

SKRIPSI

APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA BERWARNA RGB



Disusun Oleh :
ZULFIKAR KURNIAWAN
NIM : 0612546

PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013

LEMBAR PERSETUJUAN

**APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK
PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA
BERWARNA RGB**

SKRIPSI

**Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)**



Disusun Oleh :

**ZULFIKAR KURNIAWAN
NIM. 0612546**

**Mengetahui,
Ketua Program Studi Teknik Elektro S-1**

**M. Ibrahim Ashari, ST, MT
NIP. Y. 1030100358**

Diperiksa dan Disetujui,

Dosen Pembimbing I

**Ir. Eko Nurcahyo, MT
NIP. Y. 1028700172**

Dosen Pembimbing II

**Yuli Wahyuni, ST, MT
NIP. P. 1031200456**

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Zulfikar Kurniawan
NIM : 06.12.546
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer Dan Informatika

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 24 Februari 2013

Yang membuat Pernyataan,

Zulfikar Kurniawan
0612546

ABSTRAKSI

APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA BERWARNA RGB

ZULFIKAR KURNIAWAN
(06.12.546)

Dosen Pembimbing :
Ir. Eko Nurcahyo, MT
Yuli Wahyuni, ST, MT

Perkembangan komputer sekarang ini semakin mempermudah masyarakat untuk melakukan pengiriman suatu citra digital dari satu tempat ketempat lain melalui media internet, citra digital yang dikirim akan mudah untuk dimodifikasi atau dirusak dalam perjalanan pengiriman, maka perlu adanya keamanan untuk suatu citra digital agar citra tersebut tidak disalah gunakan oleh pihak yang tidak berwenang.

Untuk keamanan dari sebuah citra digital, watermark adalah salah satu cara yang tepat untuk melindungi suatu citra digital tersebut, maka dari itu dalam tugas akhir ini mengajukan skema Dual Watermark dimana skema ini meliputi dari Embedding/pelekatan, Deteksi Kerusakan dan Pemulihan Watermark. Pada tugas akhir ini metode Dual Watermaking sangat efektif untuk melakukan pelekatan, deteksi dan pemulihan citra berwatermark, metode ini juga membutuhkan sebuah kunci rahasia untuk melakukan pemulihan citra digital yang telah rusak untuk mengembalikan citra digital seperti semula.

Hasil dari penelitian adalah bahwa ukuran citra berpengaruh dalam penyisipan data semakin besar ukuran citra maka semakin lama proses pelekatan data, pengujian berhasil dilakukan pada citra berwarna berukuran 1MB membutuhkan kurang lebih waktu 4-5 detik untuk melakukan proses pelekatan/embedding.

Kata Kunci : Dual Watermark, Watermark, Deteksi, Pemulihan,

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiratMu ya Allah yang telah memberikan rahmat dan hidayahNya, sehinggga penyusun dapat menyelesaikan skripsi dengan judul **"APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA BERWARNA RGB"** ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak terlepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada:

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S-1
4. Bapak Dr. Aryuanto Soetedjo, ST, MT selaku Sekretaris Jurusan Teknik Elektro S-1.
5. Bapak Ir. Eko Nurcahyo, MT selaku Dosen Pembimbing I.
6. Ibu Yuli Wahyuni, ST, MT selaku Dosen Pembimbing II.
7. Ayah dan Ibu kami tercinta serta saudara-saudara kami yang selalu memberikan do'a restu, dorongan dan semangat.
8. Teman-teman dan semua yang telah banyak membantu dalam penyelesaian penyusunan skripsi ini yang tidak dapat kami sebut satu persatu.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penulis mengharap saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini. Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Februari 2013

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	5
2.1 Watermark	5
2.1.1 Sejarah Watermaking	5
2.1.2 Pengenalan	5
2.2 Stuktur Watermaking	8
2.3 Watermaking Pada Citra Digital	10
2.4 Citra Digital	10
2.4.1 Pengertian Citra Digital	10
2.4.2 Jenis Citra Digital	12
2.4.3 <i>Joint Photographic Expert Group (JPEG)</i>	12
2.4.4 <i>Format Portable Network Graphics (.PNG)</i>	13
2.5 Dual Watermaking.....	13
2.6 LSB (Least Significant Bit).....	14
2.7 NetBeans	15
2.8 Pemrograman Beroreintasi Obyek (OOP).....	17
2.8.1 Karakteristik OOP.....	18
2.9 Java	19
2.9.1 Kelebihan Java	19

2.9.2 Kekurangan Java	20
2.10 Java Platform Standart Edition (J2SE)	20
BAB III PERANCANGAN DAN PEMBUATAN APLIKASI	22
3.1 Analisa Masalah	22
3.2 Analisa Kebutuhan	22
3.2.1 Perangkat Keras	22
3.2.2 Perangkat Lunak	23
3.3 Perancangan Dan Pembuatan Aplikasi	23
3.3.1 Embeding Watermark	23
3.3.2 Deteksi Watermark	25
3.3.3 Pemulihan Watermark	26
3.4 Design Aplikasi	27
1. Embeding atau pelekatan watermark	27
2. Deteksi Watermark	28
3. Pemulihan Citra Berwatermark	29
BAB IV PENGUJIAN APLIKASI	30
4.1 Uji Coba Dan Analisa Aplikasi	30
4.2 Prosedur Pengujian Aplikasi	30
4.3 Uji Coba Aplikasi	30
1. Pelekatan Watermark	30
2. Deteksi Watermark	32
3. Pemulihan Citra Berwatermark	33
4.4 Hasil Uji Coba	35
1. Citra berwarna dengan <i>file size</i> 38,9KB	35
2. Citra berwarna dengan <i>file size</i> 576KB	37
3. Citra berwarna dengan <i>file size</i> 1,40MB	39
4.5 Analisis Hasil Uji Coba	41
BAB V PENUTUP	42
5.1 Kesimpulan	42
5.2 Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN	44

DAFTAR GAMBAR

2.1	Steganography dan Cryptography	7
2.2	Proses Embeding Watermark	9
2.3	Proses Deteksi Watermark	9
2.4	Proses Watermaking Pada Citra	10
2.5	Diagram Proses Pelekatan Pesan	14
2.6	Diagram Proses Deteksi Watermark	14
2.7	Diagram Proses Pemulihan Citra Berwatermark	14
3.1	Flowchart Embeding Watermark	24
3.2	Flowchart Deteksi Watermark	25
3.3	Flowchart Pemulihan Citra Berwatermark	26
3.4	Halaman Embeding	27
3.5	Halaman Deteksi Watermark	28
3.6	Halaman Pemulihan Citra Berwatermark	29
4.1	Tampilan <i>embed watermark</i> , pesan dan kunci rahasia	31
4.2	Tampilan citra berhasil dilekatkan/ <i>embedding watermark</i>	31
4.3	Tampilan deteksi <i>watermark</i> pada citra ber <i>watermark</i>	32
4.4	Tampilan deteksi pada citra yang tidak ber <i>watermark</i>	32
4.5	Tampilan citra ber <i>watermark</i> dirusak secara sengaja	33
4.6	Tampilan citra ber <i>watermark</i> yang sudah diperbaiki.....	33
4.7	Tampilan citra gagal diperbaiki.....	34
4.8	Tampilan citra tidak ber <i>watermark</i>	34
4.9	Tampilan <i>Embeding</i>	35
4.10	Tampilan citra berhasil di <i>embedding</i>	35
4.11	<i>Watermark</i> berhasil dideteksi	36
4.12	Citra ber <i>watermark</i> berhasil diperbaiki.....	36
4.13	Tampilan <i>Embeding</i>	37
4.14	Tampilan Citra berhasil di <i>embed</i>	37
4.15	Citra ber <i>watermark</i> berhasil dideteksi	38
4.16	Citra ber <i>watermark</i> yang dirusak dan berhasil diperbaiki	38
4.17	Tampilan <i>embedding</i>	39
4.18	Citra berhasil di <i>embedding</i>	39

4.19	Citra ber <i>watermark</i> berhasil dideteksi.....	40
4.20	Citra ber <i>watermark</i> yang dirusak dan berhasil diperbaiki.....	40

DAFTAR TABEL

2.1	Perbandingan JPEG, GIF, PNG	13
4.1	Hasil Pengujian Terhadap Aplikasi Lain	41

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan komputer memberikan banyak kemudahan dibidang multimedia digital yang memberikan keunggulan dibandingkan multimedia konvensional. Diantaranya adalah kemudahan pengeditan dan penyalinan sebuah citra digital, hal ini juga menimbulkan hal negatif yaitu jika pengeditan dan penyalinan citra digital dilakukan secara ilegal, ini merupakan ancaman terhadap hak kekayaan intelektual dari pemilik citra digital yang bersangkutan. Salah satu cara untuk melindungi hak kekayaan intelektual tersebut dengan cara melakukan *watermarking* pada citra tersebut.

Berdasarkan pada fungsionalitasnya, klasifikasi dari teknik watermark untuk otentifikasi citra dibagi menjadi 3 kategori: (1) *fragile watermarking*, mendeteksi adanya modifikasi pada citra. (2) *semi-fragile watermarking*, mendeteksi dan melokalisir modifikasi pada isi citra. (3) *content-based fragile watermark*, mendeteksi hanya pada perubahan yang signifikan pada citra ketika perijinan content-preserving memproses seperti koding dan scanning. Pada kategori diatas (1) dan (2) dikombinasikan sebagai *fragile watermarking*. Sedangkan kategori (3) dinamakan sebagai *semi-fragile watermarking*.

Pada proposal ini, penulis mengajukan metode digital watermarking yang efisien dan efektif untuk deteksi kerusakan citra dan memulihkan kembali. Metode ini juga merupakan ruang penyimpanan yang sangat sefektif, sepanjang hanya membutuhkan sebuah secret key dan algoritma public chaotic mixing untuk mengembalikan seperti semula area yang rusak.

Pada metode *Dual Watermaking* ini penulis mengembangkan metode Dual Watermaking yg terdahulu, yaitu *Dual Watermaking* pada citra *grayscale* 8-bit dan dikembangkan menjadi *Dual Watermaking* pada citra RGB.

1.2 Rumusan Masalah

Berdasarkan hal tersebut diatas, permasalahan yang dibahas dalam tugas akhir ini yaitu :

1. Implementasi penggabungan atau pelekatan antara citra berwarna dengan *watermark*
2. Implementasi pendeteksian kerusakan pada citra berwarna yang sudah dilekati dengan *watermark*
3. Implementasi proses pemulihan citra ber*watermark* setelah diketahui adanya kerusakan pada citra berwarna ber*watermark*

1.3 Tujuan

Adapun tujuan dari pembuatan skripsi ini adalah untuk mengimplementasikan konsep *dual watermark* pada suatu citra berwarna. Yang terdiri dari beberapa tahapan yaitu antara lain :

1. Proses *Embedding Watermark* adalah memasukkan *watermark* kedalam suatu citra berwarna sehingga diperoleh citra baru yang sudah ber*watermark*.
2. Proses pendeteksian adalah suatu proses dimana citra berwarna yang ber*watermark* dideteksi jika ada kerusakan pada saat pelekatan *watermark*.
3. Proses pemulihan citra ber*watermark* adalah proses dimana untuk memulihkan suatu citra ber*watermark* jika citra berwarna itu rusak.

1.4 Batasan Masalah

Agar permasalahan yang dibahas tidak terlalu meluas, maka ruang lingkup pembahasan adalah sebagai berikut :

1. Program yang dipakai menggunakan NeatBeans 6.9.1 yang dioperasikan pada sistem operasi Windows.
 2. Pengaplikasian *watermark* pada media *image* dengan format JPG.
 3. Ukuran gambar tidak melebihi 1,50MB.
 4. Tujuan dari pembuatan tugas akhir ini adalah hanya untuk mengimplementasikan konsep dari *Dual Watermark*.
-

1.5 Metodologi

Metologi yang digunakan untuk menyelesaikan tugas akhir ini adalah sebagai berikut :

1. Studi Kepustakaan

Pada tahap ini dilakukan pemahaman kepustakaan yang berhubungan dengan topik tugas akhir yaitu konsep *Dual Watermark*, pembuat perangkat lunak dan penggunaan Neatbeans.

2. Desain Sistem

Tahap ini melakukan perancangan sistem dengan menggunakan studi literatur dan konsep teknologi *software* yang dipakai. Pada tahap ini dilakukan desain pada tiap-tiap proses pada sistem.

3. Implementasi

Pada tahap ini dilakukan implementasi dari desain sistem yang sudah dirancang sebelumnya. Perwujudan secara nyata dari konsep yang sudah dibuat menjadi bentuk aplikasi yang siap pakai.

4. Uji Coba dan Evaluasi

Tahap ini dilakukan untuk menguji kehandalan aplikasi dalam menyelesaikan masalah. Dengan menggunakan berbagai variabel, uji coba dilakukan baik untuk menguji kebenaran algoritma maupun kemampuan aplikasi.

5. Penyusunan laporan tugas akhir

Tahap ini dilakukan untuk membuat laporan dari semua dasar teori dan metode yang digunakan serta hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.6 Sistematika Penulisan

Sistematika penulisan yang diuraikan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, tujuan, permasalahan, batasan masalah dan sistematika pembahasan.

BAB II : DASAR TEORI

Membahas teori-teori dasar penunjang perancangan dan pembuatan aplikasi.

BAB III : PERANCANGAN DAN PEMBUATAN APLIKASI

Bab ini berisi tentang perancangan dan pembuatan sistem yang digunakan untuk menjelaskan metode yang dilakukan, dan cara kerja struktur navigasi tersebut.

BAB IV : PENGUJIAN APLIKASI

Bab ini berisi pembahasan hasil pengujian dan analisa.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan pada skripsi ini.

BAB II

LANDASAN TEORI

2.1 Watermark

2.1.1 Sejarah Watermarking

Watermarking sudah ada sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi watermark atau tanda air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan terbentuklah suatu kertas yang berwatermark. Kertas ini biasa digunakan oleh seniman atau sastrawan untuk menulis karya mereka. Kertas yang sudah dibubuhi tanda air tersebut sekaligus dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka.

Ide Watermarking pada data digital (sehingga disebut digital Watermarking) dikembangkan di Jepang pada tahun 1900 dan di Swiss tahun 1993. Digital watermarking semakin berkembang seiring dengan semakin meluasnya penggunaan internet.

2.1.2 Pengenalan

Perkembangan komputer memberikan banyak kemudahan dibidang multimedia digital yang memberikan keunggulan dibandingkan multimedia konvensional. Diantaranya adalah kemudahan pengeditan dan penyalinan sebuah citra digital, hal ini juga menimbulkan hal negatif yaitu jika pengeditan dan penyalinan citra digital dilakukan secara ilegal, ini merupakan ancaman terhadap hak kekayaan intelektual dari pemilik citra digital yang bersangkutan. Beberapa faktor yang membuat data digital (seperti audio, citra, video dan text) banyak digunakan antara lain :

- Mudah diduplikasi dan hasilnya sama dengan aslinya.
- Murah untuk penduplikasian dan penyimpanan.
- Mudah disimpan dan kemudian untuk diolah atau diproses lebih lanjut.
- Serta mudah didistribusikan, baik dengan media disk maupun melalui jaringan internet.

Watermarking merupakan suatu bentuk dari Steganography, yaitu ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain. Watermarking (tanda air) ini agak berbeda dengan tanda air pada uang kertas. Tanda air

pada uang kertas masih terlihat oleh indera manusia (dalam posisi kertas tertentu), tetapi Watermarking pada media digital tak akan dirasakan kehadirannya oleh manusia tanpa alat bantu mesin pengolah digital seperti komputer .

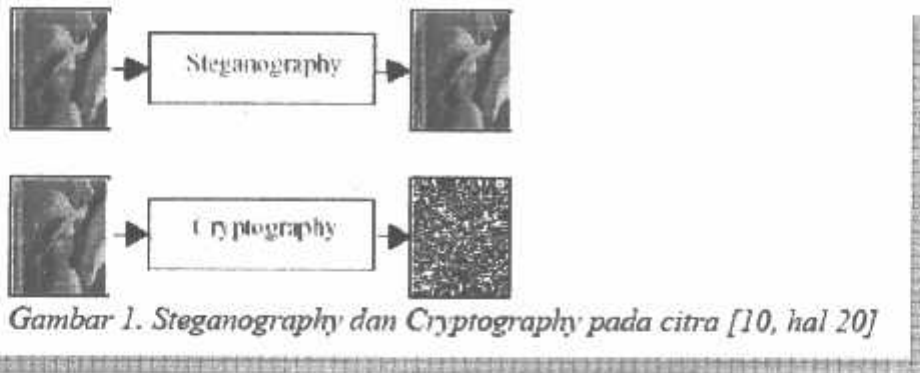
Watermarking ini memanfaatkan kekurangan-kekurangan sistem indera manusia seperti mata dan telinga. Dengan adanya kekurangan inilah, metode Watermarking ini dapat diterapkan pada berbagai data digital. Jadi Watermarking merupakan suatu cara untuk menyembunyikan atau menanam suatu data/informasi tertentu ke dalam suatu data digital lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia.

Istilah *watermarking* ini muncul dari salah satu cabang ilmu yang disebut dengan *steganography*. *Steganography* merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi “rahasia” di dalam suatu informasi lainnya. Perbedaan *steganography* dengan *cryptography* terletak pada bagaimana proses penyembunyian data dan hasil akhir dari proses tersebut. *Cryptography* melakukan proses pengacakan data aslinya sehingga menghasilkan data terenkripsi yang benar-benar acak / seolah-olah berantakan (tetapi dapat dikembalikan ke bentuk semula) dan berbeda dengan aslinya, sedangkan *steganography* menyembunyikan dalam data lain yang akan ditumpanginya tanpa mengubah data yang ditumpanginya tersebut sehingga data yang ditumpanginya sebelum dan setelah proses penyembunyian hampir sama. Dengan kata lain keluaran *steganography* ini memiliki bentuk persepsi yang sama dengan bentuk aslinya, tentunya persepsi disini oleh indera manusia, tetapi tidak oleh komputer atau perangkat pengolah digital lainnya.

Watermarking (tanda air) dapat diartikan sebagai suatu teknik penyembunyian data atau informasi “rahasia” kedalam suatu data lainnya untuk “ditumpanginya” (kadang disebut dengan *host data*), tetapi orang lain tidak menyadari kehadiran adanya data tambahan pada data *host*-nya. Jadi seolah-olah tidak ada perbedaan antara data *host* sebelum dan sesudah proses *watermarking*.

Watermarking ini memanfaatkan kekurangan-kekurangan sistem indera manusia seperti mata dan telinga. Jadi *watermaking* merupakan suatu cara untuk penyembunyian atau penanaman data/informasi tertentu (baik hanya berupa catatan umum maupun rahasia) kedalam suatu data digital lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia (indera penglihatan atau indera pendengaran), dan mampu menghadapi proses-proses pengolahan sinyal digital yang tidak merusak kualitas data yang ter-*watermark* sampai pada tahap tertentu.

Disamping itu data yang ter-*watermark* harus tahan (*robust*) terhadap serangan-serangan baik secara sengaja maupun tidak sengaja untuk menghilangkan data *watermark* yang terdapat didalamnya.



Gambar 1. *Steganography dan Cryptography pada citra [10, hal 20]*

Gambar 2.1 : Steganography dan Cryptogrhapyp

Mutu dari teknik *watermarking* meliputi beberapa parameterparameter utama yang berikut ini :

a. *Fidelity*

Perubahan yang disebabkan oleh tanda (*mark*) semestinya tidak mempengaruhi nilai isi, idealnya tanda harusnya tidak dapat dilihat, sehingga tidak dapat dibedakan antara data yang ter-*watermark* dan data yang asli. Salah satu *trade-off* antara karakteristik *watermarking* yang sangat kelihatan adalah antara *robustness* dengan *fidelity*. Dalam beberapa literatur *fidelity* kadang disebut dengan *invisibility* untuk jenis data citra dan video. Yang dimaksud dengan *fidelity* disini adalah derajat degradasi *host* data sesudah diberi *watermark* dibandingkan dengan sebelum diberi *watermark*.

Biasanya bila *robustness* dari *watermark* tinggi maka memiliki *fidelity* yang rendah, sebaliknya *robustness* yang rendah dapat membuat *fidelity* yang tinggi. Jadi sebaiknya dipilih *trade-off* yang sesuai, sehingga keduanya dapat tercapai sesuai dengan tujuan aplikasi. Untuk *host* data yang berkualitas tinggi maka *fidelity* dituntut setinggi mungkin sehingga tidak merusak data aslinya, sedangkan *host* data yang memiliki *noise* (kualitas kurang) maka *fidelity*nya bisa rendah.

b. Robustness

watermark di dalam *host* data harus tahan terhadap beberapa operasi pemrosesan digital yang umum seperti pengkonversian dari digital ke analog dan dari analog ke digital, dan manipulasi data. Pada *robust watermark*, data disisipkan dengan sangat kuat, sehingga jika ada yang berusaha menghapusnya maka gambar atau suara yang disisipi akan ikut rusak dan tidak punya nilai komersial lagi.

c. Security

Watermarking harus tahan terhadap usaha segaja memindahkan/mencopy *watermark* dari satu multimedia data ke multimedia data lainnya. Pada ketiga kriteria diatas, *fidelity* merupakan kriteria paling tinggi.

2.2 Struktur dari Watermaking

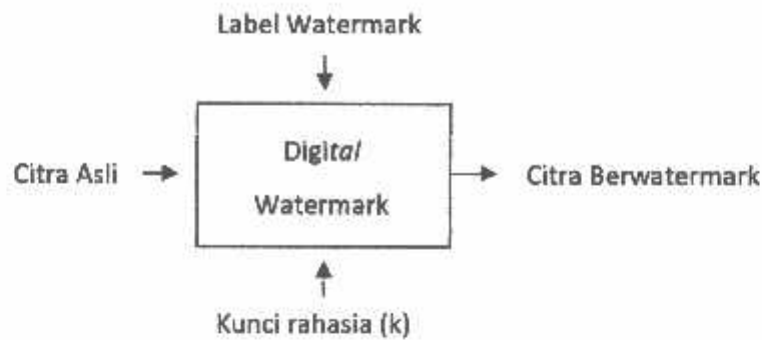
Penerapan *watermarking* pada data digital seperti text, citra, video dan audio, dilakukan langsung pada jenis data digital tersebut (misalnya untuk citra dan video pada domain spasial, dan audio pada domain waktu) atau terlebih dahulu dilakukan tranformasi ke dalam domain yang lain.

Berbagai transformasi yang dikenal dalam pemrosesan sinyal digital seperti: FFT (*Fast Fourier Transform*), DCT (*Discreate Cosine Transform*), DWT (*Discreate Wavelet Transform*), dan sebagainya.

Penerapan *watermaking* pada berbagai domain dengan berbagai transform turut mempengaruhi berbagai parameter penting dalam *watermarking*.

Terdapat 3 sub-bagian *watermarking* yang membentuknya yaitu:

1. Penghasil Label Watermark
 2. Proses penyembunyian Label
 3. Menghasilkan kembali Label Watermark dari data yang ter*watermark*.
-



Gambar 2.2 Proses Embedding Watermark



Gambar 2.3 Proses Deteksi Watermark

Label *watermark* adalah sesuatu data/informasi yang akan kita masukkan ke dalam data digital yang ingin di-*watermark*. Ada 2 jenis label yang dapat digunakan :

a. Text biasa

Label watermark dari text biasanya menggunakan nilai-nilai ASCII dari masing-masing karakter dalam text yang kemudian dipecahkan atas bit-per-bit, kelemahan dari label ini adalah kesalahan pada satu bit saja akan menghasilkan hasil yang berbeda dengan text sebenarnya.

b. Logo atau Citra atau Suara

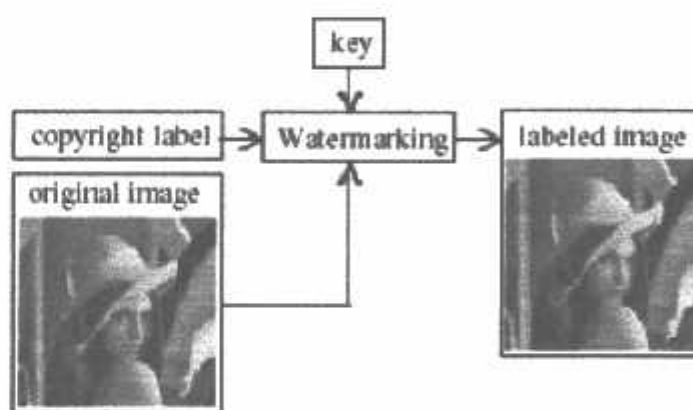
Berbeda dengan text, kesalahan pada beberapa bit masih dapat memberikan persepsi yang sama dengan aslinya oleh pendengaran maupun penglihatan kita, tetapi kerugiannya adalah jumlah data yang cukup besar. *Key* pada gambar diatas digunakan untuk mencegah penghapusan secara langsung watermark oleh pihak tak bertanggung jawab, dengan menggunakan metoda enkripsi yang sudah ada. Sedangkan ketahanan

terhadap proses-proses pengolahan lainnya, itu tergantung pada metoda *watermaking* yang digunakan.

2.3 Watermarking Pada Citra Digital

Terdapat banyak metoda *watermarking* untuk citra digital yang sudah diteliti. Teknik *watermarking* pada image digital dapat diklasifikasikan dalam dua kategori, yaitu teknik domain spasial (*spatial watermark*) dan teknik domain frekuensi (*spectral watermark*).

Pada *watermarking* untuk citra yang dilakukan pada domain spasial, penyisipan dilakukan dengan sedikit mengubah nilai pixel- pixel tertentu



Gambar 3. Proses watermaking pada citra

Gambar 2.4 Proses Watermaking Pada Citra

Sedangkan jika menggunakan domain frekuensi, maka citra tersebut diubah dahulu ke dalam domain transform (biasanya dengan DFT atau DCT) kemudian penyisipan data dilakukan dengan sedikit mengubah nilai koefisien tertentu yang dipilih.

2.4 Citra Digital

2.4.1 Pengertian Citra Digital

Citra digital merupakan citra yang berbentuk array dua dimensi yang terdiri dari blok-blok kecil yang disebut piksel. Piksel merupakan elemen penyusun warna terkecil yang menyusun suatu citra. Citra dibentuk dari kotak-kotak persegi yang teratur

sehingga jarak horizontal dan vertikal antara piksel adalah sama pada seluruh bagian citra. Setiap piksel diwakili oleh bilangan bulat (*integer*) untuk menunjukkan lokasinya dalam bidang citra. Sebuah bilangan bulat juga digunakan untuk menunjukkan cahaya atau keadaan terang gelap piksel tersebut.

Berdasar sifat dari nilai terkuantisasinya, citra digital dapat diklasifikasikan sebagai berikut:

1. Citra Biner

Citra biner adalah citra digital yang tiap pikselnya memiliki dua kemungkinan nilai, yaitu 1 dan 0.

2. Citra *Grayscale*

Citra *grayscale* adalah citra digital yang setiap pikselnya merupakan sampel tunggal, yaitu informasi intensif. Citra jenis ini terbentuk hanya dari warna abu-abu pada tingkatan yang berbeda beda, mulai dari warna hitam pada tingkat intensitas terendah hingga warna putih pada tingkat intensitas tertinggi. Citra ini juga disebut citra hitam putih atau citra monokromatik.

3. Citra Warna

Citra warna adalah citra digital yang setiap pikselnya mengandung informasi warna. Informasi warna ini biasanya dibentuk dari paling sedikit 3 sampel (saluran warna). Saluran warna yang umum dipakai dalam komputer adalah *Red-Green-Blue* (RBG), tetapi dalam konteks lain sering juga digunakan saluran warna lain seperti *Cyan-Magenta-Yellow-Black* (CMYK) atau YCbCr.

Citra berwarna dapat dibagi menjadi beberapa macam yaitu sebagai berikut :

1. *Red-Green-Blue* (RGB) dan *Cyan-Magenta-Yellow-Black* (CMYK)

Pada RBG, tiap elemen warna memiliki nilai dari 0 sampai dengan 255, dimana nilai 0 menyatakan tidak ada elemen warna tersebut pada piksel dan 255 menyatakan nilai maksimum elemen tersebut pada piksel. Model RBG digunakan pada citra yang ditampilkan dilayar monitor sedangkan CMYK digunakan pada citra yang akan dicetak pada printer.

2. *Hue-Saturation-Brightness* (HSB)

Dalam model HSB terdapat 3 karakteristik utama pada warna yaitu:

- *Hue* adalah warna yang dipantulkan dari suatu objek atau yang dipancarkan melalui suatu objek dan sering dinyatakan dengan lingkaran warna standar dari 0 sampai 360 derajat.

- *Saturation* atau yang biasa disebut *chroma* adalah kekuatan warna yang dan dinyatakan dengan presentase dari 0 sampai dengan 100 persen.
- *Brightness* adalah nilai relatif dari gelap-terang dari warna dan biasanya dinyatakan dengan presentase dari 0 (hitam) sampai dengan 100 (putih).

3. *Bitmap*

Pada model ini menggunakan dua nilai (hitam atau putih) untuk mengisi piksel pada citra.

4. *Grayscale*

Model *grayscale* adalah model warna hitam-putih yang tiap piksel memiliki warna dari 0 sampai dengan 255 dimana warna 0 adalah hitam dan warna 255 adalah warna putih.

2.4.2 Jenis Citra Digital

Citra digital dapat dibedakan menjadi 2 tipe yaitu :

1. Tipe *Raster*

Raster merupakan suatu struktur data yang berbentuk dalam persegi-persegi kecil (piksel) yang memiliki warna tersendiri. *Raster* dapat disimpan dalam suatu tipe data citra dengan berbagai format (*.PSD, *.JPG, *.GIF, *.TIF, *.BMP). Kelebihan dari tipe *raster* adalah dapat menampilkan suatu citra secara lebih realistis yang membutuhkan variasi warna yang kompleks. Dan kekurangan dari tipe *raster* adalah apabila diperbesar maka ketajamannya akan berkurang dan terlihat blok-blok piksel.

2. Tipe *Vector*

Vector image adalah citra yang berbentuk vektor atau titik-titik yang telah disusun dan terhubung untuk membentuk suatu objek.

2.4.3 *Joint Photographic Expert Group (.JPEG)*

JPEG merupakan format yang biasa digunakan untuk menyimpan foto. JPEG ini adalah hasil dari kompresi, JPEG merupakan teknik kompresi. Tingkat kompresi dapat diatur menyesuaikan besar file dan kualitas gambar.

Model-model kompresi JPEG :

- *Sequential*
 - *Progressive*
 - *Hierarchical*
-

2.4.4 Format *Portable Network Graphics* (.PNG)

PNG (*Portable Network Graphics*) adalah salah satu format penyimpanan citra yang menggunakan metode pemadatan yang tidak menghilangkan bagian dari citra tersebut (Inggris *lossless compression*).

Untuk keperluan pengolahan citra, meskipun format PNG bisa dijadikan alternatif selama proses pengolahan citra - karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah (sehingga penyimpanan berulang ulang dari citra tidak akan menurunkan kualitas citra)

PNG (Format berkas grafik yang didukung oleh beberapa web browser. PNG mendukung transparansi gambar seperti GIF, berkas PNG bebas paten dan merupakan gambar bitmap yang terkompresi.

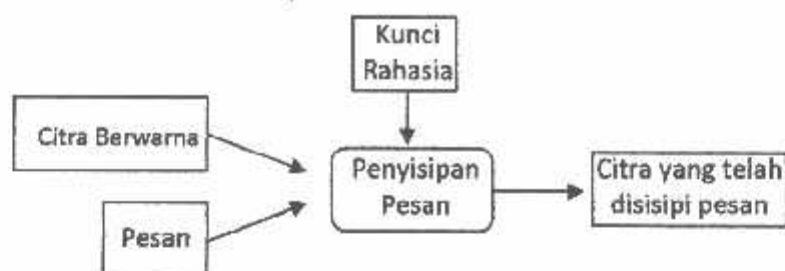
	<i>JPEG</i>	<i>GIF</i>	<i>PNG</i>
Teknik Kompresi	<i>Huffman, DCT</i>	<i>LZW</i>	<i>Deflate</i>
Lossy atau Lossless	<i>Lossy</i>	<i>Lossless</i>	<i>Lossless</i>
Warna	<i>RGB, grayscale</i>	<i>Indexed color</i>	<i>Indexed color, grayscale, RGB</i>
Warna Transparan	Tidak	Ya	Ya

Tabel 2.1 Perbandingan JPEG, GIF, PNG

2.5 Dual Watermaking

Dual Watermark adalah suatu metode watermarking yang dapat digambarkan dalam 3-tahap :

1. *Watermark embeding*, menyembunyikan *watermark* pada suatu citra
2. Deteksi kerusakan *watermark* dimana dalam deteksi ini melakukan pendeteksian pada citra yang sudah ber*watermark*
3. Pemulihan kerusakan pada citra ber*watermark* yaitu melakukan tahap pemulihan pada citra ber*watermark*.



Gambar 2.5 Diagram Proses Pelekatan Pesan



Gambar 2.6 Diagram Proses Deteksi Watermark

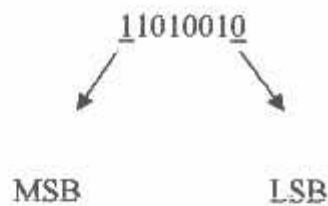


Gambar 2.7 Diagram Proses Pemulihan Citra Berwatermark

2.6 LSB (Least Significant Bit)

Pada metode *Dual Watermark* ini menggunakan metode modifikasi LSB (*Least Significant Bit Modification*) yang merupakan teknik penyembunyian data yang bekerja pada domain spasial atau waktu. Untuk menjelaskan teknik penyembunyian LSB yang dipakai ini kita menggunakan citra digital sebagai *coverttext*. Setiap *pixel* yang ada di dalam file citra berukuran 1 sampai 3 *byte*. Pada susunan bit dalam setiap *byte* (1 *byte* = 8 bit) , ada bit yang paling berarti (*most significant bit* atau *MSB*) dan bit yang paling kurang berarti (*least significant bit* atau *LSB*)

Mengganti bit LSB dengan bit data



LSB = *Least Significant Bit*

MSB = *Most Significant Bit*

Dari contoh *byte* 11010010 diatas bit 1 pertama yang (di garis bawah) adalah bit MSB dan bit 0 terakhir yang digaris bawah adalah bit LSB. Bit yang cocok untuk diganti dengan bit pesan adalah bit LSB, karena modifikasi hanya mengubah nilai *byte* tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya.

Misalkan *byte* tersebut menyatakan warna keabuan tertentu, maka perubahan satu bit LSB tidak mengubah warna keabuan tersebut secara berarti. Lagi pula, mata manusia tidak dapat membedakan perubahan yang kecil.

Misalkan segmen data sebelum perubahan:

00110011101000101110001001101111

Segmen data setelah '0111' disembunyikan:

00110010101000111110001101101111

Untuk memperkuat teknik penyembunyian data, bit-bit data rahasia tidak digunakan mengganti byte-byte yang berurutan, namun dipilih susunan byte secara acak. Misalnya jika terdapat 50 byte dan 6 bit data yang akan disembunyikan, maka byte yang diganti bit LSB-nya dipilih secara acak, misalkan byte nomor 36, 5, 21, 10, 18, 49.

2.7 Netbeans

Netbeans merupakan sebuah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu

aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu debugger.

Netbeans juga digunakan oleh sang programmer untuk menulis, meng-compile, mencari kesalahan dan menyebarkan program netbeans yang ditulis dalam bahasa pemrograman java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat professional desktop, enterprise, web, and mobile applications dengan Java language, C/C++, dan bahkan dynamic languages seperti PHP, JavaScript, Groovy, dan Ruby.

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Dan saat ini pun netbeans memiliki 2 produk yaitu Platform Netbeans dan Netbeans IDE. Platform Netbeans merupakan framework yang dapat digunakan kembali (reusable) untuk menyederhanakan pengembangan aplikasi deskto dan Platform NetBeans juga menawarkan layanan-layanan yang umum bagi aplikasi desktop, mengijinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi.

Fitur-fitur dari Platform Netbeans antara lain:

- Manajemen antarmuka (misal: menu & toolbar)
- Manajemen pengaturan pengguna
- Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
- Manajemen jendela
- Wizard framework (mendukung dialog langkah demi langkah)

Netbeans IDE merupakan sebuah IDE open source yang ditulis sepenuhnya dengan bahasa pemrograman java menggunakan platform netbeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi mobile)

Fitur fitur yang terdapat dalam netbeans antara lain:

1. Smart Code Completion: untuk mengusulkan nama variabel dari suatu tipe, melengkapi keyword dan mengusulkan tipe parameter dari sebuah method.
2. Bookmarking: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
3. Go to commands: fitur yang digunakan untuk jump ke deklarasi variabel, source code atau file yang ada pada project yang sama.
4. Code generator: jika kita menggunakan fitur ini kita dapat meng-generate constructor, setter and getter method dan yang lainnya.
5. Error stripe: fitur yang akan menandai baris yang eror dengan memberi highlight merah.

2.8 Pemrograman Beroreintasi Obyek (OOP)

Object-Oriented Programming (OOP) adalah sebuah pendekatan untuk pengembangan / development suatu software dimana dalam struktur software tersebut didasarkan kepada interaksi object dalam penyelesaian suatu proses/tugas. Interaksi tersebut mengambil form dari pesan-pesan dan mengirimkannya kembali antar object tersebut. Object akan merespon pesan tersebut menjadi sebuah tindakan /action atau metode. Bahasa pemrograman berbasis object menyediakan mekanisme untuk bekerja dengan:

§ kelas dan object

§ methods

§ inheritance

§ polymorphism

§ reusability

Object-oriented programs terdiri dari objects yang berinteraksi satu sama lainnya untuk menyelesaikan sebuah tugas. Seperti dunia nyata, users dari software programs dilibatkan dari logika proses untuk menyelesaikan tugas. Contoh, ketika kamu mencetak sebuah halaman diword processor, kamu berarti melakukan inisialisasi tindakan dengan mengklik tombol printer. Kemudian kamu hanya menunggu respon apakah job tersebut sukses atau gagal, sedangkan proses terjadi internal tanpa kita ketahui. Tentunya setelah kamu menekan tombol printer, maka secara simultan object tombol tersebut berinteraksi dengan object printer untuk menyelesaikan job tersebut.

2.8.1 Karakteristik OOP

Dalam section ini, kita akan melihat beberapa konsep dasar dan term-term yang umum untuk seluruh bahasa OOP

1. *Object*

Object adalah sebuah struktur yang menggabungkan data dan prosedur untuk bekerja bersama-sama. Contohnya Jika kamu ingin kemampuan mencetak dalam aplikasi kamu, kamu harus bekerja dengan sebuah object printer yang bertanggung jawab untuk data serta metode yang digunakan untuk berinteraksi dengan printermu.

2. *Abstraction*

Ketika berinteraksi dengan object-object di dunia ini, sering hanya konsentrasi dengan sebuah bagian dari *properties*nya. Tanpa kemampuan untuk mensarikan/*abstract* atau menyaring untuk dibuang *properties object* yang asing / tidak ada hubungannya, maka akan menemukan kesulitan untuk memproses informasi yang kebanyakan membombarding. Sebagai hasil *abstraction*, ketika 2 orang berbeda berinteraksi dengan *object* yang sama, mereka sering setuju dengan bagian yang berbeda atas atribut. Ketika kami mengendarai mobil, kami perlu tahu kecepatan serta tujuan yang akan dicapai, karena mobil itu otomatis, maka kami tidak perlu tahu berapa RPMs dari mesinnya, jadi kami akan membuang informasi ini. Tapi informasi ini sangat diperlukan oleh mekanik atau pembalap.

3. *Encapsulation*

Encapsulation adalah sebuah proses dimana tidak ada akses langsung ke data yang diberikan, bahkan *hidden*.

4. *Polymorphism*

Polymorphisms adalah kemampuan 2 buah object yang berbeda untuk merespon pesan permintaan yang sama dalam suatu cara yang unik.

5. *Inheritance*

Banyak objects diklasifikasikan menurut hirarki. Contoh, mengklasifikasikan sebuah mobil yang mempunyai karakteristik umumnya mobil, seperti mempunyai ban, mesin, serta body.

Keturunan berikutnya diklasifikasikan dengan atribut umum seperti ukuran, jumlah roda, isi silinder dll atau mengklasifikasikan mereka atas dasar daya angkutnya.

6. *Aggregation*

Aggregation adalah kondisi ketika object berisi gabungan dari object-object yang berbeda dan bekerja bersama. Contoh mesin pemotong rumput terdiri dari object roda, objects mesin, object pisau dll.

2.9 Java

Java adalah bahasa pemrograman yang multi platform dan multi device. Sekali anda menuliskan sebuah program dengan menggunakan Java, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support Java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya.

Aplikasi dengan berbasis Java ini dikompulasikan ke dalam p-code dan bisa dijalankan dengan Java Virtual Machine. Fungsionalitas dari Java ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik.

Platform Java terdiri dari kumpulan *library*, JVM, kelas-kelas loader yang dpaket dalam sebuah lingkungan rutin Java, dan sebuah kompiler, debugger dan kakas lain yang dipaket dalam Java Development Kit (JDK). Java2 adalah generasi yang sekarang berkembang dari *platform* Java. Agar sebuah program Java dapat dijalankan, maka file dengan ekstensi java harus dikompilasi menjadi file *bytecode*. Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (Java Runtime Environment) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi.

Platform Java memiliki tiga buah edisi yang berbeda diantaranya :

1. J2EE (Java2 Enterprise Edition)
2. J2ME (Java2 Micro Edition)
3. J2SE (Java2 Second Edition)

2.9.1 Kelebihan Java

Adapun kelebihan-kelebihan dari java itu sendiri diantaranya adalah :

1. Multiplatform. Java dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda.
 2. OOP atau Object Oriented Programming. Java memiliki library yang lengkap. Library disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal
-

ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan library semakin beragam jika ditambah dengan karya komunitas Java.

2.9.2 Kekurangan java

Ada kelebihan pasti ada juga kekurangan, maka kekurangan dari java ini sendiri adalah penggunaan memori yang cukup banyak, lebih besar daripada bahasa tingkat tinggi sebelum generasi Java.

Namun hal ini memang sesuai dengan fitur beragam yang dimiliki oleh Java. Masalah memori ini juga tidak dialami oleh semua pengguna aplikasi Java. Mereka yang sudah menggunakan perangkat keras dengan teknologi terbaru tidak merasakan kelambatan dan konsumsi memori Java yang tinggi.

Namun apapun kelemahan yang dimiliki Java, faktanya adalah Java merupakan bahasa pemrograman yang populer dan digunakan di seluruh dunia saat ini.

2.10 Java Platform Standard Edition (J2SE)

Banyak digunakan sebagai platform untuk pemrograman dalam bahasa Java. Platform ini digunakan untuk mendeploy sebuah aplikasi. Java SE terdiri dari virtual machine yang digunakan untuk menjalankan program java bersama-sama dengan library atau paket.

J2SE atau *Java 2 Standard Edition* merupakan bahasa pemrograman Java untuk aplikasi desktop yang merupakan *object-oriented programming*. Pada J2SE, terdiri dari dua buah produk yang dikeluarkan untuk membantu dalam membuat aplikasi tanpa tergantung dari *platform* yang digunakan, yaitu :

1. *Java SE Runtime Environment (JRE)*

Java Runtime Environment (JRE) menyediakan perpustakaan, *Java Virtual Machine (JVM)*, dan komponen lain untuk menjalankan *applet* dan aplikasi yang ditulis dengan bahasa pemrograman Java. Selain itu, terdapat dua buah kunci teknologi yang merupakan bagian JRE, yaitu : *Java Plug-in*, yang memungkinkan menjalankan *applet* di browser populer dan *Java Web Start*, yang menyebarkan aplikasi mandiri melalui jaringan. JRE tidak mengandung utilitas seperti *compiler* atau *debugger* untuk mengembangkan *applet* dan aplikasi.

2. *Java Development Kit (JDK)*

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi Java. *JDK* merupakan superset dari *JRE*, berisikan segala sesuatu yang ada di *JRE* ditambahkan *compiler* dan *debugger* yang diperlukan untuk membangun aplikasi.

BAB III

PERANCANGAN DAN PEMBUATAN APLIKASI

Dalam pembuatan perancangan Aplikasi Penggunaan Pendeteksi Dan Pemulihan *Dual Watermark* ini adalah :

3.1 Analisa Masalah

Pada perancangan dan pembuatan aplikasi *dual watermark* ini diterapkan dengan menjalankan program yang dibangun dengan menggunakan Netbeans 6.9.1 yang dimana pada aplikasi ini menyediakan tiga tahap atau proses yang diantaranya adalah *Embeding/pelekatan watermark* yang berupa pesan atau data selain melekatkan pesan pada proses pelekatan ini diminta untuk memberikan kunci rahasia, kemudian Deteksi *watermark* yang dimana proses pendeteksian *watermark* yang sudah dilekati oleh *watermark*, dan tahap atau proses yang terakhir adalah proses Pemulihan citra ber*watermark* dimana proses ini memulihkan citra ber*watermark* jika pada citra tersebut mengalami kerusakan dan pada saat pemulihan diminta untuk memberikan kunci rahasia yang sama dengan kunci rahasia pada saat melekatkan *watermark*.

3.2 Analisa Kebutuhan

Analisa kebutuhan meliputi dua hal yaitu : meliputi hardware (perangkat keras) dan software (perangkat lunak).

3.2.1 Perangkat Keras

Perangkat keras yang digunakan untuk membuat aplikasi penggunaan *dual watermark* untuk pendeteksi kerusakan dan pemulihan pada citra berwarna adalah sebagai berikut :

- a. Tipe : Compaq Presario CQ42
- b. Processor : Pentium(R) DualCore 2,3GHz
- c. Memory : 1GB
- d. Hard Disk : 280GB
- e. VGA : Mobile Intel(R) 4series Express 285MB

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan untuk membuat aplikasi penggunaan *dual watermark* untuk pendeteksi kerusakan dan pemulihan pada citra berwarna adalah sebagai berikut :

a. Sistem operasi Windows Seven

Sistem operasi yang digunakan untuk membuat aplikasi penggunaan *dual watermark* untuk pendeteksi kerusakan dan pemulihan pada citra berwarna dengan NetBeans Windows Seven.

b. NetBeans versi 6.9.1

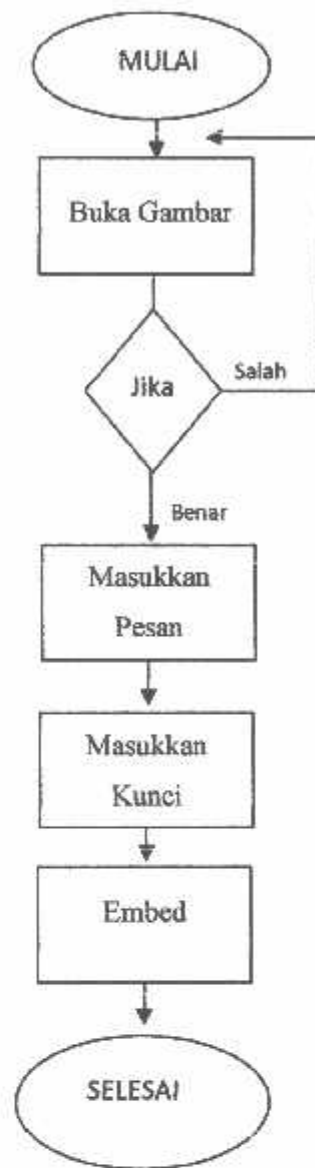
NetBeans versi 6.9.1 adalah software yang digunakan untuk membuat aplikasi penggunaan *dual watermark*.

3.3 Perancangan Dan Pembuatan Aplikasi

Dalam pembuatan perancangan Aplikasi Penggunaan Pendeteksi Dan Pemulihan *Dual Watermark* ini adalah :

3.3.1 Embedding Watermark

Dalam proses *embedding watermark* ini adalah proses untuk memasukkan pesan pada citra berwarna dan memasukkan kunci rahasia untuk keamanan dari isi pesan tersebut dan fungsi dari kunci rahasia ini disamping dari keamanan adalah untuk waktu pemulihan citra yang ber*watermark* apabila citra ber*watermark* itu rusak dengan tidak disengaja maupun disengaja.

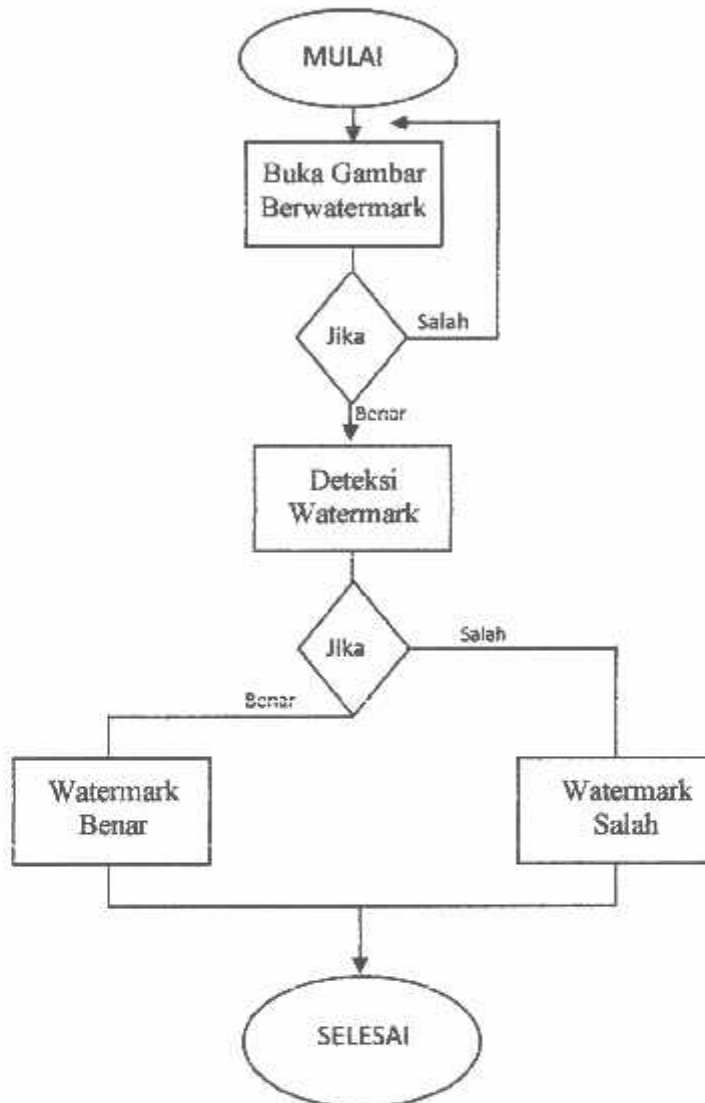


Gambar 3.1 Flowchart Embeding Watermark

Pada proses peletakan *watermark* ini citra yang awalnya berekstensi JPG setelah proses pelekatan citra tersebut akan berekstensi PNG dan ukuran filenya berubah. Hal ini dikarenakan citra berekstensi PNG mempunyai kelebihan untuk menyimpan file dalam bit deph hingga 24 bit serta mampu menghasilkan gambar transparansi (lihat Tabel 2.1)

3.3.2 Deteksi Watermark

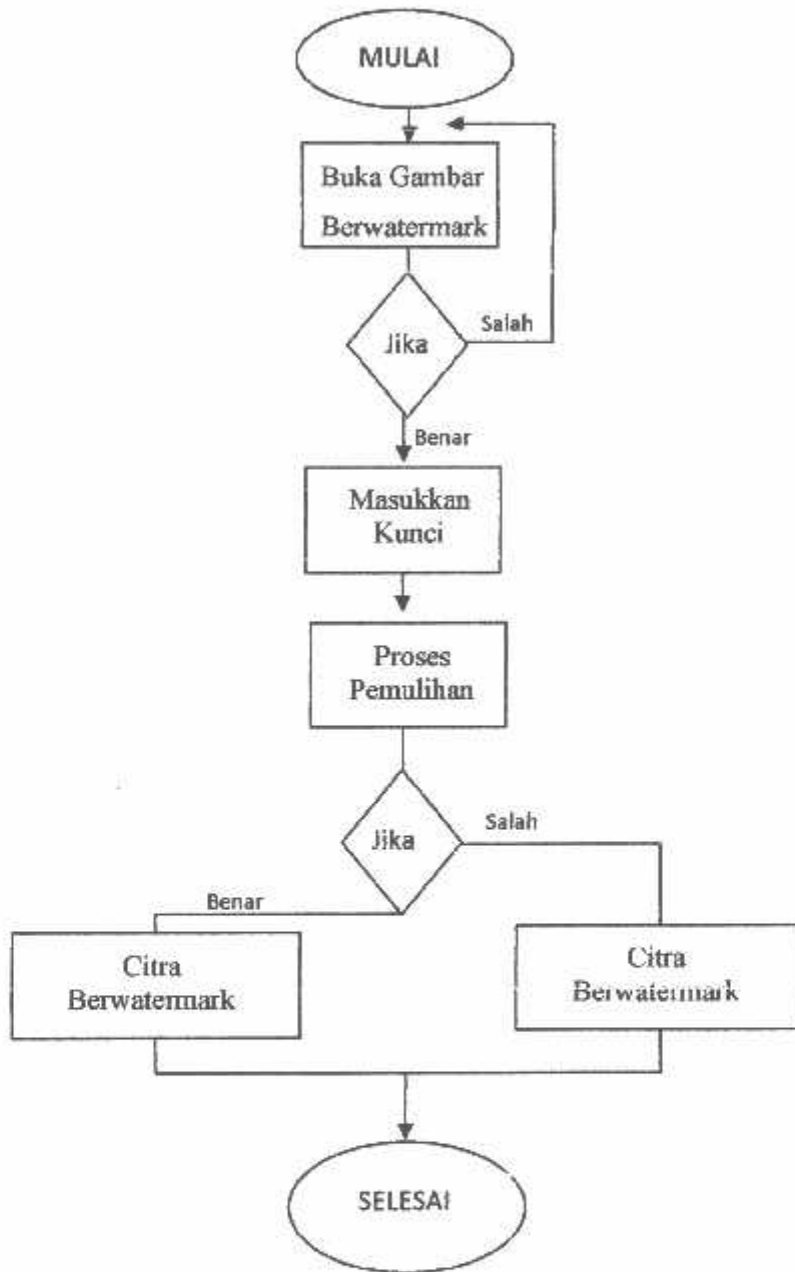
Pada proses deteksi watermark ini adalah mendeteksi adanya *watermark* pada citra berwarna



Gambar 3.2 Flowchart Deteksi Watermark

3.3.3 Pemulihan Watermark

Pada proses pemulihan ini adalah proses untuk memulihkan kembali citra berwatermark apabila citra itu rusak disengaja maupun tidak disengaja.



Gambar 3.3 Flowchart Pemulihan Citra Berwatermark

3.4 Design Aplikasi

Pada tahap pembuatan aplikasi ini ada beberapa tahap untuk memudahkan pengguna untuk melakukan *embedding watermark*, deteksi *watermark*, dan pemulihan *watermark* diantaranya ada panel *embedding*, deteksi dan pemulihan.

1. Embedding atau pelekatan *watermark*

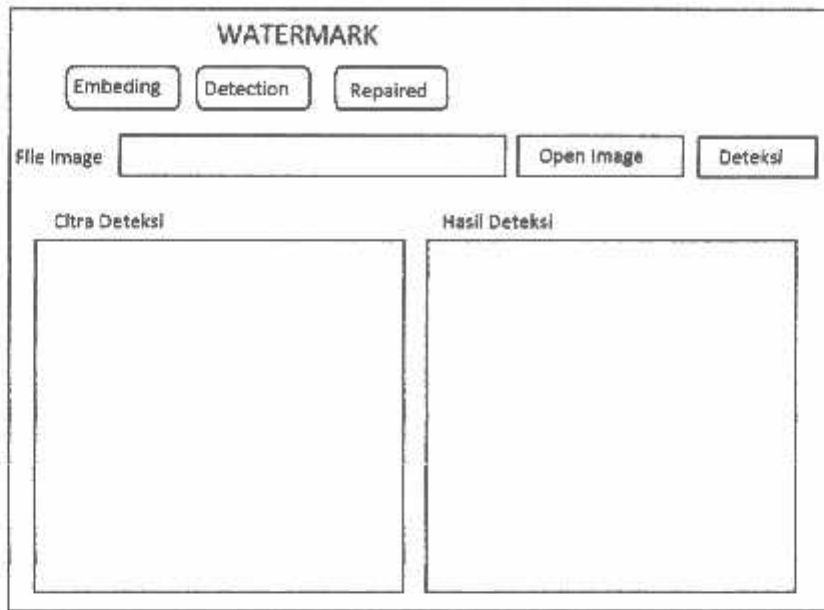
The screenshot shows a web application interface titled "WATERMARK". At the top, there are three buttons: "Embedding", "Detection", and "Repaired". Below these, there is a "File Image" label followed by a text input field and an "Open Image" button. In the center, there is a large empty rectangular box for image display. Below this box, there are two labels: "Pesan :" followed by a text input field, and "Kunci :" followed by another text input field. At the bottom center, there is an "Embed" button.

Gambar 3.4 Halaman Embedding

Pada halaman *embedding* (Gambar 3.4) ini menunjukkan bahwa didalam halaman ini ada :

- *File Image* : untuk membuka file citra yang akan di *embedding* / dilekati *watermark*
- Tampilan untuk menampilkan citra
- Pesan : untuk mengetik pesan yang akan dilekatkan pada citra. Pesan ini yang akan menjadi *watermark* pada citra tersebut.
- Kunci : pada kunci ini adalah memasukkan kunci rahasia yang diketahui oleh orang yang melakukan *embedding*
- Tombol *Embed* : tombol untuk melakukan perintah *embedding*

2. Deteksi Watermark



Gambar 3.5 Halaman Deteksi Watermark

Pada halaman Deteksi Watermark (Gambar 3.8) menunjukkan tampilan pada waktu deteksi *watermark* pada citra ber*watermark*, didalam halaman ini terdapat beberapa tombol diantaranya :

- File Image : untuk membuka citra yang ber*watermark* atau citra yang sudah dilekatkan *watermark*.
- Tombol Deteksi : melakukan proses pendeteksian *watermark* pada citra ber*watermark*
- Tampilan Citra Deteksi : menampilkan citra ber*watermark* atau citra yang akan dideteksi *watermark*nya.
- Tampilan Hasil Deteksi : menampilkan citra yang sudah dideteksi apakah citra ini ada *watermark*nya apa tidak ada.

3. Pemulihan Citra Berwatermark

The image shows a web application interface for watermark management. The title is "WATERMARK". At the top, there are three buttons: "Embedding", "Detection", and "Repaired". Below these, there are two input fields: "File Image" and "Kunci". To the right of the "File Image" field are two buttons: "Open Image" and "Deteksi". Below the input fields, there are two large empty rectangular boxes. The left box is labeled "Citra Image Watermaking" and the right box is labeled "Citra Image Repaired".

Gambar 3.6 Halaman Pemulihan Citra Berwatermark

Pada halaman Pemulihan (Gambar 3.6) menunjukkan :

- File Image : membuka citra yang *berwatermark*
- Kunci : untuk memasukkan kunci rahasia yang sudah dimasukkan pada citra berwarna
- Tombol Repair : melakukan perintah untuk memulihkan citra yang *berwatermark*.
- Tampilan Citra Image Watermaking : menampilkan citra yang *berwatermark*.
- Tampilan Citra Watermark Repair : menampilkan citra *berwatermark* yang sudah dipulihkan / repair

BAB IV PENGUJIAN APLIKASI

4.1 Uji Coba Dan Analisa Aplikasi

Pada pengujian ini bertujuan untuk mengetahui sejauh mana keberhasilan aplikasi ini.

Pengujian ini juga untuk mengimplementasikan konsep dari *Dual Watermark* yang terdiri dari 3 tahap yaitu :

1. Proses pelekatan / embeding *watermark*
2. Proses deteksi kerusakan citra berwatermark
3. Proses pemulihan citra berwatermark

Uji coba ini bertujuan untuk membuktikan bahwa 3 proses tersebut bisa diimplementasikan.

4.2 Prosedur Pengujian Aplikasi

Pada prosedur pengujian pada aplikasi ini adalah melakukan tiga tahapan yang dimana konsep dari *Dual Watermark* yaitu *Embeding / pelekatan watermark* pada citra yang berformat JPG, Deteksi kerusakan *Watermark* pada citra yang sudah dilekati *watermark*, dan Pemulihan citra yang berwatermark. Citra yang diujikan adalah tiga citra yang mempunyai ukuran file 38,9KB, 576KB dan 1,40MB.

4.3 Uji Coba Aplikasi

Pada uji coba aplikasi ini akan mengimplementasikan metode *Dual Watermark* yang diantaranya adalah :

1. Pelekatan *watermark* adalah proses dimana melekatkan/embed pesan dan kunci rahasia pada citra berwarna



Gambar 4.1 Tampilan *embed watermark*, pesan dan kunci rahasia

Dimana proses pelekatan pada pengujian ini memasukkan pesan yang bertuliskan 'Masukkan Pesan Disini', dan memberikan kunci rahasia dengan angka '123'



Gambar 4.2 Tampilan citra berhasil dilekatkan/*embedding watermark*

Setelah berhasil *embedding watermark* maka citra akan disimpan dengan ekstensi PNG dan *file size* dari citra akan berubah lebih besar dari citra aslinya, dikarenakan ada penambahan pesan yang dilekatkan pada citra berwarna tersebut.

2. Deteksi *Watermark* pada citra berwarna yang sudah dilekati *watermark*



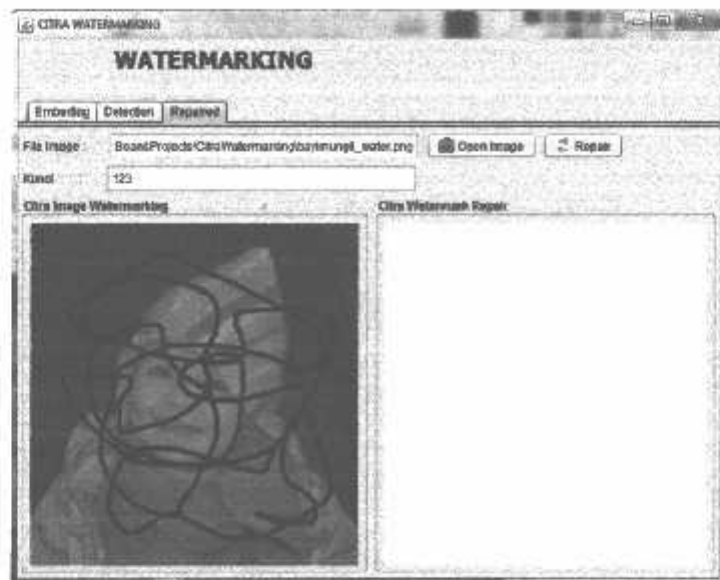
Gambar 4.3 Tampilan deteksi *watermark* pada citra ber*watermark*



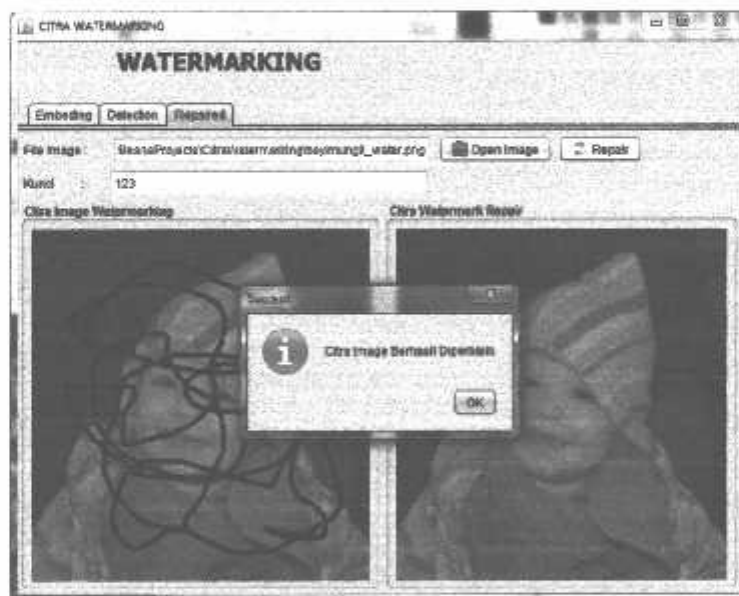
Gambar 4.4 Tampilan deteksi pada citra yang tidak ber*watermark*

Pada pengujian deteksi *watermark* ini bila tampilan menunjukkan (Gambar 4.3) membuktikan bahwa citra yang dideteksi memiliki *watermark* dan pada (Gambar 4.4) menunjukkan bahwa citra yang dideteksi adalah bukan citra ber*watermark*.

3. Pemulihan Citra Berwatermark



Gambar 4.5 Tampilan citra berwatermark dirusak secara sengaja



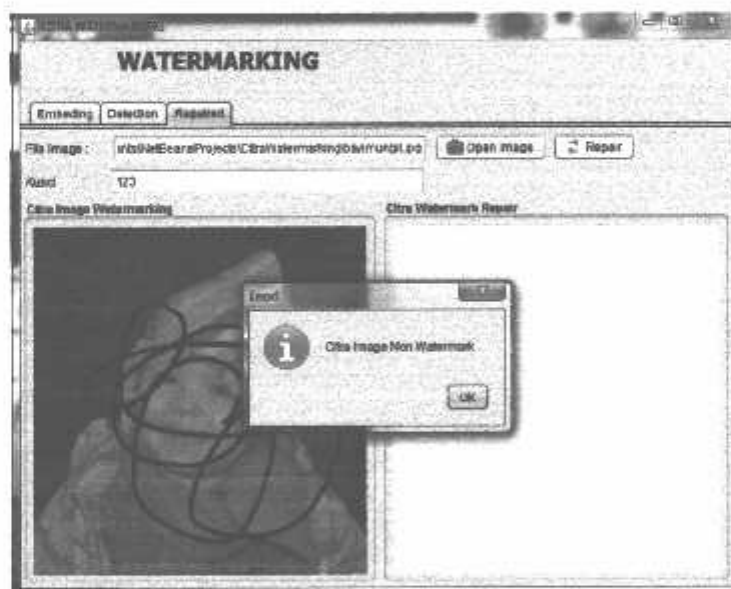
Gambar 4.6 Tampilan citra berwatermark yang sudah diperbaiki

Pada tahap pemulihan ini adalah tahapan untuk memperbaiki kerusakan pada citra yang berwatermark. Pada saat memulihkan citra berwatermark yang rusak, diminta untuk memasukkan kunci rahasia (Gambar 4.5) yang sudah diembed pada waktu *embedding watermark*.

Apabila salah memasukkan kuncinya maka citra yang dihasilkan akan rusak (Gambar 4.7)



Gambar 4.7 Tampilan citra gagal diperbaiki



Gambar 4.8 Tampilan citra tidak berwatermark

Pada pengujian tahap pemulihan apabila salah memasukkan kunci rahasia citra yang diperbaiki akan rusak (Gambar 4.7) dan jika citra tidak berwatermark maka proses pemulihan tidak akan bekerja (Gambar 4.8).

4.4 Hasil Uji Coba

Berikut ini adalah hasil uji coba fungsional yang diujikan pada 3 citra berwarna yang berbeda *file sizenya*, diantaranya adalah :

1. Citra berwarna dengan *file size* 38,9KB



Gambar 4.9 Tampilan *Embedding*

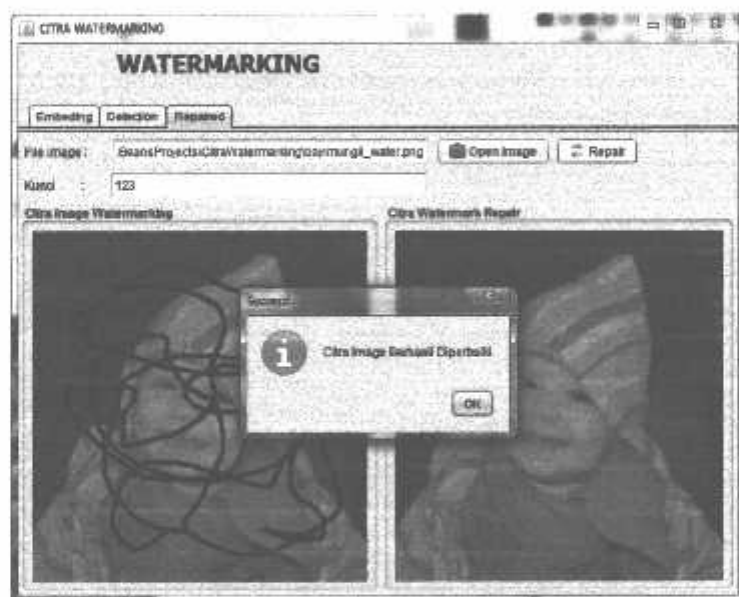


Gambar 4.10 Tampilan citra berhasil diembeding

Pada uji coba embed/pelekatan pada citra berwarna dengan *file size* 38,9KB ini citra berhasil diembed dan *file size* dari citra berwatermark ini berubah menjadi 137KB dan ekstensinya berubah menjadi PNG.

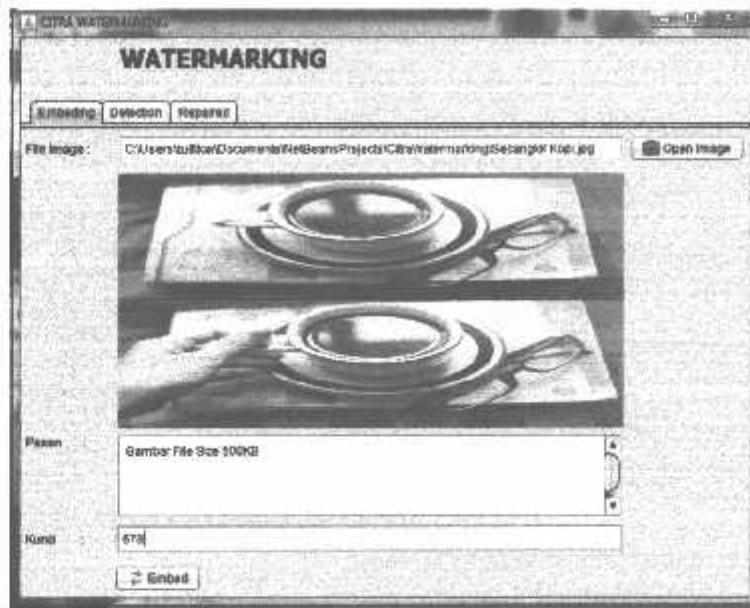


Gambar 4.11 *Watermark* berhasil dideteksi



Gambar 4.12 *Citra berwatermark* berhasil diperbaiki

2. Hasil uji coba yang kedua adalah pengujian pada citra berwarna yang memiliki *file size* 576KB

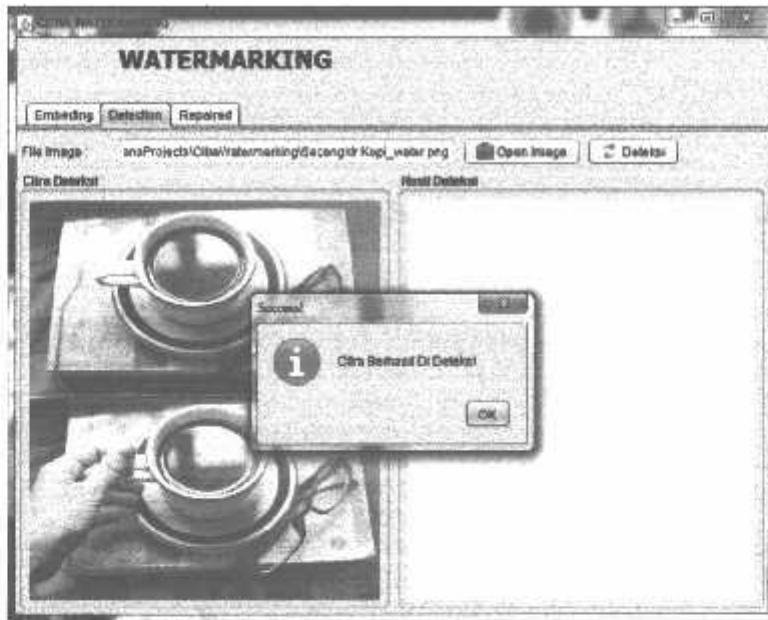


Gambar 4.13 Tampilan *Embedding*



Gambar 4.14 Tampilan Citra berhasil diembed

Pada uji coba embed/pelekatan pada citra berwarna dengan *file size* 576KB ini citra berhasil diembed dan *file size* dari citra berwatermark ini berubah menjadi 1,47MB dan ekstensinya berubah menjadi PNG.



Gambar 4.15 Citra berwatermark berhasil dideteksi



Gambar 4.16 Citra berwatermark yang dirusak dan berhasil diperbaiki

3. Hasil uji coba yang ketiga adalah pengujian pada citra berwarna dengan *file size* 1,40MB



Gambar 4.17 Tampilan *embedding*

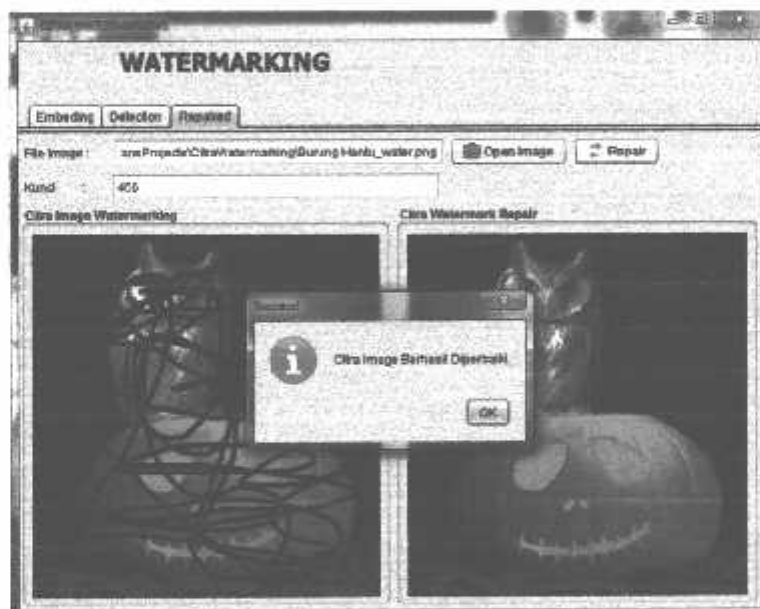


Gambar 4.18 Citra berhasil diembeding

Pada uji coba embed/pelekatan pada citra berwarna dengan *file size* 1,40MB ini citra berhasil diembed dan *file size* dari citra berwatermark ini berubah menjadi 4,99MB dan ekstensinya berubah menjadi PNG.





Gambar 4.19 Citra berwatermark berhasil dideteksi



Gambar 4.20 Citra berwatermark yang dirusak dan berhasil diperbaiki

4.5 Analisis Hasil Uji Coba

Pada analisa uji coba kali ini menguji coba pemulihan pada citra berwarna yang dirusak secara sengaja dengan menggunakan aplikasi lain.

Aplikasi	Hasil Pemulihan
Paint	
Photoshop CS 4	

Tabel 4.1 Hasil Pengujian Terhadap Aplikasi Lain

Hasil uji coba pada tabel 4.1 citra berwarna yang dirusak secara sengaja menggunakan aplikasi lain membuktikan bahwa citra tersebut tidak bisa dipulihkan menggunakan aplikasi ini.

Dapat disimpulkan bahwa saat pengrusakan hanya bisa dirusak menggunakan aplikasi ini dan dipulihkan kembali dengan aplikasi ini.

BAB V PENUTUP

5.1 Kesimpulan

Dari beberapa hasil pengujian yang telah dilakukan dengan menggunakan personal komputer dapat diambil kesimpulan sebagai berikut :

1. Implementasi *Dual Watermark* yang telah dibuat hasilnya dapat diimplementasikan dengan baik diujikan pada citra berwarna yang memiliki *file size* 38,9KB, 576KB dan 1,40MB
2. Citra yang telah diembed/dilekati *watermark file size* nya akan berubah.
3. Ekstensi citra yang awalnya JPG setelah diembed/dilekati *watermark* ekstensinya berubah menjadi PNG.
4. Untuk aspek keamanan, *watermarking* hanya bisa dipulihkan dengan menggunakan kunci rahasia yang digunakan saat pelekatan *watermark*.
5. Citra yang dirusak menggunakan aplikasi lain (misal : Paint, Photoshop) tidak bisa dipulihkan kembali menggunakan aplikasi ini.

5.2 Saran

Dual Watermarking ini masih memiliki keterbatasan yang nantinya dapat dikembangkan, berikut ini adalah saran yang diberikan untuk pengembang selanjutnya, antara lain :

1. Aplikasi *Dual Watermark* bisa ditambahkan untuk proses *ekstrak* citra yang berwatermark.
2. Aplikasi *Dual Watermark* bisa dikembangkan dengan bahasa pemrograman yang lain.

DAFTAR PUSTAKA

- [1] Munir, R, *Stenografi dan Watermaking*, Informatika, Bandung, 2004.
- [2] Armadani, Robi, *Deteksi Kerusakan Dan Pemulihan Citra Dengan Dual Watermark*, Skripsi S-1, Istitut Teknologi Sepuluh Nopember, Surabaya, 2009.
- [3] Nishom, M, *Pengertian NetBeans*, <http://www.isomwebs.com/2012/pengertian-netbeans/>, 5 September 2012.
- [4] Octaviano, S., *Aplikasi Steganografi menggunakan Metode Pensinyalan LSB (Least Significant Bit) pada File Gambar dengan Format BMP*, Skripsi S-1, Universitas Diponegoro, Semarang, 2004.
- [5] Vicky, *Pengertian pemrograman Java – Kelebihan dan Kekurangan*, <http://belajar-komputer-mu.com/pengertian-pemrograman-java-kelebihan-dan-kekurangan/>, 1 September 2012.
- [6] Saputra, H., Hadi, M. Zen Samsono, Syahroni, N., *Implementasi Algoritma Steganografi Embedding Dengan Metode Least Significant Bit (LSB) Insertion Dan Huffman Coding Pada Pengiriman Pesan menggunakan Media MMS Berbasis J2ME*, Teknik Komunikasi, Politeknik Elektronika Negri Surabaya-Institut Sepuluh Nopember, Surabaya,



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAM TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

BNI (PERSERO) MALANG : Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
BANK NIAGA MALANG : Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Zulfikar Kurniawan
NIM : 06.12.546
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika S-1
Judul Skripsi : APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK
PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA
BERWARNA RGB

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 19 Februari 2013
Dengan Nilai : 77,7 (B+) ✓

Anggota Ujian,

Penguji Pertama

M. Ibrahim Ashari, ST, MT
NIP. P. 1030100358

Penguji Kedua

Bima Aulia Firmandani, ST

Panitia Ujian Sripsi,

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP. P. 1030100358

Sekretaris Majelis Penguji

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP. Y. 1030800417



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Program Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Informatika, maka perlu adanya perbaikan Skripsi untuk mahasiswa :

Nama : Zulfikar Kurniawan
NIM : 06.12.546
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika
Masa Bimbingan : Semester Ganjil 2012 -2013
Judul Skripsi : APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA BERWARNA RGB

Tanggal	Uraian	Paraf
Penguji Kedua 19 Februari 2013	• Pengujian terhadap citra yang rusak berdasarkan isi watermark	

Mengetahui,

Dosen Pembimbing I

Ir. Eko Nurcahyo, MT
NIP. Y. 1028700172

Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP. P. 1031200456

Dosen Penguji,

Penguji Pertama

M. Ibrahim Ashari, ST, MT
NIP. Y. 1030100358

Penguji Kedua

Bima Aulia Firmandani, ST



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer & Informatika**

1.	Nim	: 0612546		
2.	Nama	: ZULFIKAR KURNIAWAN		
3.	Konsentrasi Jurusan	: Teknik Komputer & Informatika		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	05 Juni 2012	09 00	III.1.1	
5.	Judul proposal yang diseminarkan Mahasiswa	APLIKASI PENDETEKSI KERUSAKAN DAN PEMULIHAN WATERMARK PADA CITRA BERWARNA BERBASIS MATLAB		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan :			
8.	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	 (M. Ibrahim H. S. S. S.)	 (M. H. S. S. S.)	 (M. H. S. S. S.)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs		
Ir. Yusuf Ismail Nakhoda, MT NIP. Y. 1018800189	Pembimbing I	Pembimbing II		
	 (.....)	 (.....)		



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : ZULFIKAR KURNIAWAN
 NIM : 0612546
 Semester : 2
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : ~~TEKNIK ELEKTRONIKA~~
~~TEKNIK ENERGI LISTRIK~~
 TEKNIK KOMPUTER DAN INFORMATIKA
 TEKNIK KOMPUTER
 TEKNIK TELEKOMUNIKASI
 Alamat : Perum. GFA no. 19 Malang

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....) ⁻²
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro



 (..... Ahmad Fauz) ²

Malang,201

Pemohon


 (ZULFIKAR KURNIAWAN)

Disetujui
 Ketua Jurusan Teknik Elektro


 Ir. Yusuf Ismail Nakhoda. MT
 NIP. Y. 1018800189

Mengetahui
 Dosen Wali


 (AHMAD FAUZ ST.)

Catatan :

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. $102 \times 306 / 138 = 2,85$
2.



DAFTAR PRESTASI AKADEMIK PRAKTIKUM
 KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

Nama Mahasiswa	:	ZULFIKAR KURNIAWAN
NIM	:	0612546
Tempat, Tanggal Lahir	:	SIDARSO, 13 Juli 1987
Jenjang	:	Strata 1 (S1)
Fakultas	:	Teknologi Industri
Jurusan / Program Studi	:	Teknik Elektro
Konsentrasi	:	Teknik Komputer dan Informatika

Praktikum Laboratorium	Kode	Nama Praktikum	SKS	Nilai
I	EL-2215 26	Fisika	1	A
		Rangkaian Listrik		A
		Rangkaian Logika dan Digital		B+
		Dasar Komputer dan Pemrograman		B+
II	EL-4216 110	Dasar Elektronika	1	Pat.
		Dasar Sistem Telekomunikasi		B+
		Mikrokontroler		B+
		Sistem Pengukuran		B+
III	EL-5316 110	Dasar Sistem Kendali	1	A
		Basis Data		A
		Administrasi Jaringan		B+
IV	EL-6317 110	Sistem Operasi	1	B
		Pemrograman Internet		B+
		Pemrograman Objek		B+
V	EL-7318 110	Rekayasa Perangkat Lunak: Sistem Informasi	1	B
		Peripheral dan Antar Muka		A
		Pemrosesan Sinyal Digital		B+
		Multimedia		B+
		Pemrograman Jaringan		B+

55 0

393 --
 138

Malang, _____

Recording
 Jurusan Teknik Elektro S1

Puji Handayani



FORMULIR BIMBINGAN SKRIPSI

m : 0612546
nama : Zulfikar Kurniawan
asas bimbingan : Semester Genap 2011-2012
judul : APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA BERWARNA RGB

Tanggal	Uraian	Paraf Pembimbing
10 Sept 2012	Bab I	Euf
20 Sept 2012	Bab II	Euf
10 Okt 2012	Bab II	Euf
20 Okt 2012	Bab III	Euf
30 Okt 2012	Bab III	Euf
10 Nov 2012	Pengujian Aplikasi	Euf
15 Nov 2012	Bab IV	Euf
10 Desember 2012	Kesimpulan, Saran	Euf
15 Desember 2012	Acc Laporan	Euf

Malang,

Dosen Pembimbing


Ir. Eko Nurcahyo, MT

NIP. X. 1028700172



PROGRAM STUDI TEKNIK ELEKTRO S-1

**FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

FORMULIR BIMBINGAN SKRIPSI

m : 0612546

nama : Zulfikar Kurniawan

semester : Semester Genap 2011-2012

judul : APLIKASI PENGGUNAAN DUAL WATERMARK UNTUK PENDETEKSI KERUSAKAN DAN PEMULIHAN PADA CITRA BERWARNA RGB

Tanggal	Uraian	Paraf Pembimbing
14/05/2012	Uraian I	
15/05/2012	Uraian II, Uraian III, Uraian IV, Uraian V, Uraian VI	
16/05/2012	Uraian VII, Uraian VIII	
17/05/2012	Uraian IX, Uraian X	
18/05/2012	Uraian XI, Uraian XII, Uraian XIII	
19/05/2012	Uraian XIV, Uraian XV	
20/05/2012	Uraian XVI, Uraian XVII	
21/05/2012	Uraian XVIII, Uraian XIX	
22/05/2012	Uraian XX, Uraian XXI	
23/05/2012	Uraian XXII, Uraian XXIII	

Malang,
Dosen Pembimbing

Yuli Wahyuni, ST, MT
NIP.P. 1031200456



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karangrejo, Km 2 Telp. (0341) 417638 Fax. (0341) 417634 Malang

Nomor Surat : ITN/340/EL-FIU/2012

Lampiran : -

Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu Ir. Eko Nurcahyo, MT
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **ZULFIKAR KURNIAWAN**
Nira : **0612546**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil tahun Akademik 2012 - 2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Direktur Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP. Y. 1018800189



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK JIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Jalan Pembangunan No. 1
 BANK KUALA MALANG

Kampus I : C. Bandung, Jl. Raya No. 2 Telp. (0341) 551133 (Hunting) Fax. (0341) 553015 Malang 65145
 Kampus II : Raya Kalitrem, Jln. 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-PROBL-FTE-2012
 Ampiran : -
 Perihal : Bimbingan Skripsi

Cepada : Yth Bapak/Ibu Yuli Wahyuni, ST, MT
 Dosen Teknik Elektro S-1
 ITN MALANG

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : ZULFIKAR KURNIAWAN
 Nim : 412546
 Fakultas : Teknologi Industri
 Program Studi : Teknik Elektro S-1
 Konsentrasi : Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara selama masa waktu :

" Semester Ganjil Tahun Akademik 2012 - 2013 "

Demiikian agar meklum dan atas perhatian serta bantuannya kami sampaikan terima kasih



Mengetahui

Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus I : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

Jumlah : 1 (satu) berkas
Pembimbing Skripsi

Untuk : Yth. Bapak /Ibu Ir. Eko Nurcahyo, MT
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **ZULFIKAR KURNIAWAN**
Nim : **0612546**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"APLIKASI PENDETEKSI KERUSAKAN DAN PEMULIHAN WATERMARK PADA CITRA BERWARNA BERBASIS MATLAB"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Dua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189

Hormat Kami

ZULFIKAR KURNIAWAN

NIM. 0612546



Jumlah : 1 (satu) berkas

Pembimbing Skripsi

Kepada : Yth. Bapak/Ibu **Yuli Wahyuni, ST, MT**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan di bawah

Nama : **ZULFIKAR KURNIAWAN**
Nim : **0612546**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

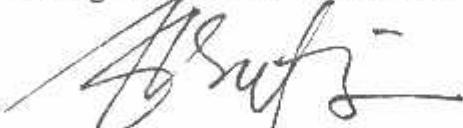
Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"APLIKASI PENDETEKSI KERUSAKAN DAN PEMULIHAN WATERMARK PADA CITRA BERWARNA BERBASIS MATI AB"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Dua Program Studi Teknik Elektro S-1


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Hormat Kami


ZULFIKAR KURNIAWAN
NIM. 0612546



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : ZULFIKAR KURNIAWAN
Nim : 0612546
Semester : XII (Duabelas)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI PENDETEKSI KERUSAKAN DAN PEMULIHAN WATERMARK PADA CITRA BERWARNA BERBASIS MATLAB"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

Ir. Eko Nurcahyo, M.T

NIP.Y. 1028700172

*) Coret yang tidak perlu



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **ZULFIKAR KURNIAWAN**
Nim : **0612546**
Semester : **XII (Duabelas)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI PENDETEKSI KERUSAKAN DAN PEMULIHAN WATERMARK PADA CITRA BERWARNA BERBASIS MATLAB"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami



Yuli Wahyuni, ST, MT

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA
N I M
Perbaikan melalui

Zulfikar K
0611596.

Pengujian terhadap
efisiensi file yang berbeda.

~~Pengujian~~ ~~efisiensi~~ ~~ujian~~
Pengujian terhadap cara yg
baik berdasarkan isi watermark

Malang,

19-02-2013

(_____)

Lampiran

```
package citrawatermarking.engine;

import citrawatermarking.*;
import java.awt.Graphics2D;
import java.awt.image.BufferedImage;
import java.awt.image.DataBufferByte;
import java.awt.image.WritableRaster;
import java.io.File;

import javax.imageio.ImageIO;
import javax.swing.JOptionPane;

/**
 *
 * @author UserXP
 */

public class Watermarking
{
    public Watermarking()
    {

    }

    public boolean encode(String path, String original, String ext1, String watermark, String
message)
    {
        String file_name = image_path(path,original,ext1);
        BufferedImage image_orig = getImage(file_name);

        BufferedImage image = user_space(image_orig);
```

```
        image = add_text(image,message);

        return(setImage(image,new File(image_path(path,watermark,"png")), "png"));
    }

    private String image_path(String path, String name, String ext)
    {
        return path + "/" + name + "." + ext;
    }

    private BufferedImage getImage(String f)
    {
        BufferedImage image = null;
        File file = new File(f);

        try
        {
            image = ImageIO.read(file);
        }
        catch(Exception ex)
        {
            JOptionPane.showMessageDialog(null,
                "Image could not be read!", "Error",JOptionPane.ERROR_MESSAGE);
        }
        return image;
    }

    private boolean setImage(BufferedImage image, File file, String ext)
    {
        try
        {
            file.delete();
            ImageIO.write(image,ext,file);
        }
    }
}
```

```

        return true;
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,
            "File could not be saved!", "Error", JOptionPane.ERROR_MESSAGE);
        return false;
    }
}

private BufferedImage add_text(BufferedImage image, String text)
{
    byte img[] = get_byte_data(image);
    byte msg[] = text.getBytes();
    byte len[] = bit_conversion(msg.length);
    try
    {
        encode_text(img, len, 0);
        encode_text(img, msg, 32);
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,
"Target File cannot hold message!", "Error", JOptionPane.ERROR_MESSAGE);
    }
    return image;
}

private BufferedImage user_space(BufferedImage image)
{
    BufferedImage new_img = new BufferedImage(image.getWidth(), image.getHeight(),
BufferedImage.TYPE_3BYTE_BGR);
    Graphics2D graphics = new_img.createGraphics();

```

```

        graphics.drawRenderedImage(image, null);
        graphics.dispose();
        return new_img;
    }

    private byte[] get_byte_data(BufferedImage image)
    {
        WritableRaster raster = image.getRaster();
        DataBufferByte buffer = (DataBufferByte)raster.getDataBuffer();
        return buffer.getData();
    }

    private byte[] bit_conversion(int i)
    {
        byte byte3 = (byte)((i & 0xFF000000) >>> 24); //0
        byte byte2 = (byte)((i & 0x00FF0000) >>> 16); //0
        byte byte1 = (byte)((i & 0x0000FF00) >>> 8 ); //0
        byte byte0 = (byte)((i & 0x000000FF) );
        return(new byte[]{byte3,byte2,byte1,byte0});
    }

    private byte[] encode_text(byte[] image, byte[] addition, int offset)
    {
        if(addition.length + offset > image.length)
        {
            throw new IllegalArgumentException("File not long enough!");
        }
        for(int i=0; i<addition.length; ++i)
        {
            int add = addition[i];
            for(int bit=7; bit>=0; --bit, ++offset)
            {

```

```
        int b = (add >>> bit) & 1;
        image[offset] = (byte)((image[offset] & 0xFE) | b);
    }
}
return image;
}
}
```
