BAB II

LANDASAN TEORI

Teori merupakan dasar yang digunakan dalam pembangunan perangkat lunak. Pada bab ini akan dijelaskan mengenai landasan teori yang berkaitan erat dengan perangkat lunak yang dibangun.

2.1. Sejarah Steganografi

Steganografi berasal dari bahasa Yunani yaitu Steganós yang berarti menyembunyikan dan Graptos yang artinya tulisan sehingga secara keseluruhan artinya adalah tulisan yang disebunyikan. Secara umum steganografi merupakan seni atau ilmu yang digunakan untuk menyembunyikan pesan rahasia dengan segala cara sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari pesan rahasia tersebut.

Steganografi sudah digunakan sejak dahulu kala sekitar 2500 tahun yang lalu untuk kepentingan politik, militer, diplomatik, serta untuk kpntingan pribadi sebagai alat. Beberapa contoh penggunaan steganografi pada masa lampau:

1. Pada tahun 480 sebelum masehi, seseorang berkebangsaan Yunani yaitu Demaratus mengirimkan pesan kepada polis Sparta yang berisi peringatan mengenai penyerangan Xerxes yang ditunda. Teknik yang digunakan adalah dengan menggunakan meja yang telah diukir kemudian diberi lapisan lilin untuk menutupi pesan tersebut, dengan begitu pesan dalam meja dapat disampaikan tanpa menimbulakn kecurigaan oleh para penjaga.

2. Pada abad ke 5 sebelum masehi, Histaiacus mengirimkan pesan kepada Aristagoras Miletus untuk memberontak terhadap raja Persia. Pesan disampaikan dengan cara mencukur kepala pembawa pesan dan mentato kepalanya dengan pesan tersebut. Kemudian saat rambutnya tumbuh kembali, pembawa pesan dikirimkan dan pada tempat tujuan rambutnya kembali digunduli dan pesan akan terbaca.
3. Penggunaan tinta yang tidak terlihat pada pesan lainnya.
4. Pada perang dunia II, Jerman menggunakan *microdots* untuk berkomunikasi. Penggnaan teknik ini biasa digunakan pada *microfilm* chip yang harus diperbesar sekitar 200 kali.
5. Pada perang dunia II, Amerika Serikat menggunakan suku Indian Navajo sebagai media untuk berkomunikasi.

2.2. Steganografi

Steganografi (*steganography*) adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indera manusia. Kata steganorafi berasal dari Bahasa Yunani yang berarti "tulisan tersembunyi" (*covered writing*). Steganografi membutuhkan dua properti wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video. 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. Ilustrasi mengenai perbedaan kriptografi dan steganografi dapat dilihat pada gambar 2.1.

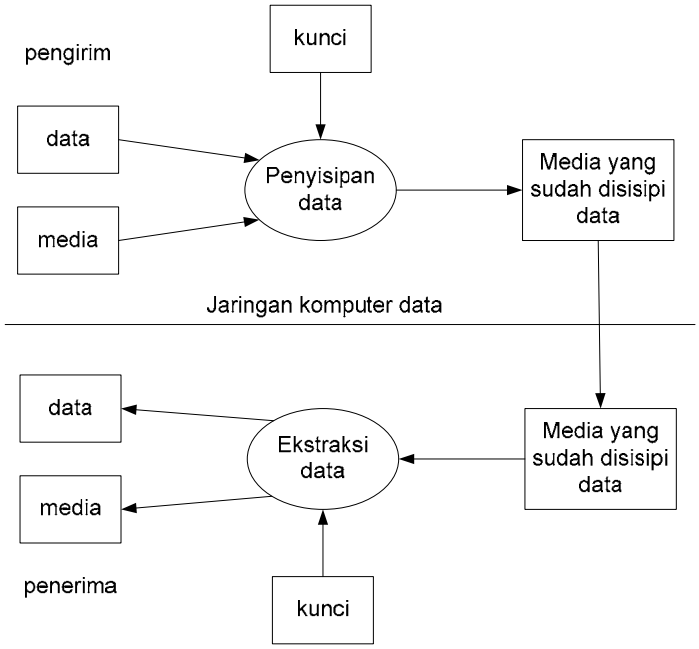


Gambar 2.1. Ilustrasi kriptografi dan steganografi pada citra digital.

Steganografi membahas bagaimana sebuah pesan dapat disisipkan ke dalam sebuah berkas media sehingga pihak ketiga tidak menyadarinya. Steganografi memanfaatkan keterbatasan sistem indera manusia seperti mata dan telinga. Dengan adanya keterbatasan 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 sebatas oleh kemampuan indera manusia, tetapi tidak oleh komputer atau perangkat pengolah digital lainnya. Ilustrasi mengenai proses steganografi dapat dilihat pada gambar 2.2.

Penyembunyian data rahasia ke dalam media digital mengubah kualitas media tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data diantaranya adalah :

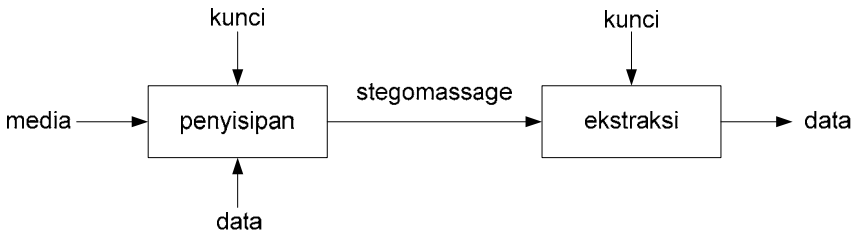
1. *Fidelity*. Mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2. *Recovery*. Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*). Karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.



Gambar 2.2. Diagram Sistem Steganografi

Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks, dan video. Sedangkan data rahasia yang disembunyikan dapat berupa berkas 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 2.3. Penambahan kunci yang bersifat opsional dimaksudkan untuk lebih meningkatkan keamanan.

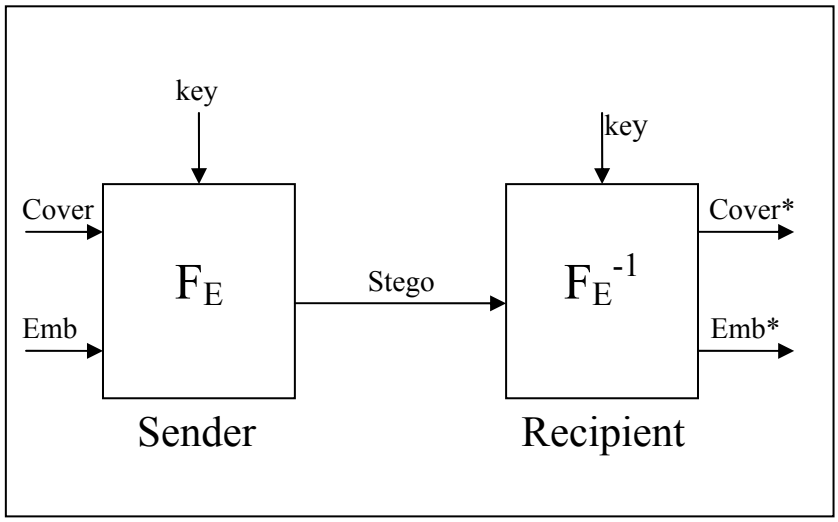


Gambar 2.3. Proses penyisipan dan ekstraksi dalam steganografi

2.3. Sistem Steganografi

Steganografi atau stego biasanya berhubungan dengan komunitas IT (*Information Technology*), Kata steganografi (*steganography*) berasal dari bahasa Yunani *steganos*, yang artinya 'tersembunyi/terselubung', dan *graphein*, 'menulis' sehingga kurang lebih artinya "menulis (tulisan) terselubung". Teknik ini meliputi banyak sekali metoda komunikasi untuk menyembunyikan pesan rahasia. Steganografi didefinisikan oleh Markus Khan sebagai berikut, “Steganografi adalah seni dan ilmu pengetahuan dalam berkomunikasi dengan cara menyembunyikan keberadaan dari komunikasi itu sendiri. Berbeda dengan kriptografi, dimana musuh bisa mendeteksi, menangkap dan memodifikasi pesan tanpa mampu mengganggu keamanan dari kriptosistemnya, tujuan dari steganografi adalah untuk menyembunyikan pesan didalam pesan “yang tidak membahayakan” dengan cara tertentu untuk menghindari musuh agar tidak dapat mendeteksi bahwa adanya pesan tersebunyi.

Gambar 2.4 dibawah ini menunjukkan metode dasar bagaimana cara kerja steganografi :



Gambar 2.4. Cara kerja steganografi secara umum

Keterangan :

F_E : *Embedding* (Penggabungan berkas *cover* dengan berkas rahasia)

F_E^{-1} : Ekstraksi (Pengekstrakan berkas rahasia dari berkas *cover*)

Cover : Berkas data yang menutup *emb* yang akan disembunyikan

Emb : Pesan yang akan disisipkan

Stego : Berkas data yang sudah sisipi pesan tersembunyi

Cover adalah berkas gambar, audio atau bisa juga video. **Emb** adalah pesan rahasia yang akan disisipi. **Key** adalah parameter yang mengontrol dari proses penyembunyian pesan dan **Stego** adalah berkas yang sudah berisi pesan tersembunyi. Yang harus diperhatikan adalah jumlah dari informasi yang akan disisipi, karena semakin besar ukuran pesan, maka semakin besar pula pengaruh

pada modifikasi pada data *cover*nya dan akibatnya akan semakin tinggi pula kemungkinan akan terdeteksi.

2.4. Aplikasi Steganografi

Steganografi sebagai suatu teknik penyembunyian informasi pada data digital lainnya dapat dimanfaatkan untuk berbagai tujuan seperti :

1. *Tamper-proofing* dimana steganografi digunakan sebagai alat untuk mengidentifikasikan atau alat indikator yang menunjukkan data *host* telah mengalami perubahan dari aslinya.
2. *Feature location* dimana steganografi digunakan sebagai alat untuk mengidentifikasikan isi dari data digital pada lokasi-lokasi tertentu, seperti contohnya penamaan objek tertentu dari beberapa objek yang lain pada suatu citra digital.
3. *Annotation/caption* dimana steganografi hanya digunakan sebagai keterangan tentang data digital itu sendiri.
4. *Copyright-Labeling* dimana steganografi dapat digunakan sebagai metoda untuk penyembunyian label hak cipta pada data digital sebagai bukti otentik kepemilikan karya digital tersebut.

2.5. Manfaat Steganografi

Steganography adalah sebuah pisau bermata dua, ia bisa digunakan untuk alasan-alasan yang baik, tetapi bisa juga digunakan sebagai sarana kejahatan. Steganography juga dapat digunakan sebagai salah satu metode watermarking

pada image untuk proteksi hak cipta, seperti juga digital watermarking (fingerprinting). Steganography juga dapat digunakan sebagai pengganti hash. Dan yang terutama, seperti disebutkan sebelumnya, steganography dapat digunakan untuk menyembunyikan informasi rahasia, untuk melindunginya dari pencurian dan dari orang yang tidak berhak untuk mengetahuinya. Sayangnya, steganography juga dapat digunakan untuk mencuri data yang disembunyikan pada data lain sehingga dapat dikirim ke pihak lain, yang tidak berhak, tanpa ada yang curiga. Steganography juga dapat digunakan oleh para teroris untuk saling berkomunikasi satu dengan yang lain. Sehubungan dengan keamanan sistem informasi, steganography hanya merupakan salah satu dari banyak cara yang dapat dilakukan untuk menyembunyikan pesan rahasia. Steganography lebih cocok digunakan bersamaan dengan metode lain tersebut untuk menciptakan keamanan yang berlapis. Sebagai contoh steganography dapat digunakan bersama dengan enkripsi. Windows dan Unix juga menggunakan steganography dalam mengimplementasikan hidden directory.

2.6. Multimedia

Komputer sebagai alat pengolahan *digital* saat ini hampir dimiliki oleh setiap orang, seperti halnya televisi, komputer juga banyak digunakan sebagai media hiburan. Dengan perkembangan komputer (serta pengetahuan dalam pengolahan sinyal *digital*), maka data-data dalam bentuk *digital* semakin banyak digunakan.

Penggunaan data *digital* baik berupa text, suara, citra maupun video sangat pesat, ditambah dengan perkembangan teknologi jaringan antar komputer didunia (internet), pertukaran data *digital* semakin mudah untuk dapat menggabungkan semua data *digital* yang ada (multimedia).

Dalam pembuatan aplikasi-aplikasi multimedia diperlukan beberapa komponen pendukung seperti *text,* suara / audio, dan citra */ image.*

2.6.1. 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.*

Teknik steganografi yang menggunakan teks sebagai *cover* adalah hal yang menantang. Ini dikarenakan berkas teks memiliki ukuran data yang kecil untuk bisa digantikan dengan berkas rahasia. Dan kekurangan lainnya adalah teks yang mengandung teknik steganografi ini dengan mudah dapat diubah oleh pihak yang tidak diinginkan dengan cara mengubah teks itu sendiri maupun mengubah format dari teksnya (misal .TXT menjadi .PDF). Ada beberapa metode yang digunakan pada media teks ini yaitu, *Line-Shift Encoding, Word-shift Encoding* dan *Feature Coding*. Ketiganya merupakan metode *encoding* yang membutuhkan berkas asli dan juga format aslinya untuk dapat di*decode* atau diekstrak kembali.

2.6.2. Data *Image*

Citra / *images* adalah gambar *digital* dua dimensi yang dihasilkan dari gambar analog, dibagi kedalam N baris dan M kolom sehingga menjadi gambar diskrit melalui proses *sampling* (proses pengambilan *sample* warna sebelum melakukan proses digitalisasi citra), dan persilangan antara baris dan kolom disebut dengan *pixel. Images* diklasifikasikan ke dalam dua jenis yaitu :

1. *Images* Tetap

   Images / gambar mati (grafik statis) yang tidak mengalami proses animasi biasanya berupa format BMP, JPG, JPEG, ICO dan lain-lain.

2. *Images* Bergerak /Animasi

   Image bergerak adalah *image* tetap yang diberikan efek seolah-olah seperti bergerak, atau sering disebut animasi. Animasi adalah proses kejadian dalam pembuatan film (*movie*), saat ini banyak film animasi tiga dimensi daripada dua dimensi.

Steganografi pada gambar adalah metode yang paling banyak digunakan secara luas didunia digital saat ini. Hal ini dikarenakan keterbatasan kemampuan dari visual atau *Human Visual System* (HVS). Format gambar yang biasanya digunakan adalah format bitmap (bmp), gif, pcx, jpeg dan format gambar lainnya.

Hampir semua *plain* teks, *cipher* teks, gambar dan media lainnya dapat di*encode* kedalam aliran bit untuk disembunyikan didalam gambar digital. Perkembangan dari metoda ini sangat pesat, didukung dengan semakin canggihnya komputer grafik yang *powerful*, dan software steganografi yang sekarang sudah banyak tersebar luas di internet. Pendekatan yang paling sering

dilakukan pada media jenis ini adalah : *Least Significant Bit Insertion, Masking and Filtering* dan *Algorithm and Transformation*. Sebagai catatan, masih ada banyak lagi teknik-teknik yang digunakan untuk *encoding* pada media gambar ini.

2.6.3. Citra Digital

Citra merupakan fungsi intensitas dalam bidang dua dimensi. Intensitas yang dimaksud berasal dari sumber cahaya. Pada hakekatnya citra yang dilihat oleh mata manusia terdiri atas berkas-berkas cahaya yang dipantulkan oleh benda-benda di sekitar kita.

Suatu citra digital adalah suatu gambar kontinu yang diubah dalam bentuk diskrit, baik koordinat maupun intensitas cahayanya. Kita dapat menganggap suatu citra digital sebagai suatu matriks dimana indeks baris dan kolomnya menyatakan koordinat sebuah titik pada citra tersebut dan masing-masing elemennya menyatakan intensitas cahaya pada titik tersebut. Suatu titik pada sebuah citra digital sering disebut sebagai "image-element" (elemen citra) "picture-element" (elemen gambar) ataupun "*pixel*".

Untuk mengubah suatu citra kontinu ke dalam suatu representasi numerik dilakukan dengan proses digitalisasi oleh suatu *digitizer*, misalnya *scanner*, sehingga citra ini dapat diproses oleh sebuah komputer.

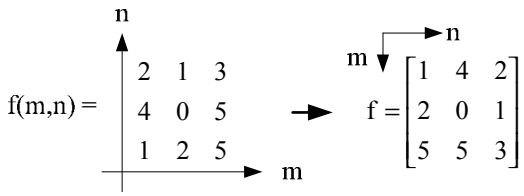
Digitalisasi sebuah citra dilakukan baik terhadap ruang (koordinat (x,y)), maupun terhadap skala keabuannya (f(x,y)). Proses digitalisasi koordinat (x,y) dikenal sebagai "pencuplikan citra" *(image sampling)*, sedangkan proses

digitalisasi skala keabuan f(x,y) disebut sebagai "kuantisasi derajat keabuan" *(grey-level quantization)*.

Sebuah citra kontinu f(x,y) akan didekati oleh cuplikan-cuplikan yang seragam jaraknya dalam bentuk matriks MxN, M adalah baris dan N adalah kolom. Nilai elemen-elemen matriks menyatakan derajat keabuan citra, sebangkan posisi elemen tersebut (dalam baris dan kolom) menyatakan koordinat titik-titik (x,y) dari citra. Bentuk matriks di bawah ini dikenal sebagai suatu citra digital.

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ f(M-1,0) & . & \dots & f(M-1,N-1) \end{bmatrix}$$

Matriks di atas dapat disajikan dalam bentuk 2 dimensi dalam sistem koordinat Cartesius dengan memutar posisi matriks di atas sejauh 90° derajat searah jarum jam.



Sedangkan derajat keabuan [0,L] dibagi kedalam G selang dengan panjang selang yang sama, yaitu: $\mathbf{G = 2^m}$ dimana m adalah kedalaman *bit* dan m bilangan bulat positif, bila hal ini diterapkan pada penyimpanan maka sebuah citra digital membutuhkan sejumlah b *bit*, dengan :

$$\mathbf{b = M \times N \times m.}$$

Dalam suatu proses pencuplikan dan kuantisasi sering terjadi permasalahan, yaitu jumlah cuplikan dan derajat keabuan yang diperlukan untuk memperoleh suatu citra yang "baik", makin tinggi nilai MxN dan m, maka citra kontinu f(x,y) akan makin didekati oleh citra digital yang dihasilkan. Tapi hal ini seringkali dibatasi oleh kemampuan hardware dari suatu komputer.

2.6.4. Bitmap (BMP)

BMP atau *DIB* (*device independent bitmap*), adalah sebuah format grafik yang digunakan secara internal oleh microsoft windows dan subsistem OS/2, dan sering digunakan sebagai format file grafik sederhana pada *flatform-flatform* tersebut.

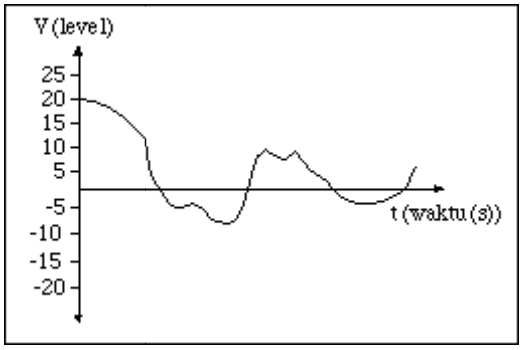
Gambar secara umum disajikan dalam ketajaman warna 2 (1-*bit*), 16 (4-*bit*), 256 (8-*bit*), 65.536 (16-*bit*), 16.7 juta (24-*bit*) warna (*bit-bit* ini mempresentasi *bit-bit* per-*pixel*). Sebuah gambar 8-*bit* juga dapat dirubah ke warna *grayscale* disamping warna indeks. Sebua *channel alpha* (untuk warna transparan) boleh disajikan dalam *file* terpisah, dimana sama dengan gambar *grayscale*. Versi 32-*bit* dengan *channel alpha* terintegrasi telah diperkenalkan oleh Windows XP dan digunakan untuk sistem *login* dan *theme*.

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* 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*.

2.6.5. Data Suara

Suara (audio) atau bunyi merupakan sebuah sinyal analog yang dihasilkan oleh getaran yang membentuk sebuah gelombang *longitudinal* dengan frekuensi 20 hz sampai 20.000 hz (berdasarkan tingkat pendengaran manusia).



Gambar 2.5. Suara dalam bentuk sinyal analog

(Sumber : Drs Sunomo [9])

Gelombang suara yang masih berbentuk sinyal analog dapat diubah kedalam bit-bit *digital* menjadi sinyal *digital*, melalui proses *sampling* dan digitalisasi.

2.6.6. File Audio Digital

Sebenarnya, file audio digital itu tidak hanya Compact Disk (CD) audio atau MP3 saja. Ada bermacam jenis ekstensi file audio digital, namun tidak semuanya dapat dimainkan di suatu player. Adapun ekstensi file audio digital yang dapat dimainkan adalah :

1. WAV

File audio ini diciptakan oleh Microsoft. File ini menjadi standar *file* audio dalam *Personal Computer*(PC) baik untuk sistem operasi, *game*, maupun *file* suara lain yang kualitas suaranya setara dengan CD. File ini memiliki ukuran yang cukup besar, karena suara yang disimpan dalam format ini tidak mengalami proses kompresi (*raw data*). Meski buatan Microsoft, bukan berarti *file* ini tidak dapat dijalankan oleh sistem operasi selain Windows. Setidaknya, Linux dan Macintosh juga dapat menjalankan *file* ini.

2. AIFF

Merupakan audio digital tidak terkompresi buatan Macintosh. CD audio komersial umumnya menggunakan format Audio Interchange File (AIFF), salah satu format audio digital dengan kualitas terbaik. Wajar kalau ukuran *file* dalam format ini rata-rata cukup besar. Untuk lagu berdurasi tiga menit saja, *file* .AIFF dapat menggunakan ruang *hard disk* sebesar 30 sampai 50 MB.

3. MP3

Ditemukan tahun 1992, MP3 merupakan *file* audio terkompresi pertama yang memiliki kualitas bagus dan berukuran kecil, dimana pemadatan *file* MP3 dilakukan dengan cara membuang bagian-bagian suara yang tidak dapat didengar telinga manusia. Hasilnya, *file* MP3 jauh lebih kecil hingga skala 12:1. Misalnya, suara berdurasi 3 menit yang tadinya berukuran 30 MB, jika di-MP3-kan bisa susut jadi 2,5 MB saja. Kecilnya ukuran MP3 membuatnya menjadi *file* audio favorit yang digunakan dalam internet.

4. WMA

Windows Media Audio (WMA) sebetulnya sudah diperkenalkan sejak akhir 1990. Namun, pamornya masih kalah dibanding MP3. Akhir-akhir ini, WMA mulai merebak di kalangan pengguna Windows, terutama dihubungkan dengan hak cipta. WMA memiliki keunggulan dalam melindungi *file* audio. WMA tidak bisa dimainkan atau disalin dengan sembarangan. Sama seperti MP3, WMA juga menggunakan *bit rate* 128 Kbps.

2.6.7. Audio Digital Tidak Terkompresi (WAV)

File wave (WAV) adalah format file audio yang diciptakan oleh Microsoft, dan telah menjadi standar format audio bagi PC. File WAV dikenali sebagai sebuah file dengan ekstensi *.wav.

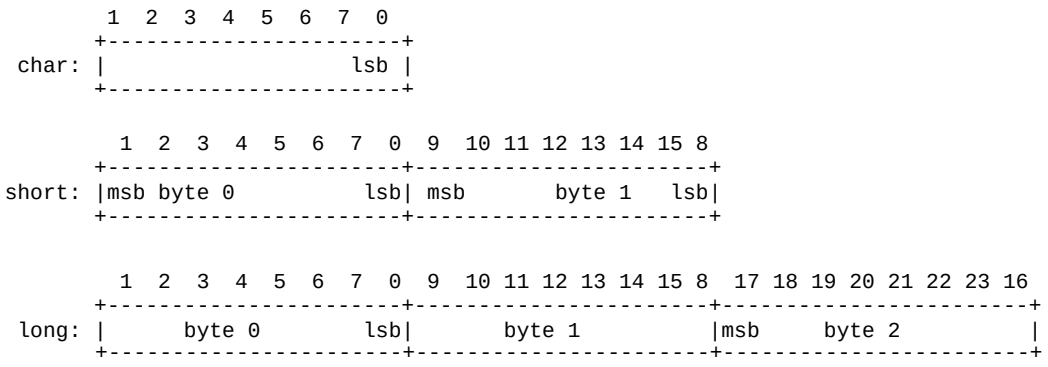
File WAV memiliki format 1-N saluran dan 16 bit per *sample,* serta memiliki *sampling rate* 44.1 KHz. File WAV digunakan di dalam PC sebagai suatu medium pertukaran antara komputer dengan platform yang berbeda.

Sound sample juga disimpan sebagai data mentah dengan 1-N saluran dalam file yang sama. Saluran tersebut harus diselipkan dengan cara yang sama seperti dalam file *AIFF-C*. Sebagai tambahan untuk data mentah audio yang tidak terkompresi, format file WAV menyimpan informasi mengenai jumlah track (mono atau stereo), *sample rate,* dan *bit depth.*

Keuntungan dengan adanya file WAV adalah kita dapat mengedit file tersebut karena file tersebut masih dalam kondisi tidak terkompresi, tetapi ketika file telah dimampatkan, kita akan mengalami kesulitan dalam mengedit.

2.6.8. Organisasi Data

Semua data disimpan pada byte berukuran 8-bit. Byte yang merupakan nilai dari sekumpulan bit disimpan dengan *least significant bit* diletakkan pada bagian awalnya. Pengaturan bit-bit datanya adalah seperti pada gambar 2.6 :



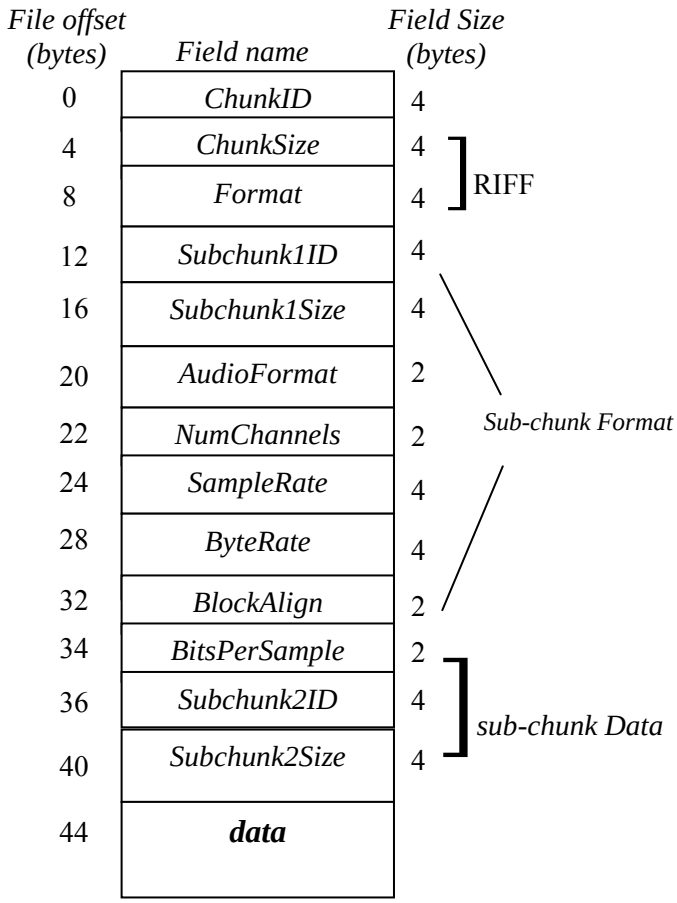
Gambar 2.6. Organisasi Bit Data pada File WAV

2.6.9. Format WAV

WAV adalah file audio digital yang tidak terkompresi yang banyak digunakan pada PC dan memiliki kualitas suara audio CD. WAV adalah format audio digital yang paling murni, oleh karena itu format ini membutuhkan *space* / ruang yang sangat besar di dalam hard-disk. Misalkan saja untuk sebuah lagu berdurasi 3 menit, file wave menghabiskan sekitar 30 MB ruang di dalam hard-disk.

Format WAV adalah suatu format file untuk penyimpanan data audio digital (*waveform*). Format WAV mampu mendukung berbagai resolusi bit, *sample rates*, dan banyak kanal audio. Format ini sangat populer untuk *platform* IBM PC dan kloningnya, serta secara luas digunakan untuk program profesional

yang memproses audio digital *waveform*. Format ini menggunakan versi microsoft sebagai metoda untuk menyimpan data di dalam *chunks*.



Gambar 2.7. Format File WAV

Keterangan :

offset panjang isi

0 4 bytes "RIFF"

4 4 bytes <panjang file - 8>

8 4 bytes "WAVE"

12 4 bytes "fmt "

16 4 bytes <panjang data fmt> // (=16)

20 2 bytes <aturan encoding file wave>

22 2 bytes <saluran > //saluran: 1 = mono, 2 = stereo

24 4 bytes <Sample rate> // Sample per detik: 44100

28 4 bytes <byte/detik > // *sample rate* * *block align*

32 2 bytes <block align> // saluran * bit/sample / 8

34 2 bytes <bit/sample> // 8 atau 16

36 4 bytes "data"

40 4 bytes <ukuran *sample data* (n)>

44 (n)bytes <*Sample data*>

Dari gambar 2.7, data *sample* harus berakhir pada batas antara byte. Semua data *numeric* adalah dalam format *Intel low-high byte ordering*, *sample* 8-bit disimpan sebagai *byte* yang tidak dikenal, rangenya antara 0-255, dan *sample* 16-bit disimpan sebagai *2's-complement* integer yang dikenal, rangenya antara -32768 untuk 32767.

2.7. Steganografi pada Audio

Penyembunyian data pada audio merupakan teknik yang paling menantang pada steganografi ini. Hal ini disebabkan *Human Auditory System* (HAS) memiliki jangkauan yang dinamis. HAS memiliki kemampuan mendengar lebih dari satu sampai 1 miliar. Dan jangkauan frequensi lebih dari satu hingga seribu. *Auditory System* ini juga sangat peka pada gangguan suara (*noise*) yang halus sekalipun. Sedikit saja terdapat gangguan pada sebuah berkas audio maka

dengan mudah akan terdeteksi. Satu-satunya kelemahan yang dimiliki HAS dalam membedakan suara adalah kenyataan bahwa suara keras bisa menenggelamkan suara pelan. Terdapat dua konsep yang harus dipertimbangkan sebelum memilih metoda mana yang akan dipakai. Yaitu format digital audio dan media transmisi dari audio.

Ada 3 format audio digital yang paling sering digunakan yaitu *Sample Quantization, Temporal Sampling Rate* dan *Perceptual Sampling. Sample Quantization* merupakan 16-bit *linear sampling architecture* yang digunakan pada audio format seperti (.WAV dan .AIFF). *Temporal Sampling Rate* digunakan pada frequensi tertentu (dalam KHz) sebagai sampel dari audionya. Umumnya, semakin tinggi *sampling rate* maka semakin besar pula ukuran data yang digunakannya. Audio format yang terakhir adalah *Perceptual Sampling*. Format ini mengubah statistik dari audio secara drastis dengan melakukan *encoding* hanya pada bagian yang didengar saja. Artinya mengubah sinyal namun tetap menjaga suara. Format ini digunakan oleh digital audio yang paling terkenal pada saat ini yaitu MP3.

2.7.1. Formula LSB

Dari keempat teknik tersebut, penulis ingin memfokuskan penyembunyian informasi pada audio dengan metode LSB. Mengubah nilai LSB tidak akan menghasilkan perubahan yang signifikan pada media digital. Tanpa perbandingan langsung antara file wav original dan file wav terlabel, sulit untuk mengatakan bahwa ada sesuatu yang berubah.

Menyembunyikan data biasanya dilakukan dengan mengubah beberapa informasi yang kurang esensial di dalam cover audio. Salah satu pendekatan adalah dengan menggunakan LSB dari setiap sample di dalam cover audio dalam tujuannya untuk menyembunyikan satu bit dari informasi rahasia, dan hal ini tidak memberikan efek yang sangat besar terhadap cover audio.

Misalkan sebuah cover (C) dan informasi rahasia (M), sebuah skema steganography harus mengandung fungsi F_E, dan untuk mendapatkan kembali informasi menggunakan fungsi F_E^{-1}, seperti di bawah ini :

$$C' = F_E (C, M, K)$$

$$F_E^{-1} (C' , K) = F_E^{-1} (F_E (C, M, K)) = M$$

Dimana K adalah sebuah kunci rahasia. Dari persamaan dapat dilihat F_E^{-1} mengekstrak informasi yang di-*embedding* pada cover audio C yang sebelumnya disembunyikan oleh F_E.

Untuk menyembunyikan sebuah informasi M di dalam cover audio C menggunakan algoritma LSB, *embedding* prosesnya terdiri atas pemilihan sebuah subset (j_1,....,$j_{l(m)}$) dari elemen cover dan operasi substitusi dengan menukar LSB tiap sample C_{ji} dengan m_i, dimana m_i bisa 0 atau 1. Operasi substitusi merubah lebih dari satu bit dari cover audio. Pada proses ekstrak, LSB elemen cover yang telah dipilih akan di ekstrak dan disusun untuk mendapatkan kembali informasi rahasia.

2.7.2. Perhitungan PSNR

Perhitungan kualitas video digital yang merupakan hasil modifikasi, terhadap video digital yang asli, dapat dilakukan dengan menghitung nilai MSE (*Mean Square Error*) dan juga nilai PSNR (*Peak Signal-to-noise ratio*). Perhitungan nilai MSE dari video digital berukuran N x M, dilakukan sesuai dengan rumus berikut:

$$MSE = \frac{1}{N.M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left[\left| f(i,j) - f'(i,j) \right|^2 \right] \qquad \textbf{(2.1)}$$

f(i,j) menyatakan citra digital yang asli sebelum dikompresi, sedangkan *f'(i,j)*, merupakan citra digital hasil kompresi nilai MSE yang besar, menyatakan bahwa penyimpangan atau selisih antara video hasil modifikasi dengan video aslinya cukup besar. Sedangkan untuk perhitungan nilai PSNR, dapat dilakukan dengan rumus berikut:

$$PSNR = 10 \log \left(\frac{(255)^2}{MSE} \right) \qquad \textbf{(2.2)}$$

Semakin besar PSNR, maka kualitas video hasil modifikasi akan semakin baik, sebab tidak banyak data yang mengalami perubahan, dibandingkan aslinya.

2.7.3. FFT (*Fast Fourier Transform*)

Citra pada umumnya digambarkan dalam dimensi spasial antara amplitude dengan posisi spasial dua dimensi. Dalam operasi analisis citra, seringkali dilakukan operasi konvolusi pada matriks (citra) yang berdimensi besar. Hal ini

menjadi salah satu penghambat karena ketika dilakukan operasi konvolusi, maka dibutuhkan resource computer yang sangat besar

Agar operasi analisis citra dapat dilakukan dengan efisien, maka sebaiknya citra dirubah dahulu kedalam dimensi frekuensi. Hal ini sangat penting dilakukan terlebih dalam operasi yang mengunakan konvolusi karena konvolusi dalam dimensi spasial berarti perkalian produk (dot product) dalam dimensi frekuensi. mempercepat waktu proses. Untuk melakukan transformasi dari dimensi spasial menjadi dimensi frekuensi, salah satunya adalah dengan menggunaka transformasi Fourier. Inti dari transformasi Fouries adalah memecah signal (citra) menjadi gelombang-gelombang sinusoidal dimana jumlahnya sama dengan signal asalnya.

Dalam fungsi yang kontinyu, transformasi Fourier dinyatakan dalam bentuk sebagai berikut:

$$\Im\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) e^{1i2\pi ux} du$$

Dan fungsi balikannya adalah sebagai berikut

$$\Im\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) e^{1i2\pi ux} du$$

Dalam konteks dua dimensi, fungsi Fourier menjadi seperti berikut:

$$\Im\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{1i2\pi ux} du$$

Dan fungsi balikannya adalah sebagai berikut:

$$\Im\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v) e^{1i2\pi ux} du$$

Untuk F(u,v) bilangan kompleks, F(u,v) dapat dinyatakan sebagai berikut:

$$F(u,v) = R(u,v) + iI(u,v) = |F(u,v)|e^{i\phi(u)}$$

Sehingga spektrumnya dapat dinyatakan sebagai:

$$|F(u,v)| = \sqrt{R^2(u,v) + i\,I^2(u,v)}$$

Karena citra bukan merupakan signal yang kontinyu, maka fungsi transformasi Fourier diatas harus dinyatakan dalam bentuk diskrit juga.

Fungsi fourier dalam bentuk diskrit disebut juga Discrete Fourier Transform (DFT).

Jika f(x) yang kontinyu dibuat menjadi diskrit dengan mengambil N buah sampel berjarak Δx, yaitu himpunan nilai

$$\{f(\chi_0), f(\chi_0 + \Delta x), f(\chi_0 + 2\Delta x), ..., f(\chi_0 + (N-1)\Delta x)\}$$

Jadi

$$f(\chi_0 + x\Delta x), x = 0,1,2,3,..., N-1$$

Sehingga dalam bentuk diskrit, fungsi Fourier bervariabel dua menjadi sebagai berikut:

$$F_{x,y} = \frac{1}{NM}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} F_{u,v} e^{-i2\pi(ux/N + vy/M)}$$

Sedangkan fungsi balikannya adalah sebagai berikut:

$$f_{x,y} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} F_{u,v} e^{i2\pi(ux/N + vy/M)}$$

Algoritma DFT standar tidak bagus jika digukanan pada citra berukuran besar, karena proses komputasinya memakan waktu yang lama. Untuk mempercepat proses transformasi, digunakan algoritma FFT (Fast Fourier

Transform). Algoritma ini sangat cepat mengerjakan matriks yang berukuran besar. Perbedaan kompleksitas waktu DFT dengan FFT adalah

$$0(N^2) : 0(N^2 \log N)$$

Dengan FFT proses komputasi biasa direduksi dari N2 menjadi N log2N. Misalnya dengan menggunakan DFT kita akan melakukan transformasi sebanyak N=1024 titik, maka kita memerlukan perkalian sebanyak N2 =1.048.567. Sedangkan dengan menggunakan FFT perkalian yang diperlukan sebanyak N log2N = 5120 perkalian.

2.7.4. LSB *(Least Significant Bit)*

LSB (Least Significant Bit) Coding. Metoda ini merupakan metoda yang sederhana. Metoda ini akan mengubah nilai LSB (*Least Significant Bit*) komponen luminansi atau warna menjadi *bit* yang bersesuai dengan *bit* label yang akan disembunyikan. Memang metoda ini akan menghasilkan video rekontruksi yang sangat mirip dengan aslinya, karena hanya mengubah nilai *bit* terakhir dari data. Metoda ini paling mudah diserang, karena bila orang lain tahu maka tinggal membalikkan nilai dari LSB-nya maka data label akan hilang seluruhnya.

1. Penyembunyian data dilakukan dengan mengganti *bit-bit* data di dalam segmen teks dengan *bit-bit* data rahasia. Metode yang paling sederhana adalah metode **LSB** (*Least Significant Bit Modification*).
2. Pada susunan *bit* di dalam sebuah *byte* (1 *byte* = 8 *bit*), ada *bit* yang paling berarti (*most significant bit* atau *MSB*) dan *bit* yang paling kurang berarti (*least significant bit* atau *LSB*).

3. Perhatikan contoh sebuah susunan *bit* pada sebuah *byte*:

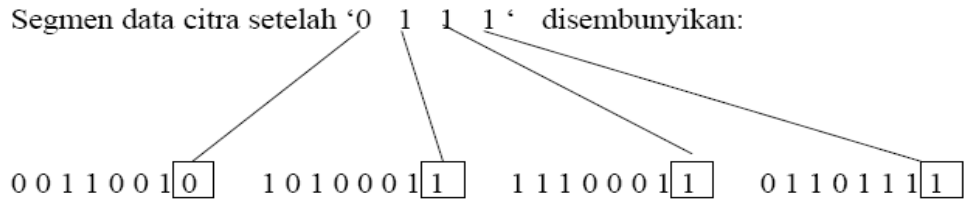
   MSB ◊1010010 ⇓ *LSB* *LSB = Least Significant Bit*

   MSB = Most Significant Bit

   Bit yang cocok untuk diganti adalah *bit LSB*, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan 6 *byte* tersebut menyatakan warna merah, maka perubahan satu *bit* LSB tidak mengubah warna merah tersebut secara berarti. Lagi pula, mata manusia tidak dapat membedakan perubahan yang kecil.

4. Misalkan segmen data citra sebelum perubahan:

   0 0 1 1 0 0 1 1 1 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 1 1 1

5. Untuk memperkuat teknik penyembunyian data, *bit-bit* data rahasia tidak digunakan mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 *bit* data yang akan disembunyikan, maka maka *byte* yang diganti *bit LSB*-nya dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49.

6. Bilangan acak dapat dibangkitkan dengan program *pseudorandom- number-generator* (*PRNG*). *PRNG* menggunakan kunci rahasia untuk membangkitkan posisi *pixel* yang akan digunakan untuk menyembunyikan *bit-bit*.

7. *PRNG* dibangun dalam sejumlah cara, salah satunya dengan menggunakan algoritma kriptografi berbasis blok (*block cipher*). Tujuan dari enkripsi adalah menghasilkan sekumpulan bilangan acak yang sama untuk setiap kunci enkripsi yang sama. Bilangan acak dihasilkan dengan cara memilih *bit-bit* dari sebuah blok data hasil enkripsi.