WATERMARKING PADA CITRA DIGITAL MENGGUNAKAN *DISCRETE WAVELET TRANSFORM*

Dean Fathony Alfatwa – NIM : 13503003

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail : if13003@students.if.itb.ac.id

Abstrak

Kemudahan penyebaran citra *digital* melalui internet memiliki sisi positif dan negatif terutama bagi pemilik asli citra *digital* tersebut. Sisi positif dari kemudahan penyebaran adalah dengan cepatnya pemilik citra tersebut menyebarkan *file* citra *digital* ke berbagai alamat situs di dunia. Sedangkan sisi negatifnya adalah jika tidak ada hak cipta yang berfungsi sebagai pelindung citra yang disebarkan tersebut, maka citra *digital* ini, yang misalakan adalah hasil foto komersil, atau hasil karya lukisan *digital*, akan sangat mudah diakui kepemelikannya oleh pihak lain.

Watermark merupakan salah satu solusi untuk melindungi hak cipta terhadap foto *digital* yang dihasilkan. Dengan diterapkannya *Digital Image Watermarking* ini maka hak cipta foto *digital* yang dihasilkan akan terlindungi dengan cara menyisipkan informasi tambahan seperti informasi pemilik, keaslian, dsb ke dalam foto *digital* tersebut. *Discrete Wavelet Transform* (*DWT*) merupakan salah satu kakas yang banyak digunakan dalam teknik *watermarking* dalam *domain transform*.

Penyisipan *watermark* ke dalam citra *digital* menggunakan *DWT* dijalankan dengan menggunakan aplikasi yang dibangun di lingkungan *desktop*. Aplikasi ini diberi nama Cammar. Cammar dibangun menggunakan bahasa pemograman Java versi 1.6. Sedangkan kompilator sekaligus IDE yang digunakan untuk memudahkan pengembangan perangkat lunak adalah NetBeans 6.0. Pemberian beberapa jenis serangan terhadap citra hasil penyisipan *watermark* menggunakan Cammar mengubah *watermark* yang telah disisipkan. Serangan yang mengubah *watermark* tersebut antara lain *bluring, sharpening,* dan penambahan *noise*.

Kata kunci: citra *digital*, *watermarking*, *Discrete Wavelet Transform* (*DWT*).

1. Pendahuluan

Kemudahan penyebaran citra *digital* melalui internet memiliki sisi positif dan negatif terutama bagi pemilik asli citra *digital* tersebut. Sisi positif dari kemudahan penyebaran tersebut adalah dengan cepatnya pemilik citra tersebut menyebarkan *file* citra *digital* tersebut ke berbagai alamat di dunia. Sedangkan sisi negatifnya adalah jika tidak ada hak cipta pelindung citra yang disebarkan tersebut, maka citra *digital* ini, yang misalakan adalah hasil foto komersil, atau hasil karya lukisan *digital*, akan sangat mudah diakui kepemelikannya oleh pihak lain.

Watermark merupakan salah satu solusi untuk melindungi hak cipta terhadap foto *digital* yang dihasilkan. Dengan diterapkannya *Digital Image Watermarking* ini maka hak cipta foto *digital* yang dihasilkan akan terlindungi dengan cara menyisipkan informasi tambahan seperti informasi pemilik, keaslian, dsb ke dalam foto *digital* tersebut. *Watermarking* adalah salah satu teknik penyembunyian data yang fungsinya untuk melindungi data yang disisipi dengan informasi lain dengan tujuan untuk melindungi hak milik, *copyright,* dsb. Teknik penyembunyian data sendiri terbagi menjadi dua *domain* yaitu, *domain* spasial yang jika dihubungkan dengan *Digital Image Watermarking* yaitu penyisipan *watermark* dilakukan secara langsung ke dalam *pixel* citra, dan *domain transform* yang menyisipkan *watermark* ke dalam koefisien transformasi [RIN06].

Digital Image Watermarking sendiri memliki beberapa jenis teknik yang memiliki keunggulan dan kelemahan masing-masing. Biasanya teknik *watermarking* yang kuat (susah dipecahkan oleh berbagai serangan) memiliki kualitas gambar ber-*watermark* yang kurang memuaskan, demikian juga sebaliknya, teknik *watermarking* yang menghasilkan kualitas gambar yang memuaskan biasanya kurang kuat menghadapi serangan [KUT99]. Secara garis besar teknik *watermarking* dibedakan menjadi dua yaitu [GIL00]:

1. *Private Watermarking* / *Incomplete Watermarking* / *Escrow Watermarking*
   Merupakan teknik *watermarking* yang membutuhkan citra asli dan citra ber-*watermark* untuk megekstraksi *watermark*.
2. *Public Watermarking* / *Complete Watermarking* / *Oblivious Watermarking* / *Blind Watermarking*
   Teknik *watermarking* yang tidak membutuhkan citra asli atau *watermark* yang disisipkan untuk melakukan ekstraksi.

Discrete Wavelet Transform (*DWT*) merupakan salah satu kakas yang banyak digunakan dalam teknik *blind watermarking* dan *escrow watermarking* dengan *domain transform*. *Watermarking* yang berbasis *wavelet* adalah pendekatan yang populer karena kekutannya melawan *malicious attack* [KEJ04]. *DWT* membagi sebuah dimensi sinyal menjadi dua bagian, biasanya bagian dengan frekuensi tinggi dan frekuensi rendah, yang disebut dengan dekomposisi [TER06]. Sebuah sinyal dilewatkan melalui *highpass filter* untuk menganalisis frekuensi tinggi, dan dilewatkan melalui *lowpass filter* untuk menganalisis frekuensi rendah. Keluaran dari *highpass filter* dan *lowpass filter* ini menghasilkan koefisien *DWT*, dengan menggunakan koefisien ini citra asli dapat direkonstruksi. Proses rekonstruksi ini disebut *Inverse Discrete Wavelet Transform* (*IDWT*). Secara umum penyisipan *watermark* ke dalam citra dilakukan dengan cara membandingkan koefisien *DWT* dari dekomposisi citra, dimana koefisien yang memiliki nilai terbesar adalah tempat yang paling signifikan untuk menyisipkan *watermark*.

Watermarking dalam *Discrete Wavelet Transform* (*DWT*) *domain* ini dipilih karena beberapa alasan yaitu :

1. *DWT* merupakan yang paling dekat terhadap *HVS* (*Human Visual System*) [TER06].
2. Distorsi yang disebabkan oleh *wavelet domain* dalam perbandingan kompresi tinggi tidak terlalu mengganggu dibandingkan *domain* lain dalam *bit rate* yang sama [TER06].
3. *Bit-error rate* yang rendah. *Bit-error rate* merupakan perbandingan antara *bit* yang salah diekstraksi dengan total *bit* yang disisipkan [KUT99].

2. Citra *Digital*

Citra *digital* sebenarnya bukanlah sebuah data *digital* yang normal, melainkan sebuah representasi dari citra asal yang bersifat analog [TEC06]. Citra *digital* ditampilkan pada layar komputer dengan berbagai macam susunan warna dan tingkat kecerahan. Susunan warna inilah yang menyebabkan sebuah citra bersifat analog. Hal ini disebabkan karena susunan warna yang dimiliki dalam sebuah citra mengandung jumlah warna dan tingkat kecerahan yang tidak terbatas [TEC06]. Citra yang ditampilkan pada layar komputer ini, yang sebenarnya merupakan sebuah representasi analog, juga tersusun dari sebuah rentang tak terbatas dari nilai cahaya yang dipantulkan atau cahaya yang ditransmisikan. Jadi secara umum citra memiliki sifat kontinu dalam tampilan warna dan tingkat kecerahannya.

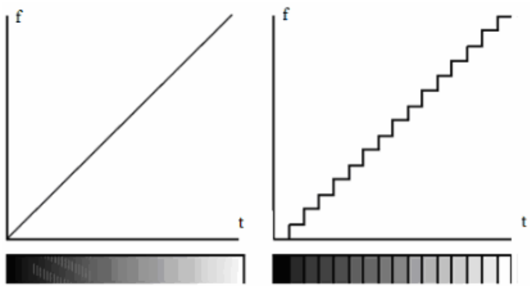
2.1. Pembentukan Citra *Digital*

Komputer merupakan alat yang beroperasi dalam sistem *digital* yang menggunakan *bit* atau *byte* dalam pengukuran datanya, dan yang terpenting dalam sistem *digital* adalah sifatnya yang diskrit, bukan kontinu. Hal ini berlawanan dengan citra *digital* yang sebenarnya merupakan representasi citra asal yang bersifat kontinu. Untuk mengubah citra yang bersifat kontinu diperlukan sebuah cara untuk mengubahnya dalam bentuk data *digital*. Komputer menggunakan sistem bilangan biner dalam pemecahan masalah ini [TEC06]. Dengan penggunaan sistem bilangan biner ini, citra dapat diproses dalam komputer dengan sebelumnya mengekstrak informasi citra analog asli dan mengirimkannya ke komputer dalam bentuk biner. Proses ini disebut dengan digitisasi

[TEC06]. Digitisasi dapat dilakukan oleh alat seperti kamera *digital* atau *scanner*. Kedua alat ini selain dapat mengambil atau menangkap sebuah citra, juga dapat bertindak sebagai alat *input* (masukan) bagi komputer. Alat penangkap citra *digital* ini dapat menyediakan aliran data biner bagi komputer yang didapatkan dari pembacaan tingkat kecerahan pada sebuah citra asli dalam interval sumbu *x* dan sumbu *y* [TEC06].

Citra *digital* merupakan citra yang tersusun dari piksel diskrit dari tingkat kecerahan dan warna yang telah terkuantisasi [OSU07]. Jadi, pada dasarnya adalah sebuah citra yang memiliki warna dan tingkat kecerahan yang kontinu perlu diubah dalam bentuk informasi warna, tingkat kecerahan, dsb yang bersifat diskrit untuk dapat menjadi sebuah citra *digital*. Pada Gambar 1 diperlihatkan kurva tingkat kecerahan yang kontinu dengan nilai hitam dan putih yang tidak terbatas (a) dan kurva tingkat kecerahan setelah mengalami kuantisasi dalam 16 tingkatan diskrit (b).

Tingkat kecerahan pada Gambar 1 (a) yang bersifat kontinu dapat diubah menjadi tingkat kecerahan seperti Gambar 1 (b) dengan pembacaan tingkat kecerahan menggunakan interval tertentu pada sumbu x dan y seperti yang telah disebutkan di atas. Pembagian seperti pada pembagian tingkat kecerahan ini juga berlaku untuk warna agar nilai warna dapat menjadi diskrit.



(a) (b)

Gambar 1. (a) Tingkat kecerahan yang kontinu, (b) tingkat kecerahan setelah mengalami kuantisasi 16 tingkatan diskrit [TEC06]. Sumbu *f* merupakan ukuran frekuensi, dan sumbu t merupakan waktu

2.2. Perbedaan Antara Format File dan Kompresi

Citra *digital* adalah sebuah *file* yang tersimpan sebagai nilai numerik dalam media magnetik atau media optikal [IMA07]. Ditinjau dari bentuknya yang merupakan sebuah *file*, citra *digital* memiliki berbagai jenis format, antara lain JPEG, GIF, PNG, BMP, dsb. Format-format *file* untuk citra *digital* ini memiliki keunggulan, kelemahan, dan tingkat komersialitasnya maing-masing. Format *file* merupakan rangkaian data yang teratur dan digunakan untuk mengkodekan informasi dalam penyimpanan atau pertukaran data [TEC05]. Format *file* dapat digambarkan sebagai sebuah bahasa tulis yang memiliki aturan-aturan sendiri dalam penulisannya. Jika digambarkan, setiap format *file* citra memiliki cara pembentukan struktur yang berbeda dimana setiap struktur ini memiliki *header* dan *body* [TEC05]. Umumnya *header* diikuti dengan *body* yang mengandung sebagian besar data.

Kompresi merupakan cara pengkodean data *file* agar lebih ringkas dan efisien [TEC05]. Seperti yang diketahui, kompresi terhadap sebuah *file* memerlukan algoritma juga. Algoritma ini berguna dalam mendefinisikan langkah-langkah yang diperlukan untuk mengurangi ukuran *file*, yang dalam hal ini merupakan tujuan dari kompresi.

Kesalahan yang sering muncul adalah pembedaan antara format *file* dengan kompresi. Contoh yang paling sering muncul adalah pembedaan antara kompresi JPEG dengan JFIF (JPEG *File Interchange Fomat*). JFIF yang diberi ekstensi *file* .jpg sering disebut *file* dengan format JPEG, bukan *file* yang dikompresi menggunakan jenis kompresi JPEG.

3. *Discrete Wavelet Transform*

Transformasi *wavlet* adalah sebuah transformasi matematika yang digunakan untuk menganalisis sinyal bergerak. Sinyal bergerak ini dianalisis untuk didapatkan informasi spektrum frekuensi dan waktunya secara bersamaan. Salah satu seri pengembangan transformasi *wavelet* adalah *Discrete Wavelet Transform* (*DWT*) [SRI03].

3.1. *Domain* dalam Transformasi Sinyal

Bentuk mentah dari penggambaran waktu dan amplitudo disebut dengan sinyal [SRI03]. Penggambaran dengan waktu dan amplitudo yang dikategorikan dalam *domain* waktu sering kali perlu ditransformasikan dalam *domain* lain untuk analisis dan pemrosesan sinyal. *Domain* lain selain *domain* waktu misalnya *domain* frekuensi, *domain* waktu-frekuensi, dsb. Dengan adanya transformasi sinyal ini maka informasi yang kemungkinan masih tersimpan di dalam sinyal asal dapat diidentifikasi. Informasi di dalam sinyal ini dapat ditampilkan melalui transformasi dengan cara mendapatkan spektrumnya. Spektrum yang bisa diperoleh dari sebuah sinyal dapat berupa frekuensi atau waktu tergantung dari jenis transformasi yang digunakan.

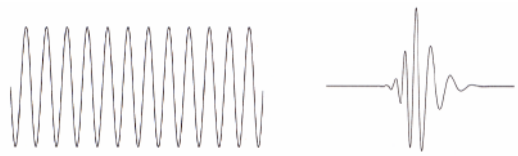
Sinyal sendiri dibagi menjadi dua jenis, yaitu sinyal tidak bergerak (*stationary signals*) dan sinyal bergerak (*non-stationary signals*). Citra dan suara merupakan salah satu contoh dari sinyal yang dapat bergerak. Contoh lain dari jenis sinyal bergerak adalah sinyal dalam bidang biologi seperti *electrocardiogram*, *electromyography*, dsb.

Untuk mendapatkan informasi dari sinyal tidak bergerak, khususnya sinyal dengan representasi frekuensi, dapat digunakan transformasi Fourier. Karena sinyal ini tidak bergerak, maka hanya perlu untuk mendapatkan spektrum frekuensi sebuah sinyal saja agar informasi dari sinyal tersebut bisa ditampilkan.

Berbeda dengan sinyal tidak bergerak, untuk menampilkan informasi dari sinyal bergerak perlu sebuah transformasi yang bisa mendapatkan spektrum frekuensi dengan keterangan waktunya. Dalam transformasi Fourier, spektrum frekuensi dari sebuah sinyal bisa didapatkan, namun, transformasi ini tidak dapat memberi tahu kapan terjadinya frekuensi sinyal tersebut. Sehingga transformasi Fourier hanya cocok untuk jenis sinyal tidak bergerak. Untuk itulah diperlukan transformasi lain untuk menampilkan informasi dari jenis sinyal bergerak ini, transformasi *Wavelet* adalah salah satunya. Transformasi ini bisa mendapatkan spektrum frekuensi dan waktu secara bersamaan. Sehingga sinyal bergerak khususnya sinyal dengan representasi waktu-frekuensi bisa diproses menggunakan transformasi ini.

3.2. *Wavelet*

Gelombang (*wave*) adalah sebuah fungsi yang bergerak naik turun ruang dan waktu secara periodik (Gambar 2 a). Sedangkan *wavelet* merupakan gelombang yang dibatasi atau terlokalisasi [SRI03] (Gambar 2 b). Atau dapat dikatakan sebagai gelombang pendek [ANS07]. *Wavelet* ini menkonsentrasikan energinya dalam ruang dan waktu sehingga cocok untuk menganalisis sinyal yang sifatnya sementara saja.



(a) (b)

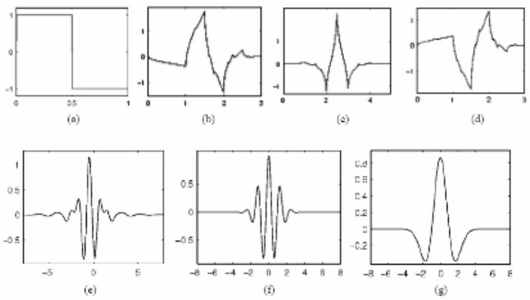
Gambar 2. (a) Gelombang (*wave*), (b) *wavelet* [SRI03]

Wavelet pertama kali digunakan dalam analisis dan pemrosesan *digital* dari sinyal gempa bumi, yang tercantum dalam literatur oleh A. Grossman dan J. Morlet [KIS07]. Penggunaan *wavelet* pada saat ini sudah semakin berkembang dengan munculnya area sains terpisah yang berhubungan dengan analisis *wavelet* dan teori transformasi *wavelet*. Dengan munculnya area sains ini *wavelet* mulai digunakan secara luas dalam filterasi dan pemrosesan data, pengenalan citra, sintesis dan pemrosesan berbagai variasi sinyal, kompresi dan pemrosesan citra, dll.

3.3. Transformasi *Wavelet* (*Wavelet Transform*)

Transformasi sinyal merupakan bentuk lain dari penggambaran sinyal yang tidak mengubah isi infomasi dalam sinyal tersebut. Transformasi *wavelet* (*wavelet transform*) menyediakan penggambaran frekuensi waktu dari sinyal. Pada awalnya, transformasi *wavelet* digunakan untuk menganalisis sinyal bergerak (*non-stationary signals*). Sinyal bergerak ini dianalisis dalam transformasi *wavelet* dengan menggunakan teknik *multi-resolution analysis*. Secara umum teknik *multi-resolution analysis* adalah teknik yang digunakan untuk menganalisis frekuensi dengan cara frekuensi yang berbeda dianalisis menggunakan resolusi yang berbeda. Resolusi dari sinyal merupakan ukuran jumlah informasi di dalam sinyal yang dapat berubah melalui operasi filterisasi [POL98].

Transformasi *wavelet* memiliki dua seri dalam pengembangannya yaitu *Continous Wavelet Transform* (*CWT*) dan *Discrete Wavelet Transform* (*DWT*). Semua fungsi yang digunakan dalam transformasi *CWT* dan *DWT* diturunkan dari *mother wavelet* melalui translasi/pergeseran) dan penskalaan/kompresi. *Mother wavelet* merupakan fungsi dasar yang digunakan dalam transformasi *wavelet* [SRI03]. Karena *mother wavelet* menghasilkan semua fungsi *wavelet* yang digunakan dalam transformasi melalui translasi dan penskalaan, maka *mother wavelet* juga akan menentukan karakteristik dari transformasi *wavelet* yang dihasilkan. Oleh karena itu, perlu pencatatan secara teliti terhadap penerapan *wavelet* dan pemilihan yang tepat terhadap *mother wavelet* harus dilakukan agar dapat menggunakan transformasi *wavelet* secara efisien. Fungsi-fungsi yang termasuk di dalam keluarga *wavelet* dipaparkan pada Gambar 3.



Gambar 3. Keluarga *Wavelet* (a)Haar, (b)Daubechies, (c)Coiflet, (d)Symlet, (e)Meyer, (f)Morlet, (g)Mexican Hat. Dengan sumbu *x* merupakan waktu, *t* dan sumbu *y* merupakan $\psi(t)$ [SRI03].

Seri pengembangan *Continous Wavelet Transform* (*CWT*) dipaparkan pada persamaan 1.

$$X_{WT}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^{*}\left(\frac{t-\tau}{s}\right) dt \qquad \textbf{(1)}$$

$x(t)$ merupakan sinyal yang akan dianalisis, $\psi(t)$ adalah *mother wavelet* atau fungsi dasar yang dipilih. τ merupakan parameter translasi yang berhubungan dengan informasi waktu pada transformasi *wavelet*. Parameter skala s didefinisikan sebagai |1/frekuensi| dan berhubungan dengan informasi frekuensi. Dengan adanya penskalaan ini sinyal dapat diperbesar atau dikompresi. Penskalaan besar (frekuensi rendah) menyebabkan sinyal diperbesar dan dapat memberikan informasi detil yang tersembunyi di sinyal, sedangkan penskalaan kecil (frekuensi tinggi) menyebababkan kompresi sinyal dan memberikan informasi global dari sinyal.

Seri pengembangan kedua dari transformasi *wavelet* adalah *Discrete Wavelet Transform* (*DWT*). Seri pengembangan ini merupakan seri *CWT* yang didiskritkan. Dengan pendiskritan *CWT* ini maka perhitungan dalam *CWT* dapat dibantu dengan menggunakan komputer.

3.4. *Discrete Wavelet Transform* (*DWT*)

Dasar dari *DWT* dimulai pada tahun 1976 dimana teknik untuk mendekomposisi sinyal waktu diskrit ditemukan [SRI03]. Di dalam *CWT*, sinyal dianalisis menggunakan seperangkat fungsi dasar yang saling berhubungan dengan penskalaan dan transisi sederhana. Sedangkan di dalam *DWT*, penggambaran sebuah skala waktu sinyal *digital* didapatkan dengan menggunakan teknik filterisasi *digital*. Secara garis besar proses dalam teknik ini adalah dengan melewatkan sinyal yang akan dianalisis pada filter dengan frekuensi dan skala yang berbeda.

Filterisasi sendiri merupakan sebuah fungsi yang digunakan dalam pemrosesan sinyal. *Wavelet* dapat direalisasikan menggunakan iterasi filter dengan penskalaan. Resolusi dari sinyal, yang merupakan rata-rata dari jumlah detil informasi dalam sinyal, ditentukan melalui filterasi ini dan skalanya didapatkan dengan *upsampling* dan *downsampling* (*subsampling*).

Sebuah sinyal harus dilewatkan dalam dua filterisasi *DWT* yaitu *highpass filter* dan *lowpass filter* agar frekuensi dari sinyal tersebut dapat dianalisis. Analisis sinyal dilakukan terhadap hasil filterisasi *highpass filter* dan *lowpass filter* di mana *highpass filter* digunakan untuk menganalisis frekuensi tinggi dan *lowpass filter* digunakan untuk menganalisis frekuensi rendah. Analisis terhadap frekuensi dilakukan dengan cara menggunakan resolusi yang dihasilkan setelah sinyal melewati filterisasi. Analisis frekuensi yang berbeda dengan menggunakan resolusi yang berbeda inilah yang disebut dengan *multi-resolution analysis*, seperti yang telah disinggung pada bagian Transformasi *Wavelet*.

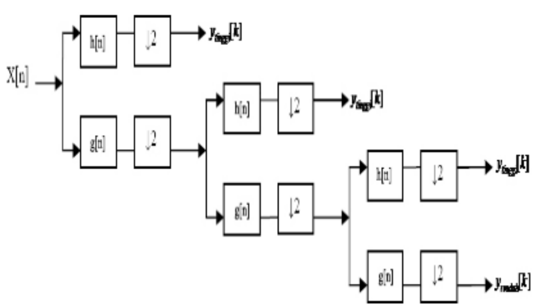
Pembagian sinyal menjadi frekuensi tinggi dan frekuensi rendah dalam proses filterisasi *highpass filter* dan *lowpass filter* disebut sebagai dekomposisi [TER06]. Proses dekomposisi dimulai dengan melewatkan sinyal asal melewati *highpass filter* dan *lowpass filter*. Misalkan sinyal asal ini memiliki rentang frekuensi dari 0 sampai dengan π rad/s. Dalam melewati *highpass filter* dan *lowpass filter* ini, rentang frekuensi di-*subsample* menjadi dua, sehingga rentang frekuensi tertinggi pada masing-masing *subsample* menjadi π/2 rad/s. Setelah filterisasi, setengah dari *sample* atau salah satu *subsample* dapat dieliminasi berdasarkan aturan Nyquist [TER06, SRI03]. Sehingga sinyal dapat selalu di-*subsample* oleh 2 ($\downarrow 2$) dengan cara mengabaikan setiap *sample* yang kedua.

Proses dekomposisi ini dapat melalui satu atau lebih tingkatan. Dekomposisi satu tingkat ditulis dengan ekspresi matematika pada persamaan 2 dan 3.

$$y_{tinggi}[k] = \sum_{n} x[n]h[2k-n] \qquad \textbf{(2)}$$

$$y_{rendah}[k] = \sum_{n} x[n]g[2k-n] \qquad \textbf{3)}$$

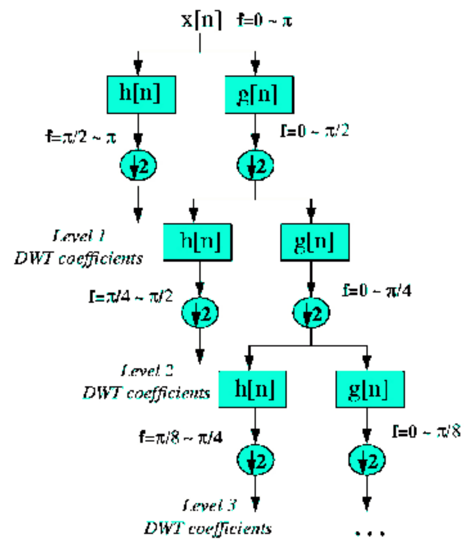
$y_{tinggi}[k]$ dan $y_{rendah}[k]$ adalah hasil dari *highpass filter* dan *lowpass filter*, $x[n]$ merupakan sinyal asal, $h[n]$ adalah *highpass filter*, dan $g[n]$ adalah *lowpass filter*. Untuk dekomposisi lebih dari satu tingkat, prosedur pada rumus 2 dan 3 dapat digunakan pada masing-masing tingkatan. Contoh penggambaran dekomposisi dipaparkan pada Gambar 4 dengan menggunakan dekomposisi tiga tingkat.



Gambar 4. Dekomposisi *wavelet* tiga tingkat [SRI03]

Pada Gambar 4, $y_{tinggi}[k]$ dan $y_{rendah}[k]$ yang merupakan hasil dari *highpass filter* dan *lowpass filter*, $y_{tinggi}[k]$ disebut sebagai koefisien *DWT* [POL98]. $y_{tinggi}[k]$ merupakan detil dari informasi sinyal, sedangkan $y_{rendah}[k]$ merupakan taksiran kasar dari fungsi pensakalaan. Dengan menggunakan koefisien *DWT* ini maka dapat dilakukan proses *Inverse Discrete Wavelet Transform* (*IDWT*) untuk merekonstruksi menjadi sinyal asal.

DWT menganalisis sinyal pada frekuensi berbeda dengan resolusi yang berbeda melalui dekomposisi sinyal sehingga menjadi detil informasi dan taksiran kasar. *DWT* bekerja pada dua kumpulan fungsi yang disebut fungsi penskalaan dan fungsi *wavelet* yang masing-masing berhubungan dengan *lowpass filter* dan *highpass filter* [POL98]. Seperti yang telah dijelaskan sebelumnya dekomposisi ini didasarkan pada aturan Nyquist yang salah satunya mengatakan bahwa frekuensi komponen *sample* harus kurang atau sama dengan setengah dari frekuensi *sampling* [AGI07]. Jadi diambil frekuensi *sample* π/2 dari frekuensi *sampling* π dalam *subsample* oleh 2 pada dekomposisi *wavelet*. Sebagai penggambaran dekomposisi *wavelet* dengan sinyal asal $x[n]$ yang memilki frekuensi maksimum f = π dipaparkan pada Gambar 5.



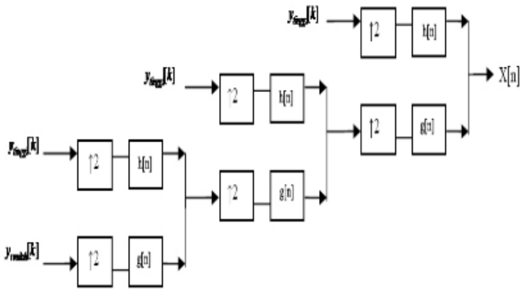
Gambar 5. Dekomposisi *wavelet* dengan frekuensi sinyal asal *f=0*~π [SRI03]

Proses rekonstruksi diawali dengan menggabungkan koefisien *DWT* dari yang berada pada akhir dekomposisi dengan sebelumnya meng-*upsample* oleh 2 ($\uparrow 2$) melalui *highpass filter* dan *lowpass filter*. Proses rekonstruksi ini sepenuhnya merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi. Sehingga persamaan rekonstruksi pada masing-masing tingkatan dapat ditulis sbb:

$$x[n] = \sum_{k} \left(y_{tinggi}[k]h[-n+2k] + y_{rendah}[k]g[-n+2k] \right)$$

(4)

Proses rekonstruksi *wavelet* untuk mendapatkan sinyal asal dengan tiga tingkatan digambarkan pada Gambar 6.



Gambar 6. Rekonstruksi *wavelet* tiga tingkat [SRI03]

3.5. Penerapan *DWT* dalam Kompresi Citra

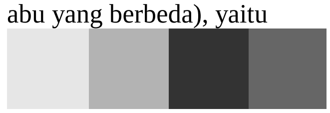
Kompresi dalam citra menggunakan *DWT* berhubungan dengan dekomposisi terhadap citra tersebut. Citra yang merupakan sinyal bergerak ini didekomposisi sama seperti cara dekomposisi sinyal yang telah dipaparkan pada bagian sebelumnya.

Secara umum, citra (sinyal bergerak) merupakan rangkaian gelombang yang memiliki banyak puncak dan lembah. Masing-masing gelombang dalam rangkaian gelombang dari sebuah citra biasanya mewakili *channel* warna (Merah, Hijau, dan Biru). Puncak dan lembah citra dipusatkan ke titik nol, selanjutnya transformasi sinyal menyimpan jarak dari titik nol menuju titik sepanjang gelombang, jarak ini disebut dengan koefisien. Koefisien yang berdekatan kemudian dirata-rata untuk mendapatkan gelombang yang lebih sederhana dan menghasilkan citra dengan resolusi atau tingkat kedetilan setengah dari semula. Koefisien yang telah dirata-rata kemudian dibagi lagi seterusnya hingga mendapatkan gelombang yang sangat sederhana. Proses ini merupakan dekomposisi pada citra.

Transformasi *wavelet* dapat menghasilkan versi resolusi citra yang sangat sederhana, oleh karena itu diperlukan perkiraan bentuk umum serta warna (informasi) dari citra untuk dapat merekonstruksi sebuah citra.

Transformasi *wavelet* dapat mengidentifikasi variasi yang signifikan dalam sebuah citra. Variasi ini berhubungan dengan tempat di mana proses penyederhanaan terjadi. Pada saat dekomposisi citra menggunakan koefisien yang dirata-rata, selisih dari koefisien tersebut dicatat. Semakin kecil selisih dari koefisien maka variasi di dalam citra tersebut sedikit, dan ini merupakan kandidat yang bagus untuk proses penyederhanaan. Semakin besar selisih koefisien maka ini menandakan detil dari citra tersebut sangat signifikan dan perlu untuk dipertahankan, biasanya yang memilki detil ini adalah garis atau tepi dari citra.

Contoh dari proses dekomposisi dan rekonstruksi citra adalah, misalkan ada sebuah citra satu dimensi yang memiliki empat nilai saja (empat piksel dalam sebuah baris, memiliki tingkat abu-abu yang berbeda), yaitu



1, 3, 8, 6

Selanjutnya diambil rata-rata dari pasangan pertama dan kedua hingga menghasilkan tingkat abu-abu sbb

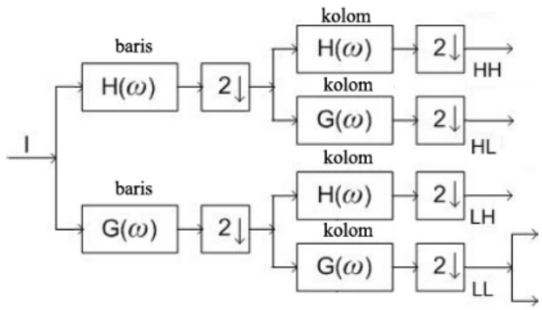
2, 7

Setelah citra telah disederhanakan, perlu untuk mencatat informasi dari citra ini yaitu berupa selisih dari koefisien rata-rata. Selisih ini perlu dicatat karena setelah citra disederhanakan maka resolusinya berkurang menjadi setengah dan ada informasi yang hilang. Padahal informasi ini dibutuhkan utnuk merekontruksi citra tersebut. Selisih dari koefisien rata-rata ini disebut dengan koefisien detil, dalam kasus ini koefisien detilnya adalah 1 dan -1. Dengan bukti sebagai berikut:

$2 + -1 = 1$
$2 – (-1) = 3$
$7 + 1 = 8$
$7 – 1 = 6$

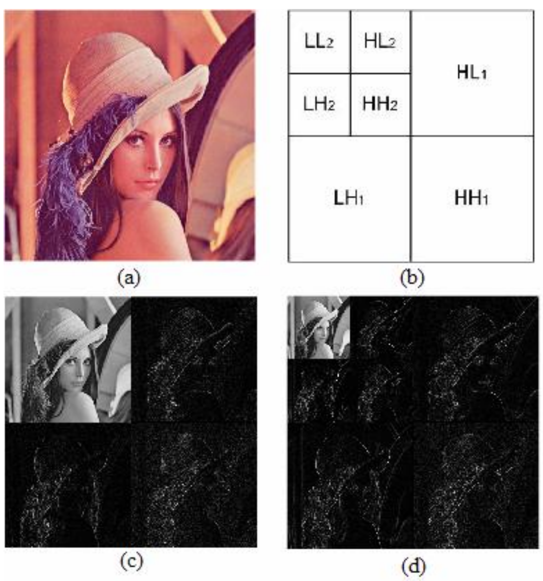
Jika dekomposisi dilanjutkan ke tingkat selanjutnya maka akan diperoleh koefisien rata-rata 4.5 dan koefisien detil 2.5. Dengan mencatat koefisien detil pada masing-masing tingkatan dkomposisi ini, maka rekonstruksi citra unruk menjadi citra asal dapat dilakukan [TEC07].

Secara teknis, dekomposisi citra yang merupakan sinyal bergerak dapat digambarkan seperti dekomposisi sinyal menggunakan transformasi *wavelet*. Citra dengan dua dimensi (baris dan kolom) dapat didekomposisi seperti Gambar 7. Dengan I adalah citra, $H(\omega)$ adalah *highpass filter*, dan $G(\omega)$ adalah *lowpass filter*.



Gambar 7. Dekomposisi *wavelet* satu tingkat terhadap citra [TER06]

Dekomposisi pada citra seperti pada Gambar 7 menghasilkan informasi rentang frekuensi yang berbeda yaitu LL, frekuensi rendah-rendah (*low-low frequency*), LH, frekuensi rendah-tinggi (*low-high frequency*), HL, frekuensi tinggi-rendah (*high-low frquency*), dan HH, frekuensi tingi-tinggi (*high-high frequency*). Kesemuanya ini membentuk struktur piramid (Gambar 8) dari sebuah citra [TER06]. Rentang frekuensi LL merupakan rentang taksiran penskalaan, sedangkan rentang frekuensi LH, HL, dan HH merupakan rentang frekuensi detil informasi [TER06].



Gambar 8. (a) Citra Lena asli, (b) Struktur piramid dua tingkat, (c) Dekomposisi Lena menggunakan Daubechies *Wavelet* satu tingkat, (d) Dekomposisi Lena menggunakan Haar *Wavelet* dua tingkat [TER06].

4. *Watermarking*

Watermark merupakan sebuah informasi yang disisipkan pada media lain dengan tujuan melindungi media yang disisipi oleh informasi tersebut dari pembajakan, penyalahgunaan hak cipta, dsb. *Watermarking* sendiri adalah cara untuk menyisipkan *watermark* ke dalam media yang ingin dilindungi hak ciptanya.

Watermarking berkembang seiring perkembangan zaman dengan munculnya *watermarking* pada media *digital* atau disebut dengan *digital watermarking*. *Digital watermarking* dapat dijalankan pada berbagai media *digital* seperti citra *digital*, *file* suara, dan video.

Salah satu prinsip dalam *digital watermarking* adalah informasi yang disisipkan pada media *digital* tidak boleh mempengaruhi kualitas media *digital* tersebut. Jadi pada citra *digital*, mata manusia tidak bisa membedakan apakah citra tersebut disisipi *watermark* atau tidak. Demikian pula jika diterapkan pada *file* suara atau musik, telinga manusia tidak bisa mendengar sisipan informasi tadi. Sehingga pada *digital watermarking* terdapat persyaratan bahwa *digital watermark* atau informasi *digital* yang disisipkan dalam dalam media *digital* haruslah *imperceptible* atau tidak terdeteksi oleh sistem penglihatan manusia (*Human Visual System*) atau sistem pendengaran manusia (*Human Auditory System*) [AGU01]. *Digital watermark* sendiri adalah sebuah kode identifikasi yang secara permanen disisipkan ke dalam data *digital* dengan membawa informasi yang berhubungan dengan perlindungan hak cipta dan otentikasi data [LUM01].

Digital watermarking memanfaatkan kelemahan sistem penglihatan dan sistem pendengaran manusia untuk dapat menyisipkan *digital watermark* atau dapat disebut dengan *watermark* saja. Jadi, *digital watermarking* dapat diartikan sebagai suatu cara untuk penyembunyian atau penanaman data/informasi tertentu, baik hanya berupa catatan umum maupun rahasia, ke dalam suatu data *digital* lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia, indera penglihatan atau indera pendengaran, dan mampu menghadapi proses-proses pengolahan sinyal *digital* sampai pada tahap tertentu [SUP00].

4.1. Sejarah *Watermarking*

Sejarah *watermarking* sudah dimulai sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi *watermark* atau tanda-air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan, terbentuklah suatu kertas yang ber-*watermark*. Kertas ini biasanya digunakan oleh seniman dan sastrawan untuk menulis karya mereka. Kertas yang sudah dibubuhi *watermark* tersebut sekaligus dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka [RIN06].

Perkembangan *watermarking* selanjutnya adalah *watermarking* pada media *digital*. *Watermarking* pada media *digital* ini mulai dikembangkan pada tahun 1990 di Jepang dan tahun 1993di Swiss [RIN06].

4.2. Jenis *Digital Watermarking*

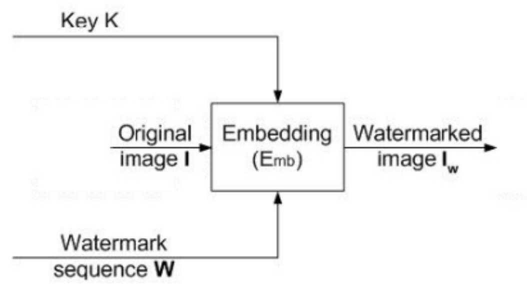
Watermarking pada data *digital* dilakukan untuk dapat melindungi kepemilkian data *digital* yang dapat dengan mudah disebarkan melalui internet atau media *digital* lain seperti *magnetic disk*. Perlindungan kepemilikan ini dapat dilakukan dengan cara penyisipan dan penyembunyian informasi pada data *digital* sehingga kepemilkan data *digital* dapat dibuktikan.

Secara umum *digital watermarking* adalah proses untuk menyisipkan data yang disebut dengan *watermark* ke dalam objek multimedia dengan sebuah cara sehingga *watermark* nantinya dapat dideteksi atau diekstraksi dengan tujuan penegasan kepemilkan [TER06]. *Digital watermarking* ini dibagi menjadi empat jenis berdasarkan media *digital* yang disisipi, yaitu:

1. *Text Watermarking*
   Watermark disisipkan pada media *digital* jenis dokumen atau teks.
2. *Image Watermarking*
   Watermark disisipkan pada citra *digital*.
3. *Audio Watermarking*
   Watermark disisipkan pada *file* audio *digital* seperti mp3, mpeg, dsb.
4. *Video Watermarking*
   Watermark disisipkan pada gambar bergerak atau disebut dengan video *digital*.

4.3. *Digital Image Watermarking*

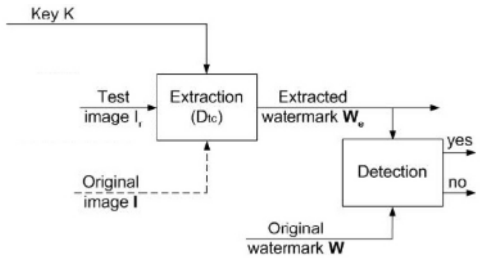
Kebutuhan terhadap perlindungan kepemilikan citra *digital* mendorong untuk dikembangkannya teknik penyembunyian data pada citra *digital*. Data yang disembunyikan atau disisipkan pada citra *digital* dapat berupa teks, citra, atau suara. Secara umum proses *watermarking* pada citra *digital* dipaparkan pada Gambar 9 dimana citra *digital* disisipi dengan *watermark* menggunakan kunci sebagai sarana kepemilikan untuk dapat membuka *watermark* yang disisipkan melalui *encoder* yang berisi algoritma penyisipan *watermark* ke dalam citra *digital*.



Gambar 9. Penyisipan *watermark* [KUT99]

Citra ber-*watermark* yang dihasilkan dari proses *watermarking* tidak berbeda jauh secara visual dengan citra *digital* asalnya. Hal ini disebabkan karena pengubahan dari citra *digital* asal ke citra ber-*watermark* hanya berpengaruh sedikit terhadap perubahan warna dari citra *digital*, sehingga sistem penglihatan manusia (*Human Visual System*) tidak dapat mempersepsi perubahan tersebut.

Proses *watermarking* perlu didukung dengan proses ekstrasi *watermark* dari citra ber-*watermark*. Proses ekstraksi ini bertujuan untk mendapatkan kembali citra *digital* asal dan *watermark* yang disisipkan dalam citra *digital* tersebut. Umumnya proses ekstraksi melibatkan proses pembandingan citra *digital* asal dengan citra ber-*watermark* untuk mendapatkan *watermark* yang disisipkan, seperti yang digambarkan pada Gambar 10.



Gambar 10. Ekstraksi *watermark* [KUT99]

Teknik di dalam *Digital Image Watermarking* terbagi menjadi dua *domain* yaitu, *domain* spasial, penyisipan *watermark* dilakukan secara langsung ke dalam *pixel* citra, dan *domain transform* yang menyisipkan *watermark* ke dalam koefisien transformasi [RIN06].

Digital Image Watermarking sendiri memliki beberapa jenis teknik yang memiliki keunggulan dan kelemahan masing-masing. Biasanya teknik *watermarking* yang kuat (susah dipecahkan oleh berbagai serangan) memiliki kualitas gambar ber-*watermark* yang kurang memuaskan, demikian juga sebaliknya, teknik *watermarking* yang menghasilkan kualitas gambar yang memuaskan biasanya kurang kuat menghadapi serangan [KUT99]. Secara garis besar teknik *watermarking* dibedakan menjadi dua yaitu [GIL00]:

1. *Private Watermarking* / *Incomplete Watermarking* / *Escrow Watermarking*
   Merupakan teknik *watermarking* yang membutuhkan citra asli dan citra ber-*watermark* untuk megekstraksi *watermark*.
2. *Public Watermarking* / *Complete Watermarking* / *Oblivious Watermarking* / *Blind Watermarking*
   Teknik *watermarking* yang tidak membutuhkan citra asli atau *watermark* yang disisipkan untuk melakukan ekstraksi.

4.4. Teknik dalam *Digital Image Watermarking*

Seperti yang telah dibahas pada subbab sebelumnya tentang *domain* dalam teknik *Digital Image Watermarking* yaitu *domain* spasial dan *domain transform*. Penyisipan *watermark* dalam *domain* spasial dilakukan secara langsung pada piksel-piksel penyusun sebuah citra *digital*. Contoh metode yang termasuk dalam teknik dengan *domain* spasial adalah *LSB* (*Least Significant Bit*) yang me-*watermark* sebuah citra *digital* dengan mengganti *bit LSB*-nya dengan *bit* data, metode lain dalam *domain* spasial yaitu metode *patchwork* yang menanamkan *watermark* sebesar 1 *bit* pada citra *digital* dengan menggunakan pendekatan statistik [SUP00].

Untuk metode yang digunakan pada teknik dalam *domain transform*, biasanya berhubungan dengan transformasi sinyal yang digunakan dalam bidang matematika. *Watermark* disisipkan ke dalam koefisien transformasi tergantung dari jenis transformasi yang digunakan. Beberapa jenis transformasi yang sering digunakan yaitu *Discrete Fourier Transform* (*DFT*), *Discrete Cosine Transform* (*DCT*), *Discrete Wavelet Transform* (*DWT*), dan *Discrete Laguerre Transform* (*DLT*). Inti *watermarking* dalam *domain transform* adalah sebuah transformasi balikan (*inverese transform*) harus dijalankan untuk mendapatkan citra ber-*watermark* [TER06].

Spread spectrum merupakan salah satu contoh metode dalam *domain* transform. Metode ini memanfaatkan transformasi sinyal dengan cara mentransformasikan citra *digital* ke dalam *domain* frekuensi, kemudian *bit watermark* disisipkan pada koefisien transformasi. Penyisipan *watermark* ini dilakukan dengan cara menyebarkan *watermark* diantara banyak komponen frekuensi [RIN06].

4.5. Serangan terhadap Citra Ber-*watermark*

Serangan terhadap citra ber-*watermark* umumnya bertujuan untuk menghilangkan *watermark* yang disisipkan di dalam citra *digital* tersebut. Serangan ini disebut sebagai serangan yang disengaja. Serangan yang tidak disengaja biasanya berhubungan dengan pengubahan citra *digital*, pengubahan ini dapat berupa *cropping*, *rotation*, kompresi, dll.

Secara umum jenis serangan terhadap citra ber-*watermark* dibagi menjadi dua, yaitu serangan standar (*standard attack*) dan *malicious attack*. *Malicious attack* merupakan serangan yang memilki tujuan untuk menghilangkan *watermark* [VAN02]. Pengujian terhadap citra ber-*watermark* menggunakan serangan hanya dapat dilakukan dengan menggunakan *standard attack* saja. Hal ini disebabkan karena dalam *malicious attack* umumnya pihak penyerang mencari algoritma penyisipan dan kunci yang digunakan saat penyisipan *watermark*. Serangan jenis *malicious attack* ini tetntunya tidak dapat diujikan karena algoritma dan kunci yang digunakan tentunya sudah diketahui oleh penyisip *watermark*.

4.5.1. Serangan Standar (*Standard Attack*)

Serangan standar biasanya merupakan serangan yang tidak disengaja untuk merusak atau mendapatkan *watermark* di dalam citra ber-*watermark.* Contoh dari jenis serangan standar (*standard attack*) adalah sbb:

1. *Cropping*
   Cropping merupakan serangan yang umum karena banyak orang sering menginginkan bagian tertentu dari sebuah citra saja. Untuk dapat mengatasi serangan ini dapat dilakukan dengan cara menyebarkan *watermark* pada tempat-tempat yang memungkinkan terjadinya serangan [TER06].
2. Serangan geometris (*geometrical attack*)
   Serangan geometris sering tidak secara sengaja bertujuan untuk menghilangkan *watermark* pada citra yang sudah ber-*watermark*. Serangan geometris ini menyebabkan pendeteksi *watermark* kehilangan sinkronisasinya dengan citra ber-*watermark*. Beberapa yang termasuk dalam serangan geometris adalah rotasi citra, penskalaan ulang citra, pengubahan *aspect ratio*, translasi, dsb [TER06].
3. Kompresi
   Serangan ini juga merupakan serangan yang sering dilakukan secara tidak sengaja. Kompresi sering dilakukan pada *file* multimedia seperti audio, video, dan citra. *Watermark* yang disisipkan biasanya lebih tahan terhadap kompresi yang memliki *domain* sama dengan *domain* yang dipakai pada saat *watermarking*. Misalnya citra yang disisipi *watermark* menggunakan *DCT* (*Discrete Cosine Transform*) lebih tahan terhadap kompresi JPEG daripada citra yang disisipi *watermark* dalam *domain* spasial [TER06]. Atau citra yang disispi *watermark* menggunakan *DWT (Discrete Wavelet Transform)* lebih kuat terhadap kompresi JPEG2000.
4. Penambahan *noise*
   Citra *digital* sangat rentan mendapatkan serangan berbagai macam jenis *noise*. Ada beberapa cara yang menyebabkan suatu *noise* dapat berada didalam sebuah citra, bergantung bagaimana citra tersebut diciptakan. Sebagai contoh, jika citra merupakan hasil *scan* foto yang berasal dari negatif film, maka negatif film ini merupakan sumber *noise*. *Noise* juga bisa merupakan akibat dari kerusakan film atau juga bisa berasal dari *scanner* itu sendiri. Jika citra diperoleh secara langsung dalam format *digital*nya, mekanisme dalam mendapatkan data *digital* tersebut juga dapat menyebabkan adanya *noise*. Penyebaran data citra secara elektronik bisa juga menghasilkan *noise*.
5. Filterisasi
   Filterisasi umum digunakan pada citra. Beberapa filter yang sering digunakan yaitu *gaussian filter*, *sharpening filter*, dsb. Untuk mengani jenis serangan ini *watermark* dapat disisipkan pada frekuensi yang paling sedikit berubah jika terjadi kompresi, dengan memperkirakan filterisasi apa saja yang umum digunakan [TER06].

4.5.2. *Malicious Attack*

Untuk jenis serangan kedua, yaitu *malicious attack* masih dibagi lagi menjadi tiga jenis serangan yaitu penghilangan *watermark* (*watermark removal*), deteksi atau perkiraan *watermark* (*watermark detection or estimation*), dan penulisan *watermark* (*watermark writing*) [VAN02]. Penjelasan beserta contoh masing-masing jenis serangan dipaparkan sbb:

1. Penghilangan *watermark*
   Dalam penghilangan *watermark* seorang penyerang tidak perlu berhubungan langsung dengan semantik dari *watermark* yang disisipkan. Artinya seorang penyerang tidak perlu mengambil *watermark* yang disisipkan tapi hanya perlu menghilangkan pesan yang dimaksud di dalam *watermark* tersebut dengan cara memodifikasi sinyal *watermark* sehingga pendeteksi tidak berhasil mendeteksi adanya *watermark* yang disisipkan. Contoh jenis serangan ini adalah serangan kolusi (*collusion attack*). Serangan kolusi ini biasanya terjadi pada citra ber-*watermark* yang memiliki banyak salinan dengan *watermark* berbeda. Serangan kolusi dijalankan dengan cara meratakan setiap salinan dan menurunkan energi *watermark* dibandingkan dengan citra asalnya [VAN02].
2. Deteksi atau perkiraan *watermark*
   Serangan ini menitik beratkan pada pencarian modifikasi yang telah dilakukan terhadap citra asal sehingga dapat merepresentasikan *watermark* yang

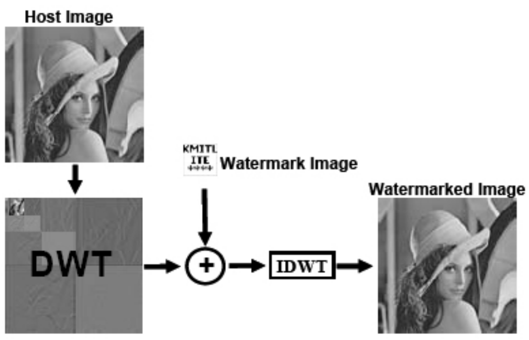
disisipkan. Serangan biasanya dilakukan dengan memperkirakan citra asal dan mengambil perbedaan antara citra asal hasil perkiraan tersebut dengan *watermark*. Sebenarnya serangan ini lebih tepat dikatakan sebagai perantara untuk melakukan serangan sesungguhnya terhadap citra ber-*watermark* [VAN02].

3. Penulisan *watermark*
   Cara yang digunakan dalam jenis serangan ini biasanya adalah redundansi *watermarking* atau memberi *watermark* kembali pada citra yang sudah ber-*watermark*.

4.6. *Watermarking* menggunakan *DWT*

Discrete Wavelet Transform (*DWT*) merupakan salah satu kakas yang banyak digunakan dalam teknik *watermarking* dengan *domain* transform. *Watermarking* yang berbasis *wavelet* adalah pendekatan yang populer karena kekuatannya melawan *malicious attack* [KEJ04].

Citra *digital* sebelumnya didekomposisi menggunakan *DWT* untuk dapat menyisipkan *watermark*, selanjutnya dijalankan *IDWT* untuk membentuk citra ber-*watermark*. Inilah proses umum *watermarking* menggunakan *Discrete Wavelet Transform* (*DWT*). Proses ini dipaparkan pada Gambar 11.



Gambar 11. Penyisipan *watermark* menggunakan *Discrete Wavelet Transform* (*DWT*) [KEJ04]

Secara umum penyisipan *watermark* dilakukan dengan cara memodifikasi koefisien pada rentang frekuensi LL, LH, HL, atau HH yang merupakan rentang frekuensi hasil dekomposisi citra menggunakan *wavelet* (tinjau kembali Gambar 9 dan Gambar 10). Data *watermark* ini dapat dianggap sebagai rangkaian bilangan *w* dengan panjang *L*, yang disisipkan pada koefisien rentang frekuensi yang dipilih *f*. Algoritma umum penyisipan *watermark* pada koefisien rentang frekuensi adalah:

$$f' = f + \alpha . w(k) + K(k), k = 1,..., L \qquad \textbf{(5)}$$

Dimana α merupakan kekuatan penyisipan yang mengontrol tingkat kekuatan penyisipan *watermark* dan f' adalah koefisien sinyal asal yang telah dimodifikasi. Penyisipan *watermark* pada citra *digital* menggunakan *DWT* ini dilakukan pada koefisien rentang frekuensi (koefisien *DWT*) sebelum direkonstruksi menggunakan *IDWT* untuk menjadi citra ber-*watermark*. Kunci *K*, berguna dalam proses penyisipan *watermark* sebagai informasi tambahan untuk citra yang disisipi *watermark*. Pemasukan kunci yang salah saat menjalankan proses ekstraksi *watermark* menyebabkan *watermark* hasil ekstraksi tidak sesuai dengan *watermark* yang disisipkan saat proses penyisipan *watermark*.

5. Kualitas Citra

Penghitungan kualitas citra dapat dilakukan dengan dua cara, yaitu menghitung *peak signal-to-noise ratio* (PSNR) sebagai pembanding kualitas citra hasil rekonstruksi dengan citra asal. Cara yang kedua adalah menghitung galat/*error* citra *watermark* yang dihasilkan dari proses ekstraksi citra.

5.1. *Peak Signal-to-Noise Ratio* (*PSNR*)

Istilah *peak signal-to-noise ratio* (PSNR) adalah sebuah istilah dalam bidang teknik yang menyatakan perbandingan antara kekuatan sinyal maksimum yang mungkin dari suatu sinyal *digital* dengan kekuatan derau yang mempengaruhi kebenaran sinyal tersebut. Oleh karena banyak sinyal memiliki *dynamic range* yang luas, maka PSNR biasanya diekspresikan dalam skala *logarithmic decibel.*

PSNR didefinisikan melalui *signal-to-noise ratio* (SNR). SNR digunakan untuk mengukur tingkat kualitas sinyal. Nilai ini dihitung berdasarkan perbandingan antara sinyal dengan nilai derau. Kualitas sinyal berbanding lurus dengan dengan nilai SNR. Semakin besar nilai SNR semakin baik kualitas sinyal yang dihasilkan. Nilai PSNR biasanya berkisar antara 20 dan 40. PSNR ini dilaporkan dengan ketepatan/presisi sebanyak dua desimal poin[HEN03].

Pertama yang dilakukan adalah menghitung nilai *mean squared error* (*MSE*) dari suatu citra hasil rekonstruksi. MSE dihitung untuk seluruh pixel dalam citra. *Root mean squared error* (*RMSE*) adalah akar dari *MSE*.

$$MSE = \frac{\sum [f(i,j) - F(i,j)]^2}{N^2} \quad \textbf{(6)}$$

N^2 menyatakan hasil perkalian panjang dan lebar citra dalam *pixel*. *F(i,j)* merupakan citra hasil rekonstruksi, sedangkan *f(i,j)* adalah citra asal.

Berdasarkan persamaan *MSE* tersebut, maka nilai *PSNR* dapat dihitung dengan persamaan 7. Nilai *PSNR* direpresentasikan dalam skala desibel (dB).

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right) \quad \textbf{(7)}$$

Nilai 255 dalam rumus 7 merupakan batas atas dari sebuah nilai pixel.

5.2. Penghitungan Galat/*Error*

Persentasi galat/*error* digunakan untuk menginformasikan jumlah bit galat dari seluruh bit *watermark*. Penghitungan galat ini merupakan penghitungan persentase jumlah kesalahan pada citra *watermark* hasil ekstraksi dibandingkan dengan citra *watermark* asal yang disisipkan.

$$G = \frac{n_{error}}{N} \times 100\% \quad \textbf{(8)}$$

dengan,

G = persentase galat

n_{error} = jumlah *bit watermark* yang berbeda dengan *bit watermark* asal

N = panjang *bit watermark* yang disisipkan

6. *Human Visual System* (*HVS*)

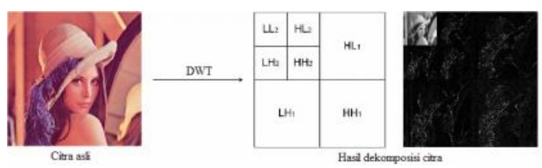
Sistem penglihatan manusia memilki tingkat sensitivitas yang berbeda terhadap warna dan tingkat kecerahan. Secara umum mata manusia lebih peka terhadap perubahan tingkat kecerahan daripada perubahan warna. Sistem penglihatan manusia ini dimulai dari mata yang digunakan untuk menangkap cahaya dan warna dari suatu benda, lalu memproyeksikannya di dalam otak.

Sistem penglihatan manusia juga memiliki sensitivitas terhadap cahaya dan warna tertentu. Kesensitivan ini berguna untuk menentukan bagian yang paling tepat di dalam sebuah citra untuk menyisipkan *watermark*.

7. Dekomposisi Citra *Digital* yang Akan Disisipi *Watermark*

Dekomposisi citra *digital* yang akan disisipi *watermark* atau citra *host* merupakan langkah pertama yang harus dilakukan untuk dapat menyisipkan *watermark* ke dalamnya. Dekomposisi citra *digital* ini dilakukan dengan menggunakan *Discrete Wavelet Transform* (*DWT*), tepatnya menggunakan Haar *wavelet*. Adapun proses yang perlu dijalankan untuk mendekomposisi citra adalah sbb:

1. Dekomposisi citra berdasarkan *Discrete Wavelet Transform* (*DWT*) sehingga menghasilkan rentang frekuensi LL, LH, HL, dan HH.
2. Dekomposisi citra *digital* satu tingkat sehingga menghasilkan LL_1, LH_1, HL_1, dan HH_1.
3. Dekomposisi terhadap citra *digital* dilakukan dua tingkat sehingga LL_1 menjadi LL_2, LH_2, HL_2, dan HH_2 (Gambar 12).



Gambar 12. Dekomposisi citra *host* [TER06]

8. Proses Penyisipan Citra *Watermark*

Penyisipan citra *watermark* diawali dengan mengubah susunan citra *watermark* ke dalam rangkaian matriks. Selanjutnya penyisipan *watermark* dilakukan dengan langkah-langkah sbb:

1. Setelah citra *host* didekomposisi dalam *l* tingkatan *DWT*, *watermark* disisipkan ke

dalam rentang frekuensi LH_l atau HL_l, ($l \in \{2,3,4\}$).

2. Mencari koefisien terbesar f_{LH} dari dari rentang frekuensi LH atau koefisien terbesar f_{HL} dari rentang frekuensi HL.
3. Menyisipkan *w* ke dalam rentang frekuensi LH atau HL dengan persamaan:

$$f'_{LH}(m,n) = f_{LH}(m,n) + \alpha w(m,n) + K(m,n), m,n = 1..L \quad \textbf{(9)}$$

$$f'_{HL}(m,n) = f_{HL}(m,n) + \alpha w(m,n) + K(m,n), m,n = 1..L \quad \textbf{(10)}$$

Dimana $f_{LH}(m,n)$ merupakan koefisien terbesar yang dipilih dan $f'_{LH}(m,n)$ merupakan koefisien yang dimodifikasi pada posisi (m,n) untuk rentang frekuensi LH. $f_{HL}(m,n)$ merupakan koefisien terbesar yang dipilih dan f_{HL} $f'_{HL}(m,n)$ merupakan koefisien yang dimodifikasi pada posisi (m,n) untuk rentang frekuensi HL. α, seperti yang telah dijelaskan di dalam bab Dasar Teori, merupakan kekuatan penyisipan *watermark* atau dapat dikatakan sebagai faktor skala persentase dari citra *host* dan citra *watermark* pada citra ber-*watermark* yang dibentuk. Bentuk kunci *K* ini adalah sebuah *array* yang memiliki panjang tertentu dengan panjang maksimal adalah sepanjang *watermark*. Kunci *K* ditambahkan dalam proses penyisipan *watermark* mealalui proses penjumlahan terhadap nilai *watermark* pada indeks yang bersesuaian.

4. Menjalankan *Inverse Discrete Wavelet Transform* (*IDWT*) untuk membentuk citra ber-*watermark*.

9. Pendeteksian dan Ekstraksi *Watermark*

Ekstraksi *watermark* dilakukan tanpa menggunakan citra asal atau citra *host*. Pendeteksian ada tidaknya *watermark* dalam citra dilakukan dengan menggunakan pembandingan koefisien yang bersesuaian pada citra ber-*watermark* dengan nilai ambang. Jika koefisien dari rentang frekuensi yang berkorelasi lebih besar daripada nilai ambang maka *watermark* terdeteksi di dalam citra. Langkah-langkah ekstraksi *watermark* adalah sbb:

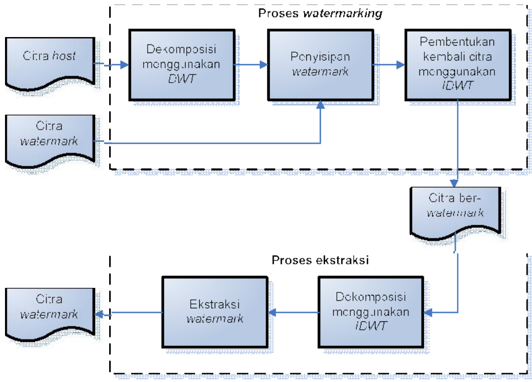
1. Citra ber-*watermark* didekomposisi dalam dua tingkatan *DWT*.
2. Memilih koefisien citra ber-*watermark* dari rentang frekuensi LH dan HL yaitu f_{LH} dan f_{HL}.
3. Mecari koefisien citra *host* dari rentang frekuensi LH dan HL yaitu f_{LH} dan f_{HL}.
4. Melakukan perbandingan koefisien citra ber-*wateremark* dengan koefisien citra *host* untuk menghasilkan *watermark*.
5. Menjalankan *Inverse Discrete Wavelet Transform* (*IDWT*) untuk membentuk citra *watermark*.
6. Secara umum proses ekstraksi *watermark* ini merupakan kebalikan dari proses penyisipan *watermark* yang telah dijelaskan sebelumnya.

10. Deskripsi Umum Sistem untuk *Watermarking*

Sistem dirancang untuk dapat melakukan proses penyisipan *watermark* ke dalam sebuah citra *digital*. *Watermark* yang disisipkan berupa citra hitam putih yang dipilih sebagai masukan bagi sistem.

Sistem ini dibangun pada satu lingkungan pengembangan saja yaitu pada *PC* (*Personal Computer*). Sehingga proses *watermarking* dan ekstraksi dilakukan pada lingkungan pengembangan yang sama. Untuk itu user diharuskan memilih proses yang diinginkan pada awal berjalannya sistem. Proses *watermarking* berguna untuk melakukan *watermarking* pada citra *host*, sedangkan proses ekstraksi berguna untuk melakukan ekstraksi *watermark* dari citra *digital* yang sudah ber-*watermark*.

Ekstraksi *watermark* dirancang untuk dapat melakukan pembandingan antara citra *host* dengan citra ber-*watermark* untuk mendapatkan *watermark*. Citra ber-*watermark* dimasukkan ke dalam sistem sehingga dapat dilakukan proses ekstraksi *watermark*. Secara umum arsitektur sistem yang dibangun pada kedua lingkungan dapat dilihat pada Gambar 13.



Gambar 13. Arsitektur Sistem

11. Implementasi Sistem

Perangkat lunak yang dibuat diberi nama Cammar. Nama Cammar diambil dari *Camera Mark* yang berarti perlindungan citra *digital* yang berasal dari pengambilan kamera.

Lingkungan yang digunakan untuk membangun perangkat lunak Cammar adalah lingkungan *desktop* berbasis Java, dan sistem operasi yang digunakan adalah Windows XP Home Edition Service Pack 2.

Perangkat keras yang digunakan dalam pengembangan perangkat lunak Cammar adalah seperangkat komputer dengan spesifikasi sebagai berikut :

1. Monitor: 15 inch
2. CPU: Intel Pentium M 1.73 GHz
3. Hard Disk: 80GB
4. Memori: 512 MB DDRAM
5. VGA Card: On board
6. Perangkat Masukan: Tetikus, Papan Kunci

Bahasa pemrograman yang digunakan untuk membangun Cammar adalah Java versi 1.6. Sedangkan kompilator sekaligus IDE yang digunakan untuk memudahkan pengembangan perangkat lunak adalah NetBeans 6.0.

12. Hasil Pengujian

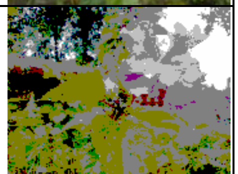
Citra yang digunakan untuk pnyisipan *watermark* ditampilkan pada Tabel 1, sedangkan citra untuk *watermark* ditampilkan pada Tabel 2.

Hasil pengujian kinerja perangkat lunak, dengan menggunakan variasi ukuran citra dan variasi format citra, dapat menghasilkan citra ber-*watermark* sebagai hasil proses penyisipan *watermark* yang tidak berebeda jauh dengan citra asalnya. Hal ini dibuktikan dengan skala nilai PSNR sekitar 50 yang merupakan nilai PSNR besar. Hal ini menandakan bahwa citra yang dihasilkan dari proses penyisipan *watermark* tidak berbeda jauh dengan citra asalnya.

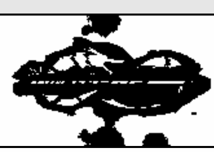
Hasil dari pengujian ketahanan perangkat lunak berjalan baik untuk pengujian kesalahan memasukkan nilai kunci. Hal ini ditandai dengan hasil ekstraksi yang tidak sesuai dengan *watermark* asalnya. Hasil pengujian terhadap serangan berupa *blur* dan *noise* hanya berajalan baik untuk skala kecil. Hasil ekstraksi dari dua serangan ini memang tidak sempurna menghasilkan citra *watermark* seperti citra *watermark* asal, namun citra ini masih dapat dikenali. Untuk pengubahan format citra, hanya berjalan baik untuk pengubahan ke format BMP. Untuk pengubahan ke format JPG yang melalui kompresi JPEG, hasilnya masih dapat dikenali, namun tidak sempurna. Hasil pengujian ketahanan perangkat lunak ini dipaparkan pada Tabel 7.

Tabel 1. Tabel *File* Citra Asal

No	Nama *file*	Ukuran Citra	Citra
1.	Bunga800.jpg	800 x 600 pixel	
2.	Bunga1024.jpg	1024 x 768 pixel	
3.	Bunga1280.jpg	1280 x 960 pixel	
4.	Bunga2048.jpg	2048 x 1536 pixel	

5.	MawarJPG.jpg	1024 x 768 pixel	
6.	MawarBMP16.bmp	1024 x 768 pixel (16 bit)	
7.	MawarBMP24.bmp	1024 x 768 pixel (24 bit)	
8.	MawarBMP32.bmp	1024 x 768 pixel (32 bit)	
9.	MawarPNG.png	1024 x 768 pixel	
10.	KeretaBMP.bmp	1024 x 768 pixel	
11.	KeretaJPG.jpg	1024 x 768 pixel	
12.	KeretaPNG.png	1024 x 768 pixel	

Tabel C- 1. Tabel Citra *Watermark*

No	Nama *File*	Ukuran Citra	Citra
1.	*Watermark*01.bmp	340 x 200 pixel	
2.	*Watermark*02.bmp	340 x 200 pixel	

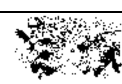
Tabel 3. Tabel Hasil Pengujian Kinerja Perangkat Lunak (Variasi Ukuran Citra Asal)

No	Citra asal	Citra *watermark*	Hasil penyisipan	Hasil ekstraksi
1.	Bunga800.jpg	*Watermark*01.bmp		
2.	Bunga1024.jpg	*Watermark*01.bmp		
3.	Bunga1280.jpg	*Watermark*01.bmp		
4.	Bunga2048.jpg	*Watermark*01.bmp	*Java Message : Out of Memory Error*	Tidak Ada

Tabel 4. Tabel Nilai PSNR Hasil Pengujian Kinerja Perangkat Lunak (Variasi Ukuran Citra Asal)

No	Citra asal	Citra *watermark*	Nilai PSNR
1.	Bunga800.jpg	*Watermark*01.bmp	48.12
2.	Bunga1024.jpg	*Watermark*01.bmp	50.27
3.	Bunga1280.jpg	*Watermark*01.bmp	52.21
4.	Bunga2048.jpg	*Watermark*01.bmp	Tidak Ada

Tabel 5. Tabel Hasil Pengujian Kinerja Perangkat Lunak (Variasi Format Citra Asal)

No	Citra asal	Citra *watermark*	Hasil penyisipan	Hasil ekstraksi
1.	MawarJPG.jpg	*Watermark*02.bmp		
2.	MawarBMP16.bmp	*Watermark*02.bmp		
3.	MawarBMP24.bm	*Watermark*02.b		

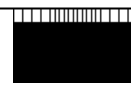
	p	mp		
4.	MawarBMP32.bmp	*Watermark*02.bmp		
5.	MawarPNG.png	*Watermark*02.bmp		
6.	KeretaBMP.bmp	*Watermark*02.bmp		
7.	KeretaJPG.jpg	*Watermark*02.bmp		
8.	KeretaPNG.png	*Watermark*02.bmp		

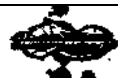
Tabel 6. Tabel Nilai PSNR Hasil Pengujian Kinerja Perangkat Lunak (Variasi Format Citra Asal)

No	Citra asal	Citra *watermark*	Nilai PSNR
1.	MawarJPG.jpg	*Watermark*02.bmp	50.24
2.	MawarBMP16.bmp	*Watermark*02.bmp	-75.43
3.	MawarBMP24.bmp	*Watermark*02.bmp	50.24
4.	MawarBMP32.bmp	*Watermark*02.bmp	50.24
5.	MawarPNG.png	*Watermark*02.bmp	50.24
6.	KeretaBMP.bmp	*Watermark*02.bmp	50.24
7.	KeretaJPG.jpg	*Watermark*02.bmp	50.24
8.	KeretaPNG.png	*Watermark*02.bmp	50.24

Pada pengujian ketahanan perangkat lunak, citra asal yang dipilih adalah Bunga1024.jpg, sedangkan citra *watermark*-nya adalah *Watermark*01.bmp.

Tabel 7. Tabel Hasil Pengujian Ketahanan Perangkat Lunak

No	Jenis serangan	Keterangan	Hasil serangan	Hasil ekstraksi
1.	Kesalahan memasukkan kunci untuk ekstraksi	Kunci yang digunakan 1		
2.	*Blur* citra *digital*	*Blur*		
3.	*Blur* citra *digital*	*Motion Blur* 2 pixel		
4.	*Blur* citra *digital*	*Motion Blur* 25 pixel		
5.	Pemberian derau (*noise*)	*Noise* 2%		
6.	Pemberian derau (*noise*)	*Noise* 10%		
7.	*Rotate* citra *digital*	Rotate 10^o		
8.	*Rotate* citra *digital*	Rotate 90^o		
10.	*Sharpen* citra *digital*			

11.	*Cropping* citra *digital*			
12.	Pengubahan format citra *digital*	Pengubahan dari PNG menuju BMP kembali ke PNG		
13.	Pengubahan format citra *digital*	Pengubahan dari PNG menuju JPG kembali ke PNG		
14.	Pengubahan format citra *digital*	Kompresi JPEG.		

13. Kesimpulan

Kesimpulan yang dapat diambil adalah:

1. Penyisipan citra *watermark* ke dalam citra asli menggunakan *Discrete Wavelet Transform* (*DWT*) adalah dengan menyisipkan citra *watermark* ke dalam koefisien *wavelet* dari citra asli.
2. Dekomposisi citra *digital* menggunakan *Discrete Wavelet Transform* (*DWT*) dilakukan dengan cara mengambil koefisien *wavelet* dari citra tersebut, koefisien *wavelet* juga yang digunakan untuk dapat merekonstruksi citra kembali menggunakan *Inverse Discrete Wavelet Transform* (*IDWT*).
3. Ekstrasi *watermark* yang disisipkan menggunakan *Discrete Wavelet Transform (DWT)* dilakukan dengan cara mengambil *watermark* dari koefisien *wavelet* dari citra tersebut.
4. Haar *wavelet* merupakan teknik dalam keluarga *wavelet* yang paling mudah penerapannya, namun hal ini diimbangi dengan kurang tahannya terhadap serangan dibandingkan teknik *wavelet* yang lain. Hal ini dibuktikan dengan pengujian menggunakan *standard attack* yang hanya menghasilkan citra *watermark* dengan kualitas baik untuk serangan minimum dan pengubahan format *file* citra saja.

14. Saran

Beberapa saran untuk pengembangan antara lain:

1. Kakas *Discrete Wavelet Transform* (*DWT)* yang digunakan adalah Haar *Wavelet*, untuk pengembangan selanjutnya dapat digunakan jenis *DWT* lain seperti Daubechies, Coiflet, Symlet, Meyer, Morlet, dan Mexican Hat. Teknik-teknik ini merupakan teknik yang lebih tahan terhadap serangan dibandingkan Haar *wavelet*.
2. Citra *watermark* yang digunakan hanya citra hitam putih, untuk pengembangan selanjutnya dapat menggunakan citra berwarna.
3. Pengujian yang dilakukan pada beberapa pengujian terhadap *standard attack*. Untuk pengembangan selanjutnya diharapkan dapat dilakukan pengujian terhadap semua jenis *standard attack*.

15. Daftar Pustaka

[AGI07] Agilent Technologies. *Evaluating Oscilloscope Sample Rates vs. Sampling Fidelity: How to Make the Most Accurate Digital Measurements*. http://www.agilent.com. diakses tanggal 10 Juni 2007

[AGU01] Agung, Wiseto P (2001). *Digital Watermarking : Teknologi Pelindung HAKI Multimedia*. Elektro Indonesia.www.elektroindonesia.com.

[ANS07] Answer.com (2007). *Wavelet.* www.answer.com. Diakses tanggal 11 April 2007.

[GIL00] Gilani, S. Asif Mahmood; A. N. Skodras (2000). *DLT-Based Digital Image Watermarking*. University of Patras.

[GIR00] Girod, Bernd (2000). *Human*

Visual Perception. Telecommunications Laboratory,University of Erlangen, Nuremberg, Germany.

[HAR99] Hartung, Frank; Jonathan K. Su; Bernd Girod (1999). *Spread Spectrum Watermarking: Malicious Attacks and Counterattacks*. Telecommunications Laboratory,University of Erlangen, Nuremberg, Germany.

[HEN03] Hendrawan, Shanty Meliani (2003). *Robust and Non Blind Watermarking* pada Citra Dijital dengan Teknik Spread Spectrum. Institut Teknologi Bandung.

[HON05] Hongjun, Wang; Li Na (2005). *An algorithm of digital image watermark based on multiresolution wavelet analysis*. Shangdong University.

[IMA07] Image Permanence Institute. www.imagepermanenceinstitute.org/ . diakses tanggal 11 April 2007.

[KEJ04] Kejriwal, Arun; Sumit Gupta; Alexandru Nicolau; Nikil Dutt; Rajesh Gupta (2004). *Proxybased Task Partitioning of Watermarking Algorithms for Reducing Energy Consumption in Mobile Devices*. University of California.

[KIS07] Kiselev, Andrey (2007). *Fundamentals of the Wavelets Transform Theory.* www.basegroup.ru. diakses tanggal 11 April 2007.

[KUT99] Kutter, Martin; Fabien A. P. Petitcolas (1999). *A Fair Benchmark for Image Watermarking Systems*. The International Society for Optical Engineering.

[LUM01] Lumini, Alessandra; Dario Maio (2001). *Approach to Digital Image Watermark*. *Watermark* Lab, Università di Bologna.

[OSU07] Oregon State University. *Adaptive-Quantization Digital Image Sensor for Low-Power Image*.osulibrary.oregonstate.edu. diakses tanggal 11 April 2007.

[POL98] Polikar, Robi (1998). *Multi Resolution Analysis : The Discrete Wavelet Transform*. Durham Computation Center, Iowa State University.

[RIN06] Munir, Rinaldi (2006). *Diktat Kuliah IF5054 Kriptografi*. Institut Teknologi Bandung.

[ROO02] Roorda, Austin (2002). *Human Visual System – Image Formation*. University of California.

[SAL06] Salomon, David. *Human Visual System* (2006). California State University. 2006.

[SRI03] Sripathi, Deepika (2003). *Efficient Implementations of Discrete Wavelet Transform using FPGAs*. Florida State University.

[SUP00] Supangkat, Suhono H.; Kuspriyanto; Juanda (2000). *Watermarking sebagai Teknik Penyembunyian Label Hak Cipta pada Data Digital*. Departemen Teknik Elektro, Institut Teknologi Bandung.

[TEC03] Technical Advisory Service for Images (2003). *New Digital Image File Formats*. http://www.tasi.ac.uk. diakses tanggal 11 April 2007.

[TEC05] Technical Advisory Service for Images (2005). *File Formats and Compression*. http://www.tasi.ac.uk. diakses tanggal 11 April 2007.

[TEC06] Technical Advisory Service for Images (2005). *The Digital Image*. http://www.tasi.ac.uk. diakses tanggal 11 April 2007.

[TEC07] Technical Advisory Service for Images. *What is Wavelet Compression?*. http://www.tasi.ac.uk/advice/*wavelet*.html. diakses tanggal 11 April 2007.

[TER06] Terzija, Nataša (2006). *Robust Digital Image Watermarking Algorithms for Copyright Protection*. Universität Duisburg-Essen.

[VAN02] Van der Veen, Michiel; Aweke Negash Lemma; Fons Bruekers; Javier Aprea; Ton Kalker (2002). *Security Issues in Digital Audio Watermarking*. Philip Research Laboratories/Philip *Digital* System Laboratories, Netehrland.