BAB II
LANDASAN TEORI

II.1.Pengertian Steganografi

Steganografi merupakan seni untuk menyembunyikan pesan di dalam pesan lainnya sehingga orang lain tidak menyadari bahwa ada sesuatu di dalam pesan tersebut (Judge, 2001). Steganografi memerlukan dua properti yaitu media penampung (*carrier file*) dan informasi yang hendak disembunyikan didalam *carrier file* yaitu *hidden file*.

Untuk dapat membaca informasi yang tersimpan di *carrier file*, dibutuhkan *password* atau *key file* (*file* sebagai kunci) yang telah di *set* di awal oleh user. Dalam *steganografi* digital terdapat berbagai macam format file yang dapat dijadikan sebagai *carrier file* antara lain (Sukmawan, 2002):

1. Format *image: bitmap (bmp), gif, pcx, jpeg, png* dan lain lain
2. Format *audio: wav, voc, mp3,wma, oog* dan lain lain
3. Format *Video: AVI, Mpeg, Mp4, H264, MOV* dan lain lain
4. Format lain: *teks file, html, pdf*, dan lain lain

Format diatas berlaku juga untuk *hidden file* asalkan *carrier file* memliki ukuran dan ruang yang cukup besar.

Banyak sekali metode yang digunakan didalam steganografi. Steganografi *image* biasanya menggunakan metode *LSB (Least Significant Bit), Masking and*

Filtering, dan *transformation (DCT, wafelet,FFT*) (Henry, 2006). Untuk steganografi video, banyak metode dari steganografi *imag*e yang bisa diterapkan di video, karena pada dasarnya video adalah kumpulan dari *image* yang bergerak (*stream*). Akan tetapi metode yang sering diterapkan di steganografi video adalah *Discrete Cosine Transform* maupun *Wavelet Compression* (Henry, 2006). Hal ini dikarenakan modifikasi *LSB* akan menghasilkan *carrier file* yang berukuran sangat besar. Sedangkan metode *masking* dan *filtering* akan mengubah tampilan visual dari video secara langsung.

Metode Steganografi sering kali disamakan dengan metode kriptografi. Padahal kedua metode ini memiliki perbedaan yang cukup besar. Prinsip dari metode Kriptografi adalah mengacak pesan atau *hidden file* sehingga tidak dapat dimengerti oleh pihak lain (Sukmawan,2002). Sedangkan prinsip dari steganografi adalah menyembunyikan pesan atau *hidden file* sehingga tidak terlihat.

II.2.Sejarah Steganografi

Steganografi merupakan seni untuk menyembunyikan pesan di dalam pesan lainnya sehingga orang lain tidak menyadari bahwa ada sesuatu di dalam pesan tersebut (Judge, 2001). Kata steganografi (*steganograph*y) berasal dari bahasa Yunani *steganos*, yang artinya ‘tersembunyi’ atau ‘terselubung’, dan *graphein* yang artinya 'menulis' sehingga kurang lebih artinya "menulis (tulisan) terselubung".

Catatan pertama tentang steganografi ditulis oleh seorang sejarawan Yunani, Herodotus, yaitu ketika

Histaeus seorang raja Yunani dipenjarakan oleh Raja Darius di Susa pada abad 5 Sebelum Masehi. Histaeus harus mengirim pesan rahasia kepada anak laki-lakinya, Aristagoras, di Militus. Histaeus menulis pesan dengan cara mentato pesan pada kulit kepala seorang budak dan ketika rambut budak itu mulai tumbuh, Histaeus mengutus budak itu ke Militus untuk mengirim pesan di kulit kepalanya tersebut kepada Aristagoras.

Teknik steganografi yang lain pada jaman dahulu adalah tinta yang tak terlihat. Teknik ini pertama digunakan pada zaman Romawi kuno yaitu dengan menggunakan air sari buah jeruk, urine atau susu sebagai tinta untuk menulis pesan. Cara membacanya adalah dengan dipanaskan di atas nyala lilin, kemudian tinta yang sebelumnya tidak terlihat, ketika terkena panas akan berangsur-angsur menjadi gelap, sehingga pesan dapat dibaca. Teknik ini pernah juga digunakan pada Perang Dunia II.

Pada abad 20, steganografi benar-benar mengalami perkembangan. Selama berlangsung perang Boer, Lord Boden Powell (pendiri gerakan kepanduan) yang bertugas untuk membuat tanda posisi sasaran dari basis artileri tentara Boer, untuk alasan keamanan, menggambar peta-peta posisi musuh pada sayap kupu-kupu agar gambar-gambar peta sasaran tersebut terkamuflase. Perang Dunia II adalah periode pengembangan teknik-teknik baru steganografi. Pada awal Perang Dunia II, walaupun masih digunakan teknik tinta yang tak terlihat, namun teknik-teknik baru mulai dikembangkan. Contohnya seperti menulis pesan rahasia ke dalam kalimat lain yang tidak berhubungan langsung dengan isi pesan rahasia tersebut,

kemudian teknik menulis pesan rahasia ke dalam pita koreksi karbon mesin ketik, dan juga teknik menggunakan pin berlubang untuk menandai kalimat terpilih yang digunakan dalam pesan. Teknik terakhir yang ditemukan adalah microdots yang dikembangkan oleh tentara Jerman pada akhir Perang Dunia II.

Dari contoh-contoh steganografi konvensional tersebut dapat dilihat bahwa semua teknik steganografi konvensional berusaha merahasiakan komunikasi dengan cara menyembunyikan pesan ataupun mengkamuflase pesan. Maka sesungguhnya prinsip dasar dalam steganografi lebih dikonsentrasikan pada kerahasian komunikasinya, bukan pada datanya (Judge, 2001).

II.3. Sejarah Audio-Video Interleaved (AVI)

Audio-Video Interleaved, atau biasa dikenal dengan kata *AVI* adalah sebuah format multimedia yang diperkenalkan oleh *Microsoft* pada bulan November 1992 sebagai bagian dari format video untuk sistem operasi *Windows* (Wikipedia, 2007).

File *AVI* terdiri dari data audio dan video dalam format standart yang memiliki sinkronisasi antara audio dan video saat dijalankan. Seperti halnya DVD, file *AVI* juga mendukung *multiple streaming* audio dan video. Saat ini hampir sebagian besar file AVI menggunakan format *extensions* yang dikembangkan oleh grup Matrox OpenDML pada bulan februari 1996. Format ini didukung juga oleh *Mirosoft* dan dikenal dengan nama "*AVI 2.0*".

Data audio dan visual dari AVI file dapat di *encoded* atau di *decoded* menggunakan *software* yang bernama *codec* (kependekan dari *coder/encoder*). *Codec*

berfungsi sebagai penterjemah antara data *raw* dan format data yang digunakan. Dengan adanya codec, maka file avi dapat dijalankan ataupun dimanipulasi (memotong file (*cutter*), *extract frame* dan lain lain). Salah satu format *codec AVI* yang terkenal dan sering digunakan adalah *DivX (.divx extension*) yang dirilis pada bulan Juni, 2005, oleh *DivX,Inc* (Wikipedia, 2007).

II.4.Citra Bitmap (*bmp*)

Terjemahan bebas *bitmap* adalah pemetaan bit. Artinya, nilai intensitas piksel di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit yang umum adalah 8, artinya setiap piksel panjangnya 8 bit. Delapan bit ini merepresentasikan nilai intensitas piksel. Dengan demikian ada sebanyak 2^8 = 256 derajat keabuan, mulai 0 -255 (Munir, 2004).

Kedalaman Citra :

Citra Biner (monokrom) : Citra biner hanya mempunyai 2 nilai keabuan yaitu 0 dan 1, dimana 0 akan merepresentasikan warna hitam dan 1 akan merepresentasikan warna putih.

a. Citra *Grayscale* : Citra *Grayscale* atau sering disebut citra 8 bit atau 16 bit. Memberikan kemungkinan warna yang lebih banyak, karena ada nilai lain diantara 0 dan 1.
b. Citra 24 bit : Pada citra 24 bit setiap piksel memiliki komponen warna yang spesifik yang merupakan kombinasi 3 warna dasar yaitu Red (R), Green (G),Blue (B) sehingga sering disebut sebagai citra RGB. Setiap komponen warna tersebut masing masing mempunyai nilai intensitas tersendiri

dengan nilai 0 – 255. Jadi masing masing piksel bernilai 3 byte dengan masing masing byte menampung 8 bit.

II.5. Binary Digit

Binary digit adalah unit informasi terkecil yang ditangani oleh sebuah komputer. Satu bit diwujudkan sebagai 1 atau 0 dalam metode numeral, atau perwujudan dari *true-false* sebagai kondisi logika. Sebuah grup dengan 8 bit akan disebut 1 *byte*, dimana ini akan memberikan banyak tipe informasi, sebagai contoh surat yang ditulis dengan alfabet, digit desimal, atau karakter lain. *Binary digit* sering disingkat dengan bit.

II.6. ASCII (American Standard Code for Information Interchange)

American Standard Code for Information Interchange (*ASCII*) adalah format yang banyak digunakan untuk file teks di dalam dunia komputer dan internet. File *ASCII* terdiri dari karakter alphabetic, numeric, atau karakter khusus seperti Return, Tab Control dan sebagainya (Febrian,1999-2007).

Dalam *ASCII*, setiap karakter mempunyai angka yang digunakan oleh komputer atau printer untuk mewakili karakter tersebut. Sebagai contoh huruf kapital A diwakili kode 65. Walaupun didalam ASCII terdapat 256 kemungkinan karakter, *ASCII* hanya memerlukan 128 karakter standar, karena *ASCII* direpresentasikan ke dalam 7-bit bilangan biner (kumpulan dari nol atau satu sebanyak tujuh angka), dimana 32 karakter pertama

adalah “karakter control”. Karakter kontrol adalah karakter yang mengendalikan komputer dan tidak ditampilkan di layar. Angka selebihnya disediakan untuk menghasilkan huruf besar dan kecil, digit, dan berbagai tanda baca yang umum.

Pada tabel ASCII terlihat bahwa fungsi dari kode tersebut dialokasikan ke dalam bentuk matriks yang disusun berdasarkan kelompok-kelompok (nilainya diperlihatkan dalam heksadesimal) sebagai berikut:

1. 00-1F *Control Codes*
2. 20-3F *Punctuation and digits*
3. 40-5F *Uppercase*
4. 60-7F *Lowercase*

II.7.Metode Penyisipan Data

Metode penyisipan *hidden file* pada steganogarfi Video *AVI* ini terdiri dari 3 tahapan utama yaitu: mengekstrak *frame-frame* dari carrier file (hasilnya berupa bitmap), menyisipkan *hidden file* ke *frame-frame* tersebut, kemudian menyusun kembali *frame-frame* tersebut sehingga menjadi *stream video* yang baru.

II.7.1.Mengekstrak *Frame* dari *AVI* File

Langkah awal dimulai dengan membuka file *AVI* dan mendapatkan Video *Stream* (aliran *frame frame* yang beruntutan sehingga membentuk video) dari file AVI tersebut. Pada umumnya file AVI terdiri dari 4 macam *streaming* file dan biasanya hanya terdiri dari satu *stream* saja untuk setiap macam file didalam satu file AVI tersebut, yaitu Video, Audio, Midi, dan Teks (John, 2003). Dalam kasus ini hanya *video stream* dan *audio*

stream yang diolah sehingga hasil akhir dari proses ini berupa *video dan audio stream* saja.

Langkah awal yang dilakukan adalah dengan mengambil audio stream dari file AVI terlebih dahulu untuk digabungkan nantinya dengan video stream yang telah dikenai proses steganografi. Setelah mendapatkan video *stream*, maka langkah selanjutnya adalah membaca informasi dari *stream* tersebut yang meliputi posisi mula-mula (*start position frame*) dari *stream* tersebut, kemudian berapa banyak *frame* yang menyusun *stream* tersebut, dan kemudian *header information* yang meliputi panjang dan lebar dari *frame* yang menyusun *stream* tersebut, dan *frame rate* dari *stream* video tersebut.

Langkah selanjutnya untuk mengekstrak *frame* dari *stream* adalah dengan menentukan *pointer* dari *object frame* yang diinginkan dan *pointer* untuk *DIB (Device Independent Bitmaps)* dari *frame* yang di inginkan. DIB adalah format yang digunakan untuk menyimpan bitmap dalam format file bmp (http://www.herdsoft.com/ti/davincie/imex3j8i.htm).

Berdasarkan *header information* yang telah didapatkan maka langkah selanjutnya adalah melakukan *decompress* terhadap *frame* untuk menyimpannya sebagai file bitmap. Proses diawali dengan menyusun header information dari bitmap yang diperoleh sebelumnya dari *stream*, selanjutnya melakukan *decompress frame* yang diambil dari *stream* AVI file yang disimpan sementara dalam DIB, kemudian setelah semua informasi yang dibutuhkan (*bitmap info header*) didapatkan dan telah tersusun secara terstruktur maka *image* dan *header info* yang didapat, disimpan (*store*) didalam format Bitmap

yang meliputi *header, info*, dan *data bitmap*. Sampai langkah ini, *frame* yang terekstrak berformat bitmap.

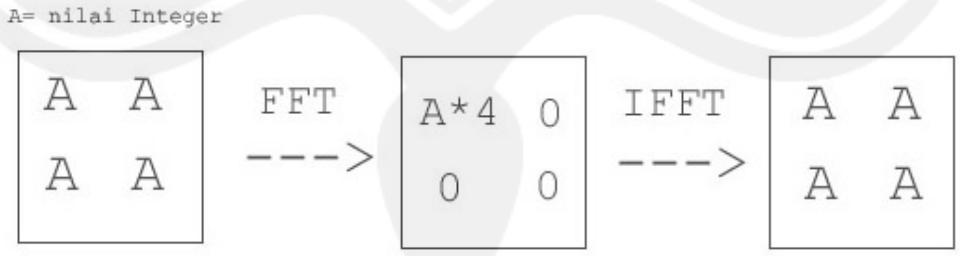
II.7.2.Menyisipkan *Hidden File* ke *frame frame* yang telah terekstrak dari *AVI* file

Proses penyisipan hidden file dilakukan secara bertahap dimana *frame* awal yang terekstrak akan tersimpan di *temporary folder* (sementara), untuk kemudian dibuka dan disisipkan sebagian data dari hidden file kedalamnya. Hidden file yang disisipkan berupa *character* dalam format *binary*, yaitu satu *character* untuk setiap frame. Setelah itu *frame* yang telah disisipi sebagian dari, akan disusun untuk membentuk *stream* video yang baru. Sedangkan *frame* selanjutnya akan disisipi *character* selanjutnya sampai semua character dapat tertampung, untuk kemudian disusun menjadi sebuah *stream* video yang baru.

Sedangkan data *frame* yang tersimpan di *temporary folder* akan terhapus ketika hendak menyisipkan data ke *frame* selajutnya. Jadi pada intinya penyisipan *hidden file* dilakukan satu persatu, dan untuk beralih ke *frame* selanjutnya, frame (bitmap) yang tersimpan di *temporary* harus dihapus terlebih dahulu. Jadi banyaknya *frame* yang digunakan akan menyesuaikan dengan besarnya *hidden data*, sehingga panjang *stream* video baru yang disisipi *hidden file* menyesuaikan dengan banyaknya data dari *hidden file*. Semakin besar *hidden file* maka *frame* dari *avi* file semakin banyak digunakan.

Metode penyisipan *hidden file* kedalam *frame* bitmaps menggunakan algoritma *Fast Fourier Transform* dimana data *hidden file* disisipkan pada bagian

frekuensi dari *frame*. Jadi terdapat mekanisme pemilihan lokasi penyisipan data pada nilai *piksel* dari suatu *frame*. Untuk setiap *frame* bitmap menampung 1 *character* yang di ubah kedalam format binary yang terdiri dari 7 nilai bit. Untuk setiap frame akan di ambil 7 blok matrik piksel berukuran 2 kali 2. Masing masing blok matrik terdiri dari nilai *RGB piksel* kawasan tersebut. Nilai *RGB* yang di ubah adalah nilai dari piksel warna *Blue*. Hal tersebut dikarenakan warna biru adalah warna yang sulit di deteksi oleh mata manusia secara umum (Dwiandiyanta, 2006). Untuk Metode mengubah nilai *blue* piksel ke nilai frekuensi, menggunakan penerapan dari algoritma *FFT (Fast Fourier Transform)*. Penyisipan data binary kedalam nilai matrik frekuensi pada dasarnya hanya terdiri dari 2 jenis nilai yaitu nilai 0 dan 1. Karena nilai *Blue* harus berformat integer (nilai piksel RGB dalam format integer) maka nilai matrik frekuensi saat di *inverse* ke nilai *Blue* tidak boleh mengandung nilai koma (*double*). Dengan adanya batasan tersebut maka penyisipan nilai 1 dan 0, menggunakan sifat dari matrik frekuensi *FFT* sebagai berikut:



Gambar 2.1 Sifat Matrik Frekuensi *FFT*

Dengan adanya sifat tersebut maka dapat dibuat 2 kondisi untuk nilai 1 dan 0 sebagai berikut:

1. Apa bila inputan bernilai 1 maka nilai untuk masing masing komponen matrik frekuensi 2 x 2 tidak boleh dalam kondisi seperti diatas, yaitu nilai selain matrik[0,0] sama dengan 0. Jika tidak terpenuhi maka nilai frekuensi matrik harus diacak agar tidak membentuk pola tersebut. Hasil *inverse* dari matrik ini akan menghasilkan nilai berbeda untuk masing masing komponen.
2. Apa bila inputan bernilai 0 maka nilai untuk masing masing komponen matrik frekuensi 2 x 2 harus dalam kondisi seperti diatas, yaitu nilai selain matrik[0,0] sama dengan 0. Jika tidak terpenuhi maka nilai frekuensi matrik harus diset sama dengan 0 untuk koordinat [0,1],[1,0] dan [1,1]. Hasil *inverse* dari matrik ini akan menghasilkan nilai sama untuk masing masing komponen matrik.

II.7.3.Menyusun *frame* menjadi *stream video* baru

Untuk menyusun kembali *frame* yang telah terbentuk diperlukan header info dari bitmap meliputi *countFrames, stride, width, height* dan lain lain untuk membentuk *stream* video yang baru (John, 2003). Setelah *header info* terpenuhi langkah selanjutnya adalah dengan memasukkan *bitmap (frame)* yang ada untuk disusun kembali menjadi *stream* video yang baru. Selanjutnya stream video yang terbentuk digabung (*merge*) dengan audio stream yang telah di ambil di langkah awal sehingga hasil akhirnya berupa video dan audio stream.

II.8. Sekilas mengenai Transformasi Fourier

Pengolahan citra tidak hanya dapat dilakukan pada kawasan ruang / spasial saja, namun pengolahan citra dapat dilakukan dalam kawasan yang lain, misalnya kawasan frekuensi (*frequency domain*) (Dwiandiyanta,2006).

Transformasi citra sendiri adalah proses mengubah suatu citra dari suatu kawasan tertentu ke kawasan yang lainnya. Transformasi citra, sesuai namanya, merupakan proses perubahan bentuk citra untuk mendapatkan suatu informasi tertentu. Pada umumnya Transformasi bisa dibagi menjadi 2 yaitu Transformasi piksel (geometris) dan Transformasi ruang (domain / *space*).

Transformasi piksel masih bermain di wilayah ruang atau domain yang sama (domain spasial), hanya posisi piksel yang kadang diubah. Contoh implementasi nya antara lain adalah rotasi, translasi, *scaling, invers, shear*, dan masih banyak lagi.Transformasi jenis ini relatif mudah diimplementasikan dan banyak aplikasi yang dapat melakukannya.

Sedangkan Transformasi ruang merupakan proses perubahan citra dari suatu ruang/domain ke ruang/domain lainnya. Contohnya adalah dari ruang spasial ke ruang frekuensi. Banyak jenis transformasi yang dapat digunakan dalam transformasi ruang antara lain Transformasi Fourier, Transformasi DCT, dan Transformasi *Wavelet.*

Transformasi Fourier pada dasarnya akan mengubah suatu citra menjadi jumlahan dari komponen eksponensial bilangan kompleks yang saling berbeda magnitude, frekuensi dan fase.

Dibawah ini adalah bentuk persamaan dari fungsi *Fourier Transform* 2 dimensi:

$$F(p,q) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n) e^{-j(2\pi/M)pm} e^{-j(2\pi/N)qn}$$

Untuk menghitung nilai e, digunakan persamaan Euler :

$$e^{-jx} = \cos x - j\sin x$$

$$e^{jx} = \cos x + j\sin x$$

Hasil penghitungan *Fourier Transform* biasanya mengandung bilangan real dan imajiner. *Fourier Spectrum* didapatkan dari magnitude kedua bilangan tersebut sehingga didapatkan persamaan sebagai berikut:

$$|F(u)| = [R\ 2(u) + I\ 2(u)]1/2$$

Dibawah ini adalah bentuk persamaan dari fungsi *invers Fourier Transform* 2 dimensi:

$$f(m,n) = \frac{1}{MN} \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} F(p,q) e^{j(2\pi/M)pm} e^{j(2\pi/N)qn}$$

Untuk mendapatkan nilai *invers Fourier Transform,* terlebih dahulu harus ditentukan nilai sudut */ teta* (θ). Nilai sudut ini digunakan dalam penentuan nilai *Real* dan *Imaginer* baru untuk proses IFFT. Nilai sudut */ teta* (θ) tersebut digunakan untuk menentukan sudut yang terbentuk dari nilai *Real* dan *Imaginer* awal. Tujuannya agar nilai *real* dan *imaginer* baru berada pada sudut dan kuadran yang sama dengan nilai awal, sehingga

nilai balikan / *invers* FFT sesuai dengan nilai awalnya. Perhitungan sudut dapat ditemukan dengan persamaan :

$$\theta(q) = arctg \frac{\mathrm{Im}(q)}{\mathrm{Re}(q)}$$

Setelah didapat nilai θ, nilai tersebut harus dikonversi berdasarkan kuadrannya. Kuadran-kuadran yang ada, yaitu :

1. Kuadran I :
   Kondisi ini terpenuhi jika nilai *real* dan *imaginer* positif.
   Nilainya : $\theta = 0 - 90^o$
2. Kuadran II :
   Kondisi ini terpenuhi jika nilai *real* negatif dan nilai *imaginer* positif.
   Nilainya : $\theta = 180^o - \theta$
3. Kuadran III :
   Kondisi ini terpenuhi jika nilai *real* dan nilai *imaginer* negatif.
   Nilainya : $\theta = 180^o + \theta$
4. Kuadran IV :
   Kondisi ini terpenuhi jika nilai *real* positif dan nilai imaginer *negatif*.
   Nilainya : $\theta = -\theta$

Pada IFFT dibutuhkan nilai *real* dan *imaginer* baru yang didapat dari persamaan :

- $\mathrm{Re}(q) = F(q)\cos\theta(q)$
- $\mathrm{Im}(q) = F(q)\sin\theta(q)$

II.9 Tinjauan Pustaka

Tinjauan pustaka dalam penelitian ini meliputi dua kelompok pembahasan. Pembahasan pertama merupakan tinjauan singkat mengenai penerapan steganografi diberbagai media baik digital maupun *non*-digital. Pembahasan kedua berkaitan dengan penerapan algoritma algoritma steganografi pada file AVI Video yang salah satunya adalah penerapan FFT (Fast Fourier Transform).

Definisi Steganografi menurut situs http://mdesian.tblog.com/post/24258 adalah sebuah teknik penyembunyian data rahasia ke dalam sebuah wadah (media / *carrier file*) sehingga data yang disembunyikan sulit dikenali oleh indera manusia. Definisi serupa juga diberikan oleh judge (2001), bahwa Steganografi merupakan seni untuk menyembunyikan pesan di dalam pesan lainnya sehingga orang lain tidak menyadari bahwa ada sesuatu di dalam pesan tersebut. Teknik ini meliputi banyak sekali metoda komunikasi untuk menyembunyikan pesan rahasia.

Seperti yang ditulis oleh Sukmawan (2002), metoda ini meliputi tinta yang tidak tampak, microdots, pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi spektrum lebar. Sedangkan didalam steganografi digital terdapat berbagai media yang dapat digunakan seperti teks *file*, *image file*, *video* dan *audio file*.

Banyak sekali algoritma yang dapat digunakan didalam teknik steganografi digital. Algoritma tersebut seperti yang dirangkum oleh Henry (2006) meliputi *Least Significant Bit (LSB), Masking and Filter*,dan *transformation*.

Untuk File Video maka algoritma yang paling tepat digunakan adalah *transformation* baik itu *Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT)*, maupun *Wafelet Compression*. Alasan penggunaan algoritma *transformation* seperti diungkapkan oleh henry (2006) adalah dikarenakan algoritma lain seperti modifikasi LSB memiliki kelemahan yaitu akan mengasilkan stego yang berukuran sangat besar sedangkan metode *masking* dan *filtering* akan mengubah tampilan visual dari video secara langsung. Hal senada juga diungkapkan oleh Soehono (2006), bahwa algoritma *transformation* memiliki kelebihan yaitu hanya mengubah sedikit gambar dari beberapa frame Video, sehingga perubahan yang terjadi tidak dirasakan. Perubahan yang dilakukan pun hanya pada daerah berfrekuensi tinggi.