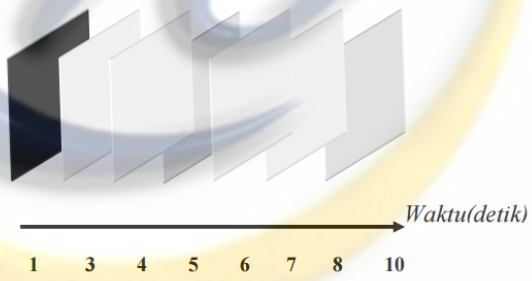
BAB II
LANDASAN TEORI

Dalam bab ini diuraikan dasar-dasar teori yang mendukung pelaksanan Tugas Akhir ini, yaitu tentang Video, *AVI*, dan Steganografi.

2.1 Video

Video merupakan gabungan gambar-gambar yang tidak bergerak dibaca berurutan dalam suatu waktu dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut dengan *frame* rate, dengan satuan fps (*frame per second*). Karena dimainkan dalam kecepatan yang tinggi maka tercipta ilusi gerak yang halus, semakin besar nilai *frame rate* maka akan semakin halus pergerakan yang ditampilkan.



Gambar 2.1 Aliran *Frame*

Berdasarkan media yang digunakan, video dapat dibagi menjadi dua macam, yaitu video analog dan video digital [6].

2.1.1 Video Digital

Video digital adalah video yang menggunakan media digital. Sumber informasi *audio* dan video berasal dari *CD audio*, *DV*, *miniDV*, *Digital8 camcorders*, dan *DVD* yang menyimpan audio dan video di dalam format digital.

Data video digital dapat diproses secara langsung oleh komputer, dan digandakan tanpa mengurangi kualitas [6].

2.1.1.1 Sinyal Video Digital

Sebuah sinyal video digital merupakan kombinasi dari tingkat keterangan (*luminance* atau *luma*) dan warna (*chrominance* atau *chroma*). Tingkat keterangan (*luminance*) adalah intensitas terang-gelapnya cahaya pada sinyal video, biasanya direpresentasikan dengan huruf *Y*. Sinyal video dipisahkan menjadi komponen *luma* dan *chroma* (warna) untuk kualitas yang lebih besar serta transmisi dan pengkodean yang efisien. Sinyal video disajikan dalam format *YUV*, di mana informasi warna direpresentasikan dalam *U* dan *V* [6].

Sebuah sinyal video digital dibagi menjadi tiga sinyal terpisah untuk mencegah penurunan kualitas dari perpaduan sinyal-sinyal. Komponen sinyal ini terdiri atas *RGB* (*red, green,* dan *blue*), *luma* (*Y*), dan dua sinyal *chroma*, sebagai contoh *Y*, *Y-R, Y-B*, atau dalam format lain dituliskan sebagai *YUV, YCbCr*, dan *Y Pr Pb* [6].

2.1.1.2 Format Video Digital[3]

Video digital dapat disimpan dalam beberapa format yang berbeda. Tiap format dari video digital memiliki karakteristik masing-masing, kelebihan dan kekurangan, serta tujuan penggunaan yang berbeda-beda.

Berikut beberapa format video digital yang umum digunakan:

1. *AVI* (*Audio Video Interleave*)

   Merupakan format video digital yang paling banyak digunakan pada *Personal Computer* dan di internet. Data disimpan di dalam blok-blok untuk mengurangi ukuran *file*, beberapa kompresor dapat mengkompresi *file* dengan format ini hingga mencapai perbandingan 100:1. *File* dengan format

ini tidak praktis jika dikirimkan melalui internet karena ukurannya yang sangat besar.

2. *DV* (*Digital Video*) dan *Mini-DV*

Merupakan format video digital kompresi untuk keperluan video profesional. Format kompresi *DV* digunakan untuk *DV* dan *Digital-8 camcorder*. Video dan audio berformat *DV* dapat di-*capture* menggunakan antarmuka *FireWire* / *IEEE* 1394, kemudian disimpan dan diedit dengan *editor* video.

3. *MPEG* (*Moving Picture Expert Groups*)

Merupakan format *file* multimedia yang populer yang ditetapkan oleh sebuah grup bernama *Moving Pictures Expert Group*. *MPEG* telah memproduksi berbagai macam standard yang masing-masing memiliki aplikasi berbeda.

4. *Quick Time* (*.mov)

File dengan format ini hanya dapat dijalankan dengan perangkat *QuickTime Player*, yang merupakan perangkat multimedia yang menawarkan kreativitas dan fleksibilitas, serta memberikan kulitas terbaik. Dapat menampilakn 99 *track* audio, video, 3D, teks, *HTML*, dan *VR*, serta dapat disisipkan dengan mudah di halaman web.

5. *Real Media* (*.rm) dan *Real Video* (*.rv)

File dengan format ini mudah di-*streaming* atau di-*download* melalui internet, memiliki kecepatan kompresi yang tinggi tetapi dapat menurunkan kualitas.

6. *Flash* dan *Shockwave* (*.swf)

File dengan format ini dibuat dengan menggunakan *Macromedia Flash* atau *Macromedia Director*, dan ditampilkan dengan perangkat

Macromedia Flash Player atau *Shockwave Player*. Dapat diintegrasikan ke dalam halaman web atau diputar sebagai film seperti pada format lainnya.

7. *WMV* (*Windows Media Video*)

   Merupakan kompresi *file* video/audio dengan Microsoft Windows Media *codec*.

2.1.2 *Frame Rate*

Ketika serangkaian gambar mati yang bersambung dilihat oleh mata manusia, maka suatu keajaiban terjadi. Jika gambar-gambar tersebut dimainkan dengan cepat maka akan terlihat sebuah pergerakan yang halus, inilah prinsip dasar film, video dan animasi. Jumlah gambar yang terlihat setiap detik disebut dengan frame rate. Diperlukan *frame rate* minimal sebesar 10 fps (*frame rate per second*) untuk menghasilkan gambar pergerakan yang halus. Film-film yang kita lihat di gedung bioskop adalah film yang diproyeksikan dengan *frame rate* sebesar 24 fps, sedangkan video yang kita lihat di televisi kira-kira memiliki *frame rate* sebesar 30 fps (tepatnya 29.97 fps) untuk negara yang memakai format standar NTSC (*National Television Standards Comitte*) yaitu Amerika Serikat, Jepang, Kanada, Meksiko dan Korea. Untuk negara Indonesia, Inggris, Australia, Eropa dan China format video standar yang digunakan adalah format PAL (*Phase Alternate Line*) dengan *frame rate* sebesar 25 fps. Sedangkan negara Perancis, Timur Tengah dan Afrika menggunakan format video standar SECAM (*Sequential Couleur Avec Memoire*) dengan *frame rate* sebesar 25 fps [3].

2.1.3 *Pixel Ratio*

Pixel aspect ratio menjelaskan tentang rasio perbandingan lebar dengan tinggi dari sebuah *pixel* dalam sebuah gambar. *Frame aspect ratio* menggambarkan perbandingan lebar dengan tinggi pada dimensi frame dari sebuah gambar. Sebagai contoh, D1 NTSC memiliki *pixel aspect ratio* 0.9 (0.9 lebar dari 1 unit tinggi) dan

memiliki pula *frame aspect ratio* 4:3 (4 unit lebar dari 3 unit tinggi). Beberapa format video keluaran menggunakan *frame aspect ratio* yang sama tetapi memakai *pixel aspect ratio* yang berbeda. Sebagai contoh, beberapa format NTSC digital menghasilkan sebuah 4:3 *frame aspect ratio*, dengan square pixel (1.0 *pixel aspect ratio*) dan dengan resolusi 640 X 480. Sedangkan D1 NTSC menghasilkan *frame aspect ratio* yang sama yaitu 4:3 tetapi menggunakan *rectangular pixel* (0.9 *pixel aspect ratio*) dengan resolusi 720 X 486. *Pixel* yang dihasilkan oleh format D1 akan selalu bersifat *rectangular*/bidang persegi, akan berorientasi *vertical* dalam format NTSC, dan akan berorientasi *horizontal* dalam format PAL. Jika kita menampilkan *rectangular pixel* dalam sebuah monitor *square pixel* tanpa alterasi maka gambar yang bergerak akan berubah bentuk atau mengalami distorsi. Contohnya lingkaran akan berubah menjadi oval. Tetapi bagaimanapun juga apabila ditampilkan pada *monitor broadcast* gambar gerak akan ditampilkan secara benar [6].

2.1.4 *Bit Depth*

Dalam dunia komputer, satuan *bit* merupakan unit terkecil dalam penyimpanan informasi. *Bit depth* menyatakan jumlah/banyaknya *bit* yang disimpan untuk mendeskripsikan warna suatu *pixel*. Sebuah gambar yang memiliki 8 *bit* per *pixel* dapat menampilkan 256 warna, sedangkan gambar dengan 24 *bit* dapat menampilkan warna sebanyak 16 juta warna. Komputer (PC) menggunakan 24 *bit* RGB (*Red Green Blue*) sedang sinyal video menggunakan standar 16 *bit* YUV sehingga memiliki jangkauan warna yang terbatas. Untuk itu kita perlu berhati-hati apabila membuat video untuk ditayangkan di TV, karena tampilan warna di layar monitor PC berbeda dengan tampilan di layar TV [6].

2.1.5 *Bit Rate*

Bit rate disebut juga dengan nama *data rate*. *Bit rate* menentukan jumlah data yang ditampilkan saat video dimainkan. Data rate ini dinyatakan dalam satuan bps (*bit per second*). *Data rate* berkaitan erat dengan pemakaian dan pemilihan

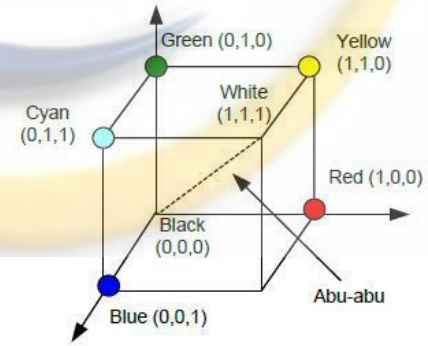
codec (metode kompresi video). Beberapa *codec* menghendaki data rate tertentu, misalnya MPEG-2 yang digunakan dalam format DVD dapat menggunakan *bit rate* maksimum 9800 kbps atau 9,8 Mbps, sedangkan format VCD hanya mampu menggunakan *bit rate* 1,15 Mbps[6].

2.1.6 Representasi Warna[3]

Pada video digital data dipisahkan menjadi beberapa komponen warna (*crominance*) dan komponen kecerahan, dimana pemisahan tiap komponen menggunakan cara-cara tertentu, beberapa cara pemisahan komponen tersebut adalah RGB, YUV, danYIQ.

1. RGB (*Red, Green, Blue*)

   Pada RGB 24 *bit*, data video dipisahkan ke dalam komponen untuk masing-masing warna merah (*red*), hijau (*green*), biru (*blue*) yang masing-masing memiliki nilai 8 *bit* atau 256 level.



Gambar 2.2 Koordinat RGB

Misalnya citra dengan 8 *bit* per piksel mempunyai 256 warna dan citra dengan 24 bit mempunyai ±16,8 juta warna, jadi tiap piksel dinyatakan dengan:

a. *Bit* 0 sampai dengan 7 untuk warna merah.
b. *Bit* 7 sampai dengan 15 untuk warna hijau
c. *Bit* 16 sampai dengan 24 untuk warna biru

Kemungkinan kombinasi warna yang ada adalah = 2563 + 2562 + 2561 = 16.843.008, dimana nilai 0 menyatkan warna hitam sedangakan nilai 16 843 008 menyatakan warna putih.

2. YUV (*Yellow Ultra Violet*)

Pemisahan komponen tidak hanya pada warna (*crominance*) tetapi juga terhadap kecerahan (*luminance*) pada format PAL, sinyal kecerahan dinyatakan dalam Y dan masing-masing komponen diperoleh dengan mentransformasikan RGB dengan rumus:

Y = 0,299 R +0,587 G + 0,114 B

U= (B-Y) x 0,493

V= (R-Y) x 0,877

3. YIQ (*Yellow Irradiation Quart*)

Pemisahan sinyal video dapat dilakukan juga dalam format NTSC di mana komponen luminance dinyatakan dalam Y dan komponen crominance dinyatakan dalam I dan Q dengan rumus:

Y = 0,299 R +0,587 G + 0,114 B

I= 0,596 R - 0,275 G - 0,321 B

Q= 0,212 R – 0,523 G – 0,311 B

2.2 AVI

Audio Video Interleave, atau disingkat *AVI*, adalah format *file* multimedia yang diperkenalkan oleh Microsoft pada bulan November 1992. *File* AVI dapat mengandung data *audio* dan video dalam suatu media, yang memungkinkan *audio* dan video dimainkan bersamaan. Seperti *DVD*, *file AVI* mendukung streaming, baik *audio* dan video, walaupun jarang dilakukan. Hampir semua *file* AVI menggunakan

format ekstensi *file* (.AVI). *File-file* ini didukung oleh Microsoft, dan disebut "AVI 2.0" [2].

AVI merupakan kasus khusus dari *Resource Interchange File Format* (*RIFF*), yang membagi *file* data ke dalam blok-blok. Setiap blok diidentifikasi dengan *tag FourCC*. Sebuah *file AVI* mengambil format blok tunggal di dalam *file RIFF*, yang kemudian dibagi menjadi dua blok perintah dan satu blok opsional. Struktur dari sebuah *file RIFF* rupanya menyalin dari format *IFF* sebelumnya yang ditemukan oleh *Electronic Arts* pada pertengahan 1980-an [2].

Bagian pertama blok pada *AVI* diidentifikasi dengan *tag* "hdrl". Blok ini merupakan *file header* dan mengandung metadata tentang video, seperti lebar, tinggi, dan laju *frame*. Bagian kedua blok diidentifikasi dengan *tag* "movi". Blok ini mengandung data audio/visual yang membangun film *AVI*. Blok opsional ketiga diidentifiki dengan *tag* "idx1" yang menunjukkan indeks alamat-alamat fisik blok data [2].

2.3 BMP

BMP atau *DIB (device-independent bitmap)* adalah sebuah format grafik *bitmap* yang digunakan secara internal oleh Microsoft Windows dan subsistem grafik OS/2, dan sering digunakan sebagai sebuah format *file* grafik sederhana pada *platform-platform* tersebut [3].

Gambar secara umum disajikan dalam ketajaman warna 2 (1-*bit*), 16 (4-*bit*), 256 (8-*bit*), 65,536 (16-*bit*), atau 16.7 juta (24-*bit*) warna (*bit-bit* ini merepresentasi *bit-bit* per *pixel*). Sebuah gambar 8-*bit* juga dapat diubah ke warna *grayscale* di samping warna index. Sebuah *channel alpha* (untuk warna transparan) boleh disajikan dalam *file* terpisah, di mana sama dengan gambar *grayscale*. Versi 32-bit dengan *channel alpha* terintegrasi telah diperkenalkan oleh Windows XP dan digunakan untuk sistem *logon* dan *theme*.

File-file BMP biasanya tidak terkompresi, sehingga dapat berukuran lebih besar daripada format *file* gambar yang terkompresi untuk gambar yang sama. Sebagai contoh, sebuah gambar 24-*bit* 800×600 *pixel* akan memiliki ukuran *file* hampir 1.4 *megabytes*. Sebagai bandingan, sebuah gambar 1058×1058 *pixel* memiliki ukuran *file* 477.6 KB dalam format PNG, setara dengan 3.2 MB dalam *file* BMP 24-*bit*.

Umumnya *file* BMP menggunakan model warna RGB. Pada model ini, sebuah warna terbentuk dari campuran intensitas yang berbeda (bervariasi dari 0 sampai 255) warna merah (R), hijau (G), dan biru (B). Dengan kata lain, sebuah warna akan didefinisikan menggunakan 3 nilai, yaitu R, G dan B.

Blok dari *bit-bit* mendeskripsikan gambar secara *pixel* per *pixel*. *Pixel* disajikan mulai dari sudut kiri bawah berjalan dari kiri ke kanan dan kemudian baris per baris dari bawah ke atas. Setiap *pixel* dideskripsikan menggunakan satu atau lebih *bit* [7].

2.4 Data *Text*

Text merupakan sekumpulan karakter terdiri dari huruf-huruf, angka-angka (A-Z, a-z, 0-9), dan simbol-simbol lainnya seperti %, &, ^, =, @, £, $, ! *, dan lain-lain, dengan menggunakan kode ASCII setiap karakter dari *text* berjumlah 8 *bit* atau 1 *byte* [6].

2.5 Perhitungan MSE[8]

Kualitas media penampung setelah ditambahkan pesan rahasia tidak jauh berbeda dengan kualitas media penampung sebelum ditambahkan pesan. Setelah penambahan pesan rahasia, kualitas video penampung tidak jauh berubah, masih terlihat dengan baik. Sehingga pengamat tidak mengetahui kalau di dalam video tersebut terdapat pesan rahasia. Untuk mengukur kualitas video steganografi

diperlukan suatu pengujian secara obyektif. Pengujian secara objektif adalah dilakukan dengan menghitung nilai PSNR.

MSE (*Mean Square Error*) adalah nilai error kuadrat rata-rata antara video asli dengan video manipulasi (dalam kasus steganografi; MSE adalah nilai error kuadrat rata-rata antara video asli (*cover-video*) dengan video hasil penyisipan (stego-video). Pada penelitian ini, MSE digunakan untuk mengetahui perbandingan kualitas video cover sebelum dan sesudah disisipkan pesan. Untuk menentukan PSNR, terlebih dahulu harus ditentukan MSE (*Mean Square Error*), secara matematis dapat dirumuskan sebagai berikut:

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

Dimana:

MSE = Nilai Mean Square Error citra steganografi

M = Panjang citra stego (dalam pixel) N = Lebar citra stego (dalam pixel)

1(x,y) = nilai piksel dari citra cover 1’(x,y) = nilai piksel pada citra stego

Setelah diperoleh nilai MSE maka nilai PSNR dapat dihitung dari kuadrat nilai maksimum dibagi dengan MSE. Secara matematis, nilai PSNR dirumuskan sebagai berikut:

$$PNSR = 10.\log\left(\frac{MAXi^2}{MSE}\right)$$

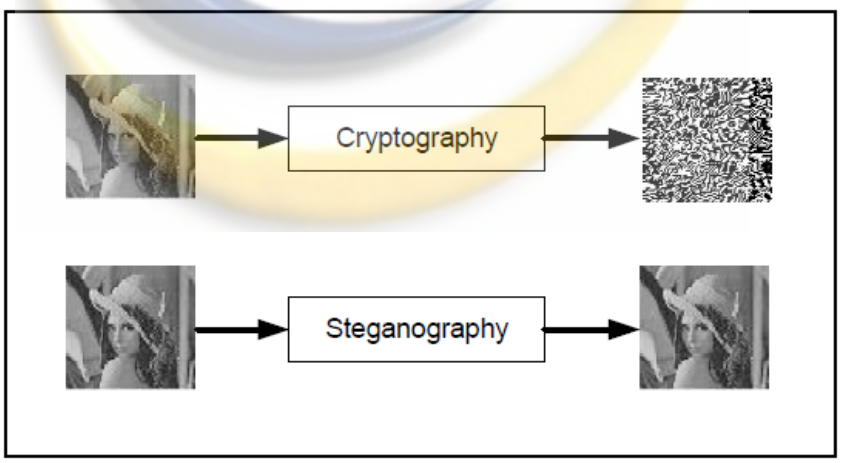
Dimana:

MSE = nilai MSE, MAXi = nilai maksimum dari pixel citra yang digunakan semakin rendah nilai MSE maka akan semakin baik, dan semakin besar nilai PSNR maka semakin baik kualitas video steganografi.

2.6 Steganografi

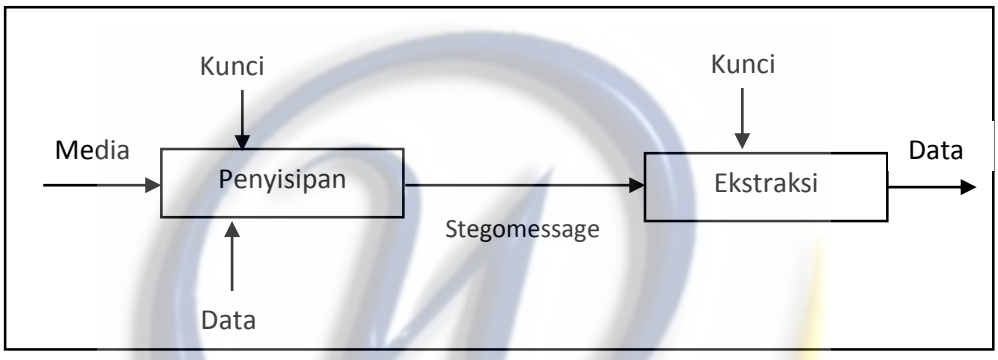
Steganografi (*steganography*) adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan (*eksistensi*) pesan tidak terdeteksi oleh indera manusia.

Steganografi berbeda dengan kriptografi, di mana pihak ketiga dapat mendeteksi adanya data (*chipertext*), karena hasil dari kriptografi berupa data yang berbeda dari bentuk aslinya dan biasanya datanya seolah-olah berantakan, tetapi dapat dikembalikan ke bentuk semula. Steganografi membahas bagaimana sebuah pesan dapat disisipkan ke dalam sebuah *file* media sehingga pihak ketiga tidak menyadarinya. Steganografi memanfaatkan kekurangan sistem indera manusia seperti mata dan telinga. Dengan adanya kekurangan inilah, metoda steganografi ini dapat diterapkan pada berbagai media digital. Hasil keluaran dari steganografi ini memiliki bentuk persepsi yang sama dengan bentuk aslinya, tentunya persepsi di sini oleh indera manusia, tetapi tidak oleh komputer atau perangkat pengolah digital lainnya. Ilustrasi mengenai perbedaan kriptografi dan steganografi dapat dilihat pada gambar 2.3[3].



Gambar 2.3 Ilustrasi Kriptografi dan Steganografi pada Citra Digital

Steganografi membutuhkan dua properti, yaitu media penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks, dan video. Sedangkan data rahasia yang disembunyikan dapat berupa *file* apapun. Media yang telah disisipi data disebut *stegomessage*. Proses penyembunyian data ke dalam media disebut penyisipan (*embedding*), sedangkan proses sebaliknya disebut ekstraksi. Proses tersebut dapat dilihat pada gambar 2.4. Penambahan kunci yang bersifat opsional dimaksudkan untuk lebih meningkatkan keamanan [3].



Gambar 2.4 Proses Penyisipan dan Ekstraksi dalam Steganografi

Steganografi agak berbeda dengan *watermarking*. *Watermarking* merupakan aplikasi dari steganografi. Jika pada steganografi informasi rahasia disembunyikan dalam media digital di mana media digital tidak berarti apa-apa, maka pada *watermarking* justru media digital tersebut yang akan dilindungi kepemilikannya dengan pemberian label hak cipta (*watermark*). Meskipun steganografi dan *watermarking* tidak sama, namun secara prinsip proses penyisipan informasi ke dalam data digital tidak jauh berbeda [3].

2.6.1 Steganografi pada Video

Seperti dikatakan sebelumnnya, video merupakan kumpulan dari *image* yang "bergerak", jadi sebagian besar metode yang digunakan pada *image* steganografi dapat digunakan pada video steganografi. Dapat dikatakan bahwa video steganografi merupakan turunan dari *image* steganografi. Pada video steganografi ini, menggunakan teknik *Random Least Significant Bit* [4].

Dalam pembuatan program ini digunakan data kualitatif yang berkaitan dengan format video yang digunakan. Dimana format video yang digunakan di ubah menjadi *frame-frame* berbentuk BMP dengan tujuan agar hasil yang tercapai bisa maksimal. *File-file* BMP biasanya tidak terkompresi, maka data dalam gambar tidak mengalami kerusakan pada saat video di kompresi sehingga dapat lebih muda di sisipkan oleh pesan. Data dalam format BMP mempunyai karakteristik warna RGB yang bernilai 24-*bit*, maka dari itu dilakukan proses pembuatan atau pengkonversian dari video ke frame BMP dengan banyak warna atau RGB, warna merah (R), hijau (G), dan biru (B).

Teknik LSB mengubah bit-bit pada *frame image cover video* secara *linier* sesuai dengan urutan warna komponen dan urutan *pixel*, dengan begitu *attacker/cracker* dapat dengan mudah mengeluarkan kembali pesan rahasia yang telah disisipkan pada video. Untuk mencegah hal-hal tersebut maka digunakan kata kunci dan penempatan *bit-bit* yang *random* pada video [4].

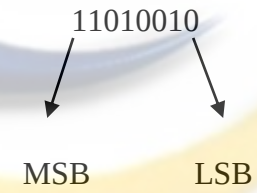
Pseudorandom Number Generator (PRNG) digunakan untuk membangkitkan suatu bilangan *random* yang akan digunakan dalam urutan penyisipan *bit-bit* pada video. *Pseudo random* menggunakan bilangan (*seed*) untuk dapat membangkitkan bilangan *random*, karena itu urutan bilangan *random* yang sama akan dihasilkan bila bilangan (*seed*) yang dimasukkan juga sama. Dengan penggunaan *pseudo random* ini maka penyisipan bit-bit pesan akan tersebar pada komponen warna maupun pada pemilihan *pixel* [4].

Untuk dapat menghasilkan bilangan (*seed*) yang akan dimasukkan pada *pseudo random*, digunakan pengisian kata kunci yang nantinya akan diambil *hash code* yang dihasilkan oleh algoritma MD5 atau SHA- 1. Dalam prosesnya kata kunci yang dimasukkan akan membangkitkan suatu *byte* yang diubah kedalam bentuk string yang kemudian diambil 16 *byte* karakter string untuk membangkitkan suatu bilangan yang akan menjadi bilangan (*seed*) yang dimasukkan pada *pseudo random*. Karena *pseudo random* dapat membangkitkan urutan bilangan *random* yang sama bila bilangan (*seed*) yang dimasukkan juga sama, maka untuk dapat mengeluarkan pesan rahasia diperlukan kata kunci yang sama. Penggunaan *pseudo*

random dan *hash code* memiliki keuntungan bahwa kata kunci yang dimasukkan tidak perlu dimasukkan dalam penyisipan *bit-bit* video cover sehingga kapasitas pada video *cover* tidak berkurang [4].

2.6.2 *Least Significant Bit (LSB)*

Teknik Steganografi Modifikasi LSB dilakukan dengan memodifikasi *bit-bit* yang termasuk bit LSB pada setiap *byte* warna pada sebuah *pixel*. *Bit-bit* LSB ini akan dimodifikasi dengan menggantikan setiap LSB yang ada dengan *bit-bit* informasi lain yang ingin disembunyikan. Setelah semua *bit* informasi lain menggantikan *bit* LSB di dalam *file* tersebut, maka informasi telah berhasil disembunyikan. Ketika informasi rahasia tersebut ingin kembali dibuka, maka *bit-bit* LSB yang sekarang ada, diambil satu per satu kemudian disatukan kembali menjadi sebuah informasi yang utuh seperti semula. Penentuan *bit-bit* LSB dilakukan secara berurutan, mulai dari *byte* awal sampai *byte* terakhir sesuai panjang dari data rahasia yang akan disembunyikan [2].



Gambar 2.5 MSB dan LSB

Mengubah *bit* LSB hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya tidak berpengaruh terhadap persepsi *visual/auditori*. Contoh Penggunaan Metode LSB pada tahap *encode:*

1. Misalkan penyisipan pada citra 24-*bit*. Setiap *pixel* panjangnya 24 bit (3 *byte*, masing-masing komponen R (1 *byte*), G (1 *byte*), dan B (1 *byte*)).
   00110011 10100010 11100010 (misal *pixel* berwarna merah)
   Misalkan *embedded message*: 010
   Encoding: 00110010 10100011 11100010

(*Pixel* berwarna "merah berubah sedikit", tidak dapat dibedakan secara visual dengan citra aslinya).

2. Jika pesan = 10 *bit*, maka jumlah *byte* yang digunakan = 10 *byte*
   00110011 10100010 11100010 10101011 00100110
   10010110 11001001 11111001 10001000 10100011

   Pesan: 1110010111
   Hasil penyisipan pada *bit* LSB:
   00110011 10100011 11100011 10101010 00100110
   10010111 11001000 11111001 10001001 10100011

Ada 2 macam teknik *Least Significant Bit*, yaitu *Extended Least Significant Bit* dan *Random Least Significant Bit*. *Extended Least Significant Bit* dan *Random Least Significant Bit* merupakan variasi dari teknik LSB dimana *Extended* LSB dapat menyisipkan pesan dalam gambar *cover* dengan besar (ukuran *file*) yang tidak terbatas sedangkan *Random* LSB menggunakan penyisipan pesan yang dilakukan secara acak berdasarkan kata kunci yang digunakan [4].

Contoh Penggunaan teknik *Random LSB* pada tahap *encode:*

1. Misalkan *bit* pada citra cover dengan ukuran 5 *pixel* sebagai berikut:
   (00011111 11101001 11001000)
   (00011111 11001000 11101011)
   (11100010 00100111 11101010)
   (11100001 00100110 11101001)
   (11100000 00100101 11101000)

   Pesan yang akan disisipkan adalah karakter 'A' yang memiliki *biner* **1000001**, maka citra stego yang akan dihasilkan adalah:

(00011111**1** 1110100**0** 1100100**0**)

(00011111 11001000 11101011)

(1110000**0** 0010011**0** 1110101**0**)

(11100001 00100110 11101001)

(1110000**1** 00100101 11101000)

2.6.3 Kriteria Steganografi yang Bagus

Penyembunyian data ke dalam *file* video digital akan mengubah kualitas *file* video digital tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data adalah:

a. Mutu video penampung tidak jauh berubah. Setelah penambahan data rahasia, video hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam video tersebut terdapat data rahasia.

b. Data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada video penampung. Bila pada video dilakukan operasi pengolahan video, maka data yang disembunyikan tidak rusak.

c. Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*) [4].

2.7 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti rahasia dan *graphia* berarti tulisan. Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain [5].

Dalam kriptografi kita akan sering menemukan baerbagai istilah yaitu:

a. Pesan, *plaintext,* dan *ciphertext*

   Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext*, agar pesan tidak dapat dimengerti oleh pihak lain yang tidak berkepentingan, maka pesan perlu disandikan kebentuk lain. Bentuk pesan yang disandikan disebut *ciphertext.*

b. Enkripsi dan dekripsi

   Proses penyandian *plaintext* menjadi *ciphertext* disebut enkripsi. Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi.

c. *Cipher* dan kunci

   Cipher adalah aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa cipher memerlukan algoritma yang berbeda. [7]

2.7.1 Tujuan Kriptografi

Kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspke keamanan sebagai berikut:

a. Kerahasiaan (*confidentiality*)
b. Integritas data (*data intergrity*)
c. Otentikasi (*authentication*)
d. *Non-repudiation.* [7]

2.7.2 *Vigenere Cipher*

Vigenere Cipher termasuk dalam cipher abjadmajemuk (*polyalphabetic substitution Cipher*) yang dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, *Blaise de Vigenere* pada abad 16 (tahun 1586). *Vigenere*

Cipher adalah metode menyandikan teks alfabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci. *Vigenere Cipher* menggunakan tabel seperti pada tabel 2.1, Vigenere Cipher dengan angka dalam melakukan enkripsi [5].

Teknik dari substitusi *vigenere cipher* bisa dilakukan dengan dua cara:

a. Angka
b. Huruf

Tabel 2.1 *Vigenere Cipher* dengan Angka

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
0	1	2	3	4	5	6	7	8	9	10	11	12

N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext: PLAINTEXT Kunci: CIPHER

Tabel 2.2 Proses *Vigenere Cipher* dengan Angka

Plain	**15**	**11**	**0**	**8**	**13**	**19**	**4**	**23**	**19**
Kunci	2	8	15	7	4	17	2	8	15
Hasil	17	19	15	15	17	10	6	5	8
Ciphertext	R	T	P	P	R	K	G	F	I

Dengan metode pertukaran angka dengan huruf di atas, diperoleh bahwa teks asli (*PLAINTEXT*) memiliki kode angka (15, 11, 0, 8, 13, 19, 4, 23, 19), sedangkan kode angka untuk teks kunci (*CIPHER*) yaitu (2, 8, 15, 7, 4, 17). Setelah dilakukan perhitungan, maka dihasilkan kode angka *ciphertext* (17, 19, 15, 15, 17,

10, 6, 5, 8). Jika diterjemahkan kembali menjadi huruf sesuai urutan awal, maka menjadi huruf RTPPRKGFI [5].

2.7.3 *Vigenere Cipher* Huruf

Tabel 2.3, *Vigenere Cipher* dengan huruf berisi alfabet yang dituliskan dalam 26 baris, masingmasing baris digeser ke kiri dari baris sebelumnya membentuk ke-26 kemungkinan sandi *Caesar* setiap huruf disediakan dengan menggunakan baris yang berbeda-beda sesuai kunci yang diulang[5].

Tabel 2.3 *Vigenere Cipher* Dengan Huruf

		Plantext																								
Kunci		***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***	***L***	***M***	***N***	***O***	***P***	***Q***	***R***	***S***	***T***	***U***	***V***	***W***	***X***	***Y***	***Z***
	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Plantext: Says Hariyanto

Kunci: Hari

Dari *plaintext* dengan kata kunci di tabel didapatkan *chipertext* sebagai berikut:

Chipertext: Zapioaiqfaebw

Proses dekripsi, dilakukan dengan mencari huruf *chipertext* pada baris *plaintext* dari kata kunci [5].

Vigenere Chipher juga dapat dilihat dalam bentuk aljabar. jika huruf A-Z yang diambil untuk menjadi nomor 0-25, maka dilakukan modular 26, maka Vigenere cipher dapat ditulis dengan:

Enkripsi: $Ci = (Pi + Ki)\ mod\ 26$
Dekripsi: $Pi = (ci - ki) mod\ 26; untuk\ Ci \geq Ki$ $Pi = (Ci + 26 - Ki) mod 26; untuk\ Ci \leq Ki$

Keterangan:

C = Ciphertext

P = Plaintext

K = Kunci

2.8 Microsoft Visual Studio.NET

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. *Visual Studio* mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN *Library*). Kompiler yang dimasukkan ke dalam paket

Visual Studio antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe* [9].

2.9 UML

UML (*Unified Modeling Language)* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan [10].

1. ***Use Case Diagram***

   Use case adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.

2. ***Class Diagram***

   Class adalah dekripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem

biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

3. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antar *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

4. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

2.9.1 Tujuan Penggunaan UML

Tujuan dari penggunaan UML adalah sebagai berikut:

a. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
b. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
c. Memberikan model yang siap pakai, bahasa pemodelan *visual* yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang *coding* program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reserve enginering*) [10].