik Skenario 5 Berdasarkan *Running Time*



Gambar 5.16 Grafik Skenario 5 Berdasarkan PSNR

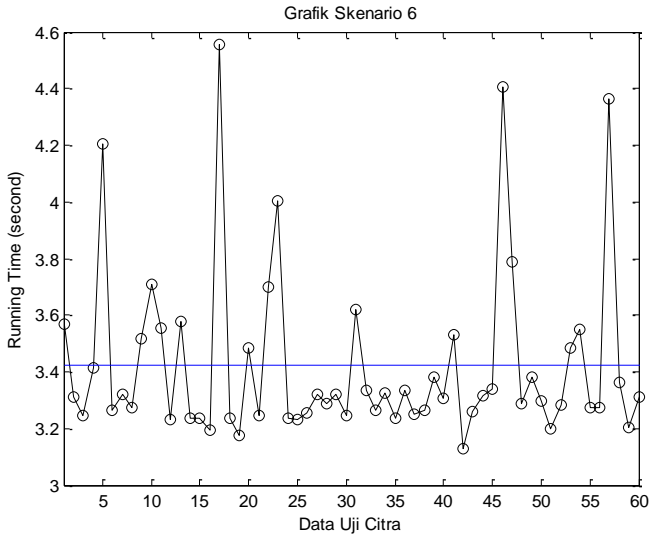
Dari hasil keluaran uji coba skenario 5 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .png. Pada Gambar 5.15 diperoleh nilai rata-rata *running time* sebesar 18.89861, sedangkan pada Gambar 5.16 diperoleh nilai rata-rata PSNR sebesar 3.454507. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.6 Uji Kinerja dan Evaluasi Skenario 6: Uji Kinerja Citra Ter-Watermark dengan Distorsi Derau Salt & Pepper (Intensitas Derau: 0.06)

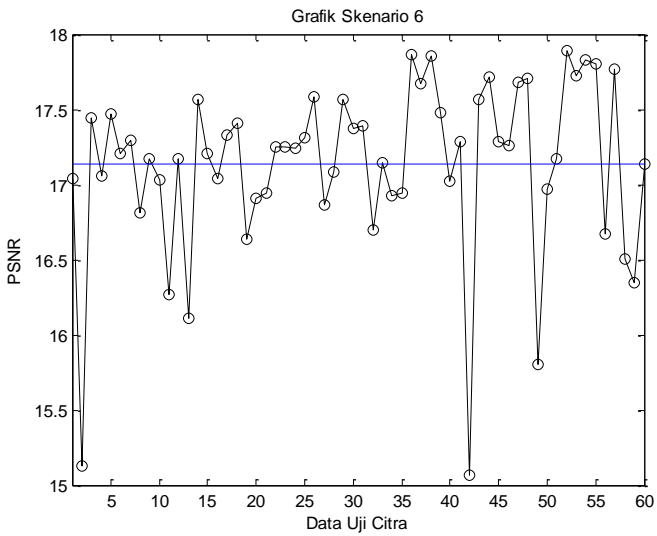


Gambar 5.17 Citra Hasil Keluaran Skenario 6

Skenario ini menggunakan citra ter-*watermark* dengan format .bmp berukuran 512×512 dengan pengujian berdasarkan pengujian terhadap distorsi derau salt & pepper dengan intensitas 0.06. Hasil PSNR dan *running time* yang didapatkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 6 dapat dilihat pada Gambar 5.18 dan Gambar 5.19.



Gambar 5.18 Grafik Skenario 6 Berdasarkan *Running Time*



Gambar 5.19 Grafik Skenario 6 Berdasarkan PSNR

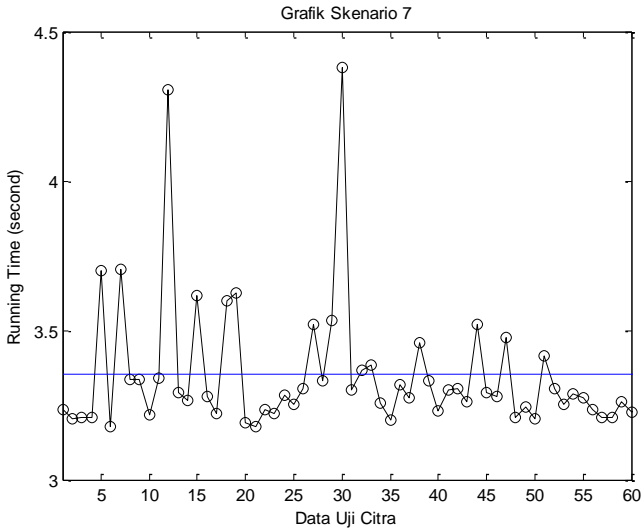
Dari hasil keluaran uji coba skenario 6 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .png. Pada Gambar 5.18 diperoleh nilai rata-rata *running time* sebesar 17.13488, sedangkan pada Gambar 5.19 diperoleh nilai rata-rata PSNR sebesar 3.422737. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.7 Uji Kinerja dan Evaluasi Skenario 7: Uji Kinerja Citra Ter-Watermark dengan Distorsi Derau Salt & Pepper (Intensitas Derau: 0.08)

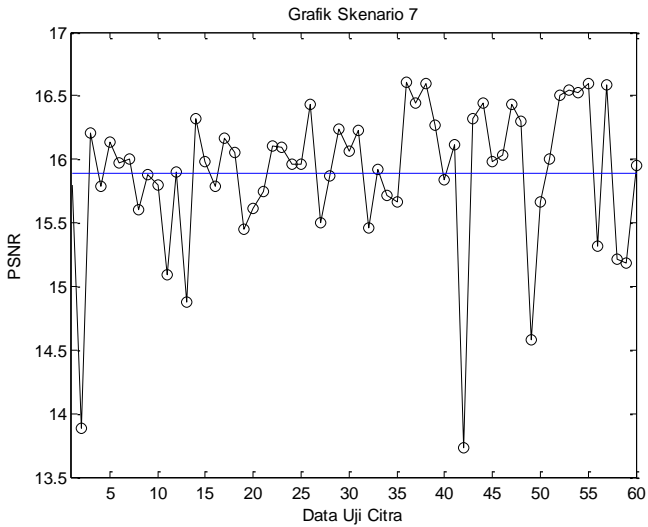


Gambar 5.20 Citra Hasil Keluaran Skenario 7

Skenario ini menggunakan citra ter-*watermark* dengan format .bmp berukuran 512×512 dengan pengujian berdasarkan pengujian terhadap distorsi derau salt & pepper dengan intensitas 0.08. Hasil PSNR dan *running time* yang didapatkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 7 dapat dilihat pada Gambar 5.21 dan Gambar 5.22.



Gambar 5.21 Grafik Skenario 7 Berdasarkan *Running Time*



Gambar 5.22 Grafik Skenario 7 Berdasarkan PSNR

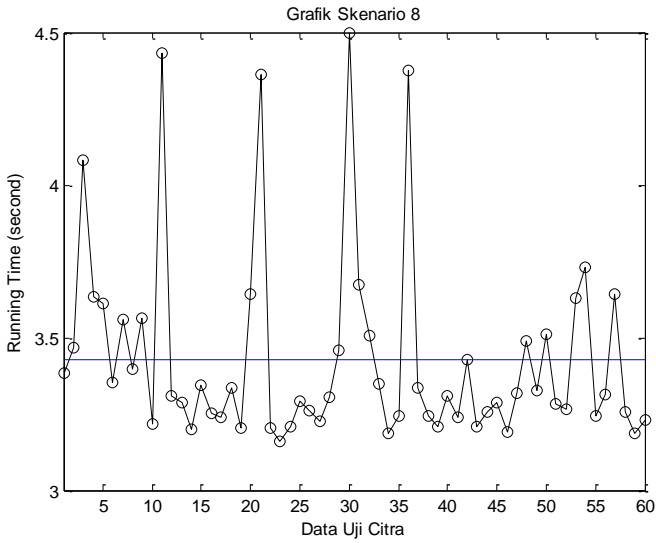
Dari hasil keluaran uji coba skenario 7 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .png. Pada Gambar 5.21 diperoleh nilai rata-rata *running time* sebesar 15.88647, sedangkan pada Gambar 5.22 diperoleh nilai rata-rata PSNR sebesar 3.35142. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

5.3.2.8 Uji Kinerja dan Evaluasi Skenario 8: Uji Kinerja Citra Ter-Watermark dengan Distorsi Derau Salt & Pepper (Intensitas Derau: 0.1)

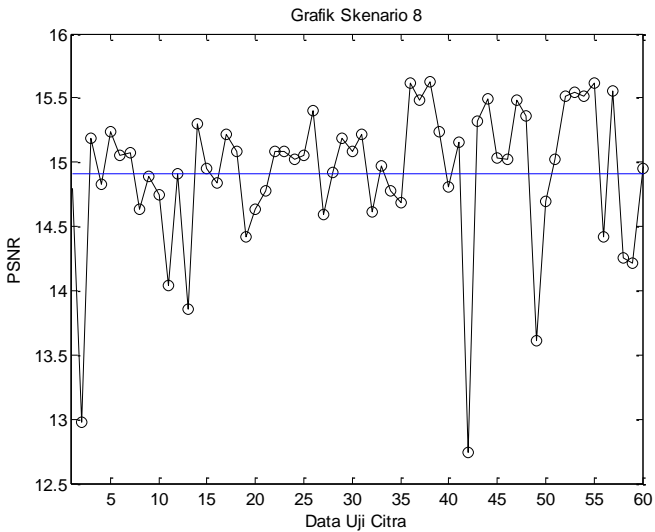


Gambar 5.23 Citra Hasil Keluaran Skenario 8

Skenario ini menggunakan citra ter-*watermark* dengan format .bmp berukuran 512×512 dengan pengujian berdasarkan pengujian terhadap distorsi derau salt & pepper dengan intensitas 0.1. Hasil PSNR dan *running time* yang didapatkan antar citra data uji coba berbeda-beda. Hasil uji kinerja skenario 8 dapat dilihat pada Gambar 5.24 dan Gambar 5.25.



Gambar 5.24 Grafik Skenario 8 Berdasarkan *Running Time*



Gambar 5.25 Grafik Skenario 8 Berdasarkan PSNR

Dari hasil keluaran uji coba skenario 8 diperoleh hasil PSNR dan *running time* yang berbeda-beda pada tiap data uji coba yang berformat .png. Pada Gambar 5.24 diperoleh nilai rata-rata *running time* sebesar 15.88647, sedangkan pada Gambar 5.25 diperoleh nilai rata-rata PSNR sebesar 3.35142. Pada kedua grafik tersebut didapatkan kesimpulan terdapat beberapa data uji coba yang menonjol di bagian PSNR dan *running time*. Ketika homogenitas piksel warna pada data uji hanya sedikit, maka semakin lama *running time*-nya. Tetapi semakin banyak piksel yang memiliki homogenitas, maka *running time*-nya semakin cepat.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi mengenai kesimpulan yang dapat diperoleh dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran dengan tujuan agar perangkat lunak ini dapat dikembangkan lebih lanjut menjadi lebih baik lagi.

6.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan hasil uji coba dan evaluasi yang telah dilakukan adalah sebagai berikut:

1. Pada proses penyisipan dan ekstraksi secara normal, artinya tanpa diberi derau apapun, citra logo *watermark* yang dihasilkan sama dengan citra logo *watermark* original.
2. Pada Tugas Akhir ini metode Integer Lifting Wavelet Transform berhasil diterapkan dalam proses filtering *lowpass* dan *highpass*, sedangkan metode Regresi Linier berhasil diterapkan dalam proses pemilihan sub-blok mana yang akan disisipi citra logo *watermark*.
3. Pada pengujian skenario dengan pemberian derau salt & pepper dengan intensitas 0.02, terdapat penurunan sebesar 59.23%. Pada intensitas 0.04 terdapat penurunan sebesar 64.49%. Intensitas 0.06 menghasilkan penurunan sebesar 67.93%, sedangkan pada intensitas 0.08 terdapat penurunan sebesar 70.27%. terakhir, pada pengujian skenario salt & pepper dengan intensitas 0.1 terdapat penurunan sebesar 59.23%.
4. Pengujian pada proses penyisipan dan ekstraksi citra logo *watermark* dengan menggunakan derau salt & pepper dengan intensitas berbeda menghasilkan hasil keluaran

citra logo *watermark* yang tidak sama dengan citra logo *watermark*. Semakin tinggi intensitas derau yang diberikan terhadap citra ter-*watermark*, maka semakin rusak citra logo *watermark* yang dihasilkan pada saat ekstraksi. Selain itu diperoleh hasil pula bahwa semakin tinggi intensitas derau yang diberikan, semakin menurun pula nilai PSNR yang dihasilkan.

5. Pada pengujian proses penyisipan citra logo *watermark* berdasarkan *running time* dengan format citra yang berbeda-beda yaitu .jpg, .bmp., dan .png, diperoleh hasil keluaran nilai PSNR pada citra berformat .bmp dan .png cenderung sama.
6. Pada proses penyisipan, penyebaran posisi penyisipan sub-blok dilihat apakah slope berada dalam rentang 0.1 sampai 1.0 atau tidak. Sehingga semakin menyebar posisi sub-blok yang disisipi maka akan semakin lama *running time* dan semakin rendah nilai PSNR yang dihasilkan.

6.2 Saran

Beberapa saran terkait dengan pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Perlu dilakukan pengembangan metode yang mampu membuat citra ter-*watermark* jauh lebih tahan terhadap beraneka macam distorsi.
2. Perlu dilakukan lebih banyak pengujian dengan berbagai citra logo *watermark*.

DAFTAR PUSTAKA

- [1] diniakape, "Lebih Baik Dibanding Tidak Sama Sekali: Pembuatan Watermarking Menggunakan Matlab 7.1," [Online]. Available: <http://tugasdiniakape.blogspot.com/2010/11/pembuatan-watermarking-menggunakan.html>. [Diakses 08 07 2014].
- [2] R. Agustina, A. dan A. M. Barmawi, "Teknik Watermarking Menggunakan AMBTC dan IWT untuk Pendeteksian dan Perbaikan Citra Digital Termanipulasi," vol. 15, no. 2, 2010.
- [3] L. Song dan G. Qialun, "A Novel Digital Watermarking Algorithm Based on Wavelet Lifting Scheme and Linear Regression," pp. 1602-1606, 2012.
- [4] K. Vij, "Comparative Study of Different Techniques of Image Enhancement for Grayscale and Colour Images," Punjab, India, 2011.
- [5] C.-T. Hsu dan J.-L. , "Hidden Digital Watermarks in Images," vol. 8, no. 1, p. 11, 1999.
- [6] J. Kapur dan A. J. Baregar, "Security Using Image Processing," vol. 5, no. 2, p. 9, 2013.
- [7] R. M. "Algoritma Enkripsi Citra Digital Berbasis Chaos dengan Penggabungan Teknik Permutasi dan Teknik Substitusi Menggunakan Arnold Cat Map dan Logistic Map," pp. 107-124, 2012.
- [8] zhuslee, "Kasembon Hall," [Online]. Available: <http://zhuslee.wordpress.com/2011/09/17/ekstrasi-ciri-menggunakan-transformasi-wavelet-diskrit/>. [Diakses 1 Juli 2014].
- [9] MathWorks, "Lifting schemes information - MATLAB linfo," The MathWorks, Inc., [Online]. Available: <http://www.mathworks.com/help/wavelet/ref/linfo.html>. [Diakses 20 Juni 2014].

BIODATA PENULIS



Rina Kharisma Juwitasari, akrab dipanggil Juwita, lahir di Surabaya pada tanggal 23 Oktober 1991, merupakan anak ketiga dari tiga bersaudara. Penulis telah menempuh pendidikan mulai dari SD Khadijah (1997-2003), SMP Khadijah (2003-2006), SMA Khadijah (2006-2009) dan terakhir sebagai mahasiswa Teknik Informastika ITS Surabaya (2009-2014). Selama kuliah, penulis pernah menjadi asisten pada mata kuliah Organisasi Komputer, mata kuliah Sistem Operasi dan mata kuliah Kecerdasan Komputasional. Selain itu, penulis juga aktif di organisasi kemahasiswaan yaitu Himpunan Mahasiswa Teknik Computer-Informatika (HMTTC) dan Studi Islam Teknik Computer-Informatika (SITC). Dalam menyelesaikan kuliahnya, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV). Selain dalam bidang akademik maupun organisasi, penulis sangat suka menghabiskan waktu dengan *travelling*, *blogging*, membaca buku terutama novel, mendengarkan musik, wisata kuliner, menonton film, dan sebagainya. Penulis dapat dihubungi melalui e-mail: rina.kharisma@gmail.com.

LAMPIRAN A

Berikut ini lampiran dari pengujian skenario 1 berdasarkan PSNR dan *Running Time*.



Gambar A. 1 Hasil Keluaran Citra Logo *Watermark* dengan Intensitas Derau 0.02

Tabel A. 1 Hasil Keluaran Skenario 1 (bagian 1)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.bmp	53.2072	3.204276 seconds.
2.	DS_3096.bmp	50.9329	3.332506 seconds.
3.	DS_12003.bmp	53.5828	3.148070 seconds.
4.	DS_15004.bmp	53.9764	3.142765 seconds.
5.	DS_27059.bmp	54.0277	3.253732 seconds.
6.	DS_35008.bmp	53.9150	3.311996 seconds.
7.	DS_35010.bmp	53.6733	3.181955 seconds.
8.	DS_35070.bmp	52.4415	3.230245 seconds.
9.	DS_38092.bmp	53.8385	3.207627 seconds.
10.	DS_42049.bmp	53.1173	3.269643 seconds.
11.	DS_42044.bmp	54.5773	3.059433 seconds.
12.	DS_376001.bmp	53.7463	3.154764 seconds.
13.	DS_42078.bmp	56.4109	3.208416 seconds.
14.	DS_45077.bmp	53.5375	3.255312 seconds.
15.	DS_46076.bmp	53.9893	3.197316 seconds.
16.	DS_55073.bmp	53.6225	3.213907 seconds.

Tabel A. 2 Hasil Keluaran Skenario 1 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
17.	DS_60079.bmp	53.3220	3.143682 seconds.
18.	DS_66075.bmp	53.5964	3.177128 seconds.
19.	DS_67079.bmp	52.6117	3.207658 seconds.
20.	DS_69015.bmp	54.4193	3.139278 seconds.
21.	DS_76002.bmp	53.8341	3.254587 seconds.
22.	DS_78019.bmp	53.8537	3.206277 seconds.
23.	DS_85048.bmp	53.7986	3.171176 seconds.
24.	DS_95006.bmp	53.3689	3.157397 seconds.
25.	DS_97017.bmp	53.8341	3.289167 seconds.
26.	DS_100075.bmp	53.6887	3.374127 seconds.
27.	DS_100080.bmp	52.8380	3.372466 seconds.
28.	DS_101085.bmp	53.8982	3.325670 seconds.
29.	DS_102061.bmp	53.9227	3.327983 seconds.
30.	DS_108005.bmp	53.8422	3.089353 seconds.
31.	DS_118020.bmp	53.8621	3.052467 seconds.
32.	DS_124084.bmp	53.9105	3.049218 seconds.
33.	DS_140075.bmp	53.9923	3.154984 seconds.
34.	DS_143090.bmp	53.3287	3.184284 seconds.
35.	DS_144067.bmp	54.7557	3.449326 seconds.
36.	DS_161062.bmp	53.9499	3.360167 seconds.
37.	DS_163062.bmp	53.5925	3.187932 seconds.
38.	DS_172032.bmp	53.5147	3.172631 seconds.
39.	DS_176019.bmp	53.7664	3.158484 seconds.
40.	DS_176035.bmp	53.7774	3.272822 seconds.
41.	DS_176039.bmp	53.8665	3.491782 seconds.
42.	DS_178054.bmp	51.3621	3.106976 seconds.
43.	DS_182053.bmp	53.9413	3.143534 seconds.
44.	DS_183087.bmp	53.7091	3.211755 seconds.
45.	DS_189011.bmp	53.8369	3.183595 seconds.
46.	DS_197017.bmp	53.8450	3.113925 seconds.
47.	DS_210088.bmp	53.7038	3.173811 seconds.
48.	DS_216041.bmp	53.8311	3.143340 seconds.

Tabel A. 3 Hasil Keluaran Skenario 1 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
49.	DS_227040.bmp	51.6831	3.172606 seconds.
50.	DS_239007.bmp	53.6775	3.187664 seconds.
51.	DS_242078.bmp	53.8348	3.178390 seconds.
52.	DS_253027.bmp	53.7028	3.192285 seconds.
53.	DS_254033.bmp	53.7621	3.071556 seconds.
54.	DS_254054.bmp	53.9463	3.236363 seconds.
55.	DS_268002.bmp	53.8156	3.128293 seconds.
56.	DS_306005.bmp	55.0083	3.156816 seconds.
57.	DS_314016.bmp	53.7736	3.185890 seconds.
58.	DS_317080.bmp	55.8793	3.231911 seconds.
59.	DS_351093.bmp	54.8347	3.254779 seconds.
60.	DS_368078.bmp	54.2164	3.342451 seconds.
61.	DS_374020.bmp	53.3374	3.223831 seconds.

Berikut ini lampiran dari pengujian skenario 2 berdasarkan PSNR dan *Running Time*.

**Gambar A. 2 Hasil Keluaran Citra Logo Watermark dengan Intensitas Derau 0.04****Tabel A. 4 Hasil Keluaran Skenario 2 (bagian 1)**

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.jpg	53.3563	3.399652 seconds.
2.	DS_3096.jpg	51.0173	3.338523 seconds.
3.	DS_12003.jpg	53.6758	3.113867 seconds.

Tabel A. 5 Hasil Keluaran Skenario 2 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
4.	DS_15004.jpg	54.1169	3.151429 seconds.
5.	DS_27059.jpg	54.0184	3.224953 seconds.
6.	DS_35008.jpg	54.0079	3.319129 seconds.
7.	DS_35010.jpg	53.7659	3.333947 seconds.
8.	DS_35070.jpg	52.5610	3.204912 seconds.
9.	DS_38092.jpg	53.9136	3.186919 seconds.
10.	DS_42049.jpg	53.3711	3.253629 seconds.
11.	DS_42044.jpg	54.2653	3.116294 seconds.
12.	DS_376001.jpg	53.6202	3.321869 seconds.
13.	DS_42078.jpg	56.9853	3.219414 seconds.
14.	DS_45077.jpg	53.7234	3.193226 seconds.
15.	DS_46076.jpg	53.8980	3.252520 seconds.
16.	DS_55073.jpg	53.8107	3.348537 seconds.
17.	DS_60079.jpg	53.0070	3.259248 seconds.
18.	DS_66075.jpg	53.7940	3.215178 seconds.
19.	DS_67079.jpg	52.8319	3.160021 seconds.
20.	DS_69015.jpg	54.3593	3.027052 seconds.
21.	DS_76002.jpg	53.8572	3.121846 seconds.
22.	DS_78019.jpg	54.0039	3.583034 seconds.
23.	DS_85048.jpg	53.8595	3.426383 seconds.
24.	DS_95006.jpg	53.2511	3.143147 seconds.
25.	DS_97017.jpg	53.7346	3.455099 seconds.
26.	DS_100075.jpg	53.6363	3.481979 seconds.
27.	DS_100080.jpg	52.9580	3.388876 seconds.
28.	DS_101085.jpg	53.8693	3.408511 seconds.
29.	DS_102061.jpg	53.9316	3.413464 seconds.
30.	DS_108005.jpg	53.6577	3.089684 seconds.
31.	DS_118020.jpg	53.7921	3.154682 seconds.
32.	DS_124084.jpg	54.0161	3.045770 seconds.
33.	DS_140075.jpg	53.9928	3.139866 seconds.
34.	DS_143090.jpg	53.3105	3.109880 seconds.
35.	DS_144067.jpg	54.5666	3.337595 seconds.

Tabel A. 6 Hasil Keluaran Skenario 2 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
36.	DS_161062.jpg	54.0807	3.405380 seconds.
37.	DS_163062.jpg	53.7192	3.133772 seconds.
38.	DS_172032.jpg	53.3906	3.241089 seconds.
39.	DS_176019.jpg	54.0021	3.360402 seconds.
40.	DS_176035.jpg	53.7567	3.570594 seconds.
41.	DS_176039.jpg	54.0412	3.496113 seconds.
42.	DS_178054.jpg	51.1944	3.111676 seconds.
43.	DS_182053.jpg	54.4229	3.319809 seconds.
44.	DS_183087.jpg	53.7344	3.232228 seconds.
45.	DS_189011.jpg	53.7818	3.226979 seconds.
46.	DS_197017.jpg	53.7981	3.311895 seconds.
47.	DS_210088.jpg	53.7493	3.284360 seconds.
48.	DS_216041.jpg	53.8168	3.220935 seconds.
49.	DS_227040.jpg	51.8748	3.334231 seconds.
50.	DS_239007.jpg	53.7444	3.231879 seconds.
51.	DS_242078.jpg	53.7414	3.211588 seconds.
52.	DS_253027.jpg	53.7989	3.258975 seconds.
53.	DS_254033.jpg	53.8508	3.238501 seconds.
54.	DS_254054.jpg	53.9034	3.248463 seconds.
55.	DS_268002.jpg	53.7192	3.180701 seconds.
56.	DS_306005.jpg	55.3484	3.114102 seconds.
57.	DS_314016.jpg	53.7130	3.151110 seconds.
58.	DS_317080.jpg	56.0808	3.195559 seconds.
59.	DS_351093.jpg	55.0160	3.015687 seconds.
60.	DS_368078.jpg	54.4117	3.217751 seconds.
61.	DS_374020.jpg	53.3917	3.306672 seconds.

Berikut ini lampiran dari pengujian skenario 3 berdasarkan PSNR dan *Running Time*.



Gambar A. 3 Hasil Keluaran Citra Logo *Watermark* dengan Intensitas Derau 0.06

Tabel A. 7 Hasil Keluaran Skenario 3 (bagian 1)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.png	53.2072	3.461069 seconds.
2.	DS_3096.png	50.9329	3.322987 seconds.
3.	DS_12003.png	53.5828	3.120182 seconds.
4.	DS_15004.png	53.9764	3.350859 seconds.
5.	DS_27059.png	54.0277	3.425133 seconds.
6.	DS_35008.png	53.9150	3.316838 seconds.
7.	DS_35010.png	53.6733	3.240973 seconds.
8.	DS_35070.png	52.4415	3.246913 seconds.
9.	DS_38092.png	53.8385	3.293295 seconds.
10.	DS_42049.png	53.1173	3.327299 seconds.
11.	DS_42044.png	54.5773	3.348565 seconds.
12.	DS_376001.png	53.7463	3.510395 seconds.
13.	DS_42078.png	56.4109	3.277137 seconds.
14.	DS_45077.png	53.5375	3.327980 seconds.
15.	DS_46076.png	53.9893	3.246369 seconds.
16.	DS_55073.png	53.6225	3.463507 seconds.
17.	DS_60079.png	53.3220	3.170635 seconds.
18.	DS_66075.png	53.5964	3.227563 seconds.
19.	DS_67079.png	52.6117	3.256116 seconds.
20.	DS_69015.png	54.4193	3.259000 seconds.

Tabel A. 8 Hasil Keluaran Skenario 3 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
21.	DS_76002.png	53.8341	3.265967 seconds.
22.	DS_78019.png	53.8537	3.371206 seconds.
23.	DS_85048.png	53.7986	3.225713 seconds.
24.	DS_95006.png	53.3689	3.180840 seconds.
25.	DS_97017.png	53.8341	3.529285 seconds.
26.	DS_100075.png	53.6887	3.430066 seconds.
27.	DS_100080.png	52.8380	3.453417 seconds.
28.	DS_101085.png	53.8982	3.417884 seconds.
29.	DS_102061.png	53.9227	3.868632 seconds.
30.	DS_108005.png	53.8422	3.214226 seconds.
31.	DS_118020.png	53.8621	3.194995 seconds.
32.	DS_124084.png	53.9105	3.248301 seconds.
33.	DS_140075.png	53.9923	3.190421 seconds.
34.	DS_143090.png	53.3287	3.273016 seconds.
35.	DS_144067.png	54.7557	3.467296 seconds.
36.	DS_161062.png	53.9499	3.903283 seconds.
37.	DS_163062.png	53.5925	3.192668 seconds.
38.	DS_172032.png	53.5147	3.224307 seconds.
39.	DS_176019.png	53.7664	3.217633 seconds.
40.	DS_176035.png	53.7774	3.445540 seconds.
41.	DS_176039.png	53.8665	3.541234 seconds.
42.	DS_178054.png	51.3621	3.205846 seconds.
43.	DS_182053.png	53.9413	3.276662 seconds.
44.	DS_183087.png	53.7091	3.314249 seconds.
45.	DS_189011.png	53.8369	3.275487 seconds.
46.	DS_197017.png	53.8450	3.341543 seconds.
47.	DS_210088.png	53.7038	3.217997 seconds.

Tabel A. 9 Hasil Keluaran Skenario 3 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
48.	DS_216041.png	53.8311	3.210590 seconds.
49.	DS_227040.png	51.6831	3.411354 seconds.
50.	DS_239007.png	53.6775	3.293652 seconds.
51.	DS_242078.png	53.8348	3.349771 seconds.
52.	DS_253027.png	53.7028	3.245316 seconds.
53.	DS_254033.png	53.7621	3.341989 seconds.
54.	DS_254054.png	53.9463	3.252717 seconds.
55.	DS_268002.png	53.8156	3.203425 seconds.
56.	DS_306005.png	55.0083	3.193632 seconds.
57.	DS_314016.png	53.7736	3.358390 seconds.
58.	DS_317080.png	55.8793	3.280396 seconds.
59.	DS_351093.png	54.8347	3.292193 seconds.
60.	DS_368078.png	54.2164	3.394286 seconds.
61.	DS_374020.png	53.3374	3.240120 seconds.

Berikut ini lampiran dari pengujian skenario 4 berdasarkan PSNR dan *Running Time*.

**Gambar A. 4 Hasil Keluaran Citra Logo *Watermark* dengan Intensitas Derau 0.08****Tabel A. 10 Hasil Keluaran Skenario 4 (bagian 1)**

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.bmp	21.8479	3.295963 seconds.
2.	DS_3096.bmp	19.8584	3.459605 seconds.
3.	DS_12003.bmp	22.1030	3.492575 seconds.
4.	DS_15004.bmp	21.7460	3.269495 seconds.

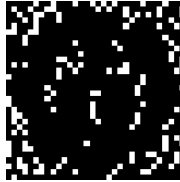
Tabel A. 11 Hasil Keluaran Skenario 4 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
5.	DS_27059.bmp	22.2008	3.578022 seconds.
6.	DS_35008.bmp	22.0661	3.220521 seconds.
7.	DS_35010.bmp	21.8930	3.146601 seconds.
8.	DS_35070.bmp	21.6468	4.290279 seconds.
9.	DS_38092.bmp	21.9601	3.303764 seconds.
10.	DS_42049.bmp	21.8828	3.170963 seconds.
11.	DS_42044.bmp	21.0385	3.546604 seconds.
12.	DS_376001.bmp	22.0121	3.464877 seconds.
13.	DS_42078.bmp	20.8939	3.179290 seconds.
14.	DS_45077.bmp	22.2268	3.344892 seconds.
15.	DS_46076.bmp	21.8252	3.306208 seconds.
16.	DS_55073.bmp	21.8562	3.251597 seconds.
17.	DS_60079.bmp	22.1105	3.470164 seconds.
18.	DS_66075.bmp	22.1761	3.376193 seconds.
19.	DS_67079.bmp	21.3588	3.260851 seconds.
20.	DS_69015.bmp	21.5166	3.525267 seconds.
21.	DS_76002.bmp	21.7717	3.380036 seconds.
22.	DS_78019.bmp	22.0956	3.140639 seconds.
23.	DS_85048.bmp	22.0722	3.482147 seconds.
24.	DS_95006.bmp	21.9284	3.247440 seconds.
25.	DS_97017.bmp	21.9501	3.185752 seconds.
26.	DS_100075.bmp	22.3378	3.163005 seconds.
27.	DS_100080.bmp	21.5874	3.193777 seconds.
28.	DS_101085.bmp	21.9306	3.208044 seconds.
29.	DS_102061.bmp	22.3081	3.227500 seconds.
30.	DS_108005.bmp	22.0441	3.241768 seconds.
31.	DS_118020.bmp	22.2021	3.207030 seconds.
32.	DS_124084.bmp	21.7549	3.494918 seconds.
33.	DS_140075.bmp	21.9359	3.267537 seconds.
34.	DS_143090.bmp	21.7860	3.323919 seconds.
35.	DS_144067.bmp	21.7430	3.254156 seconds.
36.	DS_161062.bmp	22.6105	3.275341 seconds.

Tabel A. 12 Hasil Keluaran Skenario 4 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
37.	DS_163062.bmp	22.4582	3.263164 seconds.
38.	DS_172032.bmp	22.5665	3.191200 seconds.
39.	DS_176019.bmp	22.2412	3.608907 seconds.
40.	DS_176035.bmp	21.9786	3.295770 seconds.
41.	DS_176039.bmp	22.1899	3.267743 seconds.
42.	DS_178054.bmp	19.6052	3.142170 seconds.
43.	DS_182053.bmp	22.3855	3.235575 seconds.
44.	DS_183087.bmp	22.6967	3.575238 seconds.
45.	DS_189011.bmp	22.0788	3.343212 seconds.
46.	DS_197017.bmp	22.1082	3.283091 seconds.
47.	DS_210088.bmp	22.4193	3.181924 seconds.
48.	DS_216041.bmp	22.3861	3.202224 seconds.
49.	DS_227040.bmp	20.4970	3.929080 seconds.
50.	DS_239007.bmp	21.6646	3.258807 seconds.
51.	DS_242078.bmp	21.9349	4.306286 seconds.
52.	DS_253027.bmp	22.4959	3.095803 seconds.
53.	DS_254033.bmp	22.5085	3.233988 seconds.
54.	DS_254054.bmp	22.5614	3.175972 seconds.
55.	DS_268002.bmp	22.5317	3.365935 seconds.
56.	DS_306005.bmp	21.5313	3.188547 seconds.
57.	DS_314016.bmp	22.4264	3.251141 seconds.
58.	DS_317080.bmp	21.3664	3.365619 seconds.
59.	DS_351093.bmp	21.1731	3.201617 seconds.
60.	DS_368078.bmp	22.0788	3.768197 seconds.
61.	DS_374020.bmp	22.0368	3.185896 seconds.

Berikut ini lampiran dari pengujian skenario 5 berdasarkan PSNR dan *Running Time*.



Gambar A. 5 Hasil Keluaran Citra Logo *Watermark* dengan Intensitas Derau 0.1

Tabel A. 13 Hasil Keluaran Skenario 5 (bagian 1)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.bmp	18.7895	3.248443 seconds.
2.	DS_3096.bmp	16.9760	3.206216 seconds.
3.	DS_12003.bmp	19.2215	3.196539 seconds.
4.	DS_15004.bmp	18.7194	3.233929 seconds.
5.	DS_27059.bmp	19.2177	3.190102 seconds.
6.	DS_35008.bmp	19.0749	3.235155 seconds.
7.	DS_35010.bmp	19.0041	3.176760 seconds.
8.	DS_35070.bmp	18.5708	3.262632 seconds.
9.	DS_38092.bmp	19.0290	3.310171 seconds.
10.	DS_42049.bmp	18.8307	3.226180 seconds.
11.	DS_42044.bmp	18.0456	4.239078 seconds.
12.	DS_376001.bmp	18.9207	3.137102 seconds.
13.	DS_42078.bmp	17.7699	4.637679 seconds.
14.	DS_45077.bmp	19.2908	3.202081 seconds.
15.	DS_46076.bmp	18.9099	3.929084 seconds.
16.	DS_55073.bmp	18.8439	3.337960 seconds.
17.	DS_60079.bmp	19.2468	3.317122 seconds.
18.	DS_66075.bmp	19.0334	3.232836 seconds.
19.	DS_67079.bmp	18.4371	3.296342 seconds.
20.	DS_69015.bmp	18.6852	3.310761 seconds.

Tabel A. 14 Hasil Keluaran Skenario 5 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
21.	DS_76002.bmp	18.6992	3.261287 seconds.
22.	DS_78019.bmp	19.0404	3.466576 seconds.
23.	DS_85048.bmp	19.0419	3.717618 seconds.
24.	DS_95006.bmp	19.0988	3.276326 seconds.
25.	DS_97017.bmp	18.9526	3.149616 seconds.
26.	DS_100075.bmp	19.4198	3.242909 seconds.
27.	DS_100080.bmp	18.5853	3.275657 seconds.
28.	DS_101085.bmp	18.9275	4.445602 seconds.
29.	DS_102061.bmp	19.1627	3.240334 seconds.
30.	DS_108005.bmp	19.0884	4.208079 seconds.
31.	DS_118020.bmp	19.2257	3.225125 seconds.
32.	DS_124084.bmp	18.5578	3.324676 seconds.
33.	DS_140075.bmp	18.9503	3.331724 seconds.
34.	DS_143090.bmp	18.7304	4.415716 seconds.
35.	DS_144067.bmp	18.6914	3.195580 seconds.
36.	DS_161062.bmp	19.5972	3.235639 seconds.
37.	DS_163062.bmp	19.4709	3.350367 seconds.
38.	DS_172032.bmp	19.6013	3.255683 seconds.
39.	DS_176019.bmp	19.1655	3.385584 seconds.
40.	DS_176035.bmp	18.8353	3.282290 seconds.
41.	DS_176039.bmp	19.1585	4.300171 seconds.
42.	DS_178054.bmp	16.8256	3.268837 seconds.
43.	DS_182053.bmp	19.2693	3.407029 seconds.
44.	DS_183087.bmp	19.4767	3.367380 seconds.
45.	DS_189011.bmp	18.9931	3.226598 seconds.
46.	DS_197017.bmp	19.0327	3.610488 seconds.
47.	DS_210088.bmp	19.4480	3.211827 seconds.
48.	DS_216041.bmp	19.3803	3.302668 seconds.
49.	DS_227040.bmp	17.5518	3.257184 seconds.
50.	DS_239007.bmp	18.6606	3.612671 seconds.
51.	DS_242078.bmp	19.0266	3.280472 seconds.
52.	DS_253027.bmp	19.4674	4.685878 seconds.

Tabel A. 15 Hasil Keluaran Skenario 5 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
53.	DS_254033.bmp	19.5288	3.304723 seconds.
54.	DS_254054.bmp	19.4587	3.825013 seconds.
55.	DS_268002.bmp	19.5813	3.339524 seconds.
56.	DS_306005.bmp	18.5116	3.361649 seconds.
57.	DS_314016.bmp	19.5948	3.889457 seconds.
58.	DS_317080.bmp	18.2374	3.254937 seconds.
59.	DS_351093.bmp	18.1067	3.511946 seconds.
60.	DS_368078.bmp	19.0406	3.253235 seconds.
61.	DS_374020.bmp	19.0053	3.240696 seconds.

Berikut ini lampiran dari pengujian skenario 6 berdasarkan PSNR dan *Running Time*.

Tabel A. 16 Hasil Keluaran Skenario 6 (bagian 1)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.bmp	17.0417	3.570304 seconds.
2.	DS_3096.bmp	15.1229	3.312479 seconds.
3.	DS_12003.bmp	17.4448	3.247521 seconds.
4.	DS_15004.bmp	17.0606	3.411927 seconds.
5.	DS_27059.bmp	17.4653	4.202492 seconds.
6.	DS_35008.bmp	17.2028	3.262358 seconds.
7.	DS_35010.bmp	17.2967	3.319932 seconds.
8.	DS_35070.bmp	16.8071	3.274071 seconds.
9.	DS_38092.bmp	17.1704	3.517741 seconds.
10.	DS_42049.bmp	17.0325	3.709919 seconds.
11.	DS_42044.bmp	16.2706	3.556108 seconds.
12.	DS_376001.bmp	17.1685	3.231630 seconds.
13.	DS_42078.bmp	16.1118	3.578495 seconds.
14.	DS_45077.bmp	17.5694	3.236779 seconds.
15.	DS_46076.bmp	17.2023	3.234440 seconds.
16.	DS_55073.bmp	17.0424	3.192826 seconds.

Tabel A. 17 Hasil Keluaran Skenario 6 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
17.	DS_60079.bmp	17.3272	4.557412 seconds.
18.	DS_66075.bmp	17.4038	3.237915 seconds.
19.	DS_67079.bmp	16.6363	3.174386 seconds.
20.	DS_69015.bmp	16.9059	3.484114 seconds.
21.	DS_76002.bmp	16.9406	3.245049 seconds.
22.	DS_78019.bmp	17.2473	3.699970 seconds.
23.	DS_85048.bmp	17.2540	4.004481 seconds.
24.	DS_95006.bmp	17.2435	3.238443 seconds.
25.	DS_97017.bmp	17.3073	3.232076 seconds.
26.	DS_100075.bmp	17.5850	3.255432 seconds.
27.	DS_100080.bmp	16.8618	3.321063 seconds.
28.	DS_101085.bmp	17.0877	3.288814 seconds.
29.	DS_102061.bmp	17.5691	3.319894 seconds.
30.	DS_108005.bmp	17.3710	3.246441 seconds.
31.	DS_118020.bmp	17.3912	3.621348 seconds.
32.	DS_124084.bmp	16.7013	3.334708 seconds.
33.	DS_140075.bmp	17.1448	3.264426 seconds.
34.	DS_143090.bmp	16.9233	3.324670 seconds.
35.	DS_144067.bmp	16.9442	3.235482 seconds.
36.	DS_161062.bmp	17.8663	3.333072 seconds.
37.	DS_163062.bmp	17.6748	3.249930 seconds.
38.	DS_172032.bmp	17.8558	3.264566 seconds.
39.	DS_176019.bmp	17.4774	3.383467 seconds.
40.	DS_176035.bmp	17.0261	3.305832 seconds.
41.	DS_176039.bmp	17.2840	3.532419 seconds.
42.	DS_178054.bmp	15.0685	3.127742 seconds.
43.	DS_182053.bmp	17.5625	3.261633 seconds.
44.	DS_183087.bmp	17.7134	3.316409 seconds.
45.	DS_189011.bmp	17.2884	3.338612 seconds.
46.	DS_197017.bmp	17.2609	4.404896 seconds.
47.	DS_210088.bmp	17.6755	3.786541 seconds.
48.	DS_216041.bmp	17.7022	3.286885 seconds.

Tabel A. 18 Hasil Keluaran Skenario 6 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
49.	DS_227040.bmp	15.8065	3.379798 seconds.
50.	DS_239007.bmp	16.9733	3.298787 seconds.
51.	DS_242078.bmp	17.1734	3.198703 seconds.
52.	DS_253027.bmp	17.8900	3.281270 seconds.
53.	DS_254033.bmp	17.7195	3.483472 seconds.
54.	DS_254054.bmp	17.8316	3.548854 seconds.
55.	DS_268002.bmp	17.7983	3.272002 seconds.
56.	DS_306005.bmp	16.6715	3.272568 seconds.
57.	DS_314016.bmp	17.7645	4.365024 seconds.
58.	DS_317080.bmp	16.5071	3.364316 seconds.
59.	DS_351093.bmp	16.3461	3.205749 seconds.
60.	DS_368078.bmp	17.1353	3.313284 seconds.
61.	DS_374020.bmp	17.2997	3.265986 seconds.

Berikut ini lampiran dari pengujian skenario 7 berdasarkan PSNR dan *Running Time*.

Tabel A. 19 Hasil Keluaran Skenario 7 (bagian 1)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.bmp	15.7939	3.233671 seconds.
2.	DS_3096.bmp	13.8880	3.206018 seconds.
3.	DS_12003.bmp	16.2058	3.207822 seconds.
4.	DS_15004.bmp	15.7864	3.208616 seconds.
5.	DS_27059.bmp	16.1402	3.700041 seconds.
6.	DS_35008.bmp	15.9715	3.176486 seconds.
7.	DS_35010.bmp	16.0034	3.705568 seconds.
8.	DS_35070.bmp	15.6074	3.335213 seconds.
9.	DS_38092.bmp	15.8816	3.337065 seconds.
10.	DS_42049.bmp	15.7981	3.215232 seconds.
11.	DS_42044.bmp	15.0951	3.339872 seconds.
12.	DS_376001.bmp	15.8966	4.306141 seconds.

Tabel A. 20 Hasil Keluaran Skenario 7 (bagian 2)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
13.	DS_42078.bmp	14.8732	3.290636 seconds.
14.	DS_45077.bmp	16.3199	3.264906 seconds.
15.	DS_46076.bmp	15.9852	3.616766 seconds.
16.	DS_55073.bmp	15.7913	3.278224 seconds.
17.	DS_60079.bmp	16.1614	3.223259 seconds.
18.	DS_66075.bmp	16.0583	3.600733 seconds.
19.	DS_67079.bmp	15.4473	3.624617 seconds.
20.	DS_69015.bmp	15.6181	3.191688 seconds.
21.	DS_76002.bmp	15.7466	3.175904 seconds.
22.	DS_78019.bmp	16.1064	3.235721 seconds.
23.	DS_85048.bmp	16.0953	3.221905 seconds.
24.	DS_95006.bmp	15.9648	3.284463 seconds.
25.	DS_97017.bmp	15.9641	3.250664 seconds.
26.	DS_100075.bmp	16.4274	3.302717 seconds.
27.	DS_100080.bmp	15.5025	3.518302 seconds.
28.	DS_101085.bmp	15.8704	3.332940 seconds.
29.	DS_102061.bmp	16.2426	3.534756 seconds.
30.	DS_108005.bmp	16.0665	4.380084 seconds.
31.	DS_118020.bmp	16.2257	3.300233 seconds.
32.	DS_124084.bmp	15.4549	3.365415 seconds.
33.	DS_140075.bmp	15.9192	3.382155 seconds.
34.	DS_143090.bmp	15.7117	3.255663 seconds.
35.	DS_144067.bmp	15.6594	3.201026 seconds.
36.	DS_161062.bmp	16.6059	3.316512 seconds.
37.	DS_163062.bmp	16.4440	3.273998 seconds.
38.	DS_172032.bmp	16.5944	3.460175 seconds.
39.	DS_176019.bmp	16.2676	3.330595 seconds.
40.	DS_176035.bmp	15.8348	3.231536 seconds.
41.	DS_176039.bmp	16.1139	3.299441 seconds.
42.	DS_178054.bmp	13.7300	3.305558 seconds.
43.	DS_182053.bmp	16.3174	3.261103 seconds.
44.	DS_183087.bmp	16.4401	3.519338 seconds.

Tabel A. 21 Hasil Keluaran Skenario 7 (bagian 3)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
45.	DS_189011.bmp	15.9804	3.291325 seconds.
46.	DS_197017.bmp	16.0309	3.279405 seconds.
47.	DS_210088.bmp	16.4292	3.476459 seconds.
48.	DS_216041.bmp	16.2946	3.208097 seconds.
49.	DS_227040.bmp	14.5805	3.244318 seconds.
50.	DS_239007.bmp	15.6651	3.204537 seconds.
51.	DS_242078.bmp	16.0017	3.414060 seconds.
52.	DS_253027.bmp	16.4987	3.305262 seconds.
53.	DS_254033.bmp	16.5476	3.250863 seconds.
54.	DS_254054.bmp	16.5251	3.288473 seconds.
55.	DS_268002.bmp	16.5933	3.273243 seconds.
56.	DS_306005.bmp	15.3117	3.235622 seconds.
57.	DS_314016.bmp	16.5807	3.209580 seconds.
58.	DS_317080.bmp	15.2170	3.210322 seconds.
59.	DS_351093.bmp	15.1849	3.259027 seconds.
60.	DS_368078.bmp	15.9535	3.226467 seconds.
61.	DS_374020.bmp	16.0513	3.256766 seconds.

Berikut ini lampiran dari pengujian skenario 8 berdasarkan PSNR dan *Running Time*.

Tabel A. 22 Hasil Keluaran Skenario 8 (bagian 1)

No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
1.	Cover_Lena.bmp	14.7957	3.383599 seconds.
2.	DS_3096.bmp	12.9787	3.466761 seconds.
3.	DS_12003.bmp	15.1821	4.079819 seconds.
4.	DS_15004.bmp	14.8309	3.634229 seconds.
5.	DS_27059.bmp	15.2406	3.610032 seconds.
6.	DS_35008.bmp	15.0499	3.352562 seconds.
7.	DS_35010.bmp	15.0740	3.558545 seconds.
8.	DS_35070.bmp	14.6350	3.398049 seconds.

Tabel A. 23 Hasil Keluaran Skenario 8 (bagian 2)



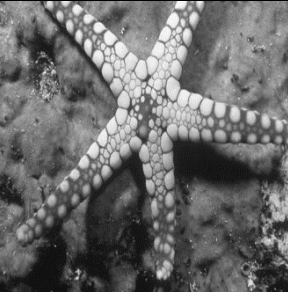
No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
9.	DS_38092.bmp	14.8887	3.564743 seconds.
10.	DS_42049.bmp	14.7414	3.219244 seconds.
11.	DS_42044.bmp	14.0354	4.431932 seconds.
12.	DS_376001.bmp	14.9055	3.309499 seconds.
13.	DS_42078.bmp	13.8520	3.287066 seconds.
14.	DS_45077.bmp	15.3019	3.198007 seconds.
15.	DS_46076.bmp	14.9514	3.342957 seconds.
16.	DS_55073.bmp	14.8431	3.253864 seconds.
17.	DS_60079.bmp	15.2161	3.237155 seconds.
18.	DS_66075.bmp	15.0829	3.334748 seconds.
19.	DS_67079.bmp	14.4216	3.203351 seconds.
20.	DS_69015.bmp	14.6366	3.643346 seconds.
21.	DS_76002.bmp	14.7720	4.362022 seconds.
22.	DS_78019.bmp	15.0846	3.202525 seconds.
23.	DS_85048.bmp	15.0889	3.160690 seconds.
24.	DS_95006.bmp	15.0183	3.206532 seconds.
25.	DS_97017.bmp	15.0508	3.289669 seconds.
26.	DS_100075.bmp	15.4015	3.259008 seconds.
27.	DS_100080.bmp	14.5893	3.227239 seconds.
28.	DS_101085.bmp	14.9154	3.306777 seconds.
29.	DS_102061.bmp	15.1862	3.456785 seconds.
30.	DS_108005.bmp	15.0857	4.498414 seconds.
31.	DS_118020.bmp	15.2184	3.672535 seconds.
32.	DS_124084.bmp	14.6109	3.504965 seconds.
33.	DS_140075.bmp	14.9756	3.346826 seconds.
34.	DS_143090.bmp	14.7788	3.184817 seconds.
35.	DS_144067.bmp	14.6836	3.243461 seconds.

Tabel A. 24 Hasil Keluaran Skenario 8 (bagian 3)




No.	Nama Citra Uji Coba	PSNR	<i>Running Time</i>
36.	DS_161062.bmp	15.6194	4.376581 seconds.
37.	DS_163062.bmp	15.4855	3.335750 seconds.
38.	DS_172032.bmp	15.6252	3.242038 seconds.
39.	DS_176019.bmp	15.2354	3.206317 seconds.
40.	DS_176035.bmp	14.8066	3.308781 seconds.
41.	DS_176039.bmp	15.1565	3.239910 seconds.
42.	DS_178054.bmp	12.7441	3.429777 seconds.
43.	DS_182053.bmp	15.3242	3.210495 seconds.
44.	DS_183087.bmp	15.4887	3.258282 seconds.
45.	DS_189011.bmp	15.0368	3.287697 seconds.
46.	DS_197017.bmp	15.0275	3.192852 seconds.
47.	DS_210088.bmp	15.4795	3.316692 seconds.
48.	DS_216041.bmp	15.3556	3.489112 seconds.
49.	DS_227040.bmp	13.6090	3.327506 seconds.
50.	DS_239007.bmp	14.6902	3.510373 seconds.
51.	DS_242078.bmp	15.0214	3.285019 seconds.
52.	DS_253027.bmp	15.5141	3.265801 seconds.
53.	DS_254033.bmp	15.5409	3.630248 seconds.
54.	DS_254054.bmp	15.5139	3.728867 seconds.
55.	DS_268002.bmp	15.6140	3.242777 seconds.
56.	DS_306005.bmp	14.4143	3.314795 seconds.
57.	DS_314016.bmp	15.5591	3.644098 seconds.
58.	DS_317080.bmp	14.2503	3.256739 seconds.
59.	DS_351093.bmp	14.2123	3.185290 seconds.
60.	DS_368078.bmp	14.9521	3.232422 seconds.
61.	DS_374020.bmp	15.0793	3.221459 seconds.

Berikut ini merupakan data uji citra yang digunakan dalam proses pengujian.




Tabel A. 25 Citra Grayscale Sebagai Citra Cover (bagian 1)

No.	Nama Citra	Citra cover grayscale
1.	Cover_Lena (.bmp, .jpg, .png)	
2.	DS_3096 (.bmp, .jpg, .png)	
3.	DS_12003 (.bmp, .jpg, .png)	




Tabel A. 26 Citra Grayscale Sebagai Citra Cover (bagian 2)

No.	Nama Citra	Citra cover grayscale
4.	DS_15004 (.bmp, .jpg, .png)	
5.	DS_27059 (.bmp, .jpg, .png)	
6.	DS_35008 (.bmp, .jpg, .png)	




Tabel A. 27 Citra Grayscale Sebagai Citra Cover (bagian 3)

No.	Nama Citra	Citra cover grayscale
7.	DS_35010 (.bmp, .jpg, .png)	 A grayscale photograph of a butterfly with dark wings and light-colored markings, perched on a cluster of small white flowers. The background consists of dark, out-of-focus foliage.
8.	DS_35070 (.bmp, .jpg, .png)	 A grayscale photograph of a dragonfly in flight. The dragonfly's body is dark, and its wings are spread, showing a light-colored membrane. The background is a blurred, natural setting.
9.	DS_38092 (.bmp, .jpg, .png)	 A grayscale photograph of a herd of bison in a grassy field. Several bison are visible, some standing and some grazing. The background shows a line of trees under a cloudy sky.



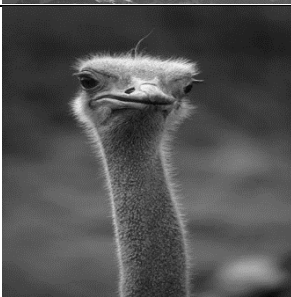
Tabel A. 28 Citra Grayscale Sebagai Citra Cover (bagian 4)

No.	Nama Citra	Citra cover grayscale
10.	DS_42049 (.bmp, .jpg, .png)	
11.	DS_42044 (.bmp, .jpg, .png)	
12.	DS_376001 (.bmp, .jpg, .png)	




Tabel A. 29 Citra Grayscale Sebagai Citra Cover (bagian 5)

No.	Nama Citra	Citra cover grayscale
13.	DS_42078 (.bmp, .jpg, .png)	
14.	DS_45077 (.bmp, .jpg, .png)	
15.	DS_46076 (.bmp, .jpg, .png)	



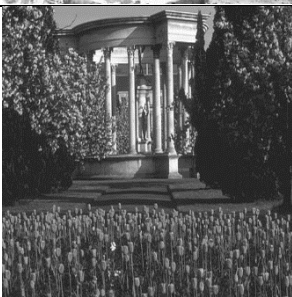
Tabel A. 30 Citra Grayscale Sebagai Citra Cover (bagian 6)

No.	Nama Citra	Citra cover grayscale
16.	DS_55073 (.bmp, .jpg, .png)	
17.	DS_60079 (.bmp, .jpg, .png)	
18.	DS_66075 (.bmp, .jpg, .png)	




Tabel A. 31 Citra Grayscale Sebagai Citra Cover (bagian 7)

No.	Nama Citra	Citra cover grayscale
19.	DS_67079 (.bmp, .jpg, .png)	
20.	DS_69015 (.bmp, .jpg, .png)	
21.	DS_76002 (.bmp, .jpg, .png)	




Tabel A. 32 Citra Grayscale Sebagai Citra Cover (bagian 8)

No.	Nama Citra	Citra cover grayscale
22.	<p style="text-align: center;">DS_78019 (.bmp, .jpg, .png)</p>	
23.	<p style="text-align: center;">DS_85048 (.bmp, .jpg, .png)</p>	
24.	<p style="text-align: center;">DS_95006 (.bmp, .jpg, .png)</p>	




Tabel A. 33 Citra Grayscale Sebagai Citra Cover (bagian 9)

No.	Nama Citra	<i>Citra cover grayscale</i>
25.	DS_97017 (.bmp, .jpg, .png)	 A grayscale photograph of a farm scene. In the foreground, there are several large, cylindrical silos with conical roofs. Behind them is a large, multi-story barn with a gabled roof. The background shows some trees and a cloudy sky.
26.	DS_100075 (.bmp, .jpg, .png)	 A grayscale photograph of two bears standing on a rocky bank next to a body of water. The bear in the foreground is larger and is looking towards the left. The second bear is partially visible behind it.
27.	DS_100080 (.bmp, .jpg, .png)	 A grayscale photograph of a bear sitting in a field of tall grass. The bear is facing the camera and appears to be eating something.



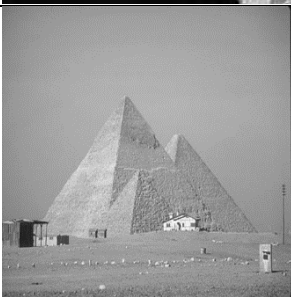
Tabel A. 34 Citra Grayscale Sebagai Citra Cover (bagian 10)

No.	Nama Citra	Citra cover grayscale
28.	DS_101085 (.bmp, .jpg, .png)	
29.	DS_102061 (.bmp, .jpg, .png)	
30.	DS_108005 (.bmp, .jpg, .png)	



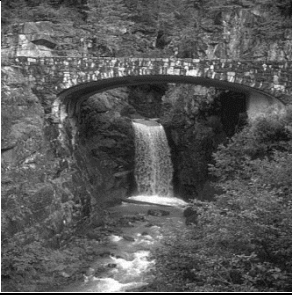
Tabel A. 35 Citra Grayscale Sebagai Citra *Cover* (bagian 11)

No.	Nama Citra	<i>Citra cover grayscale</i>
31.	DS_118020 (.bmp, .jpg, .png)	
32.	DS_124084 (.bmp, .jpg, .png)	
33.	DS_140075 (.bmp, .jpg, .png)	



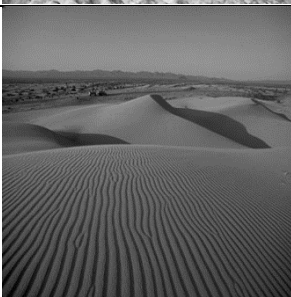
Tabel A. 36 Citra Grayscale Sebagai Citra Cover (bagian 12)

No.	Nama Citra	Citra cover grayscale
34.	DS_143090 (.bmp, .jpg, .png)	
35.	DS_144067 (.bmp, .jpg, .png)	
36.	DS_161062 (.bmp, .jpg, .png)	



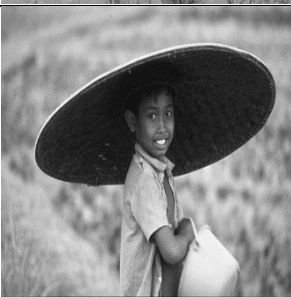
Tabel A. 37 Citra Grayscale Sebagai Citra *Cover* (bagian 13)

No.	Nama Citra	<i>Citra cover grayscale</i>
37.	DS_163062 (.bmp, .jpg, .png)	
38.	DS_172032 (.bmp, .jpg, .png)	
39.	DS_176019 (.bmp, .jpg, .png)	


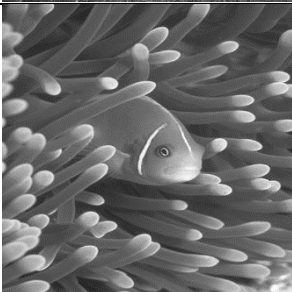

Tabel A. 38 Citra Grayscale Sebagai Citra Cover (bagian 14)

No.	Nama Citra	<i>Citra cover grayscale</i>
40.	DS_176035 (.bmp, .jpg, .png)	
41.	DS_176039 (.bmp, .jpg, .png)	
42.	DS_178054 (.bmp, .jpg, .png)	




Tabel A. 39 Citra Grayscale Sebagai Citra *Cover* (bagian 15)

No.	Nama Citra	<i>Citra cover grayscale</i>
43.	DS_182053 (.bmp, .jpg, .png)	 A grayscale photograph of a dam with multiple arches. A truck is visible on a road or bridge structure above the dam. The background shows a hazy landscape.
44.	DS_183087 (.bmp, .jpg, .png)	 A grayscale photograph of a large bear hugging a smaller cub. They are in a natural, wooded environment.
45.	DS_189011 (.bmp, .jpg, .png)	 A grayscale photograph of a woman holding a large, dark, circular umbrella. She is wearing a light-colored shirt and has a white bag slung over her shoulder. The background is a blurred outdoor setting.


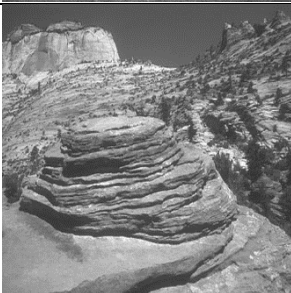
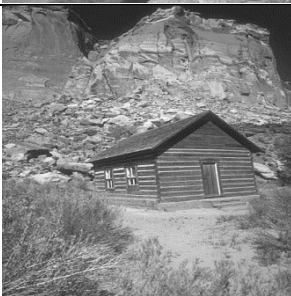
Tabel A. 40 Citra Grayscale Sebagai Citra Cover (bagian 16)

No.	Nama Citra	Citra cover grayscale
46.	DS_197017 (.bmp, .jpg, .png)	
47.	DS_210088 (.bmp, .jpg, .png)	
48.	DS_216041 (.bmp, .jpg, .png)	




Tabel A. 41 Citra Grayscale Sebagai Citra *Cover* (bagian 17)

No.	Nama Citra	<i>Citra cover grayscale</i>
49.	DS_227040 (.bmp, .jpg, .png)	
50.	DS_239007 (.bmp, .jpg, .png)	
51.	DS_242078 (.bmp, .jpg, .png)	




Tabel A. 42 Citra Grayscale Sebagai Citra Cover (bagian 18)

No.	Nama Citra	Citra cover grayscale
52.	DS_253027 (.bmp, .jpg, .png)	
53.	DS_254033 (.bmp, .jpg, .png)	
54.	DS_254054 (.bmp, .jpg, .png)	


Tabel A. 43 Citra Grayscale Sebagai Citra *Cover* (bagian 19)

No.	Nama Citra	<i>Citra cover grayscale</i>
55.	DS_268002 (.bmp, .jpg, .png)	
56.	DS_306005 (.bmp, .jpg, .png)	
57.	DS_314016 (.bmp, .jpg, .png)	


Tabel A. 44 Citra Grayscale Sebagai Citra Cover (bagian 20)

No.	Nama Citra	Citra cover grayscale
58.	DS_317080 (.bmp, .jpg, .png)	
59.	DS_351093 (.bmp, .jpg, .png)	
60.	DS_368078 (.bmp, .jpg, .png)	

Tabel A. 45 Citra Grayscale Sebagai Citra *Cover* (bagian 21)

No.	Nama Citra	Citra <i>cover grayscale</i>
61.	DS_374020 (.bmp, .jpg, .png)	

Tabel A. 46 Citra Biner Sebagai Citra Logo *Watermark*

No.	Citra logo <i>watermark</i>
1.	 <p data-bbox="445 847 706 879">LogoBiner ITS.bmp</p>