BAB II

LANDASAN TEORI

Pada bab ini akan dibahas beberapa konsep dasar yang berhubungan dengan algoritma *watermarking digital* pada domain DCT untuk citra berwarna, seperti definisi watermarking, karakteristik watermarking dan klasifikasi teknik *watermarking digital*. Kemudian pembahasan singkat mengenai Metode *discrete cosine transform* (DCT) pada *watermarking*.

2.1 Pengertian Watermarking

Ada beberapa definisi dari *watermarking,* antara lain *watermarking* berarti suatu konsep untuk menyisipkan suatu data atau pola ke dalam dokumen sehingga suatu potongan informasi seperti kepemilikan atau identitas konsumen yang berhak untuk menggunakannya berada dalam data tersebut. Selain itu *Watermarking* merupakan suatu bentuk dari *Steganography* (Ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain), dalam mempelajari teknik-teknik bagaimana penyimpanan suatu data *digital* ke dalam data *host digital* yang lain (Istilah *host* digunakan untuk data/sinyal digital yang ditumpangi.). namun ada perbedaan antara *watermarking* dan *steganografi*. Jika pada *steganografi* informasi rahasia disembunyikan di dalam media digital dimana media penampung tidak berarti apa-apa, maka pada *watermarking* justru media digital tersebut yang akan dilindungi kepemilikannya dengan pemberian suatu informasi tertentu didalamnya.

Watermarking (tanda air) ini agak berbeda dengan tanda air pada uang kertas. Tanda air pada uang kertas masih dapat kelihatan oleh mata telanjang manusia (mungkin dalam posisi kertas yang tertentu), tetapi watermarking pada media digital disini dimaksudkan tak akan dirasakan kehadirannya oleh manusia tanpa alat bantu mesin pengolah digital seperti komputer, dan sejenisnya. *Watermarking* ini memanfaatkan kekurangan-kekurangan sistem indera manusia seperti mata dan telinga. Dengan adanya kekurangan inilah, metoda watermarking ini dapat diterapkan pada berbagai *media digital.* Jadi *watermarking* merupakan 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.

2.2 *Digital Watermarking*

Digital watermarking adalah penyisipan sinyal *digital* ke dalam suatu media *digital*. *Digital watermarking* ini berangkat dari proses-proses pengolahan sinyal *digital,* dimana sinyal digital dapat berupa gambar, *audio, video*, dan teks. Seperti yang telah disebutkan sebelumnya, bahwa *digital watermarking* ini diimplementasikan dengan memanfaatkan kekurangan dari indera manusia (indera penglihatan dan indera pendengaran) dimana indera manusia ini kurang sensitif terhadap perubahan yang terjadi, misalnya saja perubahan yang terjadi pada level bit (sampai batas tertentu), perubahan pada level frekuensi (di luar frekuensi yang diterima manusia).

Ide *watermarking* pada data *digital* (sehingga disebut *digital watermarking)* dikembangkan di J*epang* tahun 1990 dan di *Swiss* tahun 1993. *Digital watermarking* semakin berkembang seiring dengan semakin meluasnya penggunaan internet, objek digital seperti *video,* citra, dan suara yang dapat dengan mudah digandakan dan disebarluaskan.

Hal yang memisahkan *watermarking* dari *steganografi* adalah dalam implementasinya, *steganografi* digunakan untuk mengamankan informasi yang ditumpangkan pada suatu media digital, sedangkan watermarking dapat dimanfaatkan untuk berbagai tujuan seperti :

- ***Tamper-proofing*;** *watermarking* digunakan sebagai indikator yang menunjukkan ada tidaknya perubahan pada data yang di*watermarking*.
- ***Feature location*;** menggunakan metode *watermarking* sebagai alat untuk identifikasikan isi dari data *digital* pada lokasi-lokasi tertentu, seperti contohnya penamaan objek tertentu dari beberapa objek yang lain pada suatu citra *digital*.
- ***Annotation/caption*;** *watermarking* yang digunakan sebagai keterangan tentang data *digital* itu sendiri atau informasi lain yang dipandang perlu untuk ditanamkan kedalam media yang bersangkutan.
- ***secure and invisible communications*** atau komunikasi yang aman.
- ***Copyright-Labeling*;** *watermarking* dapat digunakan sebagai metode untuk penyembunyikan label hak cipta pada data *digital* sebagai bukti otentik kepemilikan karya *digital* tersebut.

- *Watermarking* dapat juga dikategorikan sebagai *visible watermarking* (watermark terlihat oleh indera manusia) dan *invisible watermarking* (watermark tidak tampak).
- *Watermarking* dapat juga dikategorikan sebagai *blind watermarking* (proses verifikasi watermark tidak membutuhkan citra asal) dan *non blind watermarking* (proses verifikasi watermark membutuhkan citra asal).

Berdasarkan tipe dokumen yang diberi *watermark, watermarking* dapat diklasifikasikan menjadi:

1. Image Watermarking

2. Video Watermarking

3. Audio Watermarking

4. Text Watermarking

2.3 Karakteristik *Watermarking*

Ada beberapa karakteristik atau sifat khusus tertentu yang harus dimiliki oleh sebuah *watermark*. Sifat-sifat tersebut sangat bergantung kepada aplikasi *watermarking* yang akan dibuat, atau dengan kata lain tidak ada sekelompok sifat tertentu yang harus dipenuhi oleh semua teknik watermarking. Meskipun demikian ada beberapa sifat yang secara umum dipunyai aplikasi *watermarking*.

Sebuah *watermark* yang baik dan efektif idealnya akan memiliki karakteristik sebagai berikut:

1. *Perceptual transparency*

Sebagian besar aplikasi *watermarking* mengharuskan algoritma

watermarking digital menanamkan watermark sedemikian hingga ia tidak mempengaruhi kualitas media yang disisipi watermark. Media yang telah ditanami *watermark* haruslah sulit dibedakan dengan media aslinya oleh indera manusia, dengan kata lain penanaman *watermark* pada citra haruslah tidak terdeteksi oleh indera penglihatan manusia dan penanaman *watermark* pada audio haruslah tidak dikenali oleh indera pendengaran.

2. *Unobtrusiveness* (tak teramati)

Watermark yang baik harus tidak teramati atau tidak terlihat oleh manusia. Jadi antara *image* yang memiliki watermark dan yang tidak, secara umum tidak akan terlihat bedanya.

3. *Robustness* (kekuatan/ketahanan)

Sebuah *watermark* haruslah sulit, diharapkan mustahil untuk dihapus. Seandainya ada usaha-usaha untuk menghapus sebuah *watermark*, maka kualitas data *digital* akan mengalami penurunan, bahkan rusak. Ketahanan/*robustness* suatu *watermark* yang dibutuhkan adalah ketahanan terhadap rekayasa untuk menghapus *watermark.* Ketahanan *watermark* juga harus tinggi terhadap usaha-usaha rekayasa untuk menghapus *watermark*. Ada dua rekayasa yang biasa dilakukan yaitu penyatuan dan pemalsuan. Rekayasa dengan penyatuan misalnya adalah penggabungan beberapa salinan dari data digital dengan tujuan menghilangkan watermark. Demikian pula dengan pemalsuan yaitu dengan cara kombinasi data digital, sehingga dapat menghasilkan *watermark* yang berbeda. Kadang-kadang sebuah *watermark* hanya tahan terhadap sebuah proses tetapi rentan terhadap proses yang lain. Tetapi untungnya dalam banyak aplikasi,

ketahanan *watermark* terhadap semua proses yang mungkin tidak diperlukan dan dianggap terlalu berlebihan. Biasanya *watermark* harus tahan terhadap pemrosesan sinyal yang terjadi hanya antara proses *embedding* (penyembunyian *watermarking* dalam data) dan deteksi. Contohnya aplikasi *watermarking* pada televisi, jadi yang ditekankan disini adalah proses *kompresi lossy*, *transmisi analog,* dan sebagainya. Sedangkan aplikasi *watermarking* pada suara yang melalui kanal telepon berarti batasan bandwidth sekitar 4000 Hz, tipe data analog, dan *sampling* atau *resampling* pada beberapa *central telepon office* (CTO).

Tetapi untuk aplikasi *autenthication*, justru *watermark* diharapkan serentan mungkin terhadap proses pengolahan sinyal digital yang mungkin terjadi atau hampir seluruh proses pengolahan sinyal digital yang dapat dilakukan. Jadi ukuran *robustness* terhadap proses tertentu yang diperlukan untuk aplikasi tertentu mungkin tidak diperlukan dalam aplikasi yang lain. Untuk menentukan ukuran *robustness* harus terlebih dahulu dipikirkan aplikasi apa yang akan menggunakan sistem *watermarking*.

Ketahanan terhadap proses-proses pengolahan lainnya, itu tergantung pada metoda *watermarking* yang digunakan. Tetapi dari berbagai penelitian yang sudah dilakukan belum ada suatu metoda *watermarking* ideal yang bisa tahan terhadap semua proses pengolahan *digital* yang mungkin. Biasanya masing-masing penelitian menfokuskan pada hal-hal tertentu yang dianggap penting.

4. Tamper resistance

Yang dimaksud dengan *tamper resistance* adalah ketahanan sistem *watermarking* terhadap kemungkinan adanya serangan *(attack)* atau usaha untuk menghilangkan, merubah bahkan untuk memberikan *watermark* palsu terhadap suatu *host* data.

Ada beberapa jenis serangan *(attack)* terhadap sistem *watermarking* :

- ***Active attacks.*** Merupakan serangan dimana seseorang berusaha untuk menghilangkan *watermark* yang terdapat didalam *host* data.
- ***Passive attacks.*** Berbeda dengan *active attacks*, dimana serangannya hanya ditujukan untuk mengetahui apa isi *watermark* tersebut, jika memang ada di dalam *host* data.
- ***Collusion attacks.*** Serangan ini merupakan usaha seseorang untuk menghasilkan sebuah *copy* dari *host* data yang tidak memiliki *watermark* dengan memanfaatkan beberapa *host* data yang memiliki berbagai *watermark*, seperti pada aplikasi *fingerprinting*. Serangan ini merupakan serangan khusus yang termasuk dalam *active attacks*.
- ***Forgery attacks.*** Serangan ini tidak hanya bertujuan untuk membaca atau menghilangkan *watermark* yang ada, tetapi juga menanamkan suatu *watermark* yang baru (tentunya yang valid) ke dalam suatu *host* data. Serangan ini cukup menjadi perhatian yang serius terutama untuk aplikasi bukti kepemilikan (*proof of ownership*)

5. *Fidelity*

yaitu perbandingan antara kualitas arsip penampung setelah penyisipan *watermark* dengan kualitas arsip semula. Pada penyisipan yang baik, perubahannya tidak dapat dikenali oleh manusia.Untuk *host* data yang berkualitas tinggi maka *fidelity* dituntut setinggi mungkin sehingga tidak merusak data aslinya, sedangkan *host* data yang memiliki *noise* (kualitas kurang) maka *fidelitynya* bisa rendah seperti pada suara pada siaran radio, suara pada telepon ataupun *broadcast* acara televisi.

6. Computational Cost

Ada beberapa aplikasi yang menuntut proses *watermarking* baik *embedding* maupun *extracting* bekerja secara *real time*, ada juga yang mengharapkan salah satu baik *extracting* atau *embedding* saja yang *real time* ataupun duanya boleh tidak *real time*. Contohnya untuk aplikasi *owner identification* atau *proof of ownership*, proses *watermarking* baik *embedding* maupun *extracting* tidak perlu *real time*, sedangkan untuk aplikasi *fingerprinting* pada service *video on demand*, maka proses *embedding watermark* harus dilakukan secara *real time*.

7. *Recovery*

yaitu pengungkapan terhadap data yang disembunyikan. *Watermark* yang disisipkan harus dapat di deteksi kembali.

2.4 Jenis-jenis Watemarking

Secara garis besar, ada dua jenis *watemarking* :

1. Robust watemarking

Jenis *watermark* ini tahan terhadap serangan (*attack*), namun biasanya *watermark* yang dibubuhi ke dokumen masih dapat ditangkap oleh indera penglihatan atau pendengaran manusia.

2. Fragile watemarking

Jenis *watermark* ini akan mudah rusak jika terjadi serangan, namun kehadirannya tidak terdeteksi oleh indera manusia. Jika diinginkan untuk membuat suatu algoritma yang dapat mengimplementasikan *watemarking* yang memiliki *fidelity* yang tinggi (adanya *watermark* tidak disadari oleh pengamatan manusia) maka hasilnya akan semakin rentan terhadap serangan.

2.5 Syarat-syarat Sebuah *Digital Watemarking* yang Ideal

Untuk mendapatkan suatu teknik *digital watemarking* yang baik, maka teknik tersebut harus dapat memenuhi kondisi di bawah ini:

1. Elemen dari suatu data *digital* dapat secara langsung dimanipulasi dan informasi dapat ditumpangkan ke dalam data *digital* tersebut
2. Penurunan kualitas dari data *digital* setelah dibubuhkan *watermark*, dapat seminimal mungkin.
3. *Watermark* dapat dideteksi dan diperoleh kembali meskipun setelah data *digital* diubah sebagian, dikompresi, ataupun di-*filter*.

4. Struktur dari *watermark* membuat penyerang sulit untuk mengubah informasi yang terkandung di dalamnya.
5. Proses untuk membubuhkan *watermark* dan mendeteksinya cukup sederhana
6. Jika *watermark* dihapus, maka kualitas dari data *digital* yang ditumpanginya akan berkurang jauh atau bahkan rusak sama sekali.
7. Informasi *watermark* yang diselipkan dalam isi data *digital* dapat dideteksi ketika dibutuhkan.
8. Label hak cipta yang unik mengandung informasi pembuatan, seperti nama, tanggal, dan sebagainya, atau sebuah kode hak cipta seperti halnya ISBN (*International Standard for Book Notation*) pada buku-buku.
9. *Watermark* tidak dapat diubah atau dihapus (*robustness*) secara langsung oleh orang lain atau dengan menggunakan *software* pengolahan sinyal sampai tingkatan tertentu.
10. *Watemarking* yang diberikan lebih dari satu kali dapat merusak data *digital* aslinya. Cara ini dilakukan supaya orang lain tidak dapat melakukan pelabelan berulang terhadap data yang telah dilabel.

Sampai saat ini, belum ada teknik *watemarking* yang dapat memenuhi seluruh kriteria di atas.

2.6 Aplikasi *Watermarking*

Berikut beberapa aplikasi dari *watermarking*, antara lain :

- ***Broadcast Monitoring***

Pernah ada kejadian pada tahun 1997 di jepang mengenai periklanan di televisi, pemasang iklan dikenakan biaya terhadap iklan yang tidak pernah ditayangkan. Hal ini karena tidak ada mekanisme untuk mengawasi penayangan iklan yang sedang berlangsung. Para musisi dan aktor-aktor juga ingin dibayar atas penayangan pertunjukan mereka. Demikian juga pemilik hak cipta tidak ingin miliknya ditayangkan secara illegal oleh stasiun yang membajak.

Watermarking dapat digunakan dalam *broadcast monitoring* dengan menambahkan *watermark* yang unik kedalam tiap video ataupun suara sebelum ditayangkan oleh stasiun televisi atau disiarkan oleh stasiun radio. Dan sebuah stasiun pengamat otomatik akan menerima tayangan tersebut sehingga dapat mengekstrak informasi *watermark* yang dibawanya dan mencatat kapan dan dimana tayangan tersebut muncul.

- ***Owner Identification***

Keterangan hak cipta biasanya ditulis pada buku ataupun pada foto-foto dalam bentuk *waktu, pemilik*, tetapi keterangan ini ditulis secara eksplisit dan kelihatan. Demikian juga pada film biasanya diletakkan pada akhir film, sedangkan pada kaset atau CD *audio* diletakkan pada kotak pembungkusnya. Kekurangan dari cara diatas adalah semua informasi mengenai hak cipta tersebut dapat dihilangkan baik dengan sengaja ataupun tidak disengaja seperti hilangnya

kotak pembungkus kaset atau CD *audio*, film dipotong untuk diambil sebagian saja, gambar juga dapat dipotong (*cropping*), dan sebagainya.

Dengan demikian *watermarking* tampaknya dapat digunakan sebagai tool untuk *owner identification*, karena informasi hak cipta tersebut diletakan didalam data *host*-nya dan merupakan bagian dari data *host* tersebut, sehingga usaha untuk menghilangkan informasi hak cipta tersebut dapat menurunkan kualitas data *host*nya.

- ***Proof of Ownership***

Watermarking selain dapat digunakan untuk tanda pengenalan pemilikan (*owner identification*) seperti yang disebutkan diatas, juga dapat digunakan untuk pembuktian kepemilikan. Pembuktian kepemilikan ini diperlukan pada saat dua orang memperebutkan hak kepemilikan atau menyatakan bahwa data digital tersebut adalah miliknya. Jadi untuk membuktikannya dapat digunakan *watermarking*. Tentunya segala sesuatu relugasi hukumnya harus ditentukan secara benar dan semua ini memerlukan usaha yang sulit.

- ***Authentication***

Untuk membuktikan bahwa suatu data digital itu benar-benar asli dan tidak mengalami sedikit perubahan apapun juga dapat diterapkan dengan prinsip *watermarking*. Pembuktian bahwa data tersebut asli tanpa perubahan apapun meskipun perubahan kecil yang tidak dapat dipersepsi oleh mata atau telinga seperti penambahan pada citra diam beberapa garis halus pada tempat-tempat

tertentu atau perubahan degradasi warna yang tidak disadari oleh mata sangat perlu untuk aplikasi-aplikasi tertentu seperti pada citra medis.

Sebelumnya pembuktian keaslian ini pernah dilalukan dengan metoda *cryptography*. Dimana *signature cryptography* yang berkaitan dengan data (dalam hal ini citra) tersebut diekstrak dan disertakan dengan data asli tersebut. Bila terjadi perubahan sedikit saja pada data aslinya maka *signature cryptography*-nya otomatis akan berbeda. Tapi sayangnya signature ini merupakan data tambahan yang dikirimkan bersamaan dengan data aslinya ataupun diletakkan pada header dari data tersebut. Dengan demikian signature tersebut dapat dengan mudah dihilangkan, sehingga pembuktian keaslian tidak dapat dilakukan lagi. Dengan *watermarking*, dimana digabungkan dengan isi data tersebut, maka kemana pun data tersebut, baik di-*cropping*, diubah ke dalam format digital lain, dan sebagainya, *watermarknya* tetap akan ada bersama dengan *host* datanya.

- ***Fingerprinting***

*Fingerprintin*g atau *transactional watermark* merupakan aplikasi *watermarking* dengan menyembunyikan informasi *watermark* yang berbeda-beda kepada tiap data digital yang didistribusikan. Biasanya untuk aplikasi pelabelan hak cipta, *broadcast monitoring*, semua data *digital* yang akan didistribusikan diisikan dengan informasi hak cipta yang sama.

Fingerprinting biasanya digunakan bila terjadi transaksi sebuah data digital antara pemilik dengan seorang konsumer, misalnya sebuah provider memberikan service *Video On Demand*, maka data yang akan didistribukan

tersebut di*watermark* dengan informasi asal dan tujuan mungkin juga waktunya. Jadi bila terjadi perdagangan atau pembajakan terhadap data *digital* yang pernah ditransaksikan, maka dapat diketahui sumber kebocorannya atau pembajaknya. Penjelasan mengenai fingerprinting akan dibahas lebih lanjut pada bagian aplikasi video *watermarking* pada *broadcasting* dan *streaming video*.

- ***Copy Control***

Penggunaan *watermark* sebagai *copy control* hampir sama dengan copy protection yang digunakan pada disket-disket (*disk-protection*) beberapa tahun yang lampau. Penerapan *watermarking* sebagai *copy control* harus disertai dengan penanaman *watermarking detector* pada perangkat *hardware* untuk membaca data digital tersebut. Bila *detector* mendeteksi adanya *watermark* pada data digital yang akan dibacanya, maka beberapa proses yang dapat dilakukan hardware tersebut misalnya peng-*copy*-an akan di-*disable*-kan.

Aplikasi *copy control* juga sudah dikembangkan untuk diterapkan pada perangkat modern saat ini seperti pada *DVD player*. Tentunya penerapannya harus disepakati oleh seluruh pembuat perangkat player untuk menambahkan rangkaian detector *watermark* tersebut.

- ***Covert Communication***

Salah satu aplikasi *steganography* pada awalnya adalah untuk komunikasi rahasia terutama pada jaman perang. *Watermarking* sebagai perkembangan dari *steganography* juga dapat digunakan sebagai media untuk mengrimkan pesan-pesan rahasia kepada sekutu sehingga tidak diketahui musuh. Aplikasi

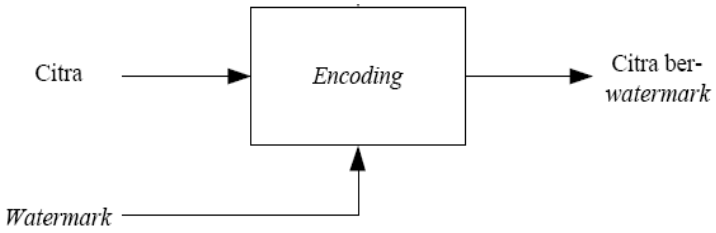
watermarking sebagai media komunikasi rahasia lebih dikenal sebagai data *hiding*.

2.7 Proses dan *Framework Watermarking*

Jika *watermark* merupakan sesuatu yang ditanamkan, maka watermarking merupakan proses penanaman *watermark* tersebut. Secara umum *framework* sebuah algoritma *watermarking* tersusun atas dua bagian, yaitu *[(Duan and King, 1999) dan (Mohanty, 1999)] :*

(1) Algoritma penyisipan *watermark (enkoder),*

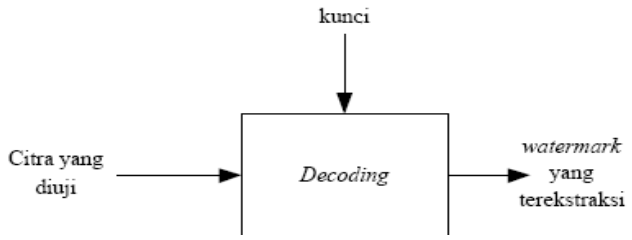
Algoritma penyisipan *watermark* menangani bagaimana sebuah *watermark* ditanamkan pada media induknya. Dibawah ini penggambaran dari proses *encoding.*



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

(2) Algoritma pendeteksian watermark (dekoder).

Deteksi *watermark* dilakukan untuk membuktikan status kepemilikan citra digital yang disengketakan. Verifikasi *watermark* terdiri atas ekstraksi *watermark.* Proses ekstraksi *watermark* disebut juga *decoding*, bertujuan mengungkap *watermark* dari dalam citra. Algoritma pendeteksian *watermark* menentukan apakah didalam sebuah media *digital* terdeteksi *watermark* yang sesuai atau tidak.



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

Watermark dapat berupa representasi identitas kepemilikan media *digital,* maupun informasi lain yang dipandang perlu untuk ditanamkan kedalam media yang bersangkutan. Algoritma penyisipan *watermark* menangani bagaimana sebuah *watermark* ditanamkan pada media induknya. Algoritma pendeteksian *watermark* menentukan apakah didalam sebuah media *digital* terdeteksi *watermark* atau tidak.

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

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

b. **Logo atau Citra atau Suara** : Berbeda dengan *text,* kesalahan pada beberapa bit masih dapat memberikan persepsi yang sama dengan aslinya oleh pendengaran maupun penglihatan kita, tetapi kerugiannya adalah jumlah data yang cukup besar.

2.8 Kalsifikasi *Watermarking* Pada Citra *Digital*

Watermarking pada citra *digital* secara umum dapat diklasifikasikan menurut warnanya menjadi 2 jenis yaitu :

2.8.1 *Watermarking* pada Citra *Gray Scale*

Citra *gray scale* merupakan jenis citra yang banyak dimanfaatkan sebagai obyek algoritma *watermarking* yang pernah diusulkan. Salah satu alasannya adalah bahwa citra ini tidak terlalu kompleks untuk diolah karena ia dapat direpresentasikan sebagi matriks dua dimensi dengan nilai masing-masing elemennya merupakan representasi tingkat intensitas keabuan dari setiap pikselnya. Informasi tingkat keabuan setiap pikselnya disimpan dalam satuan 8 bit, atau dengan kata lain disini ada 2^8 atau 256 warna keabuan yang berbeda, mulai dari 0 yang merepresentasikan warna hitam sampai dengan 255 yang mewakili warna putih.

Pada dasarnya *watermarking* pada citra *gray scale* dapat diterapkan secara langsung pada citra ini sesuai dengan jenis teknik watermarking yang digunakan. Sebagai contoh : jika teknik *spatial domain* yang digunakan maka nilai intensitas piksel-piksel dari citra ini yang akan diubah sesuai dengan prosedur yang digunakan untuk menanamkan *watermark*. Begitupun dengan teknik yang bekerja pada domain frekuensi, *transformasi* secara langsung dapat dilakukan pada citra, meskipun pada algoritma tertentu kadang-kadang diperlukan pembagian citra menjadi blok-blok yang lebih kecil. Kemudian watermark ditanamkan pada koefisien-koefisien frekuensi yang terpilih.

2.8.2 *Watermarking* pada Citra Berwarna

Berbeda dengan teknik *watermarking* pada citra *gray scale,* dalam kasus citra berwarna, *watermar*k dapat ditanamkan didalam satu atau lebih kanal warna yang ada pada ruang warna citra tersebut. Beberapa teknik *watermarking* hanya menggunakan kanal warna biru sebab sistem penglihatan manusia (*human visual system* (HVS)) tidak terlalu peka pada komponen warna ini (Swanson, 1998). Cara lain adalah dengan mentransformasikan ruang warna RGB ke ruang warna lain seperti HSL atau Yuv, kemudian *watermark* ditanamkan pada komponen *luminance*-nya saja, dan ditransformasikan kembali ke ruang warna RGB.

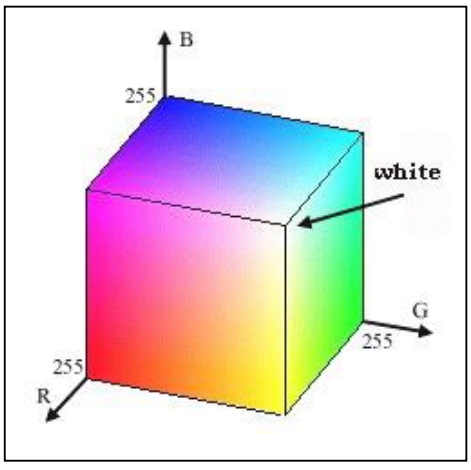
2.9 Ruang Warna

Warna pada dasarnya merupakan hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400 nm sampai dengan 700 nm (Poynton , 1997). Ruang warna atau yang sering juga disebut sebagai model warna merupakan sebuah cara atau metode untuk mengatur, membuat dan memvisualisasikan warna *(Ford and Roberts, 1998).* Untuk aplikasi yang berbeda ruang warna yang dipakai bisa juga berbeda, hal ini dikarenakan beberapa peralatan tertentu memang membatasi secara ketat ukuran dan jenis ruang warna yang dapat digunakan.

Ruang warna pada citra terbagi atas dua jenis yaitu, ruang warna citra RGB dan ruang warna citra Yuv.

2.9.1 RGB (*Red Green Blue*)

Citra berwarna yang selama ini biasa kita kenal umumnya memiliki ruang warna RGB. Ruang warna RGB dapat divisualisasikan sebagai sebuah kubus seperti pada gambar II-4, dengan tiga sumbunya yang mewakili komponen warna merah (*red*) *R*, hijau (*green*) *G* dan biru (*blue*) *B*. Salah satu pojok alas kubus ini menyatakan warna hitam ketika $R = G = B = 0$, sedangkan pojok atasnya yang berlawanan menyatakan warna putih ketika $R = G = B = 255$ (untuk sistem warna 8 bit bagi setiap komponennya). RGB sering digunakan didalam sebagian besar aplikasi komputer karena dengan ruang warna ini, tidak diperlukan transformasi untuk menampilkan informasi di layar monitor. Alasan diatas juga menyebabkan RGB banyak dimanfaatkan sebagai ruang warna dasar bagi sebagian besar aplikasi.



Gambar 2.3 Ruang warna RGB.

2.9.2 Yuv

Ruang warna Yuv merupakan standar internasional bagi pengkodean digital gambar televisi yang didefinisikan di CCIR Recommendation 601 (Ford dan Roberts, 1998). Y merupakan komponen *luminance*, u dan v adalah komponen *chrominance*. Pada monitor monokrom nilai *luminance* digunakan untuk merepresentasikan warna RGB, secara psikologis ia mewakili intensitas sebuah warna RGB yang diterima oleh mata. *Chrominance* merepresentasikan corak warna dan saturasi (*saturation*). Nilai komponen ini juga mengindikasikan banyaknya komponen warna biru dan merah pada warna (Cuturicu, 1999).

Retina mata mempunyai dua macam sel yang berfungsi sebagai analis visual, yaitu : Sel yang digunakan untuk penglihatan di waktu malam dan sel yang dipakai untuk penglihatan di siang hari. Jenis yang pertama hanya menerima corak keabuan mulai dari warna putih terang sampai dengan hitam pekat. Dan jenis kedua menerima corak warna. Jika sebuah warna RGB diberikan, sel jenis yang pertama mendeteksi tingkat keabuan (*gray level*) yang serupa dengan nilai *luminance*-nya, sedangkan sel jenis kedua yang bertanggung jawab terhadap penerimaan corak warna, mendeteksi nilai yang sesuai dengan nilai *chrominance*- nya (Cuturicu, 1999).

Yuv (256 level) dapat diperoleh dari RGB 8-bit dengan menggunakan rumus berikut *(Hamilton,1992)* :

$$Y = 0.299\ R + 0.587\ G + 0.114\ B$$

$$u = -\ 0.1687\ R - 0.3313\ G + 0.5\ B + 128$$

$$v = 0.5\ R + 0.4187\ G - 0.0813\ B + 128$$

Sedangkan untuk konversi Yuv ke RGB dapat dilakukan dengan rumus berikut *(Hamilton,1992)* :

R = Y + 1.402 (u - 128)

G = Y – 0.34414 (u - 128) – 0.71414 (v - 128)

B = Y + 1.772 (v - 128)

2.10 Teknik *Watermarking* Citra *Digital*

Terdapat banyak metoda *watermarking* untuk citra *digital* yang sudah diteliti. Ada yang bekerja pada domain *spasial atau waktu*, dan ada yang mengalami transformasi terlebih dahulu (seperti DCT, FFT, dsb) misalnya ke domain *frekuensi*. Bahkan ada yang menerapkan teknologi teknologi lain seperti *fraktal, spread spectrum* untuk telekomunikasi dan sebagainya.

2.10.1 *Discrete Cosine Transform* (DCT)

Discrete Cosine Transform (DCT) biasa digunakan untuk mengubah sebuah sinyal menjadi komponen frekuensi dasarnya. DCT adalah sebuah transformasi yag mengubah sebuah kawasan spasial menjadi kawasan fekuensi dan sebaliknya kawasan frekuensi dapat dikembalikan ke kawasan spasial dengan menggunakan invers DCT. DCT pertama kali diperkenalkan oleh Ahmed, Natarajan dan Rao pada tahun 1974 dalam makalahnya yang berjudul "*On image processing and a discrete cosine transform*" (*Watson, 1994).*

Algoritma-algoritma *watermarking* yang bekerja pada domain DCT menurut ukuran bloknya dapat dibagi kedalam 2 kelompok, yaitu algoritma yang memanfaatkan blok-blok DCT dengan ukuran tertentu yang lebih kecil dari

ukuran citra aslinya, misalnya blok dengan ukuran 8x8 atau 16x16. Kelompok kedua menerapkan DCT keseluruh dimensi citra secara langsung.

Transformasi citra dilakukan dengan menggunakan DCT (*Discrete Cosine Transform*), sehingga dapat dikatakan bahwa penyisipan dilakukan pada ranah DCT. Penyisipan dilakukan terhadap citra bitmap dengan kedalaman warna 24 bit. DCT digunakan untuk metransformasikan nilai intensitas blok 8x8 pikselnya yang berurutan dari image menjadi 64 koefisien DCT ke dalam frekuensi dasarnya, diubah koefisien-koefisiennya dan kemudian ditransformasikan kembali dengan IDCT (*Inverse Discrete Cosine Transform*). Setiap basis *matriks* dikarakterisasikan oleh frekuensi *spasial horizontal* dan *vertikal*. Frekuensi *horizontal* meningkat dari kiri ke kanan, dan dari atas ke bawah secara *vertikal.* Dalam konteks citra, hal ini menunjukkan tingkat signifikansi secara perseptual, artinya basis fungsi dengan frekuensi rendah memiliki sumbangan yang lebih besar bagi perubahan penampakan citra dibandingkan basis fungsi yang memiliki frekuensi tinggi. Nilai konstanta basis fungsi yang terletak di bagian kiri atas sering disebut sebagai basis fungsi DC, dan DCT koefisien yang bersesuaian dengannya disebut sebagai koefisien DC (*DC coefficient*).

Masukan proses DCT berupa matrik N x N. Persamaan DCT untuk blok matrik berukuran N x N dapat dituliskan sebagai berikut :

$$\mathbf{S\ (u,v)} = \frac{2}{\sqrt{nm}}\ \mathbf{C(u)\ C(v)} \sum_{y=0}^{m-1} \sum_{x=0}^{n-1} \mathbf{s(x.y)\ cos}\ \frac{(2x+1)u\pi}{2n}\ \mathbf{cos}\ \frac{(2y+1)v\pi}{2m}$$

Dengan u = 0,…,n-1,v = 0,…,m-1

Dimana :

S(u,v) = data pada domain frekuensi

S(x,y) = data pada domain ruang

Rumusan *invers* DCT sebagai berikut :

$$\mathbf{S(x,y)} = \frac{2}{\sqrt{nm}} \sum_{v=0}^{m-1} \sum_{u=0}^{n-1} \mathbf{S(u,v)\ C(u)\ C(v)\ cos}\ \frac{(2x+1)u\pi}{2n}\ \mathbf{cos}\ \frac{(2y+1)v\pi}{2m}$$

Dengan x = 0,….,n – 1, y = 0, …., m – 1

Dimana :

S(u,v) = data pada domain frekuensi

S(x,y) = data pada domain ruang

Output dari fungsi DCT adalah nilai komponen frekuensi tertentu dan *output* dari fungsi ini ditentukan oleh dua parameter, yaitu u dan v. Cara menentukan mana yang frekuensi rendah dan mana yang frekuensi tinggi adalah dengan menjumlahkan nilai u dan v. Jadi jika u+v makin tinggi berarti S(u,v) menyatakan komponen frekuensi yang makin tinggi. Input dan *output* dari fungsi DCT juga merupakan suatu matrix dengan ukuran NxN. P(x,y) adalah nilai *pixel* pada koordinat (x,y), *index* dimulai dari 0.

Fungsi C(f) adalah suatu fungsi yang mengembalikan nilai satu per akar dua bila f=0 dan mengembalikan nilai satu bila f tidak sama dengan nol. Untuk perhitungan *cosinus* dilakukan dalam *modus radian* (bukan derajat). Sedangkan *variable* N adalah *variable* yang menyatakan ukuran *matrix,* misalnya ukuran *matrix* adalah 4 x 4 berarti N=4. Untuk rumus *inverse* DCT berlaku hal yang sama. Pada rumus *inverse* DCT, S(u,v) yang digunakan adalah S(u,v) hasil dari

rumus DCT. *Output* dari fungsi *invers* DCT adalah P(x,y), nilai *pixel* pada koordinat (x,y). Untuk memperjelas penggunaan rumus diberi satu contoh, misalkan ada suatu citra yang berukuran 4x4 *pixel*, *gray scale* 8-bit, dengan nilai komponen seperti contoh 1.

Contoh 1

32	10	2	41
0	35	23	152
254	43	180	53
7	124	240	12

nilai pada baris 1 kolom 1 adalah nilai untuk P(0,0), sedangkan nilai pada baris 4 kolom 4 adalah nilai untuk P(3,3). Jika nilai tersebut dimasukkan ke rumus DCT dengan N=4, maka hasilnya adalah seperti contoh 2 berikut :

Contoh 2

302.0000	-20.0922	-26.5000	80.8428
-140.6346	-31.2787	133.9691	-105.8150
-68.0000	-18.2783	-115.5000	-14.0767
64.2059	132.6850	51.6650	103.7787

nilai baris 1 kolom 1 adalah nilai untuk frekuensi rendah, sedangkan makin ke arah kanan bawah menyatakan frekuensi yang makin tinggi. Jika hasil dari DCT

ini dimasukkan kembali ke rumus IDCT maka akan dihasilkan seperti data aslinya, contoh 3 :

Contoh 3

32	10	2	41
0	35	23	152
254	43	180	53
7	124	240	12

2.11 Rumusan Kuantisasi

Proses kuantisasi adalah proses yang digunakan untuk mengecilkan range dari *matrix* p_dct. Jika output dari kuantisasi adalah *matrix* q, maka rumusnya adalah :

q[i,j] = round(p_dct[i,j] / qua [i,j])

Matrix qua adalah *matrix* kuantisasi. Nilai koefisien *matrix* ini cenderung akan semakin besar untuk i dan j yang makin besar (umumnya qua[i2,j2] > qua[i1,j1] untuk i2,j2 > i1,j1). Pada proses kuantisasi ini terjadi proses yang lossy yang menyebabkan sebagian informasi citra hilang. Hal ini disebabkan pada proses ini akan banyak terjadi pembulatan dan sebagian besar komponen frekuensi tinggi akan berisi nilai 0.

Selain dengan menggunakan rumus, *matrix* qua dapat ditentukan secara manual menurut tabel kuantisasi standarisasi, salah satu contohnya adalah sebagai berikut :

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Gambar 2.4 Matriks Kuantitasi

2.12 *Image*

Citra atau *image* merupakan istilah lain untuk gambar, sebagai salah satu komponen multimedia yang memagang peranan sangat penting sebgai bentuk informasi *visual*. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Sebuah gambar dapat memberikan informasi yang lebih banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (Tekstual).

Istilah citra pada umumnya digunakan pada bidang pengolahan citra diartikan sebagai suatu fungsi kontinu dari intensitas cahaya f(x,y) dalam bidang dua dimensi, dengan (x,y) menyatakan suatu kordinat dan nilai f pada setiap titik (x,y) menyatakan intensitas atau tingkat kecerahan atau derajat keabuan. Dari pengertian diatas, citra dapat dibedakan atas 2 jenis, yaitu :

2.12.1 Gambar Bitmap

Gambar bitmap sering disebut juga dengan gambar raster. Gambar bitmap adalah gambar yang terbentuk dari pixel, dengan setiap pixelnya mempunyai warna tertentu. Jika gambar bitmap ini diperbesar, misalnya menjadi 4 kalinya, maka gambar akan menjadi kabur karena pixelnya juga bertambah besar menjadi 4 kalinya (kualitas gambar menurun), format gambar bitmap sering dipakai dalam foto dan gambar. Dua istilah yang perlu dipahami ketika bekerja dengan gambar bitmap adalah resolusi dan kedalaman warna. Gambar bitmap biasanya diperoleh dari proses *scanner*, *camera digital*, *video capture* dll. Contoh filenya adalah BMP, GIF dan JPG.

2.12.2 Gambar Vector

Gambar vector dihasilkan dari perhitungan matematis dan tidak berdasarkan *pixel.* Jika gambar diperbesar atau diperkecil, kualitas gambar relative tetap baik dan tidak berubah. Gambar *vector* biasanya dibuat menggunakan aplikasi – aplikasi gambar vector. Misalkan *corel draw*, *adobe illistrator, macromedia freehand, autocad,* dll. Contohnya filenya PDF.

Dari kedua jenis gambar diatas maka pengujian dilakukan pada gambar bitmap karena gambar bitmap adalah gambar yang terbentuk dari pixel-pixel dan teknik penyisipan digunakan dengan mengubah *pixel-pixel* dari citra asal.

2.13 Parameter Objektif

Untuk menentukan kualitas *image* asal dengan *image* terwatermark diperlukan suatu pengujian secara objektf. Parameter yang dijadikan kriteria pengujian secara objektif adalah sebagai berikut :

2.13.1 Mean Sequare Error

MSE adalah rata-rata kuadrat nilai error antara *image* asli dengan *image* ter*watermark*, secara matematis dapat dirumuskan sebagai berikut:

$$\text{MSE} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x,y) - f(x,y)]^2$$

Dimana :

MSE = Nilai *Mean Square Error* dari *image* ter*watermark*

M = Panjang *image* hasil (dalam *pixel*)

N = Lebar *image* hasil (dalam *pixel)*

(x,y) = Koordinat masing-masing *pixel*

f' = Nilai bit *image* ter*watermark* pada koordinat x,y

f = Nilai bit *image* asli pada koordinat x,y

2.13.2 Peak Signal to Noise Rasio

Parameter ukur yang digunakan untuk mengetahui performansi citra digital yang dihasilkan dari proses *watermark* dalam Tugas Akhir ini adalah PSNR. PSNR merupakan nilai perbandingan antara harga maksimum dari *image* ter*watermark* dengan *noise,* yang dinyatakan dalam satuan *desibel* (dB). Secara matematis, nilai PSNR dapat dirumuskan sebagai berikut :

PSNR untuk citra yang terkena ter*watermark* :

$$\text{PSNR} = 10^{10} \log \frac{f(x,y)^2}{(f(x,y) - h(x,y))^2}$$

Dimana :

f(x,y) adalah *image* asli

h(x,y) adalah *image* ter*watermark*

2.14 Parameter *Subjektif*

Untuk pengukuran secara subjektif kriteria yang ditentukan berdasarkan pengamatan mata manusia karena sifat penglihatan manusia yang mmpu melakukan pendeteksian langsung, sehingga kualitas *subjektif* tergantung kepada persepsi visual pengamat.

Krieria penilaian secara *subjektif* yang digunakan sebagai berikut :

1. *Unusable*, *image* yang diamati memiliki kualitas yang sangat rendah, sehingga sudah tidak dapat dilihat lagi.
2. *Inferior*, *image* yang diamati memiliki kualitas yang sangat rendah, tapi masih bias dilihat. Keberasaan interferensi benar – benar mengganggu.
3. *Marginal*, *image* yang diamati memiliki kualitas yang sangat rendah, interferensi tarasa cukup mengganggu.
4. *Passable*, *image* yang diamati memiliki kualitas yang cukup tinggi, dimana interferensi terasa agak mengganggu.

5. *Fine*, *image* yang diamati memiliki kualitas yang tinggi, enek dilihat, dimana interferensi belum terasa mengganggu.
6. *Excellent*, *image* yang diamati memiliki kualitas yang sangat tinggi, sebaik – baiknya kualitas sebagaimana yang diinginkan.

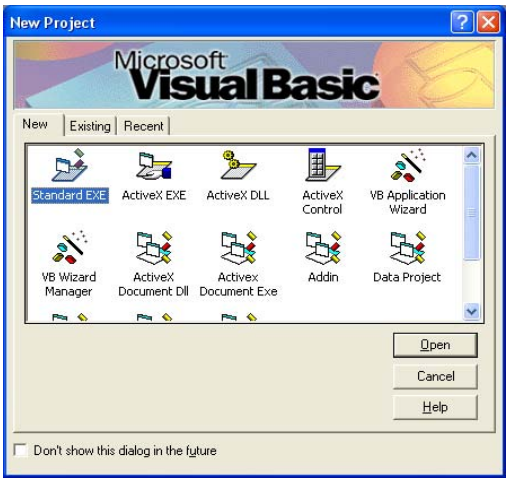
2.15 *Visual Basic* atau VB 6.0

Visual basic merupakan bahasa pemograman yamg tercepat dan termudah untuk membuat satu aplikasi dalam *microsoft windows.* Dengan menggunakan metode *Graphical User Interface (GUI). Visual basic* memudahkan pemograman untuk berinteraksi langsung dengan elemen-elemen untuk setiap bentuk pemograman. *Visual basic* dibuat sebagai langkah pengembangan untuk menyesuaikan BASIC *(Beginners All-Purpose Symbolic Intruction Code*) yang berbasis DOS yang tidak mempunyai kemampuan menggunakan metode GUI dalam basis *windows.*

Sebagai program yang berbasis *windows, Visual Basic* mempunyai kemampuan untuk berinteraksi dengan seluruh aplikasi *Windows* seperti *Microsoft Word, Microsoft Excel, Microsoft Acces* dan sebagainya. Dengan kemampuan yang hampir tidak terbatas, *Visual basic* dapat digunakan untuk semua jenis aplikasi pemograman. Anda dapat membuat aplikasi windows seperti *word* atau *exce*l ataupun *game*, *multimedia,* program penghitungan dan sebagainya. Selain itu *visual basic* juga dapat digunakan untuk menguji program *(debugging*) dan menghasilkan program akhir berakhiran *exe* yang bersifat *executable* atau dapat langsung digunakan.

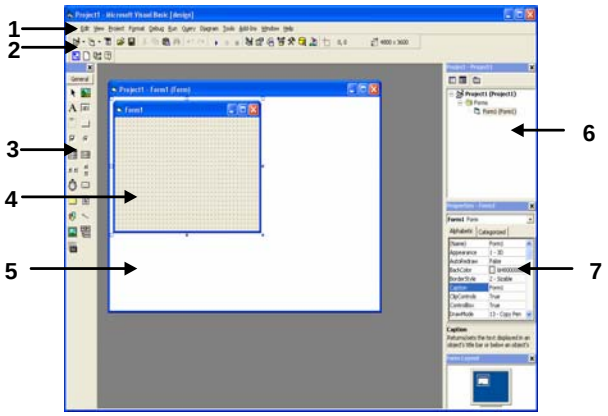
2.15.1 *Integrated Development Environment* (IDE) VB 6

Merupakan lingkungan pengembangan dari *visual basic* 6 yang terintegrasi. Ketika pertama kali visual basic diaktifkan maka akan muncul tampilan seperti pada gambar dibawah ini :



Gambar 2.5 Tampilan awal *Visual Basic*

Setelah kita memilih layar *new project standard exe*, selanjutnya akan tampil layar kerja atau area kerja dari *visual basic*, seperti gambar dibawah ini:



Gambar 2.6 Tampilan *New Project*

Keterangan :

1. *Menubar*

   Menubar berfungsi untuk memilih tugas-tugas tertentu, seprti memulai, membuka dan menyimpan *project*, mengompilasi *project* menjadi *file executable (exe)* dan lain-lain.

2. *Toolbar*

   Toolbar memiliki fungsi yang sama seperti menubar, dan juga berfungsi seperti jalan pintas karena lebih praktis dalam penggunaanya.

3. *Toolbox*

   Toolbox berisi komponen-komponen yang merupakan sarana untuk membentuk *user interface.*

4. Jendela *Form Designer*

   Form designer merupakan tempat untuk merancang *user interface,* dimana kontrol-kontrol yang dibutuhkan bisa diletakkan.

5. Jendela Kode

   Jendela kode merupakan tempat untuk menuliskan kode-kode bagi objek yang dibuat. Melalui jendela ini dapat mengatur kerakteristik dari form yang dibuat.

6. Jendela *Project*

   Jendela *project* adalah jendela yang menampilkan semua file yang berhubungan denagn aplikasi atau *project* yang saat itu sedang dijalankan.

7. Jendela *Properties*

   Jendela *properties* ini berisi daftar *property* untuk objek (form atau control) yang dipilih dan berfungsi untuk mengatur karakteristik, seperti warna, ukuran dan lain-lain.

Banyak sekali *tool-tool* dan jendela-jendela pendukung lainnya yang ada di lingkungan visual basic 6 sebagai alat untuk mengembangkan aplikasi yang akan dibuat. Seperti jendela *form layout* yang menunjukkan tampilan *form* pada saat dijalankan.

Ada juga jendela *modul* yang merupakan tempat untuk menuliskan kode-kode yang lebih bersifat global. Pada jendela modul ini, bisa mendeklarasikan *variabel* dan fungsi dengan awalan *private* maupun *public*. Jika diberi awalan private, variabel atau fungsi tersebut hanya berlaku untuk pemakaian *modul* yang bersangkutan. Sedangkan awalan *public* akan mengakibatkan variabel atau fungsi tersebut bersifat global. Jendela modul ini bisa ditampilakan denagn mengklik menu *project* pada menubar kemudian pilih *add module.*