

**HYBRID IMAGE WATERMARKING RDWT DENGAN
SVD UNTUK PERLINDUNGAN HAK CIPTA PADA
CITRA DIGITAL**

TESIS



Disusun oleh:

MUHAMMAD INNUDDIN

14917150

*Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Forensika Digital*

**MAGISTER TEKNIK INFORMATIKA
PASCASARJANA FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2016**

LEMBAR PENGESAHAN PEMBIMBING

***HYBRID IMAGE WATERMARKING RDWT DENGAN
SVD UNTUK PERLINDUNGAN HAK CIPTA PADA
CITRA DIGITAL***

TESIS



Telah disetujui oleh Dosen Pembimbing

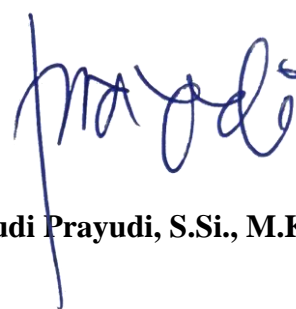
Pada tanggal 31 Desember 2016

Pembimbing I



Dr. Bambang Sugiantoro

Pembimbing II



Yudi Prayudi, S.Si., M.Kom.

LEMBAR PENGESAHAN PENGUJI

***HYBRID IMAGE WATERMARKING RDWT DENGAN
SVD UNTUK PERLINDUNGAN HAK CIPTA PADA
CITRA DIGITAL***

TESIS

Diajukan Oleh :

**MUHAMMAD INNUDDIN
14917150**

Telah dipertahankan di depan dewan penguji
Sebagai salah satu syarat memperoleh Magister Komputer
Program Pascasarjana Universitas Islam Indonesia
Konsentrasi Forensika Digital

Pada tanggal 31 Desember 2016

Penguji I Dr. Bambang Sugiantoro



Penguji II Yudi Prayudi, S.Si., M.Kom.



Penguji III Ahmad Luthfi, S.Kom., M.Kom.

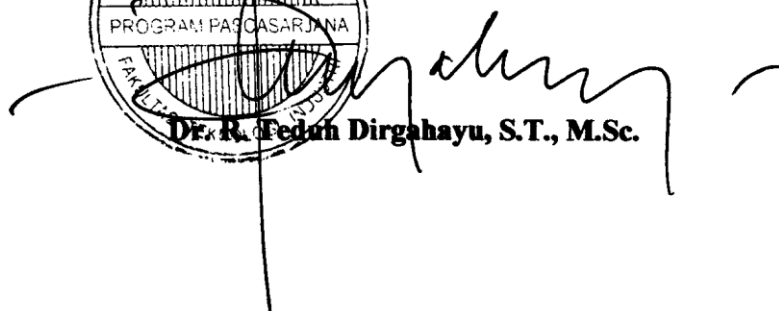


Mengetahui,

**Direktur Program Pascasarjana
Fakultas Teknologi Industri Universitas Islam Indonesia**



Dr. R. Feduh Dirgahayu, S.T., M.Sc.



LEMBAR PERNYATAAN KEASLIAN TESIS

Saya yang bertandatangan di bawah ini,

Nama : Muhammad Innuddin
NIM : 14917150
Program Studi : Magister Informatika
Konsentrasi Bidang : Forensika Digital

Menyatakan bahwa di dalam tesis saya yang berjudul “*HYBRID IMAGE WATERMARKING* RDWT DENGAN SVD UNTUK PERLINDUNGAN HAK CIPTA PADA CITRA DIGITAL” tidak terdapat karya yang pernah diajukan orang lain untuk memperoleh gelar tertentu di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan di dalam daftar pustaka. Apabila di kemudian hari terbukti bahwa pernyataan ini tidak benar, maka saya bersedia menerima sanksi atau hukuman apapun sesuai dengan peraturan yang berlaku.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 31 Desember 2016



Muhammad Innuddin

NIM : 14917150

HALAMAN PERSEMBAHAN

Alhamdulillah Rasa syukur kehadirat Allah SWT atas limpahan Rahmat dan Hidayah-Nya.

Sholawat dan salam selalu tercurahkan keharibaan Baginda Nabi Besar Muhammad SAW.

Untuk Ayah dan Ibu tercinta,

Terima kasih atas segala yang diberikan, jerih payah dan dukungan, nasihat dan pengertiannya serta do'a dan kasih sayang-mu yang tiada terhingga yang tidak mungkin dapat ku-balas dengan selembar kertas bertuliskan kata cinta dan persembahan.

Untuk Adik-adikku tersayang,

Terima kasih atas doa dan bantuan kalian selama ini.

Untuk Dindaku,

Terima kasih atas dukungan, kritik, saran dan pengertiannya selama ini, dan

Untuk Sahabatku,

Terima kasih banyak atas dukungan dan masukannya selama pembuatan laporan ini.

HALAMAN MOTTO

Sesungguhnya Allah tidak akan Mengubah keadaan suatu kaum sebelum mereka mengubah keadaan diri mereka sendiri.

(Q.S. Ar-Ra'd 11)

Dan di langit terdapat (sebab sebab) rezekimu dan apa yang dijanjikan kepadamu. Maka demi Tuhan langit dan bumi, sungguh, apa yang dijanjikan itu pasti terjadi seperti apa yang kamu ucapkan.

(Q.S. Adz-Dzariyat 22-23)

Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya, dan sesungguhnya usahanya itu kelak akan diperlihatkan (kepadanya), kemudian akan diberi balasan kepadanya dengan balasan yang paling sempurna.

(Q.S. An-Najm 39-41)

Dan apabila hamba-hamba-ku bertanya kepadamu (Muhammad) tentang Aku, maka sesungguhnya Aku dekat. Aku kabulkan permohonan orang yang berdoa apabila dia berdoa kepada-Ku....

(Q.S. Al-Baqarah 186)

“Sesungguhnya Allah ta'ala malu bila seorang hamba membentangkan kedua tangannya untuk memohon kebaikan kepada-Nya, lalu Ia mengembalikan kedua tangan hamba itu dalam keadaan hampa/gagal.”

(HR. Ahmad (5/438))

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji bagi Allah SWT atas segala rahmat, hidayah, dan inayah-Nya, sehingga penulisan laporan tesis sebagai salah satu syarat memperoleh gelar Pascasarjana Magister Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang berjudul “*HYBRID IMAGE WATERMARKING RDWT DENGAN SVD UNTUK PERLINDUNGAN HAK CIPTA PADA CITRA DIGITAL*” dapat penulis selesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para sahabat, serta pengikutnya hingga hari kiamat nanti.

Penyusunan laporan tesis ini tidak lepas dari bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu dalam kesempatan ini dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang setulus-tulusnya kepada:

1. Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis selalu diberi kesehatan dan kemudahan selama masa pengerjaan tesis ini.
2. Ayah, ibu, saudara, sahabat beserta keluarga besar yang telah memberikan do'a restu dan dukungannya.
3. Bapak Rektor dan seluruh jajaran Rektorat Universitas Islam Indonesia.
4. Dr. R. Teduh Dirgahayu, S.T., M.Sc. selaku Direktur Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Dr. Bambang Sugiantoro, M.T selaku dosen pembimbing pertama yang telah memberikan pengarahan, bimbingan, masukan, serta dorongan semangat selama pelaksanaan tesis dan penulisan laporan.

6. Yusuf Yudi Prayudi, S.Si., M.Kom. selaku dosen pembimbing kedua yang telah memberikan pengarahan, bimbingan, masukan, serta dorongan semangat selama pelaksanaan tesis dan penulisan laporan.
7. Dosen-dosen Magister Informatika dan seluruh jajaran staf program Pascasarjana. Terima kasih atas semua ilmu pengetahuan, saran, motivasi, serta bantuannya.
8. Rekan-rekan Angkatan X. Terima kasih atas semua dukungan dan kerja samanya selama ini. Selamat berjuang.
9. Keluarga besar Magister Informatika. Terima kasih atas kerja samanya.
10. Teman-teman yang jauh di sana dan selalu mendoakan, terima kasih atas semuanya.
11. Semua pihak yang telah memberikan bantuan dan dorongan yang tidak dapat penulis sebutkan satu persatu.

Semoga segala kebaikan yang telah diberikan kepada penulis, akan dibalas Allah dengan yang lebih baik. Amin.

Penulis menyadari bahwa dalam penyusunan laporan tesis ini masih banyak terdapat kekurangan. Untuk itu penulis menyampaikan permohonan maaf serta sangat mengharapkan kritik dan saran yang membangun untuk penyempurnaan di masa mendatang.

Akhir kata semoga laporan ini dapat bermanfaat bagi kita semua. Amin.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 31 Desember 2016

Muhammad Innuddin

ABSTRAK

Watermarking merupakan salah satu metode yang sering digunakan dalam perlindungan hak cipta terhadap konten multimedia untuk menanggulangi penyebaran karya seseorang secara ilegal. Teknik *Watermarking* dibagi kedalam domain frekuensi (misalnya RDWT) dan domain spasial (misalnya SVD). *Redundant Discrete Wavelet Transform* (RDWT) merupakan modifikasi dari *Discrete Wavelet Transform* (DWT) yang tahan dari berbagai serangan. Sementara *Singular Value Decomposition* (SVD) memiliki stabilitas yang baik tetapi tidak tahan terhadap serangan. Pada penelitian ini peneliti menggabungkan algoritma RDWT dengan SVD untuk menghasilkan teknik *Watermarking* yang lebih baik yakni dengan melakukan penyisipan *watermark* pada *subband horizontal detail* (LH) yang merupakan bagian karikatur dari 4 citra hasil RDWT yang sedikit mengandung informasi jika dibandingkan dengan *subband approximation* (LL) yang mengandung banyak informasi. Skema ini diterapkan supaya tidak mempengaruhi citra *host* saat disisipi *watermark* dan didukung dengan nilai α 0.1 untuk memperoleh tingkat *imperceptibility* dan *robustness* yang tinggi. Dari hasil pengujian skema menghasilkan tingkat *imperceptibility* yang tinggi dengan nilai PSNR tertinggi 49,3148 pada *image* (Splash) dan nilai Q tertinggi 4,99366 pada *image* (Brodats-Bark) dan tingkat *robustness* yang tinggi dengan nilai rata-rata 0,9221 dari hasil berbagai serangan.

Kata Kunci:

Redundant Discrete Wavelet Transform (RDWT), *Discrete Wavelet Transform* (DWT), *Singular Value Decomposition* (SVD), LH, α .

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
LEMBAR PERNYATAAN KEASLIAN TESIS	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Review Penelitian	5
1.7 Metodologi Penelitian	10
1. Studi literatur	10
2. Implementasi Hybrid Image Watermarking	10
3. Pengujian tingkat robustness metode Hybrid Image Watermarking..	10
4. Pengujian tingkat imperceptibility algoritma Hybrid Image Watermarking.....	10
5. Kesimpulan.....	11
1.8 Sistematika Penulisan	11
BAB II LANDASAN TEORI	12
2.1 Citra Digital (Image)	12
2.2 Gambaran Umum Watermarking	13
a. Autentifikasi (Tamper-proofing)	14
b. Caption	15

c.	Copyright Protection	15
d.	Fingerprinting.....	15
e.	Copy Control.....	15
f.	Medical Record	15
2.3	Sejarah Watermarking	16
2.4	Metode Watermarking	18
a.	Imperceptibility	19
b.	Robustness.....	19
a.	Blind Watermarking.....	20
b.	Non Blind Watermarking.....	20
c.	Fragile Watermarking	20
d.	Visible Watermarking	21
2.5	Wavelet	21
2.5.1	Wavelet Haar.....	23
2.5.2	Wavelet Daubechies.....	25
2.5.3	Wavelet Symlets.....	26
2.6	Transformasi Wavelet.....	27
2.7	Discrete Wavelet Transform (DWT)	28
2.8	Redundant Discrete Wavelet Transform (RDWT)	32
2.9	Singular Value Decomposition (SVD)	33
2.10	Parameter dalam Watermarking	34
2.10.1	PSNR (Peak Signal to Noise).....	34
2.10.2	Perceptual Quality Matric (Q).....	35
2.10.3	Corr.....	36
2.11	Attack pada Watermarking	36
a.	Kompresi	36
b.	Filtering	36
c.	Color Reduce.....	37
d.	Noise Addition	37
e.	Overmarking.....	37
f.	Remodulation	37
g.	MAP	37

h.	JPEG.....	37
i.	ML.....	38
j.	Resample	38
k.	Copy	38
l.	Wavelet.....	38
m.	Ratation Scala.....	38
2.12	Tingkat Ekstraksi dan Deteksi Watermark	39
BAB III METODOLOGI PENELITIAN.....		40
3.1	Rancangan Algoritma Hybrid Image Watermarking RDWT-SVD.....	40
3.1.1	Diagram Proses Penyisipan RDWT-SVD.....	40
3.1.2	Diagram Proses Ekstraksi RDWT-SVD	41
3.1.3	Desain Antarmuka Pengguna.....	42
3.2	Perangkat Pendukung Penelitian.....	43
3.2.1	Perangkat Lunak (Software).....	43
3.2.2	Perangkat Keras (Hardware)	44
3.3	Bahan Penelitian	44
3.3.1	Citra Digital (Image).....	44
3.3.2	Attacked Image	46
3.4	Pengujian.....	46
3.5	Parameter Penelitian	49
3.5.1	Tingkat Imperceptibility.....	49
3.5.2	Tingkat Robustness	50
3.6	Sampel Pengambilan Data Penelitian	50
3.6.1	Pengujian Pengaruh Nilai Alpha Terhadap Tingkat Imperceptibility pada Algoritma RDWT-SVD.....	50
3.6.2	Pengujian tingkat imperceptibility algoritma RDWT-SVD.....	51
3.6.3	Pengujian Tingkat Robustness algoritma RDWT-SVD.....	52
3.7	Perbandingan dengan penenelitian sebelumnya	55
3.8	Pengaruh Hybrid metode Watermarking terhadap HAKI	56
BAB IV IMPLEMENTASI DAN ANALISIS.....		57
4.1	Tampilan GUI Aplikasi Hybrid Image Watermarking	57
4.2	Parameter Hybrid Algoritma RDWT-SVD	58

4.3	Data Penelitian dan Pembahasan	59
4.3.1	Pengujian pengaruh alpha terhadap tingkat imperceptibility	59
4.3.2	Pengujian tingkat imperceptibility algoritma RDWT-SVD.....	63
4.3.3	Pengujian tingkat robustness algoritma RDWT-SVD	65
4.4	Perbandingan.....	69
1.	Perbandingan Tingkat Imperceptibility.....	70
2.	Perbandingan Tingkat Robustness	71
4.5	Pengaruh hybrid metode watermarking terhadap HAKI.....	73
BAB V KESIMPULAN DAN SARAN.....		75
5.1	Kesimpulan	75
5.2	Saran	76
DAFTAR PUSTAKA		76
LAMPIRAN.....		78
Lampiran 1: Matlab Coding		78
Lampiran 2: Citra Host atau Cover Image		88
Lampiran 3: Citra Ter-Watermarked		89
Lampiran 4: Citra Setelah diberikan Serangan		90
Lampiran 5: Citra Watermark dan Hasil Ekstraksi		92



DAFTAR GAMBAR

Gambar 1.1 Skema umum watermarking.....	10
Gambar 2.1 Proses penyisipan (embedder) dan ekstraksi (detector) watermark .	14
Gambar 2.2 Metode watermarking	18
Gambar 2.3 Keluarga atau jenis wavelet.....	22
Gambar 2.4 Tingkatan atau orde keluarga wavelet.....	23
Gambar 2.5 Fungsi basis pada ruang V^2	24
Gambar 2.6 Bentuk sinyal wavelet haar pada pada $W1$	24
Gambar 2.7 Bentuk sinyal dari wavelet daubechies	26
Gambar 2.8 Bentuk sinyal dari wavelet symlets	26
Gambar 2.9 Struktur proses transformasi wavelet multi-skala	29
Gambar 2.10 Struktur representasi skala 1 DWT	30
Gambar 2.11 Implementasi Skala 2 Discrete Wavelet Transform (2D-DWT).....	31
Gambar 3.1 Diagram Proses Penyisipan Watermarking RDWT-SVD.....	40
Gambar 3.2 Diagram Proses Ekstraksi Watermark.....	41
Gambar 3.3 Desain antarmuka pengguna.	43
Gambar 3.4 Histogram citra digital.....	45
Gambar 3.5 Histogram citra digital penelitian	45
Gambar 3.6 Pengujian pengaruh alpha terhadap tingkat imperceptibility metode Hybrid Image Watermarking RDWT-SVD	47
Gambar 3.7 Pengujian tingkat imperceptibility metode Hybrid Image Watermarking RDWT-SVD	48
Gambar 3.8 Pengujian tingkat robustness metode Hybrid Image Watermarking RDWT-SVD.....	49
Gambar 4.1 Tampilan GUI hybrid image watermarking algoritma RDWT-SVD	58
Gambar 4.2 Hubungan alpha dengan PSNR algoritma RDWT-SVD.....	60
Gambar 4.3 Hubungan alpha dengan Corr algoritma RDWT-SVD	61
Gambar 4.4 Hubungan alpha dengan Q algoritma RDWT-SVD.....	61

Gambar 4.5 Struktur Dekomposisi Algoritma RDWT pada Subband LL_2 dan	
LH_2	73
Gambar 4. 6 Bagan penerapan Hybrid algoritma RDWT dengan SVD pada	
HAKI.....	74



DAFTAR TABEL

Tabel 1.1 Penelitian tentang hybrid image watermarking	8
Tabel 2.1 Reting kualitas citra digital	35
Tabel 3.1 Analisis cover image berdasarkan level histogram.....	46
Tabel 3.2 Nilai PSNR, Q dan Corr untuk Watermarking RDWT-SVD	51
Tabel 3.3 Nilai PSNR, Q dan Corr algoritma RDWT-SVD	52
Tabel 3.4 Nilai Corr algoritma RDWT-SVD.....	53
Tabel 3.5 Resume Corr kelompok Attack Algoritma RDWT-SVD	53
Tabel 3.6 Deteksi dari attacked image algoritma RDWT-SVD.....	54
Tabel 3.7 Resume deteksi Corr berdasarkan kelompok attack	55
Tabel 3.8 Perbandingan tingkat imperceptibility (dB).....	55
Tabel 3.9 Perbandingan tingkat robustness (corr).....	55
Tabel 4.1 Nilai PSNR, Quality dan Corr algoritma RDWT-SVD.....	60
Tabel 4.2 Nilai PSNR, Q dan Corr algoritma RDWT-SVD	63
Tabel 4.3 Ekstraksi watermark Hybrid Image Watermarking RDWT-SVD	66
Tabel 4.4 Resume Corr ekstraksi hybrid algoritma RDWT-SVD	67
Tabel 4.5 Deteksi watermark hybrid image watermarking RDWT-SVD.....	68
Tabel 4.6 Resume tingkat deteksi watermark algoritma RDWT-SVD.....	69
Tabel 4.7 Perbandingan Tingkat Imperceptibility (dB)	71
Tabel 4.8 Perbandingan tingkat robustness (corr).....	72

BAB I

PENDAHULUAN

1.1 Latar Belakang

Era sekarang ini hampir semua orang sudah mengenal internet, seperti yang telah diketahui penggunaan internet sudah menjadi kebutuhan sehari-hari. Banyak fasilitas yang disediakan, salah satunya adalah unggah dan unduh file citra digital. Dengan adanya fasilitas ini seseorang dapat mentransmisikan dan mendapatkan file citra digital dengan mudah karena luasnya jangkauan internet.

Berdasarkan data dari *Sosial Time* (Morrison, 9 Juni 2015) terdapat 200 juta pengguna *Snapchat* dengan jumlah unggah foto 8.796 perdetik, 700 juta pengguna *Whatsapp* dengan jumlah unggah foto 8.102 perdetik, 1.39 miliar lebih pengguna *Facebook* dengan jumlah unggah foto 350 juta perhari, dan 4.501 perdetik dan 300 juta pengguna *Instagram* dengan jumlah unggah foto 70 juta perhari, dan 810 perdetik. Dilihat dari data statistiknya maka citra digital (foto/gambar) yang diunggah ke internet sangat besar secara kuantitas. Namun dampak dari perkembangan tidak selalu positif, ada dampak negatif juga.

Dampak positifnya yaitu kecepatan dalam mentransmisikan atau unggah dan unduh file citra digital ke internet. Sedangkan dampak negatifnya, yaitu penyalahgunaan Hak Atas Kekayaan Intelektual (HAKI) oleh pihak lain terhadap file citra digital asli yang sudah kita unggah ke internet, baik menggunakan aplikasi *free* atau *pro* dalam memanipulasinya. Dengan demikian, untuk menyikapi hal tersebut perlu adanya proteksi hak cipta atau hak kepemilikan terhadap suatu file citra digital untuk membuktikan jika suatu saat terjadi pengakuan hak cipta atau hak milik dari pihak lain.

Menurut Ariyus (2009) metode proteksi terhadap konten multimedia, seperti *text*, *image*, *grafik*, *audio*, dan *video* dapat dilakukan dengan menggunakan beberapa metode seperti *Digital Right Management (DRM)*, *kryptography*, *steganography*, *fingerprint*, *biometric*, *digital signatural*, *enskripsi* dan *digital watermarking*.

Untuk perlindungan hak cipta banyak sekali teknik yang bisa digunakan, salah satu teknik yang sering digunakan adalah *Watermarking* atau juga disebut tanda air, yaitu teknik penyisipan pesan rahasia ke dalam sebuah pesan lainnya yang tetap terlihat jelas dan dapat menyamarkan pesan tersebut dan keberadaan *watermark* bisa dibuktikan dengan proses ekstraksi (Ariyus, 2009). Dalam *watermarking* teknik penyisipan dibagi menjadi 2 (dua) bagian, yaitu teknik domain spasial dan teknik domain frekuensi. Teknik domain spasial penyisipan *watermark* langsung pada objek, sedangkan domain frekuensi menyisipkan *watermark* dengan mengubah nilai-nilai komponen frekuensi dengan transformasi. Dalam implementasinya teknik domain frekuensi lebih tahan terhadap serangan, tetapi teknik domain spasial lebih kompleks.

Penelitian akademik dan non-akademik di bidang *Watermarking* akhir-akhir ini sangat diminati dalam perlindungan hak cipta. Kemajuan teknologi internet dan aplikasi manipulasi dari tahun ke tahun semakin berkembang menjadi salah satu alasan bagi para peneliti untuk terus menemukan algoritma *Watermarking* yang mampu beradaptasi dengan kemajuan tersebut. Algoritma 25 tahun yang lalu mungkin dianggap masih aman pada masanya, namun sekarang algoritma tersebut sudah tidak bisa dianggap aman.

Seiring perkembangannya, para peneliti sudah banyak menemukan algoritma baru yang mampu melebihi algoritma sebelumnya. (Hien, Nakao, & Chen, 2005) melakukan modifikasi pada algoritma DWT dengan menghilangkan proses *down-sampling*, sehingga mengakibatkan *redundant* pada nilai spasialnya. Karakteristik pergeseran varian DWT yang timbul dari penggunaan *down-sampling* menghasilkan *nois* yang dapat mengurangi *imperceptibility* dari citra digital yang sudah disisipi *watermark*, sedangkan *Redundant Discrete Wavelet Transform* (RDWT) adalah pergeseran invarian tetap pada tempatnya, sehingga dapat menghilangkan *nois* dan tetap menjaga *imperceptibility* dari citra digital dan *robustness* terhadap beberapa serangan. Algoritma *Singular Value Decomposition* (SVD) merupakan teknik domain spasial yang memodifikasi koefisien yang diperoleh dari dekomposisi nilai singular citra asli dan memiliki stabilitas yang baik.

RDWT merupakan algoritma modifikasi dari DWT dengan menghilangkan proses *down-sampling* nya. Algoritma ini dalam penerapannya membagi komponen pada citra digital (gambar digital) menjadi 4 (empat) bagian atau disebut *subband*, yaitu *aproximation*, *horizontal detail*, *vertical detail*, *diagonal detail* atau sering disebut dengan *low-pass low-pass* (LL), *low-pass high-pass* (LH), *high-pass low-pass* (HL) dan *high-pass high-pass* (HH). Dalam algoritma ini terdapat banyak sekali *type wavelet* salah satunya adalah *wavelet haar* merupakan *wavelet* tertua atau sering disebut *mother wavelet* yang sering digunakan dalam implementasi citra 2D dan sederhana dalam penerapannya.

Hien et al., (2005) dalam penelitiannya mengatakan bahwa, penyisipan pada *subband low-pass high-pass* (LH) dan *high-pass low-pass* (HL) memiliki tingkat kemiripan atau *imperceptibility* yang baik karena penyisipan dilakukan pada bagian karikatur sedikit mengandung informasi sehingga tidak mempengaruhi file citra digital penampungnya dan hasil ekstrak yang baik sedangkan (Sharma dan Seema, 2012) dalam penelitiannya menyatakan bahwa penyisipan pada level dekomposisi yang tinggi dapat menghasilkan tingkat *imperceptibility* dan *robustness* yang tinggi. Untuk mendapatkan hasil yang lebih baik lagi perlu dilakukan penggabungan algoritma. SVD merupakan algoritma yang memiliki stabilitas yang baik jika diberi beberapa serangan dan sering digunakan sebagai algoritma *embedded*.

Dalam meningkatkan *robustness* dan *imperceptibility* pada citra nilai *alpha* juga memiliki peranan penting dalam hal tersebut. *Alpha* atau faktor skala menunjukkan *factor embedding* dengan skala 0 sampai dengan 1, nilai *alpha* menunjukkan tingkat ketampakan (*visible*) *watermark* yang disisipkan pada *cover image*. Semakin besar nilai *alpha* menandakan semakin tampak *watermark* pada *cover image*. Penentuan nilai faktor skala yang optimal dapat meningkatkan kemiripan (*imperceptibility*) dan ketahanan (*robustness*), sehingga *watermark* yang disisipkan dapat diekstrak kembali dan masalah hak cipta dapat dibuktikan.

Berdasarkan pemaparan di atas penelitian kali ini akan menggunakan metode *Two-Scale Redundant Discrete Wavelet Transform* (2D-RDWT) dengan *Singular Value Decomposition* (SVD), penyisipan *watermark* akan dilakukan pada *subband* LH₂ menggunakan *wavelet haar* dengan faktor skala yang bervariasi dan di

implementasikan pada *object* citra *Grayscale* dengan ukuran 512x512. Penggunaan citra *Grayscale* dengan ukuran 512x512 berdasarkan rekomendasi dari beberapa peneliti sebelumnya dalam mengimplementasikan metode *watermarking* dan citra *Grayscale* merupakan transformasi dari citra RGB dengan menghitung nilai rata-ratanya dan menjadikannya menjadi skala ke abu-abuan (Madenda, 2015).

Untuk pengujian dilakukan pengujian tingkat *imperceptibility* dengan menghitung parameter (PSNR, Q dan Corr) dan tingkat *robustness* dengan menghitung nilai *Corr* dari hasil *attack* menggunakan *StirMark* versi 4.0 merupakan aplikasi *standart* internasional dalam pengujian metode *watermarking* dan pemberi *attack* otomatis terhadap citra hasil *watermarking*. Dengan penyisipan di *subband* ini diharapkan dapat meningkatkan ketahanan dan kemiripan pada citra yang dilindungi sehingga ketika dilakukan pembuktian hak milik cipta pada suatu citra digital *watermark* yang disisipkan dapat di ekstrak kembali.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini sebagai berikut.

- a. Bagaimana pengaruh *alpha* terhadap tingkat *imperceptibility* dari *hybrid two-scale RDWT-SVD* ?
- b. Bagaimana tingkat *imperceptibility* dan *robustness* pada algoritma *hybrid two-scale RDWT-SVD* ?
- c. Bagaimana tingkat pengaruh *hybrid* metode *Watermarking* terhadap keamanan HAKI ?

1.3 Batasan Masalah

Batasan masalah pada penelitian *Hybrid Image Watermarking* metode RDWT-SVD adalah sebagai berikut.

- a. Algoritma yang akan di *hybrid* adalah RDWT dengan SVD;
- b. Menggunakan citra digital 512x512 piksel baik untuk *cover image* atau *watermark*;
- c. Menggunakan citra *Grayscale* baik untuk *cover image* atau *watermark*;

- d. Parameter yang akan diukur adalah tingkat *imperceptibility* atau kualitas citra digital hasil dari *Hybrid Image Watermarking* menggunakan PSNR (*Peak Signal to Noise Ratio*), *Perceptual Quality Metric (Q)* dan *Corr*;
- e. Menggunakan *stirmark* supaya menghasilkan *attack* secara otomatis untuk mengetahui tingkat *robustness* citra digital hasil implementasi dari *hybrid two-scale RDWT-SVD* dan *Corr* untuk mengetahui tingkat nilai korelasi hasil ekstraksi dari *watermarked* yang sudah diberi serangan; dan
- f. Menggunakan jenis *wavelet haar* dalam mengimplementasikan metode *Hybrid Image Watermarking RDWT-SVD*.

1.4 Tujuan Penelitian

Tujuan dalam penelitian *hybrid* metode *two-scale RDWT-SVD* ini adalah mengetahui:

- a. Pengaruh *alpha* terhadap tingkat *imperceptibility* dari *Hybrid Image Watermarking* dengan menggunakan algoritma *two-scale RDWT-SVD*;
- b. Tingkat *robustness* dan *imperceptibility* pada *Hybrid Image Watermarking* metode *two-scale RDWT-SVD*; dan
- c. Pengaruh *robustness* dan *imperceptibility* terhadap proteksi atau perlindungan terhadap hak cipta pada citra digital.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian *Hybrid Image Watermarking* ini adalah adanya sosialisasi dalam penggunaan metode *Watermarking* sebagai teknik proteksi atau perlindungan hak cipta pada citra digital, kemudian dapat memberikan rekomendasi teknik *Hybrid Image Watermarking* yang baik dalam proteksi atau perlindungan hak cipta pada citra digital dan menjadi bahan perbandingan bagi peneliti lain dalam mengembangkan sistem yang sama.

1.6 Review Penelitian

Keamanan citra digital dengan metode *Hybrid Image Watermarking RDWT-SVD* pertama dilakukan oleh Kourkchi dan Ghaemmaghami (2008). Implementasi metode ini diterapkan pada citra digital *Grayscale* dengan *size* 512x512, sebelum

proses *embedding* pada citra digital hal pertama yang dilakukan adalah melakukan enkripsi pada *bit watermark* menggunakan operasi bilangan XOR kemudian di-*embedding* melalui *Quantization Index Modulation* (QIM) pada nilai *singular* yang tertanam pada *subband* LL. Hasil dari penelitian ini didapatkan *image* yang *robustness* dan *imperceptibility*.

Penelitian sejenis juga dilakukan oleh Lagzian et al. (2011), penelitiannya menggunakan skema *non blind* RDWT-SVD. Implementasi metode ini menggunakan citra digital *Grayscale* dengan *size* 512x512 baik untuk citra *host* maupun citra *watermark*. Proses *embedding watermark* diterapkan pada semua *subband* RDWT setelah diterapkan SVD pada setiap semua *subband* RDWT dan *watermark*. Untuk membuktikan ada tidaknya *watermark* pada citra digital, metode ini membutuhkan citra digital *host* dan citra digital *watermarked* untuk proses ekstraksi *watermark*. Hasil dari penelitian ini didapatkan *image* yang sangat *robustness* dan *imperceptibility* dari berbagai serangan.

Untuk mencapai tingkat *robustness* dan *imperceptibility* yang tinggi Seema dan Sharma (2012) melakukan penelitian *hybrid* metode *three-scale* DWT dan SVD dengan *object* penelitian *image Grayscale* dengan *size of host* 512x512 dan *watermark* 64x64. Dalam penelitian ini *embedded watermark* diterapkan pada *subband* HL3/ frekuensi tengah ke atas pada *tree-scale* DWT kemudian penerapan nilai SVD pada *subband* HL3, dengan faktor skala 0,01-0,09 dan menggunakan *type* skema *blind watermarking* sehingga untuk membuktikan ada tidaknya *watermark* pada citra digital, metode ini tidak membutuhkan citra *host* untuk proses ekstraksi *watermark* melainkan hanya citra *watermarked* saja. Pada penelitian ini didapatkan hasil yang sangat *robustness* dan *imperceptibility* dengan waktu *embedding* yang cepat karena diterapkannya *three-scale* DWT.

Selanjutnya Padhiary (2013) melakukan pengembangan dari penelitian (Lagzian et al. 2011) dari *type* skema *non blind watermarking* ke *blind watermarking* dan *particle swarm optimizer* (PSO) untuk mencari nilai skala faktor. Pada penelitian ini didapatkan hasil yang sangat *robustness* dan *imperceptibility*, karena *embedded* dilakukan pada semua *subband* RDWT, metode tersebut adalah

penyempurnaan dari metode DWT ditambah lagi metode tersebut di-*hybrid* dengan SVD yang *imperceptibility*-nya sudah teruji.

Sementara itu, Makbol dan Khoo (2013) melakukan penelitian yang sama dengan Padhiary (2013) yaitu melakukan pengembangan dari penelitian (Lagzian et al. 2011) dari *type* skema *non blind watermarking* ke *blind watermarking* dengan skala faktor 0,05 *subband* LL dan 0,005 *subband* (LH,HL,HH). Pada penelitian ini didapatkan hasil yang sangat *robustness* dan *imperceptibility*, karena *embedded* dilakukan pada semua *subband* RDWT, metode tersebut merupakan penyempurnaan dari metode DWT ditambah lagi metode tersebut di-*hybrid* dengan SVD yang *imperceptibility*-nya sudah teruji.

Implementasi metode RDWT-SVD juga dilakukan oleh Chen dan Zhao (2015) dengan perlindungan pada *object video* dan citra digital *size* 512x512 sebagai *watermark*. Pada penelitian ini *embedding watermark* diterapkan pada *subband* tengah yaitu (LH dan HL). Parameter *embedding* menggunakan *noise visibility function* (NVF). Hasil yang didapatkan *video* yang *robustness* dan *imperceptibility*.

Ansari dan Prayudi (2015) melakukan komparasi metode *hybrid*, yaitu DWT-SVD dengan RDWT-SVD. Pada penelitian ini diterapkan *two-scale* DWT dan RDWT dengan *embedding watermark* di *subband* LL₂. Hasil dari komparasi *hybrid* ini disimpulkan bahwa *hybrid Redundant Discrete Wavelet Transform* (RDWT) dengan *Singular Value Decomposition* (SVD) lebih baik dari *hybrid* metode *Discrete Wavelet Transform* (DWT) dengan *Singular Value Decomposition* (SVD), karena adanya metode RDWT yang merupakan penyempurnaan dari metode DWT sehingga menghasilkan teknik yang sangat *robustness* dan *imperceptibility*.

Sementara pada penelitian ini akan melakukan *embedded* pada *subband* LH₂ (frekuensi tinggi) *two-scale* RDWT untuk mendapatkan hasil *watermarking* yang lebih *robustness*. Dari hasil penelitian ini dapat di jadikan sebagai acuan dalam penelitian selanjutnya dan rekomendasi teknik *Hybrid Image Watermarking* yang lebih baik pada proteksi atau perlindungan hak cipta pada citra digital.

Untuk lebih jelasnya tentang penelitian-penelitian sebelumnya dapat dilihat pada tabel penelitian dibawah ini:

Tabel 1.1 Penelitian tentang *Hybrid Image Watermarking*

No	Peneliti	Metode	Object dan Size	Scale	Penyisipan
1.	Hossein Kourkchi dan Sharokh Ghaemmaghami, 2008	<i>Semi-Fragile Watermarking</i> RDWT-SVD	- <i>Object image grayscale</i> - <i>Size of host image 512*512</i>	- <i>One-scale</i>	- <i>Subband (LL)</i>
2.	Samira Lagzian, Mohsen Soryani dan Mahmood Fathy, 2011	<i>Non blind watermarking</i> RDWT-SVD	- <i>Object image grayscale</i> - <i>Size of host image 512*512</i> - <i>Size of watermark image 512*512</i>	- <i>One-scale</i>	- <i>All subband</i>
3.	Sheetal Sharma dan Seema, 2012	<i>Blind Watermarking</i> DWT-SVD	- <i>Object image grayscale</i> - <i>Size of host image 512*512</i> - <i>Size of watermark image 64*64</i>	- <i>Three-scale</i>	- <i>Subband (HL3)</i>
4.	Sabyasachi Padhiary, 2013	<i>Blind Watermarking</i> RDWT-SVD	- <i>Object image grayscale</i> - <i>Size of host image 512*512</i> - <i>Size of watermark image 512*512</i>	- <i>One-scale</i>	- <i>All subband</i>
5.	Nasrin M.Makbol dan Bee Ee Khoo, 2013	<i>Blind Watermarking</i> RDWT-SVD	- <i>Object image grayscale</i> - <i>Size of host image 512*512</i> - <i>Size of watermark image 512*512</i>	- <i>One-scale</i>	- <i>All subband</i>
6.	Lei Chan dan Jiyang Zhao, 2015	<i>Adaptive Watermarking</i> RDWT-SVD	- <i>Object video (cover)</i> - <i>Watermark image grayscale</i> - <i>Size of watermark image 512*512</i>	- <i>One-scale</i>	- <i>Subband (LH dan HL)</i>

No	Peneliti	Metode	Object dan Size	Scale	Penyisipan
7.	Munawir Ansari dan Yudi Prayudi, 2015	Komparasi <i>Hybrid Non blind Watermarking</i> RDWT-SVD dan DWT-SVD	- <i>Object image color dan grayscale</i> - <i>Size of host image 512*512</i> - <i>Size of watermark image 512*512</i>	- <i>Two-scale</i>	- <i>Subband (LL2)</i>
8.	Usulan Penelitian	<i>Hybrid Non blind Watermarking</i> RDWT-SVD	- <i>Object image Grayscale</i> - <i>Size of host image 512*512</i> - <i>Size of watermark image 512*512</i>	- <i>Two-scale</i>	- <i>Subband (LH2)</i>
<p>Pada penelitian ini penyisipan dilakukan pada <i>subband</i> LH₂ (frekuensi tinggi) dengan <i>two-scale</i> RDWT, karena <i>subband</i> ini merupakan bagian karikatur dari empat (4) bagian <i>image</i> yang sudah diterapkan algoritma RDWT, hanya mengandung sedikit informasi dan jika penyisipan dilakukan pada <i>subband</i> ini indra penglihatan manusia tidak akan mampu mendeteksi keberadaan <i>watermark</i> yang disisipkan pada citra digital yang dilindungi dan dapat menghasilkan tingkat <i>imperceptibility</i> yang baik dan mampu mengekstrak kembali <i>watermark</i> yang disisipkan. Penentuan nilai <i>alpha</i> berdasarkan penelitian-penelitian sebelumnya, setiap melakukan penyisipan pada frekuensi tinggi selalu menerapkan nilai <i>alpha</i> tiga angka di belakang koma (,) karena frekuensi tinggi memiliki tingkat kompres yang rendah, tapi pada penelitian ini peneliti akan mencoba menerapkan beberapa nilai <i>alpha</i> untuk mengetahui nilai yang lebih baik dalam meningkatkan <i>imperceptibility</i> dan <i>robustness</i>.</p>					

1.7 Metodologi Penelitian

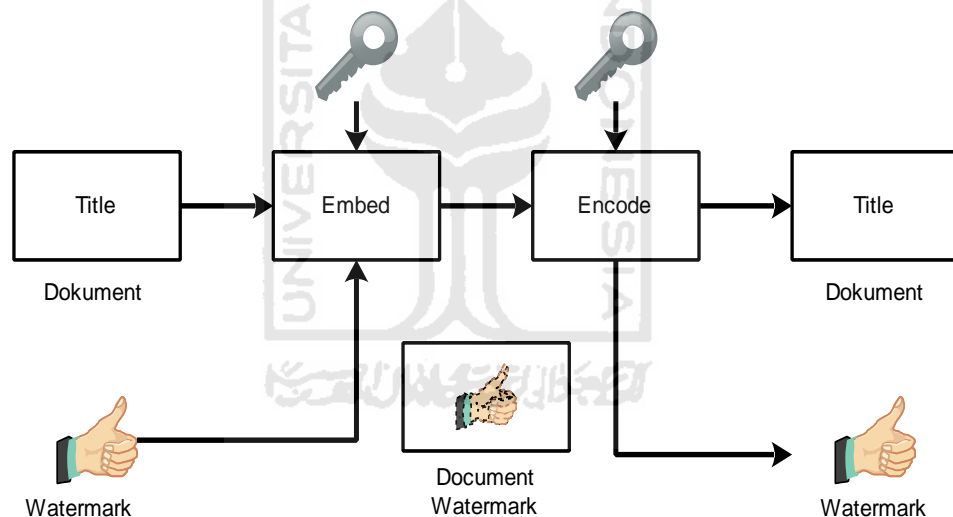
Dalam melakukan penelitian, perlu disusun langkah-langkah dalam penyelesaian penelitian. Langkah-langkah dalam penelitian ini antara lain:

1. Studi literatur

Tahapan ini adalah proses untuk mendapatkan informasi dan pendalaman konsep proteksi terhadap citra digital dengan menggunakan metode-metode *Watermarking* dan metode *hybrid* pada *Watermarking*.

2. Implementasi *Hybrid Image Watermarking*

Pada tahapan ini akan dilakukan penerapan algoritma *hybrid* yang sudah ditentukan yaitu *Redundant Discrete Wavelet Transform (RDWT)* dengan *Singular Value Decomposition (SVD)* berdasarkan algoritma mulai dari proses penyisipan hingga ekstraksi *watermark* pada citra digital.



Gambar 1.1 Proses pemberian *watermark* pada dokumen

(Sumber: Ariyus, 2009)

3. Pengujian tingkat *robustness* metode *Hybrid Image Watermarking*

Parameter tingkat *robustness* dari *hybrid* ini akan diuji menggunakan program *StirMark* sebagai pemberi *attack* otomatis dan hasil *attack* akan dihitung menggunakan *Corr* untuk mengetahui tingkat korelasinya.

4. Pengujian tingkat *imperceptibility* algoritma *Hybrid Image Watermarking*

Parameter tingkat kualitas atau *imperceptibility* akan dilakukan dengan cara membandingkan kualitas citra *watermarked* dengan citra asli atau *cover*

image menggunakan perhitungan PSNR (*Peak Signal to Nois*), *Perceptual Quality Matric* (Q) dan Corr.

5. Kesimpulan

Memberikan teknik *new hybrid* dalam proteksi citra digital sehingga bisa dijadikan rujukan dan rekomendasi oleh *user*.

1.8 Sistematika Penulisan

Untuk memberikan gambaran dalam penyusunan penelitian ini, maka peneliti menguraikan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan merupakan pengantar terhadap permasalahan yang di bahas meliputi latar belakang, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, *review* penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan kajian pustaka dan teori-teori dasar yang berkaitan dengan penelitian yang sedang diteliti.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah penelitian, kebutuhan perangkat keras, perangkat lunak dan bahan penelitian yang digunakan serta perancangan antarmuka aplikasi yang akan dibuat.

BAB IV IMPLEMENTASI DAN ANALISIS

Bab ini berisi tentang implementasi dan analisis dari algoritma yang digunakan terhadap obyek penelitian yang diteliti berdasarkan sejumlah parameter, seperti PSNR, Q, dan Corr untuk mengetahui tingkat *robust* dan *imperceptibility* dari algoritma yang digunakan.

BAB V PENUTUP

Pada bab ini memuat kesimpulan akhir dari semua proses penelitian sampai kepada hasil dari implementasi metode dan saran yang membangun untuk penelitian.

BAB II LANDASAN TEORI

2.1 Citra Digital (*Image*)

Citra digital atau *image* adalah file gambar yang digunakan pada penelitian ini. Penggunaan kata citra digital lebih pada hal yang bersifat teknis sedangkan penggunaan kata file gambar lebih banyak digunakan pada implementasi. Pada penelitian ini kata citra digital, file gambar dan *image* memiliki makna yang sama. Citra digital adalah gambar pada bidang dua dimensi, maka sebuah citra merupakan dimensi spasial atau bidang yang berisi informasi warna yang tidak bergantung waktu (Munir, 2004).

Secara umum, citra digital merupakan representasi piksel-piksel dalam ruang 2D yang dinyatakan dalam matriks berukuran N baris dan M kolom. Setiap elemen matriks citra disebut piksel. Secara matematis, sebuah citra dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$, dimana x dan y adalah koordinat spasial (*plane*) dan f adalah nilai intensitas warna pada koordinat x dan y . Nilai f , x dan y adalah nilai berhingga. Bila nilai-nilai ini bersifat kontinu maka citranya disebut citra analog, seperti yang ditampilkan pada layar monitor TV, komputer atau foto cetak. Bila nilai-nilai ini bersifat diskret maka citranya disebut citra digital, seperti yang tersimpan dalam memori komputer dan CD-ROM (Madenda, 2015).

Menurut Munir (2004), karena cahaya merupakan bentuk energi, maka intensitas cahaya bernilai antara 0 sampai tak-hingga atau $(0 \leq f(x,y) < \infty)$. Nilai $f(x,y)$ sebenarnya adalah hasil kali dari :

- a. $i(x,y)$ sebagai energi cahaya yang datang dari sumbernya (*illumination*), dengan nilai dari 0 sampai tak-hingga,
- b. $r(x,y)$ mewakili derajat kemampuan objek memantulkan cahaya (*reflection*) dengan nilai antara 0 dan 1 yaitu nilai 0 mengindikasikan penyerapan total, sedangkan nilai 1 mengindikasikan pemantulan total.

2.2 Gambaran Umum *Watermarking*

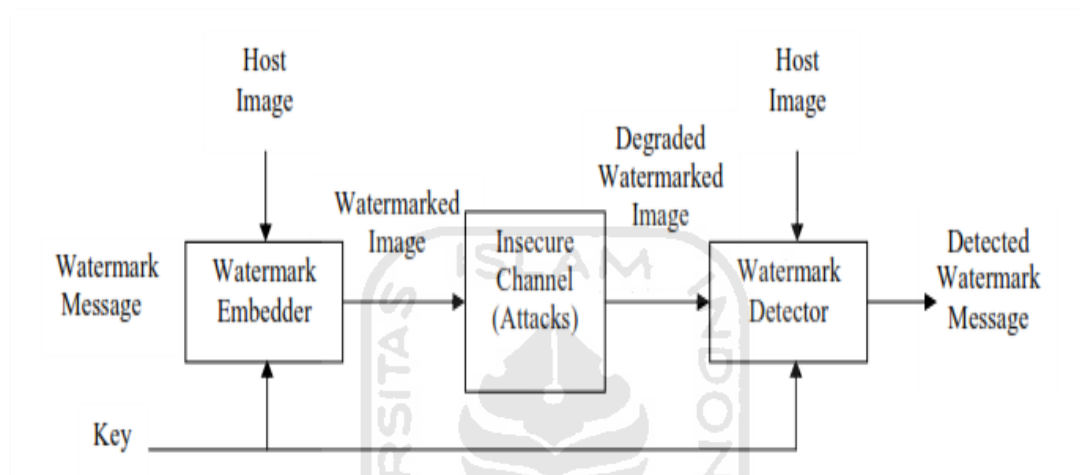
Metode proteksi konten multimedia (*text, grafik, gambar, audio, dan video*) menurut Ariyus (2009) dapat dilakukan dengan menggunakan metode-metode seperti *kriptography* (penyadian), *steganography* (pengkodean), *digital watermarking* (tanda air), *finger printing* (sidik jari digital) dan *digital signature* (tanda tangan digital). Implementasi dari metode-metode tersebut digunakan sesuai dengan kebutuhan masing-masing. *Kriptography* merupakan teknik pengamanan file multimedia dengan melakukan pengacakan data asli sehingga dihasilkan data baru yang sudah terenskripsi dan memerlukan kunci untuk bisa menyusunnya kembali sehingga bisa dipahami maksud dan artinya oleh pihak penerima data tersebut.

Steganography merupakan proses proteksi dan perlindungan file multimedia dengan cara menyisipkan pesan rahasia yang berupa informasi pada file multimedia dengan tujuan menyembunyikan pesan rahasia (*data hiding*), sedangkan *watermarking* merupakan cabang ilmu dari *steganography*, yang membedakan *steganography* dengan *watermarking* yaitu pada *steganography* yang dilindungi adalah pesan rahasianya sedangkan pada *watermarking* yang dilindungi adalah file multimediana, sedangkan pesan rahasianya cuma sebagai identitas kepemilikan file (*authentication*) untuk proteksi dan perlindungan kepemilikan (*copyright protection*) file multimedia tersebut.

Watermarking dapat diartikan sebagai teknik penyisipan informasi yang bersifat rahasia berupa identitas pada file data multimedia (*file host*), penyisipan file *watermarking* atau *watemark* diusahakan tidak mengurangi kualitas file gambar atau citra digital yang ditumpanginya sehingga file gambar atau citra digital sebelum dan sesudah proses *watermarking* memiliki kualitas yang hampir sama. *Watermarking* merupakan proses penyisipan sinyal *low level* (pesan rahasia) pada file multimedia seperti *text, grafik, image, audio* dan *video* dengan tujuan untuk proteksi dan perlindungan hak cipta (identitas) file multimedia tersebut.

Sinyal *low level* yang disisipkan pada citra induk (*file host*) disebut dengan file *watermarking* atau *watermark* yang menunjukkan identitas kepemilikan sah file

multimedia tersebut. Pada proses *watermarking* terdiri dari 2 (dua) tahap yaitu proses penyisipan (*embedder*) merupakan penyisipan file *watermarking* atau *watermark* pada file gambar atau citra digital yang menjadi citra induk dan proses *ekstraksi* atau *detector* merupakan proses mendeteksi atau menemukan kembali file *watermarking* atau *watermark* yang disipkan pada file gambar atau *file host* tersebut. Proses *watermarking* dapat dijelaskan dengan Gambar 2.1.



Gambar 2.1 Proses penyisipan (*embedder*) dan ekstraksi (*detector*) watermark
(Sumber: Saxena, 2008)

Watermarking sebagai teknik penyisipan file data pada file multimedia dapat dimanfaatkan untuk berbagai tujuan :

a. Autentifikasi (*Tamper-proofing*)

Teknik Watermarking sebagai autentifikasi (*tamper-proofing*) digunakan sebagai alat untuk mengidentifikasi keaslian citra digital yang telah mengalami perubahan dari file aslinya (*tamper-proofing*). Jika *watermark* hasil ekstraksi tidak sama dengan *watermark* aslinya, maka dapat disimpulkan bahwa citra digital yang beredar sudah tidak asli lagi atau sudah mengalami modifikasi dari file aslinya (Munir, 2004).

b. *Caption*

Teknik *Watermarking* digunakan sebagai keterangan tentang citra digital itu sendiri, seperti yang digunakan dalam dunia fotografer untuk identifikasi yang melakukan pengambilan foto (Munir, 2004).

c. *Copyright Protection*

Teknik *Watermarking* digunakan untuk proteksi pada citra digital sebagai bukti autentifikasi (kepemilikan) karya cipta terhadap data digital tersebut. Pada *copyright protection* penyisipan *watermark* biasanya berupa identitas pemilik file multimedia (nama, alamat, tulisan, logo, dan gambar). Klaim pihak lain yang mengaku sebagai pemilik data digital tersebut dapat dibantah dengan membandingkan *watermark* yang di ekstrak dengan *watermark* aslinya (Supangkat, 2000).

d. *Fingerprinting*

Teknik *Watermarking* digunakan sebagai identitas digital (*finger printing*) untuk mengendalikan peredaran data digital untuk setiap penanggungjawab. Dengan menggunakan *finger printing* bisa diketahui yang melakukan penyebaran data digital secara ilegal dengan mengetahui nomor serial atau identitas yang disisipkan pada data digital tersebut dan mengetahui yang melakukan penyebaran (Ariyus, 2009).

e. *Copy Control*

Teknik *Watermarking* digunakan untuk mencegah perangkat keras melakukan penggandaan suatu file yang tidak memiliki ijin. *Watermarking* sebagai *copy control* membutuhkan kolaborasi dengan perangkat keras dengan menyisipkan *watermark* detektor pada perangkat keras untuk membaca *watermark* pada data digital tersebut. Jika detektor mendeteksi adanya *watermark* maka proses pembacaan atau penggadaan dihentikan oleh perangkat lunak atau *hardware* (Ariyus, 2009).

f. *Medical Record*

Teknik *Watermarking* digunakan dalam dunia medis seperti penyisipan *watermark* (berupa identitas) dalam foto *rondgen* dengan maksud untuk

memudahkan mengidentifikasi identitas pasien dan hasil diagnosa penyakitnya. Hal ini akan memudahkan petugas di rumah sakit untuk memberikan data kesehatan yang sesuai dan menghindari terjadinya kesalahan data (Ariyus, 2009).

Penggunaan *watermark* memungkinkan bisa dibuktikan kepemilikan terhadap suatu media digital. Untuk mendapatkan *watermark* yang baik, ada beberapa sifat-sifat yang harus diperhatikan. Ariyus (2009) sifat-sifat *watermark* yang baik di antaranya adalah:

- a. *Resilent*, tidak mudah berubah. *Watermark* harus bisa bertahan terhadap serangan-serangan yang bisa menghancurkan *watermark*.
- b. *Cheap*, murah untuk diimplementasikan. *Watermark* tidak boleh memberikan *overhead* yang besar. Sebaliknya *watermark* harus memiliki *overhead* seminimal mungkin.
- c. *Stealthy*, tidak diketahui keberadaannya. *Watermark* harus bisa mempertahankan sifat-sifat statistik dari media penampungnya.
- d. *Unique identifying property*, keberadaan *watermark* bisa dibuktikan dengan proses ekstraksi tertentu.

2.3 Sejarah Watermarking

Ilmu *watermarking* lebih tua dibandingkan dengan ilmu komputer itu sendiri, *watermarking analog* sudah dikenal sekitar abad ke-13 dengan menggunakan perlatan manual seperti cap atau cetakan yang berupa tulisan, simbol, dan gambar dengan inisial tertentu untuk identifikasi kepemilikan atau autentifikasi karya atau ciptaan seniman pada saat itu. Penggunaan *Watermarking* yang pertama kali diketahui yaitu penggunaan *watermarking* pada pabrik kertas di Fabriano, Italia untuk membuat kertas yang diberi *watermark* dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan terbentuklah suatu kertas yang sudah memiliki *watermark*. Kertas ini biasa digunakan oleh seniman atau sastrawan untuk menulis karya mereka pada saat itu. Kertas yang sudah dibubuhi *watermark* tersebut sekaligus dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka (Ariyus, 2009).

Watermarking dengan menggunakan perangkat digital atau elektronika dikenal dengan istilah *digital watermarking*. *Digital watermarking* mulai dikembangkan pada tahun 1990 di negara Jepang dan sekitar tahun 1993 di negara Swiss. *Digital watermarking* semakin berkembang untuk mengatasi penyalahgunaan terhadap file digital seiring dengan perkembangan jaringan internet yang semakin cepat dan murah untuk transfer dan pengadaan file multimedia tanpa ijin dengan menggunakan berbagai teknologi media yang sudah canggih. Tujuan pengembangan metode *digital watermarking* untuk mendapat kualitas *watermarking* yang baik dalam hal *imperceptibility* dan *robustness*.

Diawal tahun 1970 metode *digital watermarking* menggunakan metode yang berdasarkan *domain spasial*, metode yang berdasarkan *domain spasial* proses penyisipan *watermark* langsung pada nilai bit dari piksel pada file gambar atau citra digital tersebut. Salah satu contoh metode *digital watermarking* yang berdasarkan *domain spasial* yang paling banyak digunakan adalah *Least Significant Bit (LSB)*, metode ini dilakukan dengan mengganti bit terakhir pada *byte* data dengan *bit watermark*. Sebagai ilustrasi penyisipan dengan menggunakan metode LSB pada *bit* data :

01001101 00101110 10101110 10001010 01001111 10101111 10100010
10101011

Misalkan melakukan penyisipan karakter 'H' dengan bit data (01001000) pada file gambar atau citra digital.

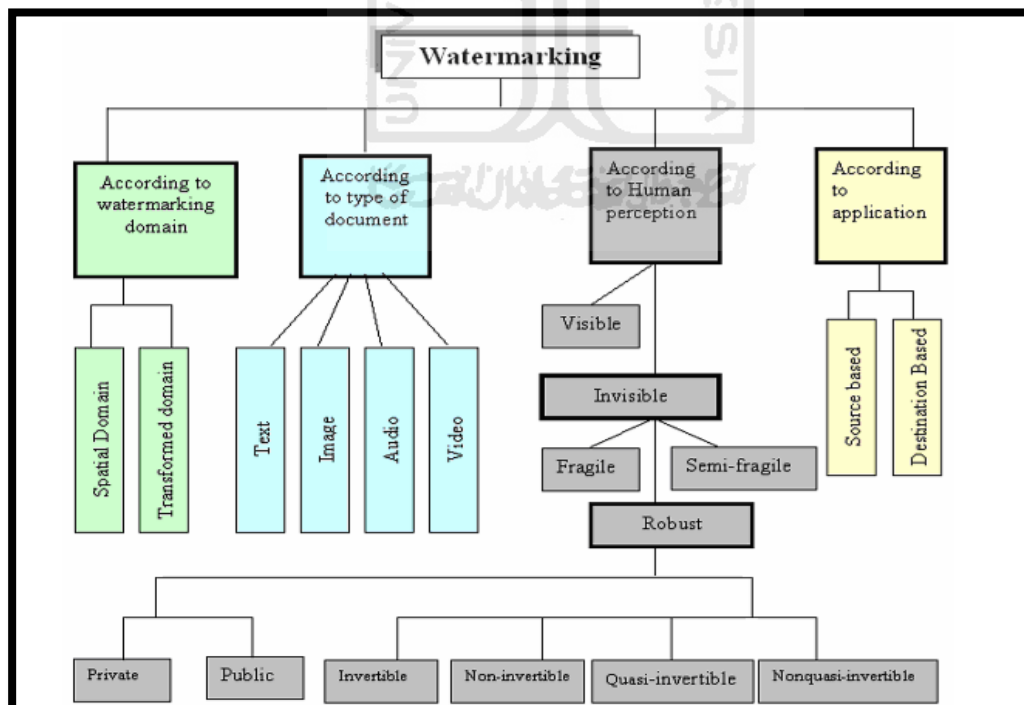
01001100 00101111 10101110 10001010 01001101 10101110 10100010
101010110

Metode *watermarking* berdasarkan *domain spasial* tidak tahan terhadap berbagai serangan karena penyisipan *watermark* dilakukan pada bit piksel dari file gambar atau citra digital sehingga tidak tahan terhadap manipulasi citra digital. Teknik *watermarking* yang berdasarkan domain spasial biasanya digunakan untuk menjaga keaslian atau keutuhan data, jika sedikit saja ada modifikasi maka data tersebut akan rusak sehingga file hasil manipulasi tidak dapat digunakan lagi.

Metode *watermarking* berdasarkan *domain spasial* memiliki kelemahan tidak tahan terhadap berbagai serangan kemudian dikembangkan lagi metode *watermarking* yang berdasarkan domain frekuensi atau transformasi, metode ini ternyata menghasilkan *Watermarking* yang lebih kuat terhadap berbagai serangan. Teknik *watermarking* yang berdasarkan domain frekuensi mulai dikembangkan pada tahun 1997, dimulai dari metode *Spread Spectrum Watermarking*. Perkembangan metode *watermarking* berikutnya dengan menggunakan metode *Discrete Wavelet Transform (DWT)*, *Discrete Cosine Transform (DCT)*, *Fast Fourier Transform (FFT)*, *Singular Value Decomposition (SVD)* dan *Fractal Transform* (Wijaya dan Prijono, 2007).

2.4 Metode *Watermarking*

Secara garis besar metode *watermarking* dapat diklasifikasikan menjadi beberapa jenis berdasarkan domain kerja, type file *watermarking*, *visibility*, dan aplikasi *watermark*. Untuk lebih jelas tentang pengelompokan metode *watermarking* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Metode *Watermarking*
(Sumber: Saxena, 2008)

Metode *watermarking* berdasarkan domain proses atau area kerjanya dapat diklasifikasikan menjadi dua kelompok, yaitu metode *watermarking* yang berdasarkan pada domain spasial dan *watermarking* yang berdasarkan pada domain frekuensi atau transformasi. Metode *watermarking* yang berdasarkan domain spasial bekerja dengan menyisipkan file *watermarking* atau *watermark* secara langsung pada bit piksel atau *spasial* dari suatu file gambar atau citra digital. Domain spasial sendiri mengacu pada piksel-piksel penyusun suatu gambar atau citra digital. Sedangkan Metode *watermarking* yang berdasarkan domain frekuensi proses penyisipan file *watermarking* atau *watermark* dilakukan pada koefisien frekuensi atau frekuensi hasil transformasi dari file gambar atau citra digital aslinya. Metode *watermarking* yang berdasarkan *domain frekuensi* atau *transformasi* antara lain *Discrete Wavelet Transform* (DWT), *Discrete Cosine Transform* (DCT), *Discrete Fourier Transform* (DFT), *Fast Fourier Transform* (FFT), *Singular Value Decomposition* (SVD) dan *Fractal Transform*.

Menurut Hidayat dan Udayanti (2011) metode *watermarking* yang baik atau efektif harus memenuhi persyaratan yaitu harus memiliki kriteria atau tingkat *imperceptibility* dan *robustness* yang tinggi.

a. *Imperceptibility*

Imperceptibility merupakan ketidaktampakan file *watermarking* atau *watermark* yang telah disisipkan pada file gambar atau citra digital. Semakin tinggi tingkat *imperceptibility* menandakan keberadaan *watermark* tidak mengganggu citra digital sebagai *cover image* (*file host*), sehingga kualitas citra digital sebelum dan sesudah proses *watermarking* hampir sama atau tidak mengalami perubahan. Tingkat *imperceptibility* suatu teknik *watermarking* bisa dilihat dari nilai PSNR, Corr, dan Q yang merupakan perbandingan kualitas *file host* dengan citra digital hasil proses *watermarking*.

b. *Robustness*

Robustness merupakan ketahanan file *watermarking* atau *watermark* dari berbagai serangan (*attack*) dan usaha-usaha untuk menghilangkan *watermark* pada citra digital hasil proses *watermarking* atau *watermarked image*. Untuk mengukur tingkat *robust* suatu teknik *watermarking* dapat dilakukan dengan memberikan

attack atau serangan berupa *noise*, jika file gambar hasil watermarking atau *watermarked image* masih tetap baik kualitasnya setelah diberi *attack* atau serangan maka tingkat *robust* dari teknik *watermarking* itu dianggap baik.

Kenyataannya pada metode *single image watermarking* pada domain transformasi tingkat *imperceptibility* berlawanan dengan tingkat *robustness*. Jika suatu metode *watermarking* memiliki nilai *imperceptibility* yang tinggi. Nilai *robust* akan rendah, begitu pula sebaliknya jika metode *watermarking* memiliki nilai *imperceptibility*-nya rendah maka nilai *robust*-nya akan tinggi. Oleh karena itu, harus dalam implementasi metode *watermarking* dilakukan proses penggabungan metode guna untuk menutupi kekurangan masing-masing metode.

Dilihat dari sifat *watermark* yang disisipkan pada citra digital, metode *watermarking* dibedakan menjadi *Blind Watermarking*, *fragile* dan *Visible Watermarking* (Ariyus, 2009).

a. *Blind Watermarking*

Blind Watermarking digunakan untuk mengetahui ada tidaknya sebuah *watermark* yang telah disisipkan pada citra digital. Dengan menggunakan metode ini tidak memerlukan adanya citra asli atau *file host* dalam proses ekstraksi *watermark*

b. *Non Blind Watermarking*

Non Blind Watermarking digunakan untuk mengetahui ada tidaknya sebuah *watermark* pada citra digital, pada metode ini membutuhkan *file host*, *watermark* dan *watermarked image* untuk proses ekstraksi *watermark*.

c. *Fragile Watermarking*

Fragile (lemah) pada metode ini *watermark* tidak tahan dari berbagai serangan atau *attack*. Metode ini biasanya digunakan untuk *autentifikasi* atau ididentifikasi kepemilikan citra digital. Jika terjadi modifikasi atau perubahan walaupun sedikit menyebabkan *watermark* akan rusak.

d. *Visible Watermarking*

Berdasarkan daya tangkap panca indra manusia atau *Human Visual System (HVS) watermarking* terdiri dari *visible watermarking* dan *invisible watermarking*. Pada *visible watermarking*, *watermark* yang disisipkan pada citra digital kelihatan atau tampak (*visibel*) oleh panca indra manusia. Sedangkan pada *invisible watermarking* *watermark* yang disisipkan tidak kelihatan (*invisible*). Pada penelitian ini menggunakan metode *invisible watermarking*.

2.5 *Wavelet*

Menurut Sydney (1998), *wavelet* merupakan gelombang mini (*small wave*) yang mempunyai kemampuan untuk mengelompokkan energi citra digital dan terkonsentrasi pada sekelompok kecil koefisien, sedangkan koefisien lainnya hanya mengandung sedikit energi yang dapat dihilangkan tanpa mengurangi informasi pada citra digital tersebut. *Wavelet* merupakan basis fungsi yaitu fungsi skala (*Scaling function*) yang memiliki sifat dapat disusun dari sejumlah salinan dirinya yang telah diskalakan (dilatasi) dan digeserkan posisinya (translasi). *Wavelet* merupakan keluarga fungsi yang dihasilkan oleh *wavelet basis* $\psi(x)$ yang disebut juga sebagai induk *wavelet* (*mother wavelet*). Dua operasi utama yang mendasari *wavelet* yaitu translasi dan dilatasi.

- Translasi merupakan pergeseran (perubahan posisi), misalnya $\psi(x-1)$, $\psi(x-2)$, $\psi(x-a)$, $\psi(x-b)$;
- Dilatasi merupakan penyekalaan (skala), misalnya $\psi(2x)$, $\psi(4x)$, $\psi(ax)$, $\psi(bx)$.

Dari pergeseran dan penskalaan menghasilkan fungsi dan keluarga *wavelet*. Secara umum persamaan *wavelet* dinyatakan dengan :

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \dots \dots \dots (2.1)$$

Keterangan :

$a, b \in \mathbb{R}$; $a \neq 0$ (\mathbb{R} =bilangan nyata),

a adalah parameter penskalaan (dilatasi),

b adalah parameter pergeseran posisi (translasi) pada sumbu x ,

$\sqrt{|a|}$ adalah normalisasi energi yang sama dengan energi induk.

Wavelet induk diskalakan dan digeser melalui pemisahan menurut frekuensi menjadi sejumlah sub-sub bagian. Untuk mendapatkan sinyal kembali dilakukan proses rekonstruksi *wavelet*. Beberapa contoh keluarga atau jenis *wavelet* antara lain *Haar*, *Daubechies*, *Symlets*, *Coiflets*, *BiorSplines*, *ReverseBior*, *Meyer*, *DMeyer*, *Gaussian*, *Mexican_hat*, *Morlet*, *Complex*, *Gaussian*, *Shannon*, *Frequency B-Spline*, *Complex Morlet*, *Riyad*, dan lain sebagainya. Data dan informasi tentang keluarga atau jenis *wavelet* dapat dilihat pada Gambar 2.3.

Haar	haar
Daubechies	db
Symlets	sym
Coiflets	coif
BiorSplines	bior
ReverseBior	rbio
Meyer	meyr
DMeyer	dmey
Gaussian	gaus
Mexican_hat	mexh
Morlet	morl
Complex Gaussian	cgau
Shannon	shan
Frequency B-Spline	fbsp
Complex Morlet	cmor
Lemarie	lem

Gambar 2.3 Keluarga atau jenis *wavelet*
(Sumber: Mathworks, 2016)

Untuk level atau tingkatan (orde) pada masing-masing jenis *wavelet* terdiri dari beberapa level seperti pada Gambar 2.4 dibawah ini .

=====				=====			
Haar				haar			
=====				=====			
Daubechies				db			
-----				-----			
db1	db2	db3	db4				
db5	db6	db7	db8				
db9	db10	db**					
=====				=====			
Symlets				sym			
-----				-----			
sym2	sym3	sym4	sym5				
sym6	sym7	sym8	sym**				
=====				=====			
Coiflets				coif			
-----				-----			
coif1	coif2	coif3	coif4				
coif5							
=====				=====			
BiorSplines				bior			
-----				-----			
bior1.1	bior1.3	bior1.5	bior2.2				
bior2.4	bior2.6	bior2.8	bior3.1				
bior3.3	bior3.5	bior3.7	bior3.9				
bior4.4	bior5.5	bior6.8					
=====				=====			
ReverseBior				rbio			
-----				-----			
rbio1.1	rbio1.3	rbio1.5	rbio2.2				
rbio2.4	rbio2.6	rbio2.8	rbio3.1				
rbio3.3	rbio3.5	rbio3.7	rbio3.9				
rbio4.4	rbio5.5	rbio6.8					
=====				=====			
Meyer				meyr			
-----				-----			
DMeyer				dmey			
=====				=====			
Gaussian				gaus			
-----				-----			
gaus1	gaus2	gaus3	gaus4				
gaus5	gaus6	gaus7	gaus8				
gaus**							
=====				=====			
Mexican_hat				mexh			
-----				-----			
Morlet				morl			
=====				=====			
Complex Gaussian				cgau			
-----				-----			
cgau1	cgau2	cgau3	cgau4				
cgau5	cgau**						
=====				=====			
Shannon				shan			
-----				-----			
shan1-1.5	shan1-1	shan1-0.5	shan1-0.1				
shan2-3	shan**						
=====				=====			
Frequency B-Spline				fbsp			
-----				-----			
fbsp1-1-1.5	fbsp1-1-1	fbsp1-1-0.5	fbsp2-1-1				
fbsp2-1-0.5	fbsp2-1-0.1	fbsp**					
=====				=====			
Complex Morlet				cmor			
-----				-----			
cmor1-1.5	cmor1-1	cmor1-0.5	cmor1-1				
cmor1-0.5	cmor1-0.1	cmor**					
=====				=====			

Gambar 2.4 Tingkatan atau orde keluarga *wavelet*
(Sumber: Mathworks, 2016)

2.5.1 Wavelet Haar

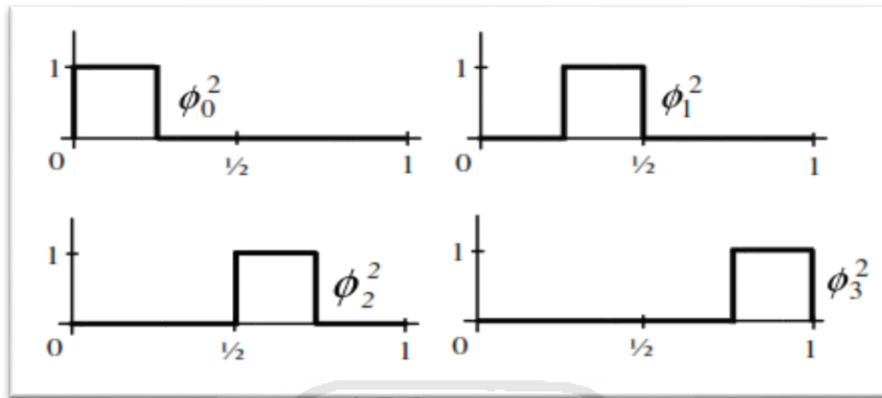
Wavelet haar merupakan salah satu tipe *wavelet* yang paling sederhana yang dapat diterapkan pada transformasi signal (1-dimensi) dan *transformasi* pada citra (signal 2-dimensi). *Transformasi Wavelet Haar 1-Dimensi* Fungsi basis ruang V^j disebut dengan *fungsi skala (Scaling function)* yang dilambangkan dengan ϕ . Salah satu fungsi basis sederhana dinyatakan dengan persamaan berikut :

$$\phi_i^j(x) := \phi(2^j x - i), i = 0, \dots, 2^j - 1 \dots \dots \dots (2.2)$$

dengan

$$\phi(x) := \begin{cases} 1, & \text{untuk } 0 \leq x \leq 1/2 \\ 0, & \text{untuk } x \text{ yang lainnya} \end{cases}$$

Fungsi diatas disebut juga sebagai fungsi kotak (*boxfunction*). Sebagai contoh fungsi kotak pembentuk basis dalam ruang V^2 akan terdapat 2^j atau $2^2 = 4$ potongan fungsi konstan pada jangkauan $[0,1]$ seperti terlihat pada Gambar 2.5 berikut ini.



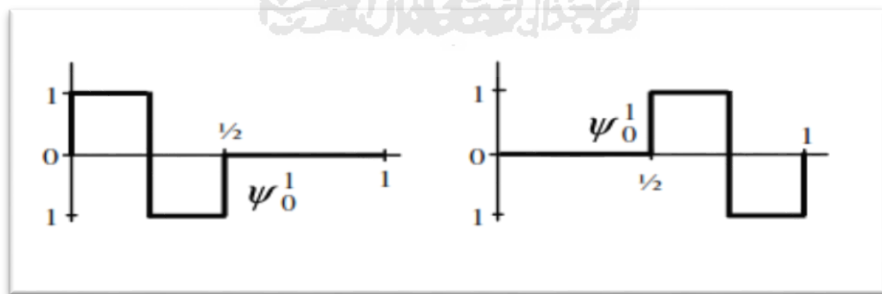
Gambar 2.5 Fungsi basis pada ruang V^2

Fungsi *wavelet* yang sesuai dengan fungsi penyekalaan diatas disebut dengan *wavelet haar*, yang diberikan dengan persamaan:

$$\psi_i^j(x) = \psi(2^j x - i), i = 0, \dots, 2^j - 1 \dots \dots \dots (2.3)$$

dengan

$$\phi(x) := \begin{cases} 1, & \text{untuk } 0 \leq x \leq 1/2 \\ -1, & \text{untuk } 1/2 \leq x \leq 1 \\ 0, & \text{untuk } x \text{ yang lainnya} \end{cases}$$



Gambar 2.6 Bentuk sinyal *wavelet haar* pada pada $W1$

Jika fungsi basis ini dinormalisasi, maka persamaan 2.2 dan persamaan 2.3 akan menjadi seperti persamaan 2.4 berikut :

$$\begin{aligned} \phi(x) &:= 2^{\frac{j}{2}} \phi(2^j x - i) \\ \psi(x) &:= 2^{\frac{j}{2}} \psi(2^j x - i), \dots \dots \dots (2.4) \end{aligned}$$

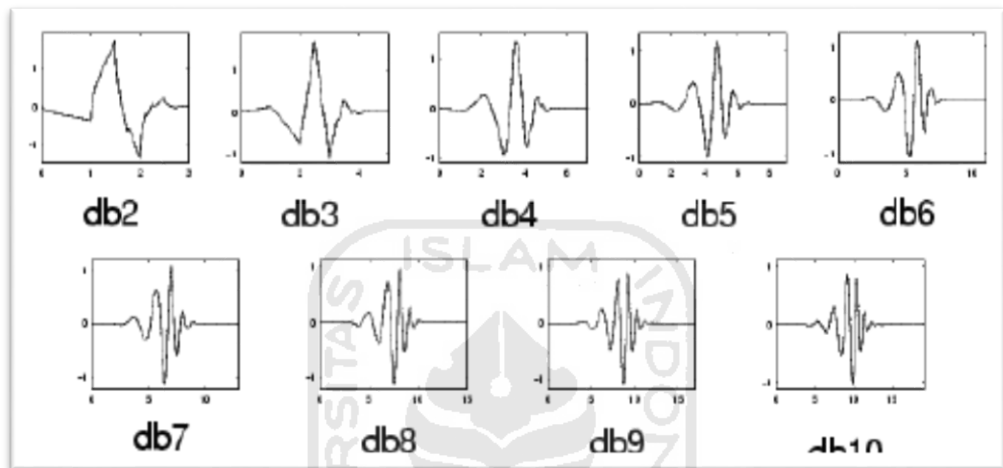
Hasil *Low Pass Filter* (LPF) disebut dengan *scale function* dan hasil *Hight Pass Filter* (HPF) disebut *wavelet function* atau *wavelet haar* yang telah dinormalisasi adalah *Low Pass Filter* (ϕ) = [0.7071 0.7071], dan *Hight Pass Filter* (ψ) = [-0.7071 0.7071].

Sedangkan *Transformasi Wavelet Haar* pada citra merupakan transformasi sinyal 2-dimensi atau citra dalam mentransformasikan nilai-nilai pikselnya dapat dilakukan dengan dua cara, yaitu metode dekomposisi standar (*standard decompositions*), dan metode dekomposisi tidak standar (*non standard decompositions*). Dekomposisi standar dimulai dengan *transformasi wavelet* 1-dimensi untuk setiap baris dari setiap nilai piksel citra hingga level yang diinginkan. Kemudian *transformasi wavelet* 1-Dimensi untuk setiap kolomnya. Sedangkan pada dekomposisi tidak standar, *transformasi wavelet* 1-dimensi untuk baris kemudian *transformasi wavelet* 1-dimensi untuk kolom levelnya, hal ini dilakukan berulang setiap levelnya hingga level yang diinginkan dengan panjang tapis *wavelet* adalah 2. *Wavelet haar* memiliki *scaling function* dengan koefisien $c_0 = c_1 = 1$.

2.5.2 Wavelet Daubechies

Wavelet daubechies merupakan salah satu jenis *wavelet orthogonal*. *Daubechies* biasanya disimbolkan dengan dbN dengan N adalah angka indeks mulai dari 1 sampai 20. *Wavelet daubechies* memiliki orde dimana orde pada *daubechies* menggambarkan jumlah koefisien filternya. Sebagaimana diketahui proses *filtering* oleh *low pass filter* (*scaling fuction*) akan menghasilkan koefisien subband dengan frekuensi rendah. Sebaliknya *filtering* dengan *high pass filter* (*wavelet function*) akan menghasilkan subband dengan frekuensi tinggi. *Wavelet daubechies* memiliki properti yang dimanakan *vanishing moment*. *Vanishing moment* menunjukkan kemampuan *wavelet* dalam merepresentasikan sifat *polinomial*. Suatu *wavelet daubechies* dengan ordo *wavelet* N , memiliki nilai *vanishing moment* sama dengan N . Sifat *polinomial* yang dimiliki oleh *wavelet* akan berpengaruh dalam penentuan jumlah koefisien filter *wavelet*. Semakin besar jumlah filter yang dimiliki oleh suatu *wavelet filter daubechies*, maka semakin baik

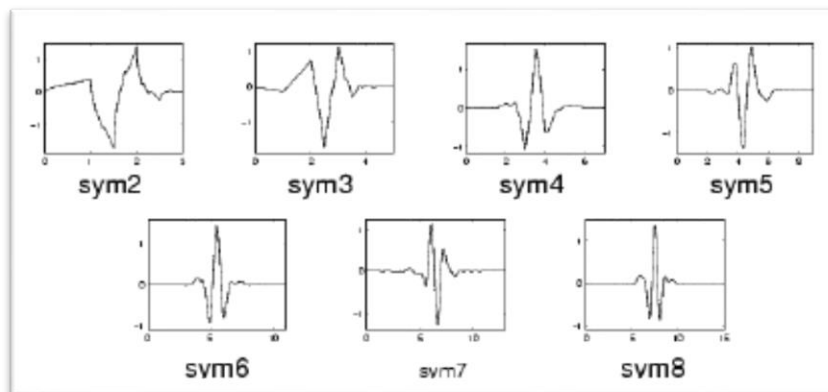
filter tersebut dalam melakukan pemilihan frekuensi. Untuk *daubechies* dengan ordo N (dbN), maka *daubechies* tersebut memiliki ukuran koefisien filter $2N$ sehingga panjang tapis *wavelet daubechies* adalah $2N$. *Wavelet daubechies* dengan 4 koefisien (DB4) memiliki *scaling function* dengan koefisien $c_0 = (1+\sqrt{3})/4$, $c_1 = (3+\sqrt{3})/4$, $c_2 = (3-\sqrt{3})/4$, $c_3 = (1-\sqrt{3})/4$. Bentuk sinyal dari *wavelet daubechies* dari berbagai orde dapat dilihat pada Gambar 2.7.



Gambar 2.7 Bentuk sinyal dari *wavelet daubechies*

2.5.3 Wavelet Symlets

Wavelet symlets dikenal dengan singkatan *sym* memiliki panjang tapis adalah $2N$. Sebagai contoh *sym4* memiliki panjang tapis 8 dan *sym10* memiliki panjang tapis 20. Bentuk sinyal dari *wavelet symlets* dari berbagai orde dapat dilihat pada Gambar 2.8.



Gambar 2.8 Bentuk sinyal dari *wavelet symlets*

2.6 Transformasi Wavelet

Transformasi wavelet merupakan pengubahan sinyal dalam berbagai *wavelet basic* dengan berbagai pergeseran (translasi) dan penyekalaan (dilatasi). Oleh karena itu, koefisien *wavelet* dari beberapa skala atau resolusi dapat dihitung dari koefisien *wavelet* pada resolusi tinggi berikutnya. Hal ini memungkinkan mengimplementasikan *transformasi wavelet* menggunakan struktur pohon yang dikenal sebagai algoritma piramida (*pyramid algorithm*).

Menurut Sutarno (2010), alasan mengapa *transformasi wavelet* menjadi begitu penting dalam berbagai bidang pengolahan citra adalah karena sifat-sifat berikut:

- a. waktu kompleksitasnya bersifat linear. *Transformasi wavelet* dapat dilakukan dengan sempurna dengan waktu yang bersifat linear;
- b. koefisien-koefisien *wavelet* yang terpilih bersifat jarang. Secara praktis, koefisien-koefisien *wavelet* kebanyakan bernilai kecil atau nol. Kondisi ini sangat memberikan keuntungan terutama dalam bidang kompresi atau pemampatan data; dan
- c. *wavelet* dapat beradaptasi pada berbagai jenis fungsi, seperti fungsi tidak kontinu, dan fungsi yang didefinisikan pada domain yang dibatasi.

Transformasi wavelet merupakan suatu proses pengubahan data dalam bentuk lain agar lebih mudah dianalisis. *Transformasi wavelet* menghasilkan energi citra yang terkonsentrasi pada sebagian kecil koefisien *transformasi* dan kelompok lain yang mengandung sedikit energi. Secara garis besar *transformasi wavelet* dapat dibagi menjadi dua jenis yaitu *Cosinus Wavelet Transform (CWT)* dan *Discrete Wavelet Transform (DWT)*. Cara kerja *Cosinus Wavelet Transform (CWT)* yaitu dengan menghitung konvolusi sebuah sinyal dengan jendela modulasi pada setiap waktu dengan setiap skala yang diinginkan. Jendela modulasi yang fleksibel inilah yang biasanya disebut dengan fungsi dasar *wavelet* atau *mother wavelet*. Sedangkan cara kerja *Discrete Wavelet transform (DWT)* dengan cara mendapatkan representasi waktu dan skala dari sebuah sinyal dengan menggunakan teknik pemfilteran sinyal digital dan operasi *sampling*.

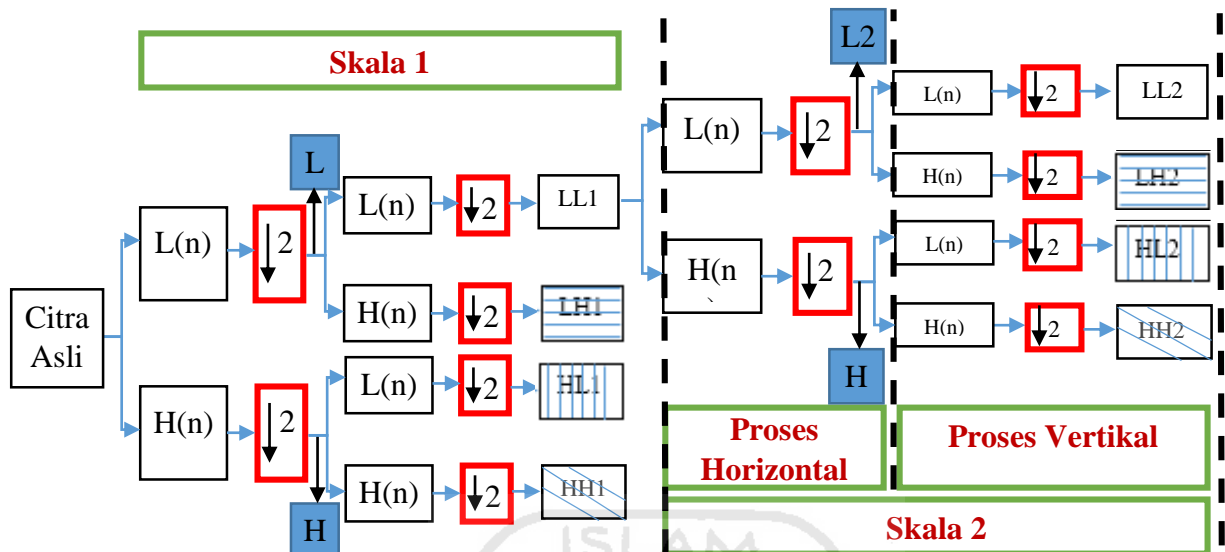
2.7 Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform (DWT) merupakan salah satu metode yang sangat terpakai dan sangat baik digunakan untuk representasi dan analisis sinyal diskret. Kelebihan dari DWT adalah pada saat yang bersamaan dapat memberikan informasi frekuensi dan informasi temporal, berbeda dengan DFT dan DCT yang hanya memberikan informasi frekuensi. Sehingga, DWT lebih sering digunakan untuk analisis *time-frequency* dari sebuah sinyal.

Transformasi wavelet diskret adalah sembarang bentuk gelombang atau sinyal elektrik yang disampel secara diskret kemudian dilakukan proses transformasi. DWT pertama kali diperkenalkan oleh seorang ahli matematika Hungaria bernama Alfred Haar. Untuk sederetan data sinyal masukan yang direpresentasikan dalam bilangan numerik 2^n , transformasi wavelet Haar ini menyatakan bahwa setiap pasang bilangan data masukan itu dapat dinyatakan dalam satu nilai selisih dan satu nilai jumlah antara keduanya. Bagian deretan nilai selisih dapat disimpan dan bagian deretan nilai jumlah dapat digunakan untuk proses transformasi yang sama, sehingga proses transformasi dapat dilakukan secara berulang atau rekursif. Transformasi secara rekursif ini disebut sebagai *multiscale transform* (transformasi multi-skala), dimana pada akhir proses transformasi ini akan menghasilkan $2^n - 1$ bagian nilai selisih dan satu bagian nilai jumlah.

Dalam DWT, bagian nilai jumlah dinyatakan sebagai *koefisien aproksimasi* yang berisikan informasi global atau komponen frekuensi rendah yang terdapat dalam sebuah citra, sedangkan bagian nilai selisih dikenal sebagai *koefisien wavelet* yang berisikan informasi detail atau informasi frekuensi tinggi.

Gambar 2.9 memperlihatkan transformasi wavelet multi-skala dari sebuah citra.



Gambar 2.9 Struktur proses *transformasi wavelet* multi-skala
(Sumber: Madenda, 2015).

Proses transformasi dimulai dari penghitungan secara bersamaan terhadap komponen dari citra asli: komponen frekuensi rendah dengan fungsi *low-pass filter* $L(n)$, komponen frekuensi tinggi dengan fungsi *high-pass filter* $H(n)$. Proses ini dilakukan pada arah *horizontal* antara dua piksel atau lebih, dimulai dari baris pertama dari kolom pertama hingga kolom terakhir, kemudian dilanjutkan ke garis berikutnya hingga baris terakhir. Simbol operator *downsampling* $\downarrow 2$ bermakna bahwa untuk setiap dua kolom citra asli akan menghasilkan satu kolom setelah proses. Dengan demikian, keluaran dari filter $L(n)$ adalah citra $L1$ yang berisikan informasi global yang mirip dengan citra aslinya namun jumlah kolomnya menjadi setengahnya, sedangkan keluaran dari filter $H(n)$ adalah citra $H1$ yang mengandung informasi detail dengan jumlah kolom yang sama dengan jumlah kolom citra keluaran filter $L(n)$.

Tahap selanjutnya adalah citra $L1$ dan $H1$ masing-masing di proses kembali dengan menggunakan filter $L(n)$ dan $H(n)$ yang sama. Proses ini dilakukan pada arah *vertikal* antara dua piksel atau lebih, mulai dari baris pertama hingga terakhir pada kolom pertama dan selanjutnya kolom berikutnya hingga kolom terakhir. Simbol $\downarrow 2$ pada arah *vertikal* menunjukkan bahwa, untuk setiap dua baris citra asli, hasil prosesnya hanya tinggal satu baris (jumlah baris dibagi 2). Dengan demikian:

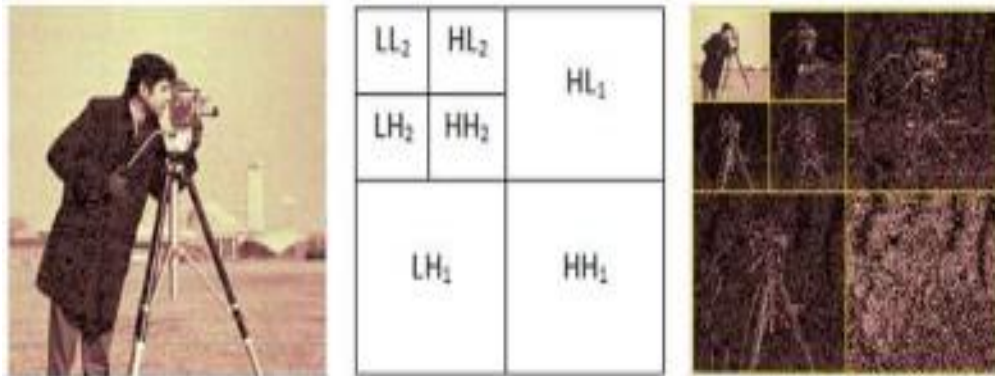
1. Untuk citra masukan citra L_1
 - Keluaran dari filter $L(n)$ adalah citra LL_1 yang berisikan informasi global yang mirip dengan citra L_1 namun jumlah barisnya menjadi setengahnya.
 - Keluaran dari filter $H(n)$ adalah citra LH_1 yang mengandung informasi detail dengan jumlah baris yang sama dengan filter $L(n)$.
2. Untuk citra masukan citra H_1
 - Keluaran dari filter $L(n)$ adalah citra HL_1 yang jumlah barisnya menjadi setengahnya.
 - Keluaran dari filter $H(n)$ adalah citra HH_1 yang jumlah barisnya sama dengan jumlah HL_1 .

Citra LL_1 adalah citra hasil *Low-Pass Filter* dari citra asli, citra LH_1 adalah hasil *low-pass high-pass filter* yang merupakan informasi detail arah *horizontal* dari citra asli. Citra HL_1 adalah hasil dari *high-pass low-pass filter* yang merupakan informasi detail arah vertikal dari citra asli, dan citra HH_1 adalah hasil *high-pass high-pass filter* yang merupakan informasi detail arah diagonal dari citra asli.

LL_1 <i>(Aproximation)</i>	HL_1 <i>(Vertical Detail)</i>
LH_1 <i>(Horizontal Detail)</i>	HH_1 <i>(Diagonal Detail)</i>

Gambar 2.10 Struktur representasi skala 1 DWT
(Sumber: Madenda, 2015).

Sedangkan untuk implementasi pada citra digital dengan menggunakan proses level 2 *Discrete Wavelet Transform (2D-DWT)* dapat diilustrasikan seperti gambar 2.11 dibawah ini.



Gambar 2.11 Implementasi Skala 2 *Discrete Wavelet Transform (2D-DWT)*
(Sumber: Hidayat & Udayanti, 2011)

1. Proses DWT skala 1 dari citra asli menghasilkan empat citra yaitu LL_1 , LH_1 , HL_1 dan HH_1 masing-masing dengan ukuran $\frac{1}{4}$ ($\frac{1}{2}$ kolom * $\frac{1}{2}$ baris) dari ukuran citra aslinya.
2. Untuk menghitung DWT skala 2 dilakukan tahapan proses yang sama dengan skala 1. Selanjutnya hasil proses DWT skala 2 juga menghasilkan empat citra yaitu: LL_2 , LH_2 , HL_2 dan HH_2 masing-masing dengan ukuran $\frac{1}{4}$ dari pada citra masukannya LL_1 .
3. Demikian seterusnya untuk DWT skala ke- j , citra masukannya adalah citra LL_{j-1} .

Indeks j ($j > 0, j \in \mathbb{Z}$) akan terus digunakan sebagai penunjuk skala, dimana $j=0$ adalah skala citra asli proses DWT. Ide dasar *wavelet haar* di atas dapat dituangkan dalam formulasi matematika 1-D berikut: andaikan sederetan N data $X = (x_1, x_2, x_3, \dots, x_n)$ dimana $N = 2^i$ dan $i > 1, i \in \mathbb{Z}$. Deretan nilai jumlah (bersimbol L) dan deretan nilai selisih (bersimbol H) dari data X ini dapat dituliskan sebagai berikut (Madenda, 2015):

$$L(n) = \frac{X(2n-1) + X(2n)}{2}, n = 1, \dots, \frac{N}{2} \dots \dots \dots (2.5)$$

$$H(n) = \frac{X(2n-1) - X(2n)}{2}, n = 1, \dots, \frac{N}{2} \dots \dots \dots (2.6)$$

Dengan demikian, *transformasi wavelet haar* terhadap data X dapat dituliskan sebagai berikut:

$$X = (x_1, x_2, x_3, \dots, x_n) \rightarrow (L/H) = (l_1, l_2, l_3, \dots, l_n | h_1, h_2, h_3, \dots, h_{N/2}) \dots \dots (2.7)$$

2.8 Redundant Discrete Wavelet Transform (RDWT)

Discrete Wavelet Transform (DWT) merupakan salah satu metode *watermarking* yang paling umum digunakan untuk proteksi pada citra digital karena kemampuannya dalam hal membagi citra digital sesuai dengan subband frekuensi sehingga bisa mengetahui bagian pada citra yang cocok untuk disisipkan *watermark*. Pada DWT memiliki sedikit kekurangan atau kelemahan akibat proses *downsampling* yang mengakibatkan pergeseran invariant sehingga mengurangi kualitas citra digital sebagai *file host* dan mengakibatkan pergeseran lokasi *watermark* pada subband *file host* sehingga pada saat ekstraksi tingkat akurasinya menjadi berkurang. Untuk mengatasi masalah tersebut *Redundant Discrete Wavelet Transform* (RDWT) coba diimplementasikan pada metode *watermarking*.

Redundant Discrete Wavelet Transform (RDWT) merupakan pergeseran invariant dan redundansi tingkat spasial sinyal *sampling* dan ukuran masing-masing *subband*. Pada *Redundant Discrete Wavelet Transform* (RDWT) memiliki ukuran *output* yang sama dengan sinyal input sehingga menyebabkan proses ekstraksi *watermark* dari *file host* menjadi lebih akurat karena *watermark* tidak mengalami pergeseran (Lagzian et al., 2011).

Pada RDWT proses *downsampling* dan *upsampling* dihilangkan, sehingga tidak ada perubahan koefisien pada setiap *subband* pada tingkat iterasi dan menjadikan *watermark* pada *file host* tidak mengalami pergeseran sehingga *shift sensitifitas* dan *poor directory* dapat diredudansi.

2.9 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) merupakan teknik komputasi numerik yang melakukan faktorisasi terhadap sebuah matriks tak nol sehingga diperoleh tiga matriks tak nol. Salah satu matriks yang diperoleh dari proses SVD akan memuat nilai-nilai *singular* dari matriks asal. Istilah “nilai singular” menyatakan jarak antara sebuah matriks dan himpunan matriks-matriks singular. Nilai-nilai singular berguna untuk suatu matriks yang merupakan *transformasi* dari sebuah ruang vektor ke ruang vektor yang lain atau dimensi berbeda. Sistem persamaan aljabar yang *overdetermined* atau *undetermined* adalah contoh *transformasi* ruang vektor berbeda.

Alkiviadis dan Gennadi (2002), memaparkan tiga karakteristik SVD, yaitu matriks *hanger* (U), *stretcher* (S) dan *aligner* (V). Matriks U merupakan matriks *hanger*, matriks S adalah matriks *stretcher* dan matriks V menjadi matriks *aligner*. Dekomposisi dari matriks A $m \times n$ piksel dengan menggunakan *Singular Value Decomposition* (SVD) dinyatakan dengan persamaan :

$$A = U S V^T \dots\dots\dots(2.8)$$

Pada dasarnya matriks apapun dapat dibagi menjadi ketiga matriks tersebut. Matriks *hanger* (U) dan *aligner* (V) merupakan matriks orthogonal yaitu matriks U merupakan matriks orthogonal $m \times m$ dan V merupakan matriks orthogonal $n \times n$, maka perkalian dengan kedua matriks tersebut tidak akan mengubah bentuk objek karena kedua matriks itu akan mempertahankan bentuk objek. Sementara itu, matriks *stretcher* (S) merupakan matriks diagonal $m \times n$ yang bernilai riil tak negatif yang disebut dengan nilai-nilai singular. Perkalian dengan matriks *stretcher* (S) akan merentangkan suatu objek. Jika suatu kurva dikalikan dengan matriks diagonal (S), semua nilai pada kurva sepanjang sumbu x dan y akan terentang. Misal, jika terdapat sebuah kurva berbentuk lingkaran, setelah kurva tersebut dikalikan dengan matriks *stretcher* (S), bentuk kurva akan menjadi elips. Untuk mengembalikan kurva ke bentuk asalnya, dapat dilakukan perkalian dengan matriks yang komponen-komponennya merupakan kebalikan dari matriks *stretcher* (S).

Sifat ketiga matriks tidak hanya berlaku saat matriks-matriks tersebut dikalikan dengan objek lain saja, namun juga dalam pembentukan suatu objek.

Ketiga karakteristik SVD tersebut dimanfaatkan dalam pengolahan citra digital. Asumsi yang digunakan yaitu sebuah citra digital $m \times n$ piksel diterjemahkan sebagai sebuah matriks berukuran $m \times n$. Informasi yang disimpan dalam matriks tersebut bilangan-bilangan yang menyatakan intensitas warna tiap pixel citra. Proses SVD pada matriks citra akan mendekomposisi matriks tersebut menjadi tiga matriks baru. Masing-masing matriks memuat informasi spesifik tentang citra yang diproses.

2.10 Parameter dalam *Watermarking*

2.10.1 PSNR (*Peak Signal to Noise*)

Peak Signal to Noise Ratio (PSNR) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya *noise* yang berpengaruh pada sinyal tersebut. PSNR merupakan parameter standar untuk menilai kualitas suatu citra secara obyektif dengan membandingkan *noise* terhadap sinyal puncak dengan satuannya dalam desibel (dB). PSNR sering digunakan untuk mengukur kualitas file gambar asli dengan file gambar terkompresi dalam hal ini file gambar hasil *watermarking*. Pada *watermarking* PSNR digunakan untuk menentukan tingkat *imperceptibility* suatu citra digital sebelum dan sesudah proses *watermarking*. Semakin tinggi nilai PSNR maka semakin baik kualitas hasil proses *watermarking*. Citra digital hasil proses *watermarking* yang baik harus memiliki nilai PSNR diatas 35dB. Untuk menghitung nilai PSNR, terlebih dahulu harus menghitung nilai MSE (*Mean Squared Error*) dengan menggunakan persamaan:

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - g(x, y))^2 \dots\dots\dots(2.9)$$

Keterangan :

$f(x,y)$ = nilai piksel citra asli

$g(x,y)$ = nilai piksel citra hasil proses *watermarking*

M, N = merupakan dimensi citra

Dengan menggunakan nilai MSE yang didapatkan dengan Rumus 2.9, maka untuk menghitung nilai PSNR menggunakan persamaan berikut:

$$\text{PSNR} = 20 \cdot \log_{10} \left(\frac{255}{\sqrt{\text{MSE}}} \right) \dots \dots \dots (2.10)$$

Pada persamaan diatas nilai 255 merupakan *level* citra atau nilai fluktuasi maksimum untuk jenis input citra digital. Jika citra input memiliki ganda-presisi floating point data maka nilai fluktuasinya adalah 1, dan jika citra inputan memiliki 8 bit data *integer* jenis *unsigned* maka nilai fluktuasinya adalah 255. Nilai MSE yang tinggi menandakan kualitas citra hasil proses *watermarking* kurang baik dan sebaliknya nilai yang tinggi PSNR menandakan kualitas citra hasil proses *watermarking* sangat baik dalam tingkat *imperceptibility*.

2.10.2 *Perceptual Quality Matric (Q)*

Merupakan rating kualitas citra digital dengan rentang dari 1 sampai dengan 5. Nilai reting *excellent* (5) menunjukkan kualitas citra hasil proses *watermarking* memiliki kualitas yang tinggi atau *imperceptibility*.

Untuk menghitung nilai *Perceptual Quality Matric (Q)* menggunakan rumus sebagai berikut:

$$Q = \frac{5}{1 + N \times E} \dots \dots \dots (2.11)$$

E adalah ukuran distorsi dan N adalah konstanta Normalisasi, nilai N standar [0.01 0.03] . Hasil yang didapatkan dibandingkan tabel kualitas citra dari ITU – R Rec 500 dengan skala 1-5 seperti tabel 2.1.

Tabel 2.1 Reting kualitas citra digital

Reting	Impairment	Quality
5	Imperceptibility	Excellent
4	Perceptibility, not annoying	Good
3	Slightly annoying	Fair
2	Annoying	Poor
1	Very Annoying	Bad

(Sumber: Prayudi, 2002)

2.10.3 *Corr*

Corr merupakan nilai korelasi hasil ekstraksi citra digital hasil proses *watermarking*. Nilai korelasi yang sempurna adalah 1 (satu), semakin mendekati nilai 1 (satu) maka nilai korelasi dianggap sempurna dan nilai minimal untuk *Corr* yang masih bisa diterima adalah 0,8. Untuk menghitung nilai *Corr* dapat menggunakan persamaan atau Rumus 2.15.

$$\text{correlation} = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2} \sqrt{\sum(y - \bar{y})^2}} \dots \dots \dots (2.12)$$

x = file host (matriks), y = *watermarked image/ ekstraksi image/ detection watermark* (matrix), \bar{x} = `mean2 (x)`, \bar{y} = `mean2 (y)`, *Correlation* atau *Corr* digunakan untuk mengetahui korelasi atau kesamaan citra hasil proses *watermarking* yaitu dengan *watermarked image* (citra hasil penyisipan *watermark*) dan *ekstraksi image* (citra hasil ekstraksi *watermark* dari *attacked iamge*).

2.11 *Attack pada Watermarking*

Serangan atau *attack* pada *watermarking* merupakan upaya oleh pihak lain untuk menghilangkan identitas kepemilikan citra berupa *watermark* dengan cara manipulasi citra hasil proses *watermarking* menyebabkan *watermark* rusak atau hilang sehingga tidak bisa terdeteksi lagi. Metode *watermarking* dikatakan *robust* jika tahan terhadap berbagai serangan atau *attack*. Jenis-jenis *attack* atau serangan pada *watermarking* dapat berupa:

a. **Kompresi**

Kompresi citra digital digunakan untuk memperkecil ukuran citra dengan tujuan supaya citra yang ditransmisikan lebih cepat. Kompresi yang berlebihan akan mengakibatkan rusaknya *watermark* yang disisipkan pada *file host*.

b. **Filtering**

Merupakan cara untuk menurunkan kualitas citra dengan menggunakan filter seperti *blur*, *distort*, *pixel*, *sharpen* dan *stylize* sehingga mengakibatkan *watermark* yang ada dalam citra menjadi kabur dan bahkan bisa tidak terdeteksi.

c. *Color Reduce*

Color Reduce merupakan proses pengurangan jumlah warna dengan cara mengambil beberapa perwakilan warna yang dapat membedakan satu bagian dengan bagian yang lain pada citra. Pengurangan warna pada citra hasil proses *watermarking* mengakibatkan warna pada *watermark* ikut berkurang sehingga menurunkan kualitas *watermark*.

d. *Noise Addition*

Penambahan *noise* secara *uniform* maupun *gaussian* biasanya digunakan untuk mengaburkan gambar. Semakin tinggi *noise* yang diberikan maka kualitas citra hasil proses *watermarking* dan *watermark* akan turun bahkan bisa mengakibatkan rusaknya *watermark*.

e. *Overmarking*

Usaha-usaha menghilangkan identitas pemilik file seperti pemberian *watermarking* ulang yang dilakukan oleh pihak lain menyebabkan *watermark* yang terdapat pada file gambar atau citra digital sulit untuk dideteksi kembali pada saat diekstraksi.

f. *Remodulation*

Merupakan type serangan dengan cara mereduksi citra hasil proses *watermarking* atau *watermarked image*, dengan mereduksi *watermarked image* maka *watermark* yang ada didalam *watermarked image* juga ikut tereduksi sehingga kualitas *watermark* bisa menurun.

g. *MAP*

Merupakan proses *filtering noise* pada citra digital, penyisipan *watermark* pada *file host* dianggap sebagai *noise*. Jika dilakukan proses *filtering noise* pada *watermarked image* maka *watermark* akan terfilter dan kemungkinan *watermark* bisa terpisah dari *watermarked image*.

h. *JPEG*

Kompresi citra berdasarkan faktor kualitas, jpeg 100 adalah kompresi yang menghasilkan kualitas citra yang paling tinggi. Sedangkan pengurangan faktor kualitas misalnya jpg 80 akan menghasilkan kualitas citra yang

penurunan kualitas sebesar 20% dari citra asli. Semakin kecil faktor kompresi maka kemungkinan rusaknya *watermark* akan semakin tinggi.

i. *ML*

Merupakan *filtering* untuk tingkat kebisingan *noise* pada citra, sehingga menghasilkan citra yang lebih kabur. Penurunan kualitas citra hasil proses *watermarking* akan diikuti pula oleh penurunan kualitas citra *watermark*.

j. *Resample*

Citra diproses dengan *downsampling* dan *upsampling*. Dengan adanya banyak iterasi mengakibatkan pergeseran posisi *watermark* pada *file host* sehingga pada proses ekstraksi *watermark* mengalami penurunan kualitas.

k. *Copy*

Citra asli digabung dengan berbagai citra lain yang merupakan hasil *attack* seperti citra yang dirotasi, diskala dan dicroping sehingga menghasilkan citra yang besar. Penggabungan citra yang banyak akan menumpuk *watermark* yang ada pada citra digital dan mengakibatkan tergantinya *watermark* dengan citra lain.

l. *Wavelet*

Citra dikompresi dengan berbagai *bitrates* menggunakan standar kompresi JPEG2000 dikenal juga dengan istilah kompresi JASPER. Semakin tinggi tingkat kompresi dengan JPEG2000 mengakibatkan kualitas *watermark* menurun.

m. *Rotation Scala*

Citra dirotasi pada sudut tertentu kemudian dicroping untuk menghapus bagian yang hitam kemudian diskalakan lagi seperti citra asli. Akibat proses cropping bisa mengakibatkan terpotongnya *watermark* yang disisipkan pada *file host*.

2.12 Tingkat Ekstraksi dan Deteksi *Watermark*

Tingkat ekstraksi merupakan seberapa banyak *watermark* yang dapat diekstrak dari *file host* setelah dilakukan manipulasi dan *attack* terhadap citra hasil proses *watermarking*. Sedangkan deteksi menunjukkan tingkat keberhasilan *watermark* untuk dideteksi berdasarkan standar tertentu yaitu dengan nilai *Corr* diatas 0.8. Berdasarkan data pada situs *checkmark* (2001), tingkat ekstraksi dan deteksi *watermark* dari *attacked image* untuk metode Wang menghasilkan tingkat ekstraksi *watermark* sebesar 61% dan deteksi *watermark* sebesar 74%, metode Cox menghasilkan tingkat ekstraksi sebesar 67% dan deteksi 90%, metode Xia menghasilkan tingkat ekstraksi sebesar 67% dan deteksi 84% dan metode Kim menghasilkan tingkat ekstraksi sebesar 38% dan deteksi sebesar 48%.

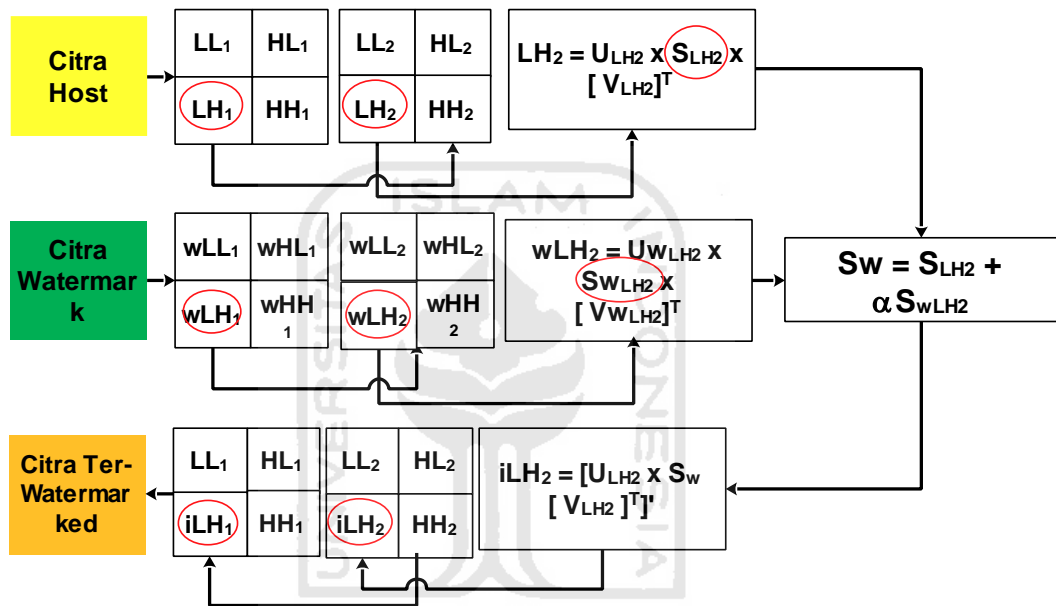


BAB III METODOLOGI PENELITIAN

3.1 Rancangan Algoritma *Hybrid Image Watermarking* RDWT-SVD

3.1.1 Diagram Proses Penyisipan RDWT-SVD

Pada penelitian ini rancangan algoritma penyisipan *watermark* dalam perlindungan hak cipta dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram proses penyisipan *Watermarking* RDWT-SVD

Prosedur penyisipan *watermark* pada *file host* dengan teknik RDWT-SVD dijelaskan sebagai berikut.

1. Menentukan nilai *alpha*
2. Menentukan citra digital yang akan dijadikan *cover watermarking* atau *cover image* dan *watermark*;
3. Dekomposisi *cover image* dan *watermark* menjadi empat bagian dengan RDWT untuk memperoleh koefisien atau *subband* LL_1, LH_1, HL_1, HH_1 ;
4. Pilih koefisien horizontal (LH_1) untuk di dekomposisi lagi jadi empat *subband* untuk mendapatkan empat *subband* LL_2, LH_2, HL_2, HH_2 ;
5. Pilih detail horizontal (LH_2) pada *cover image* dan citra *watermark* sebagai tempat penyisipan *watermark* dan yang disisipkan;

6. Terapkan SVD pada detail horizontal (LH_2) pada *file host* dengan rumus :

$$I^l = U^l S^l V^{lT}$$
7. Untuk citra *watermark* juga diproses dengan operasi SVD :

$$W = U^w S^w V^{wT}$$
8. Sisipkan *watermark* pada *cover image* dengan algoritma penyisipan dengan menambahkan *factor alpha* pada *watermark* ;

$$S_{em} = I_s + Alpha * Ew_s$$

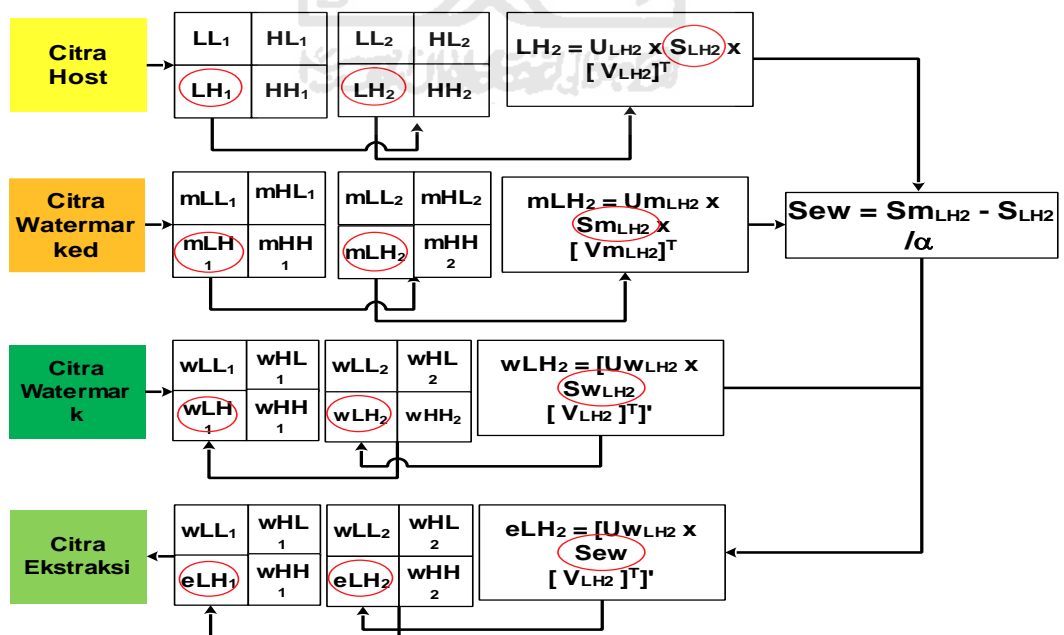
$$I_s = \text{file host}$$

$$S_{em} = \text{watermarked image}$$

$$Ew_s = \text{watermark}$$
9. Lakukan *Inverse SVD* pada *watermarked image* dengan modifikasi nilai singular;

$$I^{*l} = U^l S^* V^{lT}$$
10. Lakukan proses *Invers RDWT* untuk merekonstruksi *watermarked image* yang berupa fungsi menjadi *watermarked image* yang berupa file gambar atau citra digital.

3.1.2 Diagram Proses Ekstraksi RDWT-SVD



Gambar 3.2 Diagram proses ekstraksi *watermark*

Prosedur ekstraksi *watermark* dari citra digital (*file host*) untuk metode RDWT-SVD dijelaskan sebagai berikut.

1. Menentukan nilai *alpha*;
2. Pilih citra *watermarked*, *cover image* dan *watermark* diproses dengan RDWT untuk mendapatkan *subband* LL₁, LH₁, HL₁, HH₁;
3. Pilih koefisien horizontal (LH₁) untuk di dekomposisi lagi jadi empat *subband* untuk mendapatkan empat *subband* LL₂, LH₂, HL₂, HH₂;
4. Pilih detail horizontal (LH₂);
5. Terapkan SVD pada detail horizontal (LH₂) pada *file host* dengan rumus:

$$I^l = U^l S^l V^{lT}$$
6. Terapkan SVD pada detail horizontal (LH₂) pada *watermark*:

$$W = U^w S^w V^{wT}$$
7. Untuk citra *watermarked* juga diproses dengan operasi SVD:

$$WM = U_{wm} S_{wm} V_{wm}^T$$
8. Ekstraksi *watermark* dari *cover image* dengan algoritma ekstraksi:

$$W_{ew} = (S_{wm} - I_s) / \text{Alpha}$$

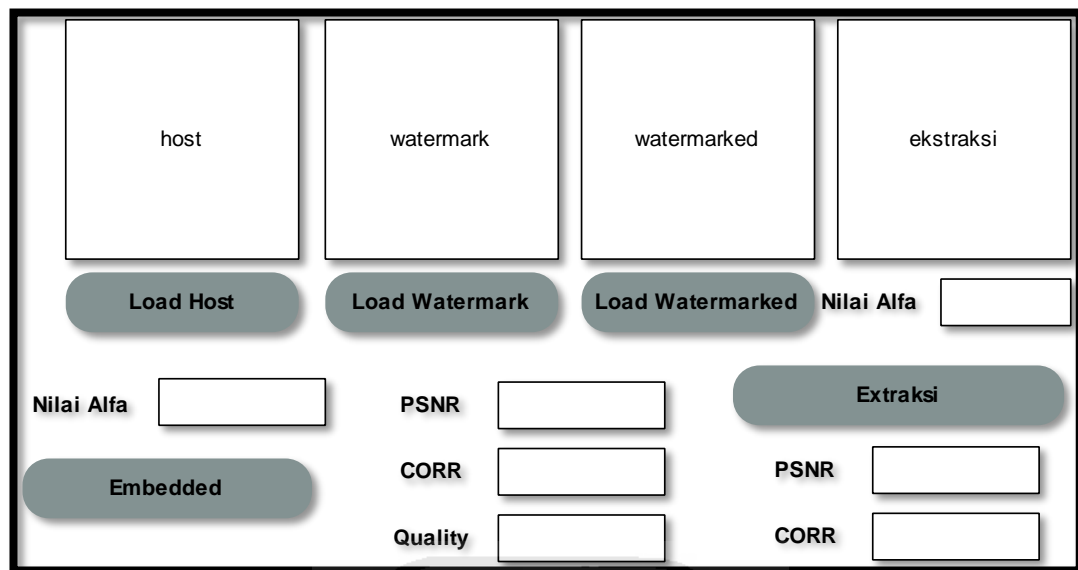
$$I_s = \text{file host}$$

$$W_{ew} = \text{watermark}$$

$$S_{wm} = \text{watermarked image}$$
9. Terapkan *Invers* SVD dengan modifikasi nilai singular untuk memperoleh koefisien frekuensi tinggi dari *watermark* ;
10. Lakukan proses *Invers* RDTW untuk merekonstruksi citra *watermark*.

3.1.3 Desain Antarmuka Pengguna

Rancangan desain antarmuka pengguna merupakan tahap akhir dari perancangan sebuah program. Perancangan antarmuka penggabungan algoritma *Watermarking* RDWT dengan SVD menggabungkan antarmuka proses *embedded* dengan ekstraksi menjadi satu (1). Rancangan antarmuka dari algoritma yang diterapkan dapat di lihat pada gambar 3.3.



Gambar 3.3 Desain antarmuka pengguna.

3.2 Perangkat Pendukung Penelitian

3.2.1 Perangkat Lunak (*Software*)

Dalam penelitian ini perangkat lunak yang digunakan adalah matlab versi R2013b untuk pengolahan dan penyisipan citra digital. Penggunaan Matlab (*Matrix Laboratory*) karena kemampuan program tersebut untuk analisis dan komputansi *numeric* dan merupakan bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks.

Matlab merupakan bahasa pemrograman tingkat tinggi yang bisa digunakan untuk menyelesaikan masalah-masalah seperti matematika, komputasi, pengembangan algoritma dan metode, pemrograman modeling, pemrograman simulasi, pembuatan *prototype*, analisis data, eksplorasi dan visualisasi, analisis numerik dan statistik dan pengembangan aplikasi teknik. Karena Kemampuan komputansi matlab yang sangat baik untuk pengolahan citra digital berbentuk matriks yaitu baris dan kolom ($m \times n$) piksel sehingga memudahkan implemetasi metode *watermarking* untuk proteksi atau perlindungan hak cipta pada citra digital dengan menggunakan metode *Hybrid Image Watermarking*.

3.2.2 Perangkat Keras (*Hardware*)

Sedangkan perangkat keras yang digunakan dalam penelitian ini adalah menggunakan *notebook* lenovo thinkpad dengan spesifikasi sebagai berikut:

1. Processor : Intel(R) Core(TM) i3-3227U CPU @1.90GHz 1.90 GHz;
2. RAM : 6,00 GB;
3. System Type : 64-bit Windows 7 Ultimate
4. Harddisk : 500 GB

3.3 Bahan Penelitian

Pada penelitian ini file gambar atau citra digital yang digunakan diambil dari situs USC-SIPI sebagai berikut :

3.3.1 Citra Digital (*Image*)

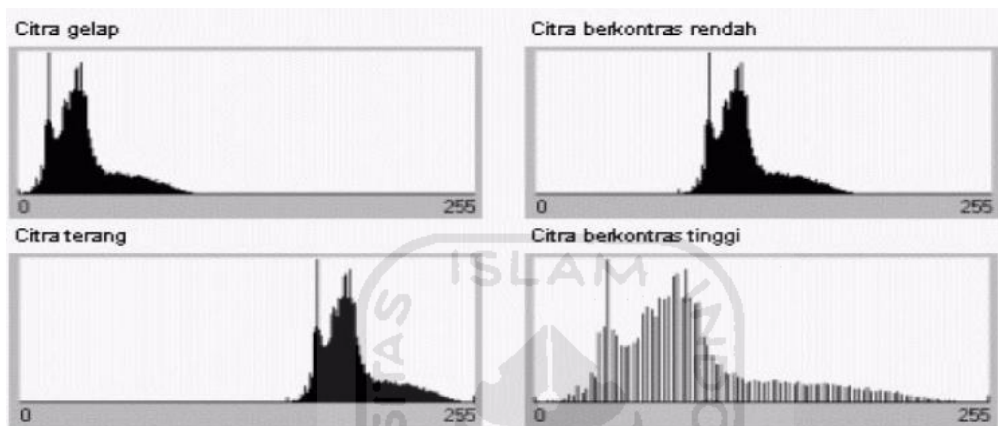
Citra digital standar yang akan digunakan sebagai *file host* dan *watermark* dalam implementasi metode *hybrid ini* diambil dari situs <http://sipi.usc.edu/database/database.php>. USC-SIPI *image* database merupakan kumpulan koleksi citra digital. USC-SIPI adalah salah satu organisasi penelitian di dunia yang didedikasikan untuk pengolahan gambar. USC-SIPI *image* database sudah menyediakan gambar untuk keperluan penelitian sejak tahun 1977 dan banyak citra digital yang ditambahkan secara berkala sampai saat ini. Database dibagi menjadi 4 (empat) kelompok berdasarkan karakteristik dasar citra digital, yaitu *Textures*, *Aerials*, *Miscellaneous* dan *Sequences* dan analisis histogram *image*.

Histogram citra dapat didefinisikan sebagai banyaknya jumlah piksel dalam sebuah citra yang memiliki nilai intensitas yang sama tanpa memperhitungkan dimana posisi piksel berada. Artinya, histogram tidak menggambarkan informasi posisi spasial setiap piksel dalam citra, tetapi lebih kepada gambaran informasi statistik. Sumbu y mempresentasikan jumlah piksel, sedangkan sumbu x adalah tingkat keabuan. Madenda (2015).

Analisis tekstur citra dengan menggunakan algoritma histogram sering dilakukan pada pengolahan citra digital. Tekstur citra berpengaruh terhadap kualitas citra hasil proses *Watermarking* atau tingkat *imperceptibility*. Nilai Pada

PSNR bergantung pada distribusi nilai histogram secara absis (sumbu x) yang menunjukkan nilai level atau intensitas citra (1-255).

Menurut Pratiwi (2014), secara probabilitas kemunculan nilai intansitas, citra dapat dikempokan menjadi citra gelap, citra kosentrasi rendah, citra terang, dan citra berkonsentrasi tinggi. Untuk lebih jelas tentang ke empat jenis citra tersebut dapat dilihat pada gambar 3.4.

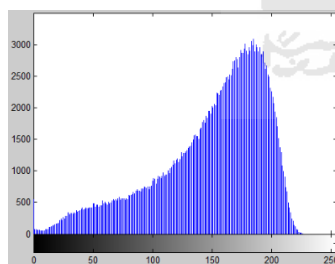


Gambar 3.4 Histogram citra digital

Untuk mengetahui perbedaan dari USC SIPI dapat dilakukan analisis histogram untuk dikelompokkan berdasarkan level histogram citranya.

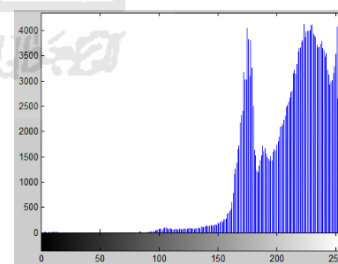
Brodatz-Bar, Brodatz-BW & Nort islan NAS

Brick wall, San Diego(PL), San Diego(SI)



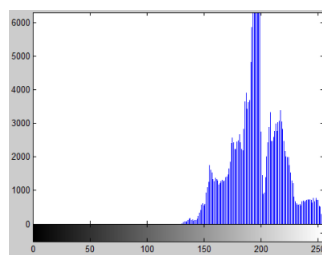
Berkonsentrasi tinggi

Motion(01,02,03), Girl Tiffany

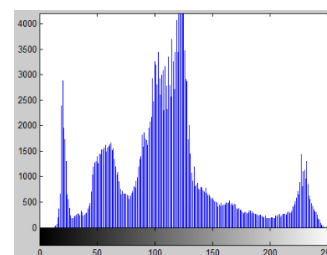


Citra terang

F1 & Splash



Citra terang



Berkonsentrasi tinggi

Gambar 3.5 Histogram citra digital penelitian

Pemetaan hasil analisis *cover image* menggunakan histogram dapat dilihat pada tabel 3.1.

Tabel 3.1 Analisis cover image berdasarkan level histogram

No	Kelompok Cover Image	Level Histogram
1	Brick wall, San Diego(PL), San Diego(SI), Motion(01,02,03) dan Girl Tiffany	Citra terang
2	Brodatz-Bar, Brodatz-BW, Nort islan NAS, F1 & Splash	Citra berkonsentrasi tinggi

Penelitian ini menggunakan citra digital dengan format .jpg dengan ukuran 512 x 512 piksel baik sebagai *file host* maupun sebagai *watermark*. Sebenarnya untuk implementasi *Watermarking* bisa dengan menggunakan gambar dengan type dan ukuran yang lain, tetapi berdasarkan rekomendasi dari peneliti-penelitian sebelumnya yang banyak menggunakan gambar dengan type .jpg yang mendukung kompresi baik secara *lossy* maupun *lossles* dan ukuran 512 x 512 piksel yang lazim digunakan oleh peneliti-peneliti dalam bidang *Watermarking* untuk proteksi pada citra digital (Ansari, 2015).

3.3.2 Attacked Image

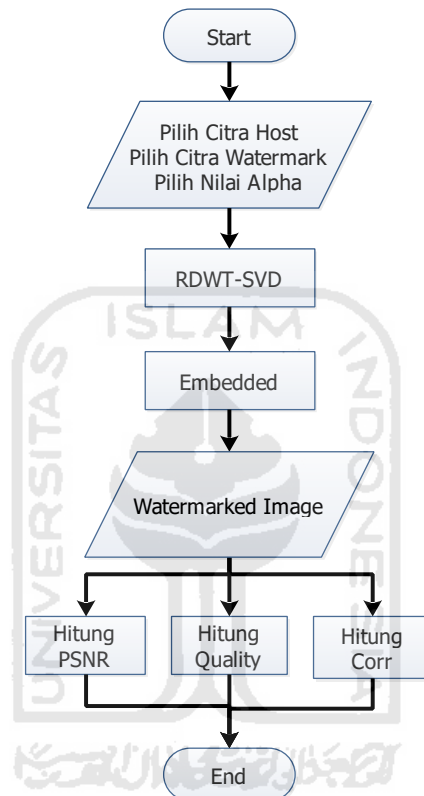
Pada penelitian ini *attack* akan dilakukan menggunakan program *freeware* yaitu *stirmark*. Penggunaan aplikasi *stirmark* karena aplikasi ini merupakan aplikasi *freeware* dan kemampuan menghasilkan *attack* secara otomatis baik *attack* yang berupa *geometris* maupun *non-geometris* dan sudah support untuk *image color* maupun *Grayscale*. Aplikasi *stirmark* ini bisa menghasilkan berbagai jenis *attack* seperti *remodulation, filtering, map, jpeg, ml, resample, color reduce, nois, rotation* dan *ration scala*.

3.4 Pengujian

Pengujian pada penelitian ini terdiri dari 3 (tiga) tahap, yaitu pengujian pengaruh *alpha*, pengujian tingkat *imperceptibility* dan pengujian tingkat *robustness*.

3.4.1 Pengujian Pengaruh Alpha

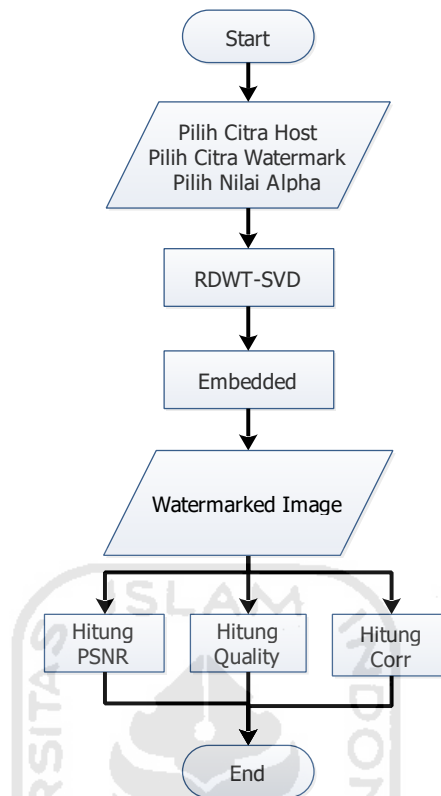
Pengujian ini dilakukan untuk mengetahui sejauh mana pengaruh *factor embedded* atau *alpha* terhadap kualitas citra hasil proses *Hybrid Image Watermarking*. Dengan menggunakan nilai *alpha* yang berbeda. Skema pengujian pengaruh *alpha* terhadap tingkat *imperceptibility* *Hybrid Image Watermarking* dapat dilihat pada gambar 3.6.



Gambar 3.6 Pengujian pengaruh *alpha* terhadap tingkat *imperceptibility* metode *Hybrid Image Watermarking* RDWT-SVD

3.4.2 Pengujian Tingkat *Imperceptibility*

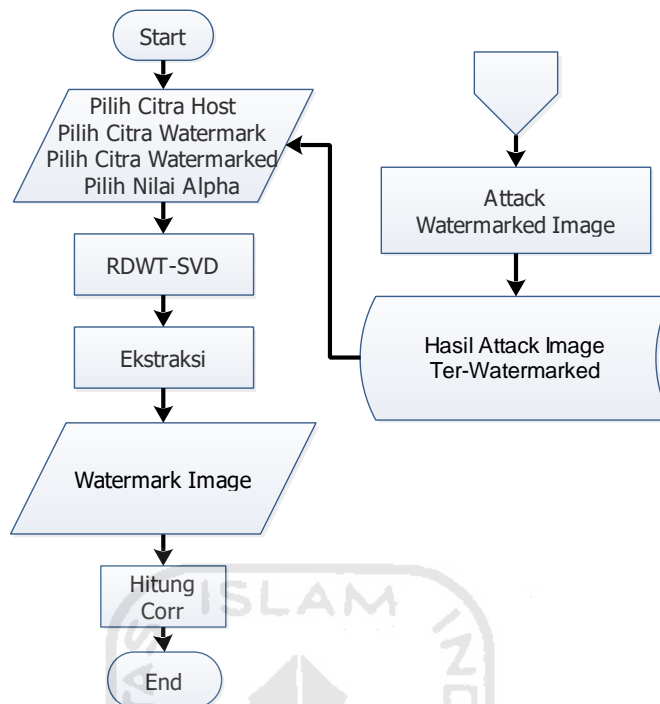
Pengukuran tingkat *imperceptibility* dilakukan dengan menggunakan nilai *alpha* yang sama. Skema pengujian tingkat *imperceptibility* dapat dilihat pada gambar 3.7.



Gambar 3.7 Pengujian tingkat *imperceptibility* metode *Hybrid Image Watermarking* RDWT-SVD

3.4.3 Pengujian Tingkat *Robustness*

Pengukuran tingkat *robustness* citra hasil proses *watermarking* dilakukan secara terpisah dengan proses penyisipan dan ekstraksi *watermarking*. Untuk mengetahui tingkat *robustness* suatu metode atau teknik *watermarking* dilakukan uji ketahanan dengan memberikan sejumlah *attack* terhadap citra hasil proses *Hybrid Image Watermarking*. Untuk menghasilkan *attack* secara otomatis dapat menggunakan aplikasi seperti *StirMark*, *Unzig*, dan *Checkmark*. Penelitian ini menggunakan *StirMark* versi 4.0. Penggunaan *StirMark* pada penelitian ini karena aplikasi ini mampu menghasilkan *attacked image* untuk citra *Grayscale* maupun citra *color* dalam jumlah yang banyak. Skema pengujian tingkat *robust* dapat dilihat pada gambar 3.8.



Gambar 3.8 Pengujian tingkat *robustness* metode *Hybrid Image Watermarking* RDWT-SVD

3.5 Parameter Penelitian

Penelitian komparasi metode *Hybrid Image Watermarking* RDWT-SVD untuk proteksi pada citra digital, pengambilan data penelitian dilakukan untuk mengetahui tingkat *imperceptibility* dan *robustness* citra hasil proses *Watermarking*.

3.5.1 Tingkat *Imperceptibility*

Pengukuran tingkat *imperceptibility* dilakukan dengan mengukur parameter:

1) PSNR (*Peak Signal to Noise Ratio*)

Deteksi *noise* dengan PSNR dapat dihitung dengan menggunakan persamaan Rumus 2.13.

2) *Perceptual Quality Metric (Q)*

Untuk mendapatkan nilai Q dapat dihitung dengan menggunakan persamaan atau Rumus 2.14.

3) *Correlation (Corr)*

Untuk mendapatkan nilai Q dapat dihitung dengan menggunakan persamaan atau Rumus 2.15.

3.5.2 Tingkat *Robustness*

Pengukuran tingkat *robustness* dilakukan pengujian terpisah dengan menggunakan program *stirmark* sebagai penghasil *attack*. Parameter yang diukur adalah parameter Corr dengan menggunakan Rumus 2.15, yang membedakan pengukuran Corr pada tingkat *imperceptibility* yaitu pada pengukuran tingkat *imperceptibility* nilai yang dikorelasi adalah *watermarked image* dengan *file host* sedangkan pada tingkat *robustness* yang dikorelasi adalah hasil ekstraksi dari *attacked image* dengan *file host*.

3.6 Sampel Pengambilan Data Penelitian

Pengukuran parameter-parameter tersebut dimuat dalam sampel cara pengambilan data dibawah ini.

3.6.1 Pengujian Pengaruh Nilai *Alpha* Terhadap Tingkat *Imperceptibility* pada Algoritma RDWT-SVD

Alpha menunjukkan *factor embedding* dengan skala 0 sampai dengan 1, nilai menunjukkan tingkat derajat atau tingkat ketampakan (*visible*) *watermark* yang disisipkan pada *file host*. Semakin besar nilai *alpha* menandakan semakin tampak *watermark* pada *file host*. Pengukuran pengaruh *alpha* dilakukan untuk mengetahui seberapa besar pengaruh *visible watermark* terhadap kualitas citra atau tingkat *imperceptibility* hasil proses *Hybrid Image Watermarking* dengan standar minimal PSNR 35db baru bisa dikatakan *imperceptibility*. Sampel tabel penelitian pengambilan data untuk pengaruh *alpha* terhadap kualitas citra hasil proses metode *Hybrid Image Watermarking* RDWT-SVD sesuai tabel 3.2.

Tabel 3.2 Nilai PSNR, Q dan *Corr* untuk *Watermarking* RDWT-SVD

Nilai Alpha	Brick wall			Splash		
	PSNR	Q	Corr	PSNR	Q	Corr
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

3.6.2 Pengujian tingkat *imperceptibility* algoritma RDWT-SVD

Pada pengujian tingkat *imperceptibility* peneliti menggunakan nilai *alpha* 0.1 untuk semua *image*. *File host* atau *cover image* pada pengujian ini menggunakan *image* setiap karakter, diantaranya *Brick wall*, *Brodatz – Brick wall*, *Brodatz – Bark*, *San diego (Point Loma)*, *North Island NAS*, *San diego (Shelter Island)*, *F1*, *Girl (Tiffany)*, *Splash*, *Motion01*, *Motion02*, *Motion03* dan *image watermark* menggunakan *Earth from space*. Parameter yang diukur adalah PSNR, Q dan *Corr* dan sampel pengambilan data penelitian seperti pada tabel 3.3.

Tabel 3.3 Nilai PSNR, Q dan *Corr* algoritma RDWT-SVD

Karakteristik	Image Grayscale	RDWT-SVD		
		PSNR	Q	Corr
Textures	Brick wall			
	Brodats – Brick wall			
	Brodats – Bark			
Aerials	San diego (Point Loma)			
	North Island NAS			
	San diego (Shelter Island)			
Miscellaneous	F1			
	Girl (Tiffany)			
	Splash			
Sequences	Motion01			
	Motion02			
	Motion03			

3.6.3 Pengujian Tingkat *Robustness* algoritma RDWT-SVD

Pengujian tingkat *robustness* dilakukan dengan 2 (dua) tahap yaitu pengujian tingkat ekstraksi dan deteksi *watermark* dari *attacked image*.

1) Tingkat Ekstraksi *Watermark*

Ekstraksi *watermark* dilakukan untuk mengukur tingkat ekstraksi *watermark* dari *file* yang ditumpangi setelah diberikan serangan. Pengukuran dilakukan dengan menghitung nilai korelasi citra hasil ekstraksi dari *attacked image* dengan citra asli (*file host*). Parameter yang diukur pada *test image* terhadap *attack* adalah nilai *Corr* minimal 0,8 dengan menggunakan rumus 2.15. Rancangan tabel pengambilan sampel dapat dilihat pada tabel 3.4.

Tabel 3.4 Nilai *Corr* algoritma RDWT-SVD

Attack	Group Attack	Corr Algoritma RDWT-SVD		
		Image 1	Image 2	Image 3
Filtering	ConvFilter_1
	ConvFilter_2
	SS_1
JPEG	Jpeg_15
	Jpeg_25
	Jpeg_50
Remodulation	LasterNddist_0.95
	LasterNddist_1
	LasterNddist_1.05
ML	MedianCut_3
	MedianCut_5
	MedianCut_7
PSNR	PSNR_10
	PSNR_50
	PSNR_100
Rotation	Rot_90
Nois	Nois_20
Rotation Scale	RotScale_-0.5
	RotScale_-0.25
	RotScale_0.25

Untuk memudahkan dalam menganalisis hasil ekstraksi *watermark* setelah diberi *attack*, maka nilai *Corr* perlu di *resume* berdasarkan kelompok *attack*.

Tabel 3.5 *Resume* Corr kelompok Attack Algoritma RDWT-SVD

No	Kelompok Attack	Image 1	Image 2	Image 3
1	Filtering (3)
2	JPEG (3)
3	Remodulation (3)
4	ML (3)
5	PSNR (3)
6	Rotation (1)
7	Nois (1)
8	Ratation Scala (3)
Rata-rata	

2) Tingkat Deteksi Watermark

Deteksi *watermark* dilakukan untuk mengetahui tingkat deteksi *watermark* pada citra hasil proses *Watermarking* setelah diberi *attack* dengan batas nilai ekstraksi tertentu yaitu nilai *Corr* diatas sama dengan 0.8 dianggap bisa dideteksi. Pada proses deteksi menggunakan nilai *Corr* hasil pengukuran *ekstraksi watermark* terhadap *attack* yang dilakukan sebelumnya. Dari nilai *Corr* pada Tabel 3.4, dengan asumsi nilai *Corr* diatas sama dengan 0.8 dianggap 1 (terdeteksi) dan nilai 0.8 dianggap 0 (tidak terdeteksi). Menurut (Ansari dan Prayudi 2015) dalam melakukan *resume* tingkat deteksi *watermark* nilai 1 (terdeteksi) diberikan nilai 100 dan 0 (tidak terdeteksi) diberi nilai 0. Sampel data untuk tingkat deteksi *watermark* pada algoritma RDWT-SVD dapat dilihat pada tabel 3.6.

Tabel 3.6 Deteksi dari *attacked image* algoritma RDWT-SVD

Attack	Group Attack	Corr Algoritma RDWT-SVD		
		Image 1	Image 2	Image 3
Filtering	ConvFilter_1
	ConvFilter_2
	SS_1
JPEG	Jpeg_15
	Jpeg_25
	Jpeg_50
Remodulation	LasterNddist_0.95
	LasterNddist_1
	LasterNddist_1.05
ML	MedianCut_3
	MedianCut_5
	MedianCut_7
PSNR	PSNR_10
	PSNR_50
	PSNR_100
Rotation	Rot_90
Nois	Nois_20
Rotation Scale	RotScale_-0.5
	RotScale_-0.25
	RotScale_0.25

Untuk mengetahui tingkat deteksi *watermark* secara keseluruhan dapat dilihat pada Tabel 3.6 diresume berdasarkan kelompok *attack* menjadi Tabel 3.7.

Tabel 3.7 Resume deteksi *Corr* berdasarkan kelompok *attack*

No	Kelompok Attack	Image 1	Image 2	Image 3
1	Filtering (3)
2	JPEG (3)
3	Remodulation (3)
4	ML (3)
5	PSNR (3)
6	Rotation (1)
7	Nois (1)
8	Ratation Scala (3)
Rata-rata	

3.7 Perbandingan dengan penelitian sebelumnya

Pada bagian ini peneliti melakukan perbandingan dengan penelitian sebelumnya yang menerapkan *hybrid* algoritma RDWT-SVD *Two-Scale*. Parameter yang dibandingkan yaitu tingkat *imperceptibility* dan *robustness* dengan menggunakan *cover image* dan *watermark* yang sama. *Cover image* (motion01, motion02 dan motion03), *watermark image* (Earth from space) dan nilai *alpha* 0.1. Untuk sampel *robustness* mengambil data *attack* yang sama, yaitu JPEG_50. Perbandingan hasil dapat dilihat pada tabel 3.8.

Tabel 3.8 Perbandingan tingkat *imperceptibility* (dB)

Nama gambar	Skema penelitian	Ansari dan Prayudi
motion01		
motion02		
motion03		
Nilai rata-rata		

Tabel 3.9 Perbandingan tingkat *robustness* (corr)

Nama gambar	Skema penelitian	Ansari dan Prayudi
motion01		
Brodats		

3.8 Pengaruh *Hybrid* metode *Watermarking* terhadap HAKI

Watermarking atau juga disebut tanda air merupakan teknik penyisipan pesan rahasia ke dalam sebuah pesan lainnya yang tetap terlihat jelas dan dapat menyamarkan pesan tersebut dan keberadaan *watermark* bisa dibuktikan dengan proses ekstraksi. *Watermarking* adalah teknik yang sering digunakan dalam hal perlindungan hak cipta karena keunggulannya dapat menyembunyikan pesan kedalam pesan lain tanpa merubah file yang di tumpanginya. *Watermarking* akan dikatakan mampu melindungi hak cipta pada citra digital apabila memenuhi tingkat *imperceptibility* atau kemiripan antara citra *watermarked* dengan *cover image* dengan parameter nilai PSNR diatas atau sama dengan 35dB dan nilai kualitas atau Q minimal 4 (good) dan tingkat *robustness* atau ketahanan dengan parameter Corr diatas atau sama dengan 0,8 dalam mendeteksi *watermark* setelah diberikan *attack*.



BAB IV

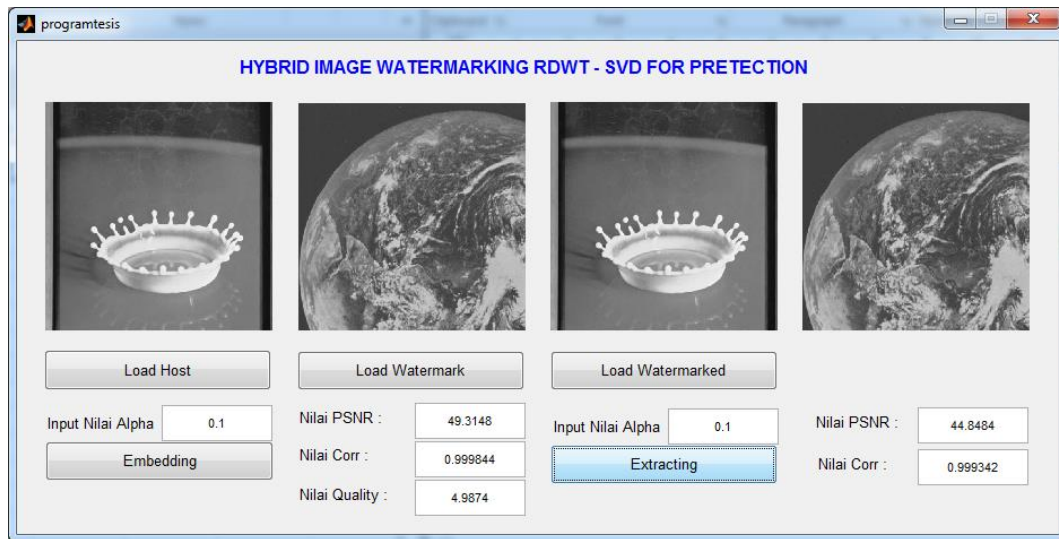
IMPLEMENTASI DAN ANALISIS

Pada penelitian tentang penggabungan algoritma *Watermarking* yaitu RDWT dengan SVD untuk proteksi terhadap citra digital dilakukan beberapa tahapan untuk meneliti tingkat keberhasilan dan keamanan dari algoritma yang diterapkan khusus pada citra digital *Grayscale*. Tahapan penelitian dimulai dari pembuatan aplikasi *Watermarking* kemudian melakukan percobaan *embedded* gambar dan pemberian serangan menggunakan aplikasi *stirmark* untuk mengetahui tingkat ketahanan algoritma, selanjutnya melakukan analisa terhadap hasil *embedded* dan ekstraksi *watermark* setelah diberikan serangan. Implementasi dan analisis pada penelitian ini menggunakan komputer dengan spesifikasi *hardware* dan *software* sebagai berikut.

- a. Processor Intel(R) Core(TM) i3-3227U CPU @1.90GHz 1.90 GHz;
- b. RAM 6,00 GB;
- c. System Type 64-bit Windows 7 Ultimate
- d. Harddisk 500 GB
- e. Matlab R2013B
- f. Stirmark 4.0

4.1 Tampilan GUI Aplikasi *Hybrid Image Watermarking*

Graphical User Interface (GUI) merupakan tampilan yang berfungsi sebagai *window* atau jendela yang menghubungkan pengguna dengan aplikasi *watermarking* untuk perlindungan hak cipta pada gambar. Pada penerapan penggabungan algoritma *watermarking* dan parameternya menggunakan pemrograman matlab R2013b. Tampilan GUI dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan GUI *Hybrid Image Watermarking* algoritma RDWT-SVD

Tombol *load host* digunakan untuk memilih gambar yang dijadikan *cover image*, tombol *load watermark* digunakan untuk memilih gambar *watermark*, tombol *embedding* digunakan untuk proses penyisipan *watermark* ke dalam *cover image*, tombol *load watermarked* digunakan untuk memilih gambar yang sudah diberi *watermark* atau gambar yang sudah di *watermarked* dan tombol *extracting* digunakan untuk proses ekstraksi *watermark* dari *cover image*.

4.2 Parameter *Hybrid* Algoritma RDWT-SVD

4.2.1 Peak Signal to Noise Ratio (PSNR)

PSNR merupakan parameter untuk mengukur kualitas atau tingkat *imperceptibility* citra digital proses *watermarking*, penyisipan *watermark* pada *cover image* dianggap sebagai *noise* dapat menurunkan kualitas citra digital *watermarked*. Nilai PSNR didapat dengan membandingkan kualitas citra asli atau *cover image* dengan citra *watermarked*, semakin tinggi nilai yang didapat menandakan deteksi *noise* semakin kecil dan kualitas *watermarked image* dianggap baik.

4.2.2 *Perceptual Quality Matric (Q)*

Quality menunjukkan kualitas citra digital *watermarked* dengan skala 1 sampai 5, nilai 5 menunjukkan kualitas citra *watermarked* yang sempurna (*excellend*), 4 (*good*), 3 (*fair*), 2 (*poor*) dan 1 (*bad*).

4.2.3 *Corr*

Corr merupakan nilai korelasi citra digital *watermarked* hasil ekstraksi dengan citra asli atau *cover image*. Nilai korelasi yang sempurna adalah 1 (satu), semaki mendekati angka 1 menandakan nilai korelasi dianggap sempurna atau mendekati citra aslinya. Nilai korelasi dapat diterima minimal 0.8.

4.3 Data Penelitian dan Pembahasan

Pada penelitian penggabungan algoritma *watermarking* RDWT dengan SVD meliputi beberapa tahap pengujian yaitu pengujian pengaruh *alpha*, pengujian tingkat *imperceptibility* dan pengujian tingkat *robustness*.

4.3.1 Pengujian pengaruh *alpha* terhadap tingkat *imperceptibility*

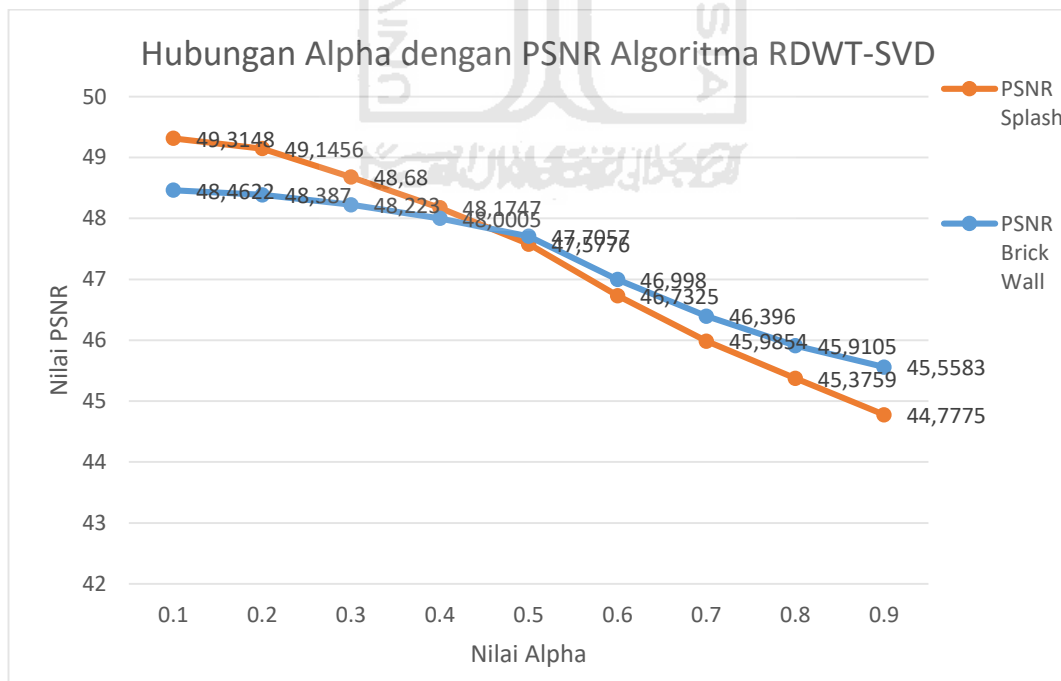
Parameter pengaruh *factor embedded (alpha)* terhadap tingkat *imperceptibility* dapat dilihat dari perubahan nilai PSNR, Q dan *Corr* dari citra hasil *embedded* atau citra *watermarked* menggunakan algoritma RDWT dengan SVD. Pengujian pengaru *alpha* dilakukan, karena *alpha* merupakan level atau derajat ketampakan (*visible watermark*) terhadap tingkat *imperceptibility* citra *watermarked*.

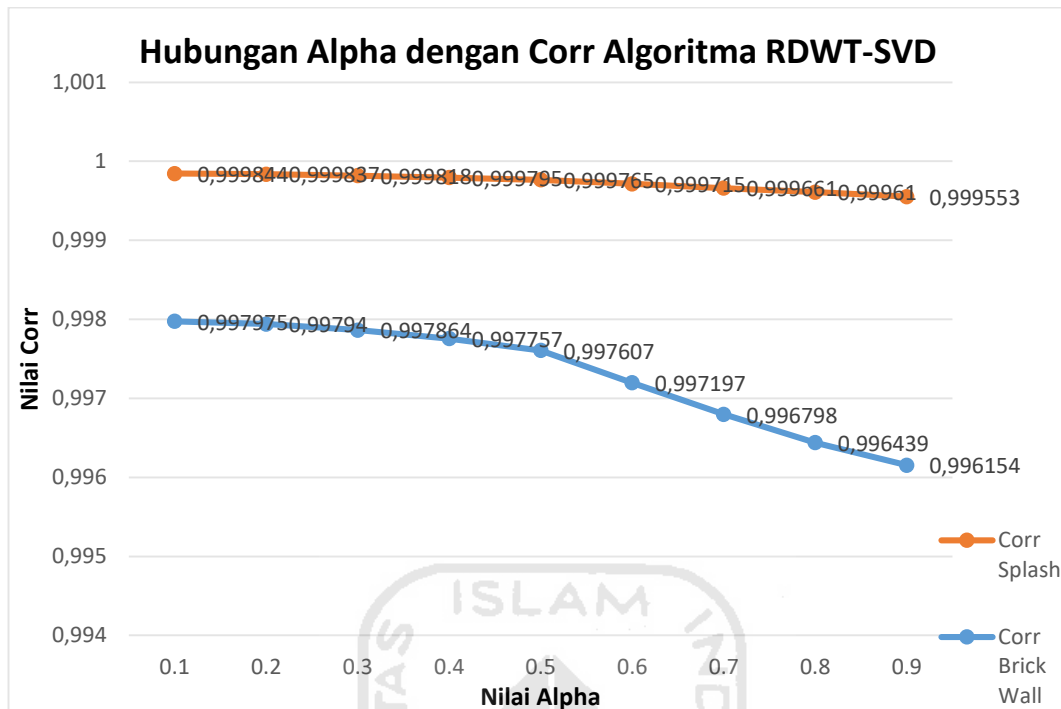
Pada pengujian ini menggunakan nilai *alpha* yang bervariasi (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9) dengan sampel citra *Grayscale (Brick Wall dan Splash)* skema pengujian sesuai dengan gambar 3.6 di dapatkan data nilai PSNR, *Corr* dan *Quality (Q)*.

Tabel 4.1 Nilai PSNR, *Quality* dan *Corr* algoritma RDWT-SVD

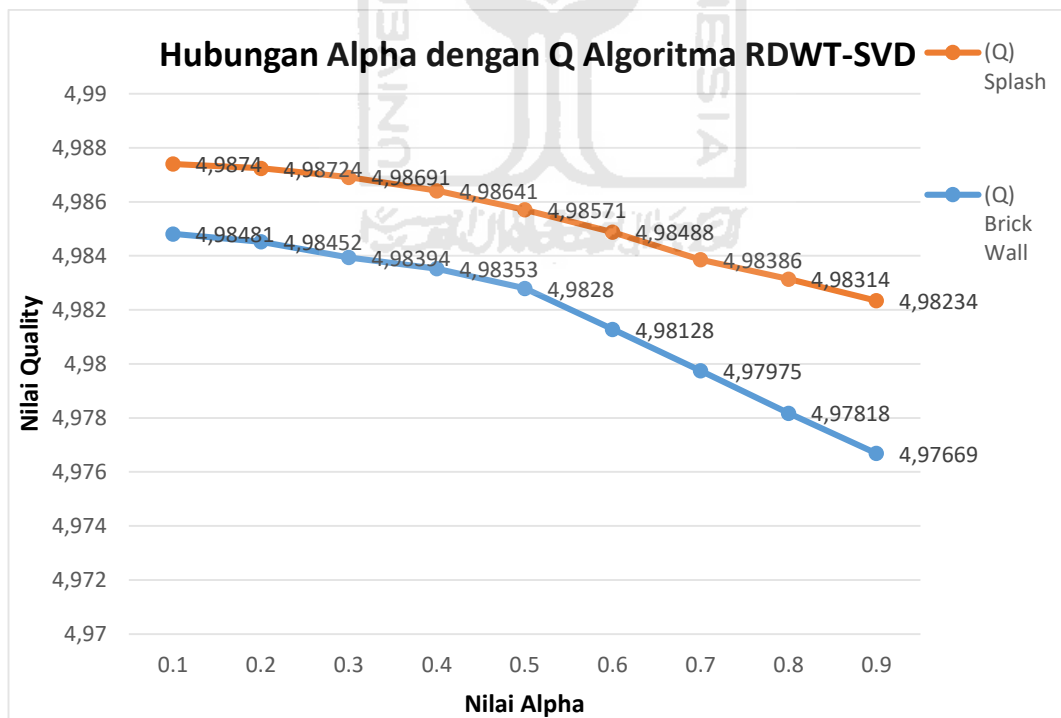
Nilai Alpha	Brick wall			Splash		
	PSNR	Q	Corr	PSNR	Q	Corr
0,1	48,4622	4,98481	0,997975	49,3148	4,9874	0,999844
0,2	48,387	4,98452	0,99794	49,1456	4,98724	0,999837
0,3	48,223	4,98394	0,997864	48,68	4,98691	0,999818
0,4	48,0005	4,98353	0,997757	48,1747	4,98641	0,999795
0,5	47,7057	4,9828	0,997607	47,5776	4,98571	0,999765
0,6	46,998	4,98128	0,997197	46,7325	4,98488	0,999715
0,7	46,396	4,97975	0,996798	45,9854	4,98386	0,999661
0,8	45,9105	4,97818	0,996439	45,3759	4,98314	0,99961
0,9	45,5583	4,97669	0,996154	44,7775	4,98234	0,999553

Dari data diatas dapat disusun sebuah grafik yang menunjukkan pengaruh α terhadap tingkat *imperceptibility watermarked* algoritma RDWT dengan SVD.

Gambar 4.2 Hubungan α dengan PSNR algoritma RDWT-SVD



Gambar 4.3 Hubungan α dengan $Corr$ algoritma RDWT-SVD



Gambar 4.4 Hubungan α dengan Q algoritma RDWT-SVD

Berdasarkan data diatas pada gambar 4.2 hubungan *alpha* dengan tingkat *imperceptibility* berdasarkan nilai PSNR dapat diketahui bahwa semakin tinggi nilai *alpha* yang diberikan dapat mempengaruhi nilai PSNR yang mengakibatkan penurunan nilai dan sebaliknya semakin rendah nilai *alpha* dapat meningkatkan nilai PSNR, sehingga dapat disimpulkan bahwa *factor alpha* mempengaruhi tingkat *imperceptibility*. Sementara pada nilai *alpha* 0.5 terjadi persilangan kurva dimana *image* Brick wall mendapat nilai lebih tinggi dibanding *image* Splash karena *image* Splash termasuk *image* yang berdiagram berkonsentrasi tinggi dimana *image* yang berkonsentrasi tinggi tidak stabil dalam menjaga tingkat *imperceptibility* nya dibandingkan *image* berhistogram citra terang yang cenderung lebih stabil.

Pada gambar 4.3 nilai *alpha* yang bervariasi memberikan nilai korelasi yang berbeda semakin tinggi nilai *alpha* maka tingkat korelasi akan semakin tinggi dan sebaliknya semakin rendah nilai *alpha* maka tingkat korelasi mengalami penurunan, tapi masih diatas 0.9 hal ini menandakan bahwa korelasi antara citra *watermarked* dengan citra asli atau *cover image* tetap tinggi dengan nilai *alpha* yang bervariasi. Pada gambar 4.3 tidak terjadi persilangan antara *image* Brick wall dengan Splash seperti gambar 4.2, karena perhitungan nilai *Corr* berbeda dengan PSNR. Nilai *corr* di dapatkan dari menghitung nilai rata-rata dari hasil penjumlahan nilai matrix kemudian membaginya sementara PSNR adalah perhitungan nilai maksimum dari sinyal yang diukur dengan besarnya *nois* yang mempengaruhi sinyal tersebut.

Sedangkan nilai *Perceptual Quality Metric* (Q) pada gambar 4.4 juga mengalami perubahan akibat perubahan *alpha*, peningkatan nilai *alpha* mengakibatkan penurunan nilai Q, namun penurunan tersebut tidak signifikan dengan nilai Q tetap diatas 4,9 yang menandakan Q ditingkat 4 (*good*) dan hampir mendekati nilai 5 (*excellent*). Perubahan nilai tersebut secara *perceptual* masih dapat diterima walaupun mengalami penurunan. Pada gambar 4.4 juga tidak terjadi persilangan seperti 4.2 karena proses perhitungan untuk mendapatkan nilai Q dengan melakukan pembagian menggunakan ukuran distorsi dan konstanta normalisasi.

4.3.2 Pengujian tingkat *imperceptibility* algoritma RDWT-SVD

Pada pengujian tingkat *imperceptibility* peneliti menggunakan nilai α 0.1 untuk semua *image*. Dari data pengujian pengaruh α terhadap tingkat *imperceptibility* nilai 0.1 memiliki nilai PSNR dan Q yang tinggi dengan menggunakan *wavelet haar* karena *wavelet haar* merupakan induk dari *wavelet* (*mother of wavelet*) dan memiliki bentuk yang sederhana sehingga mudah dalam implementasi metode *Watermarking*.

Untuk mengetahui tingkat *imperceptibility hybrid* algoritma RDWT-SVD *Watermarking* dapat dilakukan dengan mengukur beberapa parameter seperti PSNR, Q dan Corr pada citra *watermarked*.

Menurut Kutter dan Petitcola (1999), standar untuk tingkat *imperceptibility* teknik *Watermarking* diupayakan dapat menghasilkan nilai PSNR diatas 35dB, rating kualitas citra atau Q minimal 4 (*good*) dan *Corr* minimal atau sama dengan 0.8. Parameter *imperceptibility* dilakukan pada *image watermarked* atau gambar yang sudah disisipkan *watermark*. Skema pengujian sesuai dengan gambar 3.7 dan sampel data dapat dilihat pada tabel 4.2.

Tabel 4.2 Nilai PSNR, Q dan *Corr* algoritma RDWT-SVD

Karakteristik	Image Grayscale	RDWT-SVD		
		PSNR	Q	Corr
Textures	Brick wall	48,4622	4,98481	0,997975
	Brodatz – Brick wall	39,9436	4,97858	0,996817
	Brodatz – Bark	38,6931	4,99366	0,998001
Aerials	San diego (Point Loma)	47,3514	4,98227	0,998395
	North Island NAS	43,1827	4,98722	0,999291
	San diego (Shelter Island)	43,2938	4,98472	0,997607
Miscellaneous	F1	46,3787	4,98868	0,99965
	Girl (Tiffany)	46,3109	4,98559	0,999165
	Splash	49,3148	4,9874	0,999844
Sequences	Motion01	45,8573	4,96926	0,998162
	Motion02	45,6	4,9693	0,998103
	Motion03	45,5027	4,96968	0,998178
Nilai Rata - rata		44,9909	4,98176	0,99843

Berdasarkan data pada tabel 4.2 bahwa terdapat beberapa kelompok *image* berdasarkan karakteristiknya diantaranya *Textures*, *Aerials*, *Miscellaneous* dan *Sequences* merupakan kelompok karakter *image standart* internasional penelitian tentang *image* analisis. Dari setiap karakteristik di ambil tiga *image* sebagai sampel dalam menerapkan algoritma yang diusulkan dari hasil penerapan pada tabel 4.2 menunjukkan tingkat *imperceptibility* berdasarkan nilai PSNR didapatkan nilai yang bervariasi seperti nilai tertinggi sebesar 49,3148 dB untuk citra *Splash* dan nilai PSNR terendah 38,6931 dB pada citra *Brodatz – Bark* dan mendapatkan nilai rata-rata 44,9909 yang merupakan nilai di atas *standart* yaitu 35dB dan dapat dikatakan *imperceptibility* berdasarkan parameter PSNR.

Dari data diatas diketahui bahwa nilai PSNR yang bervariasi dipengaruhi oleh citra yang digunakan sebagai *cover image* atau *file host* maka untuk mengetahuinya perlu dilakukan analisis *image*. Analisis *cover image* dapat dilakukan dengan analisis histogram pada citra. Berdasarkan hasil analisis histogram pada bab 3, distribusi level citra dapat dikelompokkan menjadi dua (2) yaitu citra terang dan citra berkonsentrasi tinggi.

Dari data tabel 4.2 citra *Splash* mendapatkan nilai PSNR tertinggi merupakan kelompok citra *miscellaneous* memiliki level histogram berkonsentrasi tinggi, karena mendekati nilai maksimal yaitu 255 merupakan tingkat fluktuasi yang digunakan untuk menentukan nilai PSNR sesuai dengan persamaan 2.10, sedangkan citra *Brodatz-Bark* mendapat nilai PSNR terendah merupakan kelompok *image Texture* memiliki level histogram berkonsentrasi tinggi karena mendekati nilai minimal yaitu 0. Citra berkonsentrasi tinggi merupakan level histogram yang menyebar atau meluas sehingga *image* yang berada di kelompok tinggi ini cenderung tidak stabil karena mendekati 0 dan 255. Sedangkan citra terang berada di level histogram tengah ke tinggi mendekati nilai maksimal yaitu 255 sehingga nilai PSNR citra terang cenderung lebih stabil dan tinggi. Dari hasil analisis diketahui bahwa perubahan nilai PSNR dipengaruhi oleh level histogram pada *cover image* bukan berdasarkan karakteristiknya.

Untuk tingkat *imperceptibility* berdasarkan nilai Q dan Corr tidak mengalami perubahan yang signifikan seperti PSNR terhadap sampel setiap karakteristik *cover*

image, karena perhitungan pada parameter Q menggunakan ukuran distorsi dan konstanta dan $Corr$ perhitungan dengan menghitung nilai rata-rata dari hasil penjumlahan nilai matrix sementara PSNR adalah perhitungan nilai maksimum dari sinyal yang di ukur dengan besarnya *nois* yang mempengaruhi sinyal tersebut dapat dilihat nilai Q untuk semua karakteristik *image* masih mendapatkan nilai rata-rata 4,98176 mendekati nilai 5 atau sempurna dan $Corr$ atau tingkat korelasi citra *watermarked* dengan *cover image* dengan nilai rata-rata 0,99843 dan mendekati nilai sempurna yaitu nilai 1.

4.3.3 Pengujian tingkat *robustness* algoritma RDWT-SVD

Untuk mengetahui tingkat *robustness* pada algoritma RDWT-SVD dapat dilakukan dengan memberikan sejumlah *attack* terhadap citra ter-*watermarked*. Pengujian tingkat *robust* dengan memberikan *attack* terhadap citra *watermarked* dilakukan terpisah dari implementasi algoritma metode *Watermarking* dengan menggunakan aplikasi *freeware stirmark* sebagai pemberi *attack* otomatis.

Penggunaan *stirmark* karena mampu memberikan berbagai serangan (*attack*) terhadap citra *watermarked* secara otomatis dan sudah menjadi salah satu standarisasi pengujian metode *Watermarking*. Teknik *Watermarking* dikatakan *robust* jika *watermark* dapat bertahan setelah diberikan beberapa serangan yang dapat merusak *watermark* dan dapat diekstrak kembali dengan nilai korelasi minimal atau sama dengan 0.8.

a) Pengujian tingkat ekstraksi *watermark*

Pengujian tingkat ekstraksi dilakukan untuk mengetahui tingkat *robust* dari *Hybrid Image Watermarking* RDWT dengan SVD setelah diberikan *attack* otomatis menggunakan aplikasi *freeware stirmark* versi 4.0. Citra yang digunakan dalam pengujian tingkat *robustness* yaitu *cover image* (*image 1* (Brick wall), *image 2* (San diego (Point Loma)), *image 3* (Splash)) dan citra *watermark* menggunakan Earth from space. Citra *watermarked* diberi nama dengan (emBrick wall_0.1 (Brick wall), emSanPM_0.1 (San Diego (Point Loma)) dan emSplash_0.1 (Splash)).

Selanjutnya diberikan serangan menggunakan *stirmark* dan mendapat citra 115 dengan serangan yang berbeda-beda. Kemudian sebagian dari kelompok *attack*

diambil sebagai sampel untuk di uji tingkat *robust* nya dengan melakukan ekstraksi *watermark* dari citra *watermarked* yang sudah diberi *attack*. Ada sebagian jenis serangan yang tidak dapat ekstraksi karena hasil serangan yang diberikan merubah dimensi citra *watermarked*. Kelompok serangan yang dapat di ekstraksi dapat dilihat pada tabel 4.3.

Tabel 4.3 Ekstraksi *watermark Hybrid Image Watermarking RDWT-SVD*

Attack	Group Attack	Corr Algoritma RDWT-SVD		
		Image 1	Image 2	Image 3
Filtering	ConvFilter_1	0.991861	0.979028	0.995043
	ConvFilter_2	0.993356	0.99109	0.971233
	SS_1	0.967278	0.958508	0.977218
JPEG	Jpeg_15	0.998218	0.99798	0.999118
	Jpeg_25	0.998961	0.999148	0.999126
	Jpeg_50	0.998802	0.998458	0.999202
Remodulation	LasterNddist_0.95	0.980281	0.973678	0.99862
	LasterNddist_1	0.980798	0.969499	0.998667
	LasterNddist_1.05	0.980682	0.96735	0.99864
ML	MedianCut_3	0.996785	0.993773	0.998525
	MedianCut_5	0.995005	0.988239	0.978377
	MedianCut_7	0.993381	0.984785	0.977691
PSNR	PSNR_10	0.99903	0.999165	0.999342
	PSNR_50	0.999032	0.999165	0.999341
	PSNR_100	0.999037	0.999168	0.999332
Rotation	Rot_90	0.998109	0.99908	0.998525
Nois	Nois_20	0.396809	0.384079	0.430238
Rotation Scale	RotScale_-0.5	0.998509	0.998364	0.979443
	RotScale_-0.25	0.998566	0.998229	0.977295
	RotScale_0.25	0.998437	0.998383	0.997173

Dari data tabel 4.3 dapat di *resume* berdasarkan kelompok *attack* nya untuk mempermudah dalam pengambilan keputusan. Data *resume* korelasi ekstraksi *watermark* dapat dilihat pada tabel 4.4.

Tabel 4.4 Resume Corr ekstraksi *hybrid* algoritma RDWT-SVD

No	Kelompok Attack	Image 1	Image 2	Image 3
1	Filtering (3)	0,98417	0,97621	0,98116
2	JPEG (3)	0,99866	0,99853	0,99915
3	Remodulation (3)	0,98059	0,97018	0,99864
4	ML (3)	0,99506	0,98893	0,98486
5	PSNR (3)	0,99903	0,99917	0,99934
6	Rotation (1)	0,99811	0,99908	0,99853
7	Nois (1)	0.39681	0.38408	0.43024
8	Rotation Scala (3)	0,9985	0,99833	0,98464
Rata-rata		0,9189	0,9143	0,9221

Dari data tabel 4.4 nilai korelasi *watermark* rata-rata diatas 0.8 setelah di ekstraksi dari berbagai serangan pada citra *watermarked*, yaitu 0.9189 untuk citra *image 1* (Brick wall), 0.9143 *image 2* (San Diego (Point Loma)) dan *image 3* (Splash) 0.9221. Hal ini menandakan bahwa *Hybrid Image Watermarking* algoritma RDWT dengan SVD pada *subband* LH₂ sangat *robust* terhadap serangan *filtering*, *JPEG*, *Remodulation*, *ML*, *PSNR*, *Rotation* dan *Rotaion Scale*, karena memiliki nilai korelasi diatas 0,9 mendekati nilai 1 atau sempurna. Namun untuk jenis serangan *nois* mendapatkan nilai korelasi dibawah 0.8 menandakan *hybrid* algoritma RDWT-SVD pada *subband* LH₂ tidak tahan terhadap serangan tersebut.

Penerapan *Hybrid Image Watermarking* dalam melakukan proteksi citra digital harus *robust* sehingga *watermark* yang di sisipkan kedalam *cover image* tidak mudah untuk dimodifikasi atau dihilangkan, karena penyisipan *watermark* merupakan tanda identitas kepemilikan sah terhadap citra digital dan menjadi bukti untuk menjerat para pengguna citra digital tersebut apabila terjadi penyalahgunaan dengan melakukan ekstraksi.

Penerapan algoritma RDWT-SVD *subband* LH₂ masih bisa dikatakan efektif untuk proteksi citra digital selama citra *watermarked* tidak direkayasa dengan serangan *nois*, dan algoritma ini cenderung lebih *robust* dari upaya menghilangkan *watermark* dibandingkan dengan *hybrid* algoritma lainnya.

b) Pengujian tingkat deteksi watermark

Pengujian ini dilakukan untuk mengetahui tingkat deteksi *watermark* dengan standar tertentu. Dengan asumsi nilai Corr diatas 0.8 sama dengan 1 (terdeteksi) dan nilai Corr dibawah 0.8 sama dengan 0 (tidak terdeteksi). Data sampel diambil dari data tabel 4.4 kemudian di sederhanakan menjadi 1 dan 0 untuk tingkat deteksi *watermark* sebagaimana pada tabel 4.5.

Tabel 4.5 Deteksi *watermark hybrid image watermarking RDWT-SVD*

Attack	Group Attack	Corr Algoritma RDWT-SVD		
		Image 1	Image 2	Image 3
Filtering	ConvFilter_1	1	1	1
	ConvFilter_2	1	1	1
	SS_1	1	1	1
JPEG	Jpeg_15	1	1	1
	Jpeg_25	1	1	1
	Jpeg_50	1	1	1
Remodulation	LastesrNddist_0.95	1	1	1
	LastesrNddist_1	1	1	1
	LastesrNddist_1.05	1	1	1
ML	MedianCut_3	1	1	1
	MedianCut_5	1	1	1
	MedianCut_7	1	1	1
PSNR	PSNR_10	1	1	1
	PSNR_50	1	1	1
	PSNR_100	1	1	1
Rotation	Rot_90	1	1	1
Nois	Nois_20	0	0	0
Rotation Scale	RotScale_-0.5	1	1	1
	RotScale_-0.25	1	1	1
	RotScale_0.25	1	1	1

Data dari tabel 4.5 di-*resume* kembali sesuai kelompok *attack*-nya, sehingga terlihat seperti tabel 4.6.

Tabel 4.6 *Resume* tingkat deteksi *watermark* algoritma RDWT-SVD

No	Kelompok Attack	Image 1	Image 2	Image 3
1	Filtering (3)	3	3	3
2	JPEG (3)	3	3	3
3	Remodulation (3)	3	3	3
4	ML (3)	3	3	3
5	PSNR (3)	3	3	3
6	Rotation (1)	1	1	1
7	Nois (1)	0	0	0
8	Rotation Scala (3)	3	3	3
Total Image Terdeteksi		19	19	19
Presentase (%)		95	95	95

Jumlah total image = 60

nilai rata-rata deteksi = 95%

Ansari dan Prayudi (2015) mengasumsikan bahwa nilai corr diatas 0,8 dianggap 1 (terdeteksi) dan diberi nilai 100% menunjukkan tingkat deteksi sempurna, karena dapat mendeteksi *watermark* dengan nilai korelasi diatas 0.8 dari *attacked image* dan yang tidak terdeteksi diberi nilai 0. Dari data tabel 4.6 diketahui bahwa mekanisme *Hybrid Image Watermarking* RDWT dengan SVD menghasilkan nilai rata-rata tingkat deteksi sangat memuaskan terhadap *attacked image* yaitu sebesar **95%** dan tingkat rata-rata ekstraksi berdasarkan data pada tabel 4.4 sebesar **0,9189 image1**, **0,9143 image2** dan **0,9221 image3** dengan total citra digital 60 sebagai sampel yang mewakili semua jenis serangan yang dihasilkan *stirmark*.

4.4 Perbandingan

Pada bagian ini peneliti melakukan perbandingan dengan penelitian sebelumnya yaitu (Ansari dan Prayudi, 2015) dalam penelitiannya *Hybrid Image Watermarking Two-scale* RDWT-SVD dengan penyisipan pada *subband LL₂* merupakan bagian konsentrasi citra yang menyerupai citra aslinya, karena mirip dengan citra aslinya maka informasi yang dimilikipun banyak, sehingga ketika

penyisipan dilakukan pada *subband* ini dapat mempengaruhi tingkat *imperceptibility* dari hasil *watermarking*.

Sementara penelitian yang diusulkan penyisipan dilakukan pada *subband* LH₂ dimana *subband* ini merupakan *subband* tengah yang termasuk frekuensi tinggi berbentuk karikatur dan sedikit mengandung informasi. Menurut (Hien, Nakao, & Chen, 2005) penyisipan pada *subband* LH memiliki tingkat *imperceptibility* yang baik karena penyisipan dilakukan pada bagian karikatur *image* yang sedikit mengandung informasi sehingga ketika dilakukan penyisipan tidak akan mempengaruhi *image* penampungnya didukung oleh penelitiannya (Sharma dan Seema, 2012) dalam penelitiannya menyatakan bahwa penyisipan pada level dekomposisi yang tinggi dapat menghasilkan tingkat *imperceptibility* dan *robustness* yang tinggi.

Pada perbandingan ini parameter yang dibandingkan yaitu tingkat *imperceptibility* dan *robustness* dengan menggunakan *cover image* dan *watermark* yang sama, karena fokus penelitian ini pada *object Grayscale* maka *image* yang dibandingkan adalah *image Grayscale* yang sama yaitu *cover image* (motion01, motion02 dan motion03), *watermark image* (Earth from space) dan nilai *alpha* 0.1. Untuk sampel *robustness* mengambil data *attack* yang sama, yaitu serangan atau *attack* JPEG_50.

1. Perbandingan Tingkat Imperceptibility

Pada perbandingan tingkat *imperceptibility* ini parameter yang digunakan berdasarkan nilai PSNR (dB) merupakan standart perbandingan dalam implementasi metode *watermarking*, karena sering digunakan oleh para peneliti sebelumnya ketika melakukan perbandingan hasil uji metode *watermarking* yang di usulkan.

Untuk lebih jelasnya dapat dilihat pada tabel 4.7 yaitu tabel perbandingan tingkat *imperceptibility* berdasarkan nilai PSNR (dB) dari *Hybrid Image Watermarking Two-scale RDWT-SVD* penyisipan pada *subband* LL₂ dengan skema yang di usulkan.

Tabel 4.7 Perbandingan Tingkat *Imperceptibility* (dB)

Nama gambar	Skema penelitian	Ansari dan Prayudi
motion01	45,8573	44,5237
motion02	45,6	44,4264
motion03	45,5027	44,3784

Berdasarkan data pada tabel 4.7 skema penelitian yang di usulkan mendapatkan nilai rata-rata PSNR diatas 45 dB dan (Ansari dan Prayudi, 2015) mendapatkan nilai rata-rata PSNR diatas 44 dB, membuktikan bahwa penyisipan pada *subband* LH₂ lebih *imperceptibility* dibandingkan dengan penyisipan di *subband* LL₂ berdasarkan penelitian sebelumnya (Hien, Nakao, & Chen, 2005) mengayatakan bahwa penyisipan pada *subband* LH memiliki tingkat *imperceptibility* yang baik karena penyisipan dilakukan pada bagian karikatur *image* yang sedikit mengandung informasi sehingga ketika dilakukan penyisipan tidak akan mempengaruhi *image* penampungnya didukung oleh penelitiannya (Sharma dan Seema, 2012) dalam penelitiannya menyatakan bahwa penyisipan pada level dekomposisi yang tinggi dapat menghasilkan tingkat *imperceptibility* dan *robustness* yang tinggi..

2. Perbandingan Tingkat *Robustness*

Untuk dapat mengetahui tingkat ketahanan atau *robust* pada *Hybrid Image Watermarking* RDWT-SVD dapat di uji dengan memberikan berbagai macam serangan terhadap citra hasil *watermarking* atau citra *watermarked* kemudian melakukan ekstraksi *watermark* dari citra *watermarked* yang sudah diberikan serangan dan hasil ekstraksi *watermark* akan di ukur menggunakan parameter *corr* untuk mengetahui tingkat korelasi citra *watermark* hasil ekstraksi dengan citra *watermark* asli. Menurut (Kutter dan Petitcola, 1999) metode *Watermarking* akan dikatakan *robust* apabila mampu menghasilkan nilai *corr* diatas 0,8.

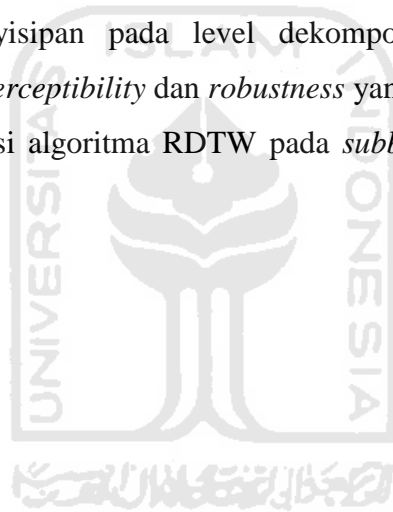
Berdasarkan penelitian sebelumnya dapat dilakukan perbandingan untuk mengetahui tingkat *robust* nya. Pada perbandingan *robust* ini peneliti mengambil sampel serangan *jpeg_50*. Perbandingan tingkat *robust* dapat dilihat pada tabel 4.8.

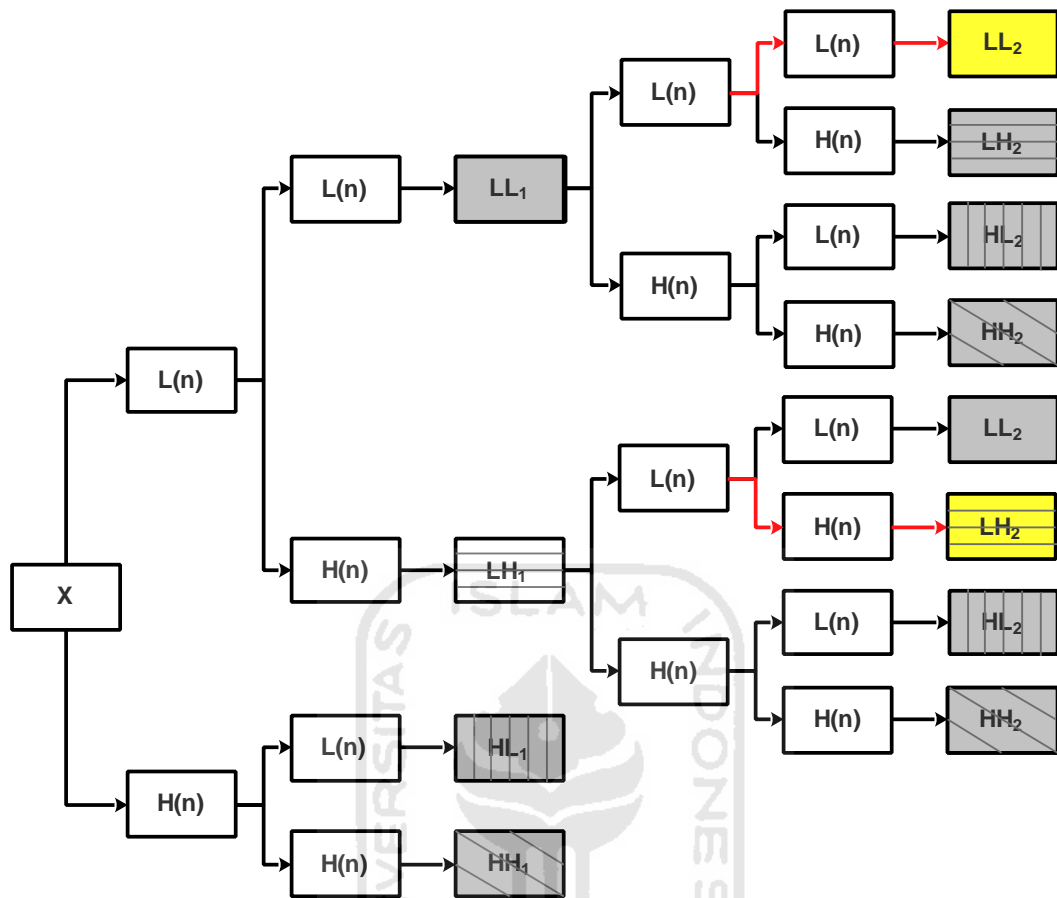
Tabel 4.8 Perbandingan tingkat *robustness* (corr)

Nama gambar	Skema penelitian	Ansari dan Prayudi
motion01	0,99899	0,98468
Brodats	0,99880	0,99246

Berdasarkan data perbandingan pada tabel 4.8 membuktikan bahwa penyisipan pada *subband* LH₂ juga tahan atau *robust* terhadap serangan yang diberikan selama tidak di rekayasa menggunakan *nois*. Hal ini dikarenakan *subband* LH₂ adalah bagian karikatur dengan tingkat dekomposisi yang tinggi berdasarkan penelitian yang dilakukan oleh (Sharma dan Seema, 2012) dalam penelitiannya menyatakan bahwa penyisipan pada level dekomposisi yang tinggi dapat menghasilkan tingkat *imperceptibility* dan *robustness* yang tinggi.

Struktur dekomposisi algoritma RDTW pada *subband* LL₂ dan LH₂ dapat dilihat pada gambar 4.5.



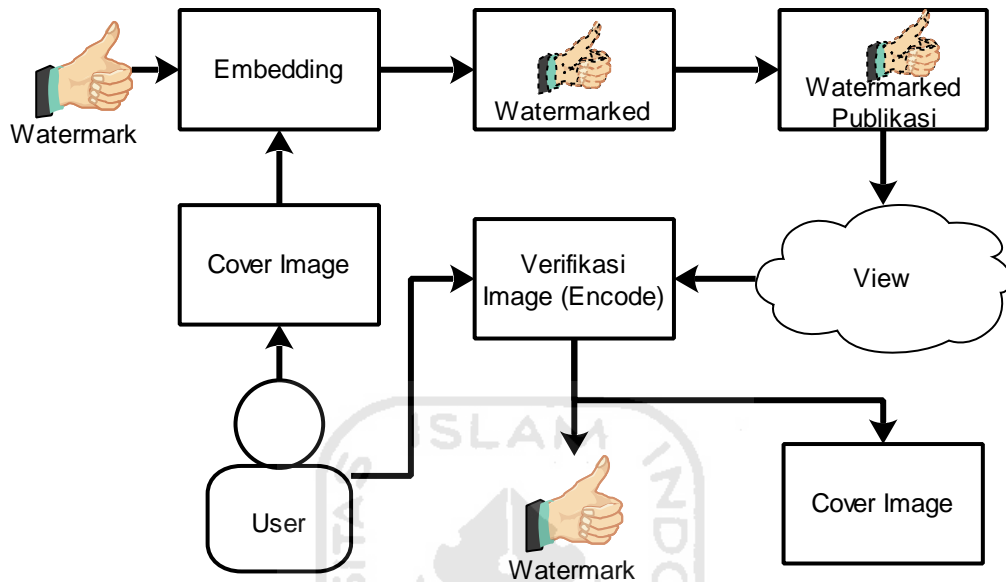


Gambar 4.5 Struktur dekomposisi algoritma RDWT pada *subband* LL_2 dan LH_2

4.5 Pengaruh *hybrid* metode *watermarking* terhadap HAKI

Watermarking merupakan teknik penyisipan pesan dalam pesan dan sering digunakan dalam perlindungan hak cipta, karena keunggulannya dapat menyembunyikan pesan kedalam pesan lain tanpa merubah file yang di tumpanginya. *Watermarking* akan dikatakan mampu melindungi hak cipta pada citra digital apabila memenuhi tingkat *imperceptibility* antara citra ter-*watermarked* dengan *cover image* mendapatkan nilai diatas standar. Parameter yang digunakan untuk mengukur tingkat *imperceptibility* yaitu nilai PSNR diatas atau sama dengan 35dB dan nilai kualitas atau Q minimal 4 (*good*) dan tingkat *robustness* atau ketahanan dengan parameter *Corr* diatas atau sama dengan 0,8 dalam mendeteksi *watermark* setelah diberikan *attack*.

Adapun bagan penerapan *Hybrid Image Watermarking* metode RDWT dengan SVD pada HAKI dapat dilihat pada gambar 4.6.



Gambar 4. 6 Bagan penerapan *Hybrid* algoritma RDWT dengan SVD pada HAKI
 Contoh perlindungan HAKI : Seorang *photographer* (A) mengunggah hasil foto nya yang sudah diberikan *watermark* ke internet untuk mengikuti lomba. Pada saat seleksi hasil Dewan juri menemukan dua gambar yang sama dengan pemilik yang berbeda, karena dewan juri bingung dan tidak bisa menentukan pemilik aslinya maka dipanggilah kedua pemilik foto tersebut untuk di selidiki, karena si A pemilik asli yang sebelumnya sudah menanamkan *watermark* kedalam hasil karyanya maka dengan mudah membuktikan bahwa gambar tersebut adalah miliknya dengan melakukan ekstraksi *watermark* dair gambar karyanya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian dengan judul “*Hybrid Image Watermarking RDWT dengan SVD untuk Perlindungan Hak Cipta pada Citra Digital*” disimpulkan sebagai berikut:

1. Pada penerapan *Hybrid Image Watermarking RDWT* dengan SVD nilai *alpha* dapat mempengaruhi tingkat *imperceptibility* terhadap citra *watermarked*, karena nilai *alpha* menunjukkan tingkat ketampakan (*visible*) *watermark* pada *cover image*. Semakin rendah nilai *alpha* maka tingkat ketampakan *watermark* semakin berkurang dan tingkat *imperceptibility* semakin tinggi. Dari pengujian yang dilakukan dengan *object grayscale* ditemukan nilai *alpha* terbaik yaitu nilai *alpha* 0.1, karena mendapatkan nilai PSNR, Q dan Corr tertinggi dari yang lainnya.
2. Berdasarkan pengujian tingkat *robustness* metode *Hybrid Image Watermarking RDWT-SVD* pada *subband* LH₂ memiliki tingkat ketahanan yang cukup tinggi terhadap upaya menghilangkan *watermark* yang menjadi identitas kepemilikan sah terhadap suatu citra digital selama citra *watermarked* tidak direkayasa dengan serangan *nois*.
3. Berdasarkan data hasil perbandingan dengan penelitian sebelumnya diketahui bahwa penyisipan pada *subband* LH₂ memiliki tingkat *imperceptibility* dan *robustness* lebih tinggi dibandingkan dengan penyisipan pada *subband* LL₂ dengan sampel citra *Grayscale*, karena *subband* LH₂ merupakan karikatur *image* yang mengandung sedikit informasi sehingga ketika penyisipan *watermark* pada *subband* ini tidak mengganggu *cover image* yang ditumpanginya.
4. Berdasarkan pengujian tingkat *imperceptibility* *Hybrid Image Watermarking RDWT* dengan SVD diketahui bahwa level histogram pada *cover image* dapat mempengaruhi tingkat *imperceptibility*. Level histogram berkonsentrasi

tinggi memiliki tingkat *imperceptibility* tertinggi dan terendah dan cenderung tidak stabil karena mendekati 0 dan 255. sedangkan histogram citra terang berada ditengah-tengahnya, karena berada dilevel histogram tengah ke tinggi sehingga lebih stabil.

5.2 Saran

1. Peningkatan tingkat *imperceptibility*, *robustness*, *capacity* dan *security* menjadi isu yang paling sering muncul jadi permasalahan dalam implementasi metode *Watermarking*, sehingga penelitian berikutnya diharapkan dapat menemukan metode *Watermarking* dengan karakteristik *imperceptibility* dan *robustness* yang tinggi.
2. Metode *Hybrid Image Watermarking* untuk pengembangan lebih lanjut perlu dilakukan kombinasi algoritma atau *hybrid* algoritma dua atau bahkan lebih untuk mendapatkan karakteristik *imperceptibility* dan *robustness* yang tinggi.
3. Penelitian *hybrid image watermarking* kedepannya dapat mengkombinasikan algoritma seperti RDWT-SVD dengan DCT, RDWT-DCT dengan SVD dan RDWT-SVD dengan RDWT-DCT.
4. Salah satu isu yang terjadi saat-saat ini adalah menyembunyikan barang bukti digital kedalam dokumen atau *image* sehingga mempersulit pihak berwenang dalam menanganinya dari itu penelitian berikutnya diharapkan dapat melakukan penelitian tentang *forensics watermark image*.

DAFTAR PUSTAKA

- Ansari, M. (2015). Komparasi Metode Hybrid Image Watermarking DWT-SVD dengan RDWT-SVD Untuk Proteksi dan Perlindungan Hak Cipta Pada Citra Digital (Master Theses). Program Studi Teknik Informatika Program Pascasarjana Fakultas Teknik Industri, Universitas Islam Indonesia, Yogyakarta.
- Ansari, M., & Prayudi, Y. (2014). Komparasi Metode Hybrid Image Watermarking DWT-SVD dengan RDWT-SVD Untuk Proteksi dan Perlindungan Hak Cipta Pada Citra Digital. *Cybermatika ITB*, 2(2), 23–29. Retrieved from <http://cybermatika.stei.itb.ac.id/ojs/index.php/cybermatika/article/view/65>
- Ariyus, D. (2009). Keamanan Multimedia : Pengenalan Konsep Multimedia, Keamanan Multimedia, Cryptography, Steganography dan Watermarking. Yogyakarta: CV. Andi Offset.
- Chen, L., & Zhao, J. (2015). Adaptive Digital Watermarking Using RDWT and SVD Lei Chen and Jiyong Zhao School of Electrical Engineering and Computer Science , University of Ottawa, 0–4.
- Hidayat, E. Y., & Udayanti, E. D. (2011). Hybrid Watermarking Citra Digital Menggunakan Teknik DWT-DCT dan SVD. Seminar Nasional Teknologi Informasi & Komunikasi Terapan (Semantik).
- Kourkchi, H., & Ghaemmaghani, S. (2008). Image adaptive semi-fragile watermarking scheme based on RDWT-SVD. *2008 International Conference on Innovations in Information Technology, IIT 2008*, 130–134. <http://doi.org/10.1109/INNOVATIONS.2008.4781744>
- Kutter, M., & Petitcola, F. A. P. (1999). A fair benchmark for image watermarking systems. *Security and Watermarking of Multimedia Contents The International Society for Optical Engineering The Computer Laboratory, University of Cambridge*, 3657, 1–14
- Lagzian, S., Soryani, M., & Fathy, M. (2011). Robust watermarking scheme based on RDWT-SVD: Embedding data in all subbands. *2011 International Symposium on Artificial Intelligence and Signal Processing, AISP 2011*, 43–47. <http://doi.org/10.1109/AISP.2011.5960985>
- Madenda, S. (2015). Pengolahan Citra & Video Digital : Teori, Aplikasi, dan Pemrograman Menggunakan MATLAB. Jakarta: Erlangga.

- Makbol, N. M., & Khoo, B. E. (2013). International Journal of Electronics and Communications (AEÜ) Robust blind image watermarking scheme based on Redundant Discrete Wavelet Transform and Singular Value Decomposition. *AEUE - International Journal of Electronics and Communications*, 67(2), 102–112. <http://doi.org/10.1016/j.aeue.2012.06.008>
- Mathworks, C. (2016). MATLAB Documentation. Retrieved from <http://www.mathworks.com/help/matlab>.
- Munir, R. (2004). *Pengolahan Citra Digital Dengan Pendekatan Algoritamtik*. Bandung: Informatika.
- Padhinary, S. (2013). International Journal of Advanced Research in Computer Science and Software Engineering Digital Watermarking Based on Redundant Discrete Wavelet Transform and Singular Value Decomposition, 3(2), 431–434.
- Prayudi, Y. (2002). *Metode Watermarking Ganda Sebagai Teknik Pengaman Pada Citra Digital* (Master Theses). Program Studi Teknik Informatik Program Pascasarjana Institut Teknologi Sepuluh November, Surabaya.
- Saxena, V. (2008). *Digital Image Watermarking* (Doctor of Philosophy). Department of Computer Science and Engineering Jaypee Institue of Information Technology University, India Seema, & Sharma, S. (2012). DWT-SVD Based Efficient Image Watermarking Algorithm to Achieve High Robustness and Perceptual Quality, 2(4), 2–5.
- Supangkat, S. H. (2000). Watermarking Sebagai Teknik Penyembunyian Label Pada Data Digital. *Departemen Teknik Elektro, Institut Teknologi Bandung*, 6(3), 19–27.
- USC-Viterbi School of Engineering Signal. (2016). The USC-SIPI Image Database Signal and Image Processing. Retrieved Juli 21, 2016, from <http://sipi.usc.edu/database/database.php>
- Wijaya, M. C., & Priyono, A. (2007). *Pengolahan Citra Digital Menggunakan Matlab*. Bandung: Informatika.

LAMPIRAN

Lampiran 1: Matlab Coding

```

function varargout = programtesis(varargin)
% PROGRAMTESIS MATLAB code for programtesis.fig
%   PROGRAMTESIS, by itself, creates a new PROGRAMTESIS or
%   raises the existing
%   singleton*.
%
%   H = PROGRAMTESIS returns the handle to a new PROGRAMTESIS
%   or the handle to
%   the existing singleton*.
%
%   PROGRAMTESIS('CALLBACK',hObject,eventData,handles,...)
%   calls the local
%   function named CALLBACK in PROGRAMTESIS.M with the given
%   input arguments.
%
%   PROGRAMTESIS('Property','Value',...) creates a new
%   PROGRAMTESIS or raises the
%   existing singleton*. Starting from the left, property
%   value pairs are
%   applied to the GUI before programtesis_OpeningFcn gets
%   called. An
%   unrecognized property name or invalid value makes property
%   application
%   stop. All inputs are passed to programtesis_OpeningFcn via
%   varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%   only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help programtesis

% Last Modified by GUIDE v2.5 15-Nov-2016 23:59:03

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @programtesis_OpeningFcn, ...
                  'gui_OutputFcn',  @programtesis_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

```

```

else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before programtesis is made visible.
function programtesis_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to programtesis (see VARARGIN)

% Choose default command line output for programtesis
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes programtesis wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = programtesis_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in btnEkstrak.
function btnEkstrak_Callback(hObject, eventdata, handles)
% hObject    handle to btnEkstrak (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

wImg = handles.gui.wmark;
[wmLL,wmLH,wmHL,wmHH] = swt2(wImg,1,'haar');
[wmLL2,wmLH2,wmHL2,wmHH2] = swt2(wmLH,1,'haar');
[wm_U, wm_S, wm_V] = svd(wmLH2);

hostImg = handles.gui.host;
[H_LL,H_LH,H_HL,H_HH] = swt2(hostImg,1,'haar');
[H_LL2,H_LH2,H_HL2,H_HH2] = swt2(H_LH,1,'haar');

```



```

[H_U, H_S, H_V] = svd(H_LH2);

alpha2 = handles.alpha2;
S_eW = (wm_S - H_S) / alpha2;

markImg = handles.gui.mark;
[W_LL, W_LH, W_HL, W_HH] = swt2(markImg, 1, 'haar');
[W_LL2, W_LH2, W_HL2, W_HH2] = swt2(W_LH, 1, 'haar');
[W_U, W_S, W_V] = svd(W_LH2);

newWLH2 = W_U * S_eW * W_V';

newWLH = iswt2(W_LL2, newWLH2, W_HL2, W_HH2, 'haar');
eImg = iswt2(W_LL, newWLH, W_HL, W_HH, 'haar');
eImg = uint8(eImg);
imwrite(eImg, 'ERDWTSVD.jpg');

extract = imread('ERDWTSVD.jpg');
handles.gui.extract = extract;

[FileName3 PathName3] = uiputfile('*.jpg', 'Save As');
if PathName3==0
return;
end
Name3 = fullfile(PathName3, FileName3);
imwrite(handles.gui.extract, Name3, 'jpg');
guidata(hObject, handles);

axes(handles.axes4);
imshow(extract);
guidata(hObject, handles);

markImg = handles.gui.mark;
eImg = handles.gui.extract;

[rows columns ~] = size(markImg);

mseCitra = (double(markImg) - double(eImg)).^2;
mse = sum(sum(mseCitra))/(rows * columns);
psnr = 10 * log10(255.^2 / mse);

set(handles.epsnr2, 'String', psnr);

cc = corr2(eImg, markImg);
set(handles.ecorr2, 'String', cc);

function ecorr2_Callback(hObject, eventdata, handles)
% hObject    handle to ecorr2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% Hints: get(hObject,'String') returns contents of ecorr2 as text
%         str2double(get(hObject,'String')) returns contents of
ecorr2 as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function ecorr2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ecorr2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in btnHost.
```

```
function btnHost_Callback(hObject, eventdata, handles)
% hObject    handle to btnHost (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
%browse file
```

```
[filename,pathname] = uigetfile('*.jpg','Pilih Host Image');
```

```
%cek file name
```

```
if isequal([filename,pathname],[0,0])
    return
```

```
else
```

```
    fullpath = fullfile(pathname,filename);
    handles.gui.fullpath = fullpath;
    handles.gui.filename = filename;
    handles.gui.pathname = pathname;
```

```
    % tampilkan citra
```

```
    addpath(handles.gui.pathname);
    host = imread(handles.gui.filename);
    handles.gui.host = host;
    axes(handles.axes1);
    imshow(host);
    guidata(hObject, handles);
```

```
end
```

```
% --- Executes on button press in btnMark.
```

```
function btnMark_Callback(hObject, eventdata, handles)
% hObject    handle to btnMark (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)

%browse file
[filename2,pathname2] = uigetfile('*.*jpg;', 'Pilih Mark Image');
%cek file name
if isequal([filename2,pathname2],[0,0])
    return
else
    fullpath2 = fullfile(pathname2,filename2);
    handles.gui.fullpath2 = fullpath2;
    handles.gui.filename2 = filename2;
    handles.gui.pathname2 = pathname2;

    % tampilkan citra
    addpath(handles.gui.pathname2);
    mark = imread(handles.gui.filename2);
    handles.gui.mark = mark;
    axes(handles.axes2);
    imshow(mark);
    guidata(hObject, handles);
end

% --- Executes on button press in btnProses.
function btnProses_Callback(hObject, eventdata, handles)
% hObject      handle to btnProses (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

hostImg = handles.gui.host;
[h_ll, h_lh, h_hl, h_hh]=swt2(hostImg,1,'haar');
[h_ll2, h_lh2, h_hl2, h_hh2]=swt2(h_lh,1,'haar');
[U_h, S_h, V_h]=svd(h_lh2);

markImg = handles.gui.mark;
[w_ll, w_lh, w_hl, w_hh]=swt2(markImg,1,'haar');
[w_ll2, w_lh2, w_hl2, w_hh2]=swt2(w_lh,1,'haar');
[U_w, S_w, V_w]=svd(w_lh2);

%proses penyisipan
alpa1 = handles.alpa1;
S_em = S_h + (alpa1 * S_w);

new_lh2 = U_h * S_em * V_h';

%output atau invers
new_lh = iswt2(h_ll2, new_lh2, h_hl2, h_hh2,'haar');
wImg = iswt2(h_ll, new_lh, h_hl, h_hh,'haar');
imwrite(uint8(wImg), 'wRDWTSVD.jpg');
wmark = imread('wRDWTSVD.jpg');
handles.gui.wmark = wmark;

%simpan

```

```

[FileName2 PathName2] = uiputfile('*.jpg', 'Save As');
if PathName2==0
return;
end
Name2 = fullfile(PathName2, FileName2);
imwrite(handles.gui.wmark, Name2, 'jpg');
guidata(hObject, handles);
%baca kembali
axes(handles.axes3);
imshow(wmark);
%deklarasian psnr
wImg = handles.gui.wmark;
hostImg = handles.gui.host;
%dijadikan matrik
[rows columns ~] = size(hostImg);
%psnr
mseCitra = (double(hostImg) - double(wImg)).^2;
mse = sum(sum(mseCitra))/(rows * columns);
psnr = 10 * log10(255.^2 / mse);

set(handles.epsnr1, 'String', psnr);
%Corr
cc = corr2(wImg, hostImg);
set(handles.ecorr1, 'String', cc);
%Quality
ssim(wImg, hostImg);
Q=ssim(wImg, hostImg);
set(handles.equality, 'String', Q);
guidata(hObject, handles);

function epsnr1_Callback(hObject, eventdata, handles)
% hObject    handle to epsnr1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of epsnr1 as text
%        str2double(get(hObject, 'String')) returns contents of
epsnr1 as a double

% --- Executes during object creation, after setting all
properties.
function epsnr1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to epsnr1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))

```

```

    set(hObject, 'BackgroundColor', 'white');
end

function ecorr1_Callback(hObject, eventdata, handles)
% hObject    handle to ecorr1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of ecorr1 as text
%        str2double(get(hObject, 'String')) returns contents of
ecorr1 as a double

% --- Executes during object creation, after setting all
properties.
function ecorr1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ecorr1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function equality_Callback(hObject, eventdata, handles)
% hObject    handle to equality (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of equality as
text
%        str2double(get(hObject, 'String')) returns contents of
equality as a double

% --- Executes during object creation, after setting all
properties.
function equality_CreateFcn(hObject, eventdata, handles)
% hObject    handle to equality (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function epsnr2_Callback(hObject, eventdata, handles)
% hObject     handle to epsnr2 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of epsnr2 as text
%         str2double(get(hObject,'String')) returns contents of
epnsr2 as a double

% --- Executes during object creation, after setting all
properties.
function epsnr2_CreateFcn(hObject, eventdata, handles)
% hObject     handle to epsnr2 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btnWMark.
function btnWMark_Callback(hObject, eventdata, handles)
% hObject     handle to btnWMark (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

%browse file
[filename3,pathname3] = uigetfile('*.jpg;', 'Pilih Watermarked
Image');
%cek file name
if isequal([filename3,pathname3],[0,0])
    return
else
    fullpath3 = fullfile(pathname3,filename3);
    handles.gui.fullpath3 = fullpath3;
    handles.gui.filename3 = filename3;
    handles.gui.pathname3 = pathname3;

    % tampilkan citra
    addpath(handles.gui.pathname3);

```

```

    wmark = imread(handles.gui.filename3);
    handles.gui.wmark = wmark;
    axes(handles.axes3);
    imshow(wmark);
    guidata(hObject, handles);
end

function input_alpa2_Callback(hObject, eventdata, handles)
% hObject    handle to input_alpa2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input_alpa2 as
text
%         str2double(get(hObject,'String')) returns contents of
input_alpa2 as a double
alpha2 = str2double(get(handles.input_alpa2, 'String'));
handles.alpha2 = alpha2;
guidata(hObject, handles)

% --- Executes during object creation, after setting all
properties.
function input_alpa2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_alpa2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_alpa1_Callback(hObject, eventdata, handles)
% hObject    handle to input_alpa1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input_alpa1 as
text
%         str2double(get(hObject,'String')) returns contents of
input_alpa1 as a double
alpha1 = str2double(get(handles.input_alpa1, 'String'));
handles.alpha1 = alpha1;
guidata(hObject, handles)

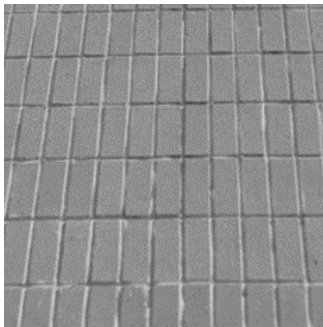
```

```
% --- Executes during object creation, after setting all
properties.
function input_alpa1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_alpa2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

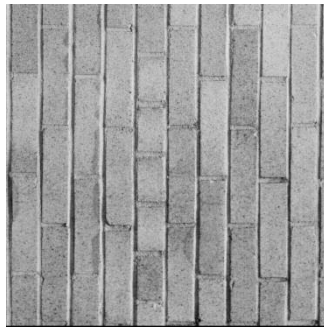
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```



Lampiran 2: Citra *Host* atau *Cover Image*



Brick wall



Brodats – Brick wall



Brodatz - Bark



San diego (Point loma)



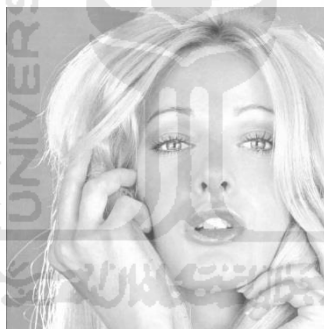
North island NAS



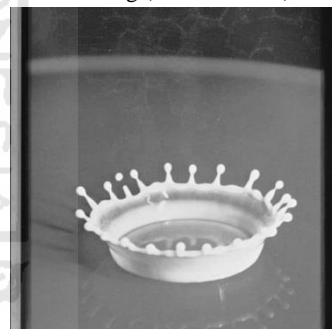
San dieg (Shelter island)



F1



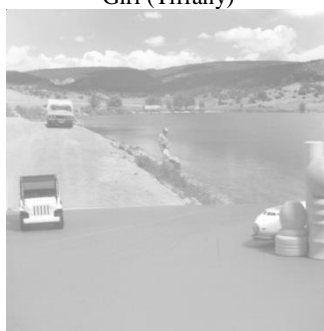
Girl (Tiffany)



Splash



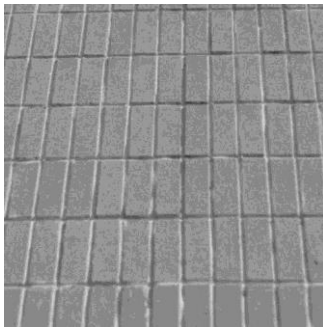
Motion01



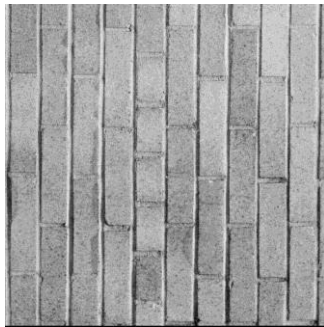
Motion02



Motion03

Lampiran 3: Citra Ter-Watermarked

emBrick wall_0.1



emBrodats-BW_0.1



emBrodatz – Bark_0.1



emSan diego (Point loma)_0.1



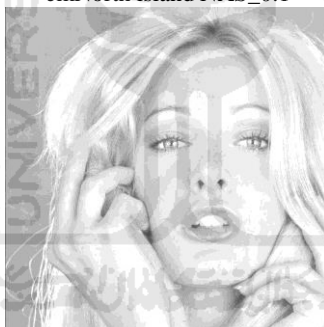
emNorth island NAS_0.1



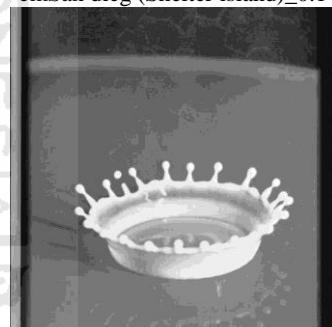
emSan dieg (Shelter island)_0.1



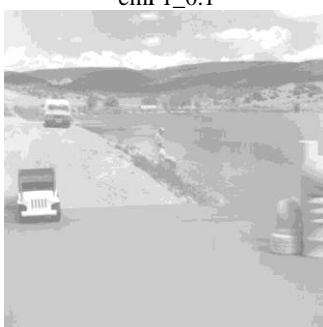
emF1_0.1



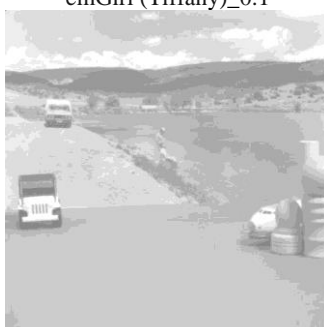
emGirl (Tiffany)_0.1



emSplash_0.1



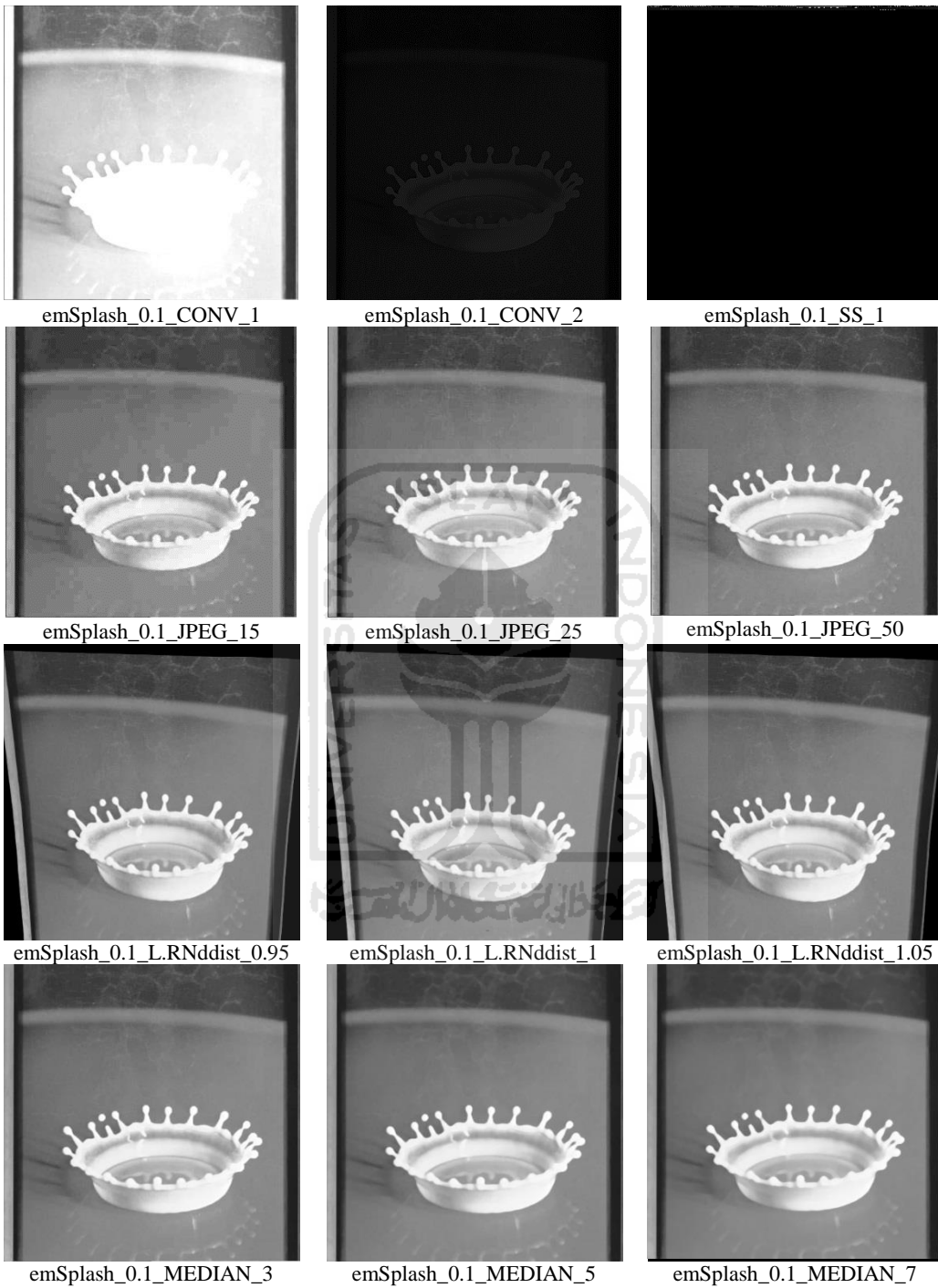
emMotion01_0.1



emMotion02_0.2



emMotion03_0.1

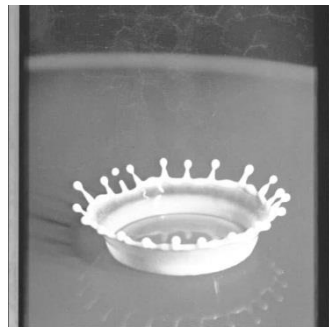
Lampiran 4: Citra Setelah diberikan Serangan



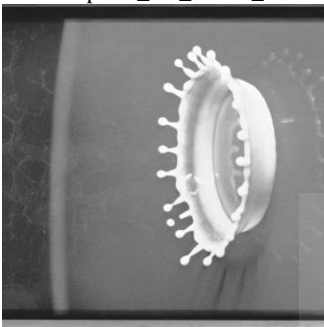
emSplash_0.1_PSNR_10



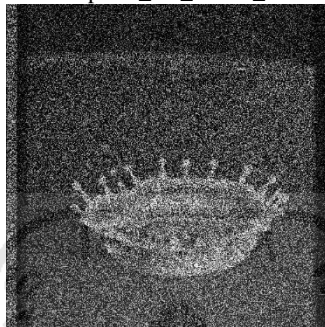
emSplash_0.1_PSNR_50



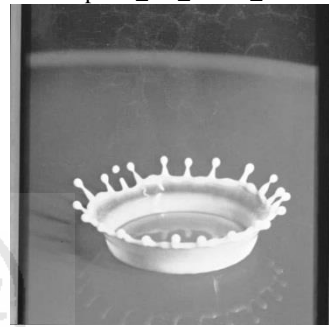
emSplash_0.1_PSNR_100



emSplash_0.1_ROT_90



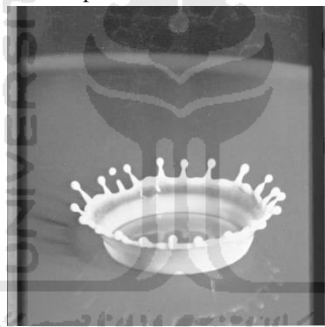
emSplash_0.1_NOISE_20



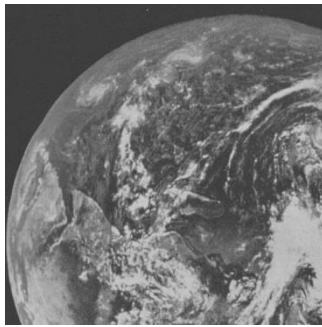
emSplash_0.1_RotSCALE_-0.5



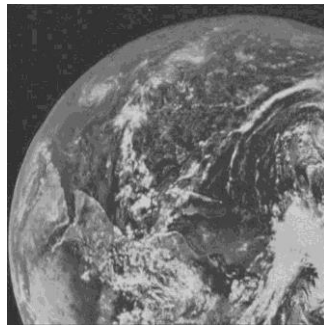
emSplash_0.1_RotSCALE_-0.25



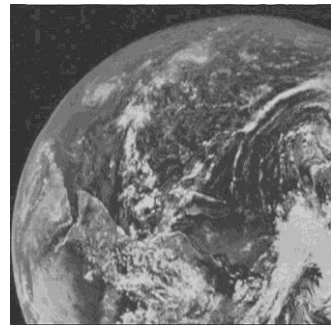
emSplash_0.1_RotSCALE_0.25

Lampiran 5: Citra *Watermark* dan Hasil Ekstraksi

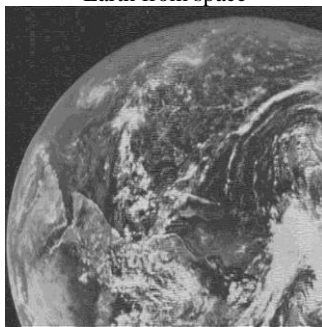
Earth from space



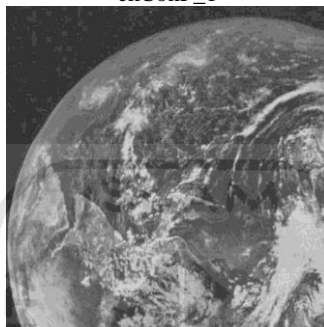
ekConF_1



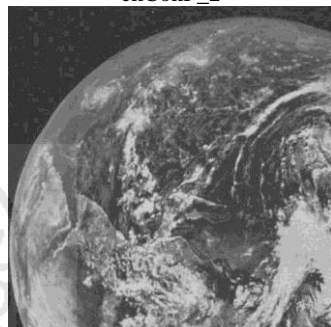
ekConF_2



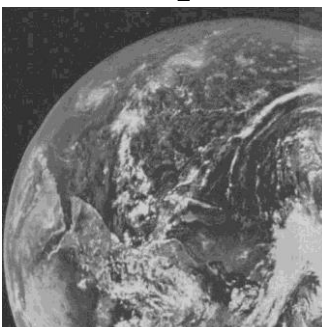
ekSS_1



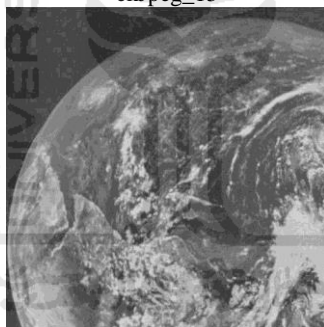
ekJpeg_15



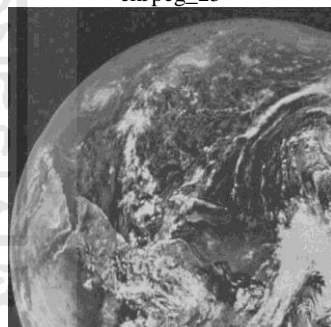
ekJpeg_25



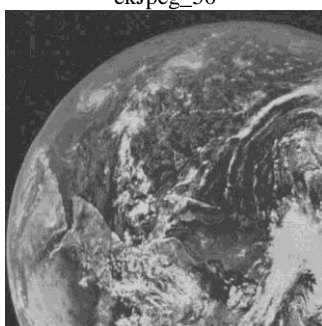
ekJpeg_50



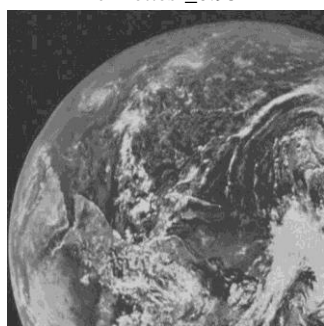
ekLettesr_0.95



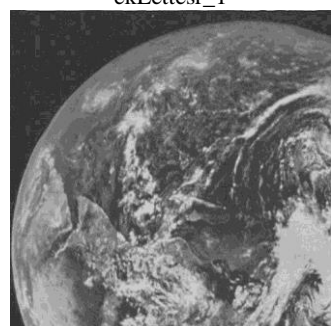
ekLettesr_1



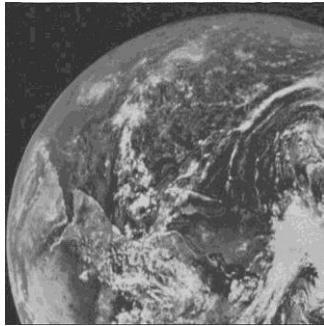
ekLastesr_1.05



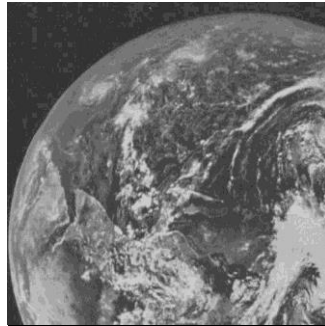
ekMedian_3



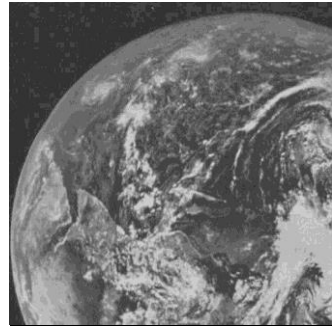
ekMedian_5



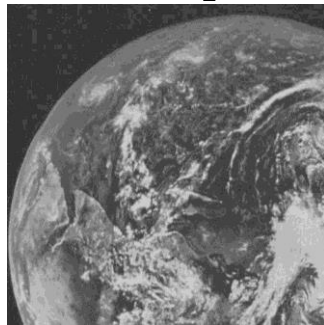
ekMedian_7



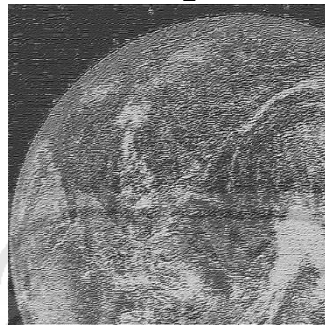
ekPsnr_10



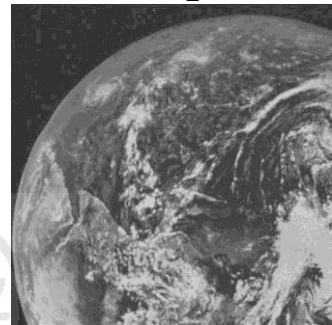
ekPsnr_50



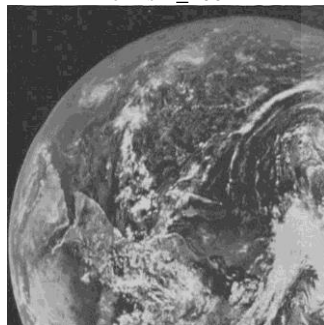
ekPsnr_100



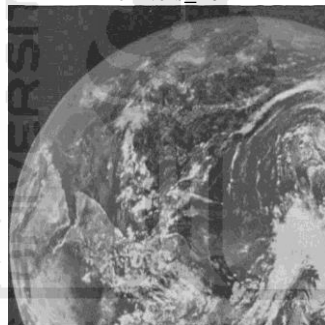
ekNois_20



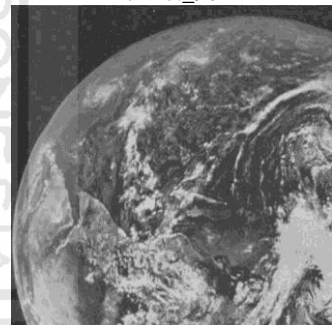
ekRot_90



ekRotScale_-0.5



ekRotScale_-0.25



ekRotScale_0.25