BAB III

LANDASAN TEORI

3.1 Steganografi

Dalam penelitian yang telah dilakukan (Johnson, et al., 2000) teknik steganografi adalah cara untuk menyisipkan pesan, jika bukan pengirim dan penerima, pihak lain tidak berhak untuk mengetahui dan menyadari bahwa terdapat sebuah pesan rahasia didalam wadah atau media yang disisipkan. Teknik *steganografi* berasal dari bahasa Yunani atau dapat ditulis *steganos*, yang berarti tersembunyi dan dapat ditulis *graphein* yang berarti menulis.

Pada jaman sekarang, istilah teknik steganografi termasuk dalam proses penyembunyian dalam bentuk data digital yang berada dalam *file* komputer. Contoh singkat yang dapat dijelaskan adalah pengirim memilih gambar yang ingin dititipi informasi, kemudian mengatur setiap warna *pixel* ke-100 untuk menyesuaikan huruf dalam *alphabet.* Setiap perubahan yang terjadi diamati dan dicatat, perubahan yang begitu halus atau tanpa cacat dalam kondisi fisik gambar tersebut sehingga tidak ada pihak lain yang menyadarinya bahwa didalam gambar tersebut terdapat sebuah informasi yang bersifat rahasia jika pihak lain tidak benar-benar memperhatikannya menggunakan aplikasi untuk mendeteksi informasi tersembunyi dalam gambar digital (Sandro, 2013). Secara umum, teknik penyisipan pesan steganografi dapat menggunakan berbagai media yang dapat dijadikan sebagai tempat atau wadah penampung informasi seperti data digital, selain itu informasi yang disisipkan juga

berbagai macam seperti artikel, daftar belanja, dokumen penting informasi lainnya. Teknik dalam steganografi mancakup banyak sekali metode untuk menyembunyikan dan merahasiakan sebuah informasi (teks atau gambar) di dalam tempat atau wadah penampung yang mengandung teks, gambar, audio dan video tanpa memperlihatkan detil perubahan yang terjadi atau terlihat dalam kualitasnya dibandingkan dengan berkas aslinya (Duric, et al., 2004)..

Tujuan utama dari teknik steganografi adalah merahasiakan atau menyembunyikan sebuah informasi tersembunyi agar tidak tersebar ke pihak lain yang tidak berkepentingan. Dalam penelitian yang telah dilakukan, pesan yang telah dititipkan tidak membuat perubahan yang *significant* terhadap data digital lain. Secara fisik tidak akan menarik perhatian dari penyerang potensial atau *hacker*, sebagai contoh sebuah gambar yang terlihat tidak berbahaya atau tidak berpotensi untuk diserang, pilihlah gambar yang umum yang telah diketahui oleh orang lain (Reddy & Raja, 2011).

Menurut (Nazelliana & Hapsari, 2015) untuk teknik steganografi bermanfaat jika digunakan tepat sasaran pada kasus yang berhubungan dengan perangkat keras komputer karena terdapat banyak sekali format berkas digital yang dapat dijadikan media atau wadah penampung untuk menyembunyikan pesan yang ingin dititipkan. Format yang biasa digunakan di antaranya:

- Format gambar (*image*): bitmap, gif, pcx, jpeg, dan lainnya.
- Format suara (*audio*) : wav, voc, mp3, dan lainnya.

- Format video : mp4, avi, h264, dan lainnya.
- Format lain : teks *file*, html, pdf, dan lainnya.

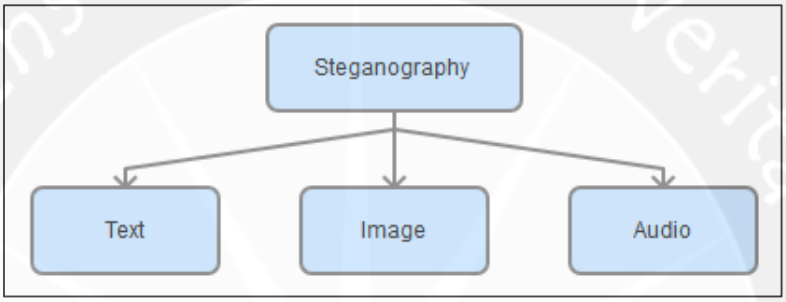
Keunggulan dari teknik steganografi jika dibandingkan dengan teknik penyembunyian lainnya adalah pesannya tidak menarik perhatian orang lain, karena dititipkan pada setiap *bit* pada *pixel* dalam gambar di R, G dan B nya. Sebagai contoh, jika dibandingkan dengan teknik kriptografi, setiap pesan memliki sandi tersendiri dan tidak disembunyikan. Walaupun sulit untuk dipecahkan setiap sandinya, tetapi akan menimbulkan kecurigaan bagi pihak lain yang ingin mengetahui isi dari informasi tersebut. Pada umumnya, teknik steganografi dan teknik kriptografi sering digunakan secara bersamaan atau kombinasi untuk menjamin keamanan informasi rahasia yang terdapat didalamnya. Metode yang digunakan pada teknik steganografi sangat beragam, tetapi secara keseluruhan pada teknik ini menggunakan *redundant bits* sebagai wadah atau tempat penampung untuk menyembunyikan pesan yang disisipi ketika dilakukan kompresi dat, karena lemahnya indera pada manusia yang sangat tidak sensitif terhadap perubahan pixel dalam gambar sehingga secara kasat mata media yang dititipi seolah olah tidak ada perbedaan yang dapat dilihat atau didengar (Bender, et al., 1996).

Setiap teknik penyembunyian informasi menggunakan media penampung dapat mengubah kualitas dari media yang telah dititipi tersebut. Sebagai perancang dalam merancang sebuah aplikasi perlu juga untuk diperhatikan kriteria dalam teknik penyembuyiannya, sebagai berikut :

- *Imperceptibility*. Keberadaan sebuah informasi rahasia tidak dapat diolah oleh indera manusia. Sebagai contoh, jika *covertext* berupa gambar digital, maka teknik penyisipan membuat gambar digital *stegotext* sulit untuk dibedakan oleh mata dengan gambar *digital covertext* nya. Jika *covertext* berupa suara *(audio)*, maka indera telinga sangatlah sulit mendeteksi perubahan yang terjadi pada audi *stegotext-nya* (Rahayu, et al., 2012).
- *Fidelity*. Kualitas media penampung tidak mempengaruhi banyak perubahan akibat penyisipan yang telah dilakukan. Perubahan tersebut sangatlah sulit untuk dipersepsi oleh indera manusia. Sebagai contoh, jika *covertext* berupa gambar digital, maka penyisipan informasi yang dilakukan membuat *citra stegotext* sulit untuk dibedakan oleh mata dengan *citra covertext-nya*. Jika covertext berupa suara *(audio)*, maka *audio stegotext* yang dihasilkan tidak rusak, sehingga indera telinga manusia tidak dapat mendeteksi perubahan tersebut (Aliwa, et al., 2013).
- *Recovery*. Informasi yang telah disembunyikan harus dapat tampil atau dimunculkan kembali sesuai aslinya *(reveal)*, karena tujuan dari teknik steganografi adalah *data hiding*, maka sewaktu-waktu informasi rahasia dalam *stegotext* harus dapat tampil dan diambil kembali agar dapat digunakan lebih lanjut (Alatas, 2009).
- *Robustness*. Informasi yang telah disembunyikan harus tahan atau kuat terhadap serangan dan manipulasi yang dilakukan pada wadah penampung.

Bila pada wadah penampung dilakukan operasi pengolahan, maka data atau informasi yang disembunyikan tidak rusak (Sejpal & Shah, 2015).

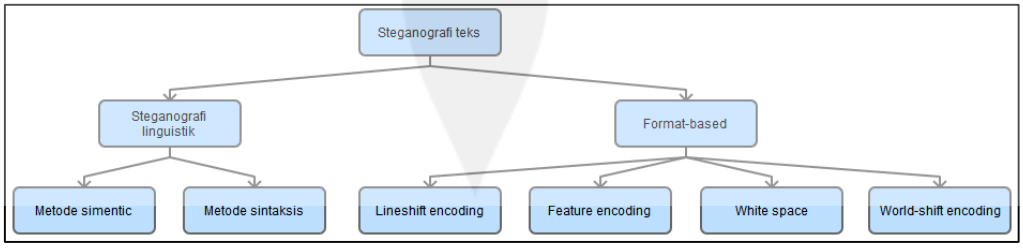
Dalam teknik steganografi membutuhkan media yang digunakan sebagai perantara dan terbagi menjadi tiga kategori menurut (Nosrati , et al., 2011) yang dapat terlihat pada (Gambar 1), yaitu:



Gambar 1. Diagram tipe media steganografi

a. Steganografi teks

Steganografi teks dapat di bagi menjadi dua kategori, seperti terlihat pada (Gambar 2), steganografi linguistik yang kemudian dibagi kembali ke dalam metode semantik dan sintaksis. Kategori lain adalah *Format-Based Steganography* yang dibagi menjadi beberapa kategori, *line-shift encoding*, *wordshift encoding*, *open-space encoding* dan *feature encoding*.



Gambar 2. Tipe steganografi teks

b. Steganografi citra

Menyembunyikan informasi dalam gambar merupakan teknik yang populer saat ini. Sebuah gambar dengan pesan rahasia di dalamnya dengan mudah dapat menyebar melalui *world wide web* atau di *news group*. Untuk menyembunyikan pesan dalam gambar tanpa mengubah sifat yang terlihat, media penutup dapat diubah di dalam wilayah "*noisy*" dengan variasi warna yang lebih banyak, sehingga lebih sedikit perhatian pada daerah modifikasi tersebut. Metode yang paling umum digunakan pada media gambar adalah dengan *least significant bit* atau LSB, *masking*, *filtering* serta transformasi pada gambar *cover* (Aboalsamh, et al., 2008). Teknik ini dapat digunakan dengan berbagai tingkat keberhasilan yang berbeda pada berbagai jenis berkas gambar.

c. Steganografi audio

Dalam steganografi audio, pesan rahasia disisipkan ke dalam sinyal *audio digital* dengan cara sedikit mengubah urutan biner yang sesuai dari berkas audio (Atoum, et al., 2011). Ada beberapa metode yang tersedia untuk steganografi audio, secara singkat akan dijelaskan sebagai berikut :

- LSB *Encoding* (Begum & Venkataramani, 2011)

  Teknik sampling di ikuti dengan proses kuantisasi untuk mengkonversi sinyal *audio* analog ke dalam biner digital. Dalam

teknik LSB ini, urutan biner dari masing-masing sampel berkas *audio digital* diganti dengan setiap *biner* dari pesan rahasia.

- *Phase Encoding* (Kaur & Behal, 2014)

  Sistem Auditory Manusia tidak dapat dengan mudah mengenali perubahan fasa dalam sinyal audio, metode *Phase Encoding* mengeksploitasi fakta ini. Teknik ini mengkodekan *bit* pesan rahasia sebagai pergeseran fase dalam spektrum fase dari sinyal digital.

- *Spread Spectrum* (Reddy, et al., 2013)

  Ada dua pendekatan yang digunakan dalam teknik ini: *Direct Sequence Spread Spectrum* (DSSS) dan *Frequency Hopping Spread Spectrum* (FHSS). DSSS adalah teknik modulasi yang digunakan di bidang telekomunikasi. Seperti dengan teknologi *spread spectrum* lain, sinyal yang ditransmisikan membutuhkan *bandwidth* yang lebih dari sinyal informasi yang sedang dimodulasi. Transmisi DSSS mengalikan data yang dikirim oleh sinyal "*noise*". Sinyal *noise* ini adalah sebuah urutan pseudorandom yang memiliki frekuensi lebih tinggi dibandingkan dengan sinyal asli, sehingga menyebarkan energi dari sinyal asli menjadi band yang lebih luas (Kaur & Behal, 2014). Sinyal yang dihasilkan menyerupai *white noise*. Namun, sinyal yang mirip seperti *noise* ini dapat digunakan dengan tepat untuk

merekonstruksi data asli di sisi penerima, dengan mengalikan urutan *pseudorandom* yang sama.

- *Echo Hiding* (Das , et al., 2008)

  Dalam metode ini, pesan rahasia disisipkan ke dalam sinyal *audio* sebagai sebuah gema (*echo*). Tiga parameter sinyal gema yaitu *amplitudo, decay rate* dan *offset* dari sinyal asli yang bervariasi untuk mewakili pesan biner dikodekan secara rahasia. Tiga parameter tersebut diatur sedemikian sehingga berada di bawah ambang Sistem Pendengaran Manusia, *Human Auditory System* (HAS) sehingga gema tidak dapat dengan mudah dipecahkan.

3.2 Mobile Application

Pengertian dari *mobile application* atau *mobile apps*, adalah istilah yang digunakan untuk menjelaskan aplikasi yang terhubung dengan internet dan jaringan dan dapat berjalan pada *smartphone* atau piranti *mobile* lainnya. Aplikasi *mobile* sangat membantu penggunanya untuk dapat terhubung dengan layanan jaringan internet yang biasanya diakses melalui komputer, dengan kata lain mempermudah para penggunanya agar dapat mengakses jaringan internet pada perangkat yang dapat digenggam, yaitu *smartphone* (Holla & Katti, 2012).

3.3 Android

Menurut (Lessard & Kessler , 2010), (Bharathi, et al., 2010) dan (Holla & Katti, 2012) pengertian dari android adalah sistem operasi untuk telepon genggam yang

berbasis Linux. Dalam sistem operasi android, menyediakan *platform* terbuka bagi para pengembangnya untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam perangkat. Awalnya, *Google Inc*. membeli *android Inc*., pendatang baru yang membuat peranti lunak untuk telepon genggam. Kemudian untuk mengembangkan *android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana *android*, 5 November 2007, *android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat telepon genggam. Di lain pihak, *Google* merilis kode-kode *android* di bawah lisensi perangkat lunak dan standar terbuka perangkat seluler. Terdapat beberapa versi pada sistem opersai *android*, mulai dari versi 1.5 *(CupCake)*, versi 1.6 *(Donut)*, versi 2.1 *(Eclair)*, versi 2.2 *(Froyo)*, versi 2.3 *(GingerBread)*, versi 3.0 *(HoneyComb)*, versi 4.3 *(JellyBean)*, versi 4.4 *(KitKat)*, versi 5.0 *(Lollipop)*, versi 6.0 *(Marshmallow)*, hingga versi yang terbaru yaitu versi 7.0 *(Nougat).*

3.4 Bahasa pemograman java

Bahasa pemrograman *java* adalah bahasa pemrograman umum *(general purpose programming language)* yang berorientasi objek (Cahyono, 2006). Pada saat ini bahasa *java* sedang populer dikalangan akademis dan praktisi komputer. Bahasa pemograman *java* pertama kali dimunculkan oleh seorang ahli bernama James Gosling dari *Sun Microsystems* pada tahun 1990-an. Bahasa pemograman *java* untuk pertama kalinya dikembangkan dengan tujuan memenuhi kebutuhan bahasa komputer yang dapat

ditulis satu kali tetapi dapat dijalankan pada banyak sistem komputer yang berbeda tanpa perubahan yang *significant*. Sebelumnya, bahasa pemograman komputer yang ada memiliki keterbatasan migrasi sistem yang berbeda. Bahasa *java* hadir dan diciptakan sebagai sebuah bahasa pemograman yang baru dengan implementasi yang berbeda. Bahasa *java* merupakan bahasa pemograman berorientasi objek yang dapat diturunkan dari *C++* dengan banyak penyempurnaan yang telah terjadi. Pada umumnya, para ahli pemrograman berpendapat bahwa bahasa *java* memiliki konsep yang konsisten dengan teori pemrograman objek dan aman untuk digunakan. Saat ini setiap universitas di berbagai negara berpaling dari *Pascal* atau *C++* kemudian memilih bahasa pemograman *java* sebagai bahasa komputer untuk belajar pemrograman (Irawan, 2007).

Bahasa pemograman *java* memiliki banyak kemampuan dalam menciptakan aplikasi berbasis PC, *web* maupun berbasis *handheld devices*, serta kelebihan *java* yang mampu berjalan pada sistem operasi apapun. Dalam praktiknya, untuk mengembangkan sebuah aplikasi berbasis *java* yang berjalan pada sebuah sistem atau jaringan, diperlukan *Java Development Kit (JDK)* dan *web server* (Supardi, 2007). Bahasa pemograman *java* memiliki beberapa karakteristik sebagai berikut (Supardi, 2007):

- Bersifat sederhana dan simpel

  Bahasa pemrograman *java* menerapkan algoritma atau sintaks yang mirip dengan bahasa *C++*, tetapi sintaks pada *java* telah banyak diperbaharui

terutama menghapus penggunaan pointer yang menyusahkan atau *multiple inheritance*. Bahasa *java* juga mempunyai *automatic memory allocation* dan *memory garbage collection*.

- Merupakan bahasa pemograman yang berorientasi objek

  Bahasa *java* adalah bahasa pemrograman berorientasi objek yang menjadikan program dapat dibuat secara modular dan dapat dipergunakan kembali jika dibutuhkan. Sifat dari pemrograman berorientasi objek adalah memodelkan keadaan dunia nyata kedalam sebuah objek dalam sistem kemudian melakukan interaksi antar objek tersebut.

- Dapat terdistribusi dengan baik

  Bahasa *java* diciptakan untuk membuat program aplikasi dapat terdistribusi secara baik dengan *libraries networking* yang terintegrasi langsung pada *Java*.

- Aman untuk sistem komputer *(Safety)*

  Sebagai bahasa pemrograman untuk aplikasi jaringan internet dan dapat didistribusikan, bahasa *java* memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

- Netral Arsitektur (*Architecture Neutral*)

  Bahasa *java* merupakan platform yang *independent*. Dalam implementasinya, program cukup mempunyai satu buah versi yang dapat

dijalankan pada platform yang berbeda sesuai dengan *Java Virtual Machine*.

- Dapat digunakan diberbagai *platform (Portable)*

  Source code maupun program *java* dapat dengan mudah dibawa ke *platform* yang berbeda tanpa harus dilakukan kompilasi ulang atau perubahan *code*.

- Simultan terhadap program lain *(Multithreading)*

  Bahasa *java* memiliki kemampuan untuk menjalankan program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan dalam waktu yang bersamaan.

- Dinamis

  Bahasa *java* dirancang agar dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa menggangu program yang menggunakan *class* tersebut.

- Robust

  Bahasa *java* mempuyai reliabilitas yang tinggi, karena c*ompiler* pada *Java* mempunyai kemampuan untuk mendeteksi *error* yang terjadi secara lebih teliti dibandingkan bahasa pemrograman lain. *Java* mempunyai *runtime-Exception handling* untuk membantu mengatasi *error* yang terjadi pada saat pemrograman sedang berlangsung.

3.5 Citra Digital

Dalam penelitian yang telah dilakukan, citra adalah suatu representasi kemiripan atau imitasi dari sebuah objek digital. Citra terbagi menjadi 2 kategori yaitu citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, hasil CT Scan dan lainnya. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = picture element), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f (x,y), yaitu besar intensitas atau warna dari piksel di titik itu.Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut (Sutoyo, et al., 2009).

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

Berdasarkan persamaan (1) dapat dijelaskan bahwa, dalam perhitungannya citra digital dapat dituliskan sebagai fungsi intensitas f (x,y), dimana harga x (baris) dan y (kolom) merupakan koordinat posisi dan f (x,y) adalah nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (sampling dan kuantitas) diperoleh besar baris M dan kolom N hingga citra membentuk matriks M x N dan jumlah tingkat keabuan piksel G (Sutoyo, et al., 2009).

3.6 Metode Least Significant Bit

Pada umumnya, metode yang digunakan untuk menyembunyikan pesan pada wadah penampung digital adalah dengan memanfaatkan metode *least significant bit* (LSB). Metode *least significant bit* adalah metode penyisipan informasi dengan cara mengganti *bit* ke-8, 16 dan 24 pada representasi *biner* berkas gambar dengan representasi *biner* pesan rahasia yang akan disembunyikan. Dengan demikian pada setiap *pixel* berkas gambar 24 *bit* dapat disisipkan 3 *bit* pesan (Por, et al., 2008). Walaupun banyak kekurangan pada metode ini, tetapi kemudahan implementasinya membuat metode ini tetap digunakan sampai sekarang. Kekurangan dari metode modifikasi *least significant bit* adalah metode ini membutuhkan tempat penyimpanan yang relatif besar dan pesan yang ingin dimasukkan harus memiliki panjang data yang lebih sedikit daripada media penampung datanya (Reddy, et al., 2013). Sebagai contoh, data raster original berkas gambar adalah sebagai berikut :

00100111 11101001 11001000

00100111 11001000 11101001

11001000 00100111

Sedangkan representasi *biner* huruf A adalah 01100101, dengan menyisipkannya kedalam *pixel* di atas maka akan dihasilkan,

00100110 11101001 11001001

00100110 11001000 11101000

11001000 00100111

Dari contoh diatas dapat terlihat jelas bahwa pada *bit* ke-8, 16 dan 24 diganti dengan representasi *biner* huruf A, dan hanya tiga *bit* rendah yang berubah (cetak tebal), indera manusia sangat sulit untuk dapat membedakan warna pada berkas gambar yang sudah disisipi pesan jika dibandingkan dengan berkas gambar asli sebelum disisipi dengan pesan rahasia (Das , et al., 2008).

3.7 Metode End of File

Menurut penelitian yang telah dilakukan, metode ini digunakan dengan cara menambahkan data atau pesan rahasia pada akhir *file* (Sandro, 2013). Metode *end of file* merupakan salah satu metode yang digunakan dalam teknik steganografi. Dalam implementasinya, metode ini dapat digunakan untuk menambahkan data yang ukurannya sesuai dengan kebutuhan pengguna. Perhitungan kasar ukuran *file* yang telah disisipkan data sama dengan ukuran *file* sebelum disisipkan data ditambah ukuran data tersembunyi yang telah diubah menjadi *encoding file* . Metode *end of file* tidak akan mengubah isi awal dari *file* yang disisipi. Sebagai contoh, jika akan menyisipkan sebuah pesan kedalam sebuah *file* dokumen, isi dari dokumen tersebut tidak akan berubah. Ini yang menjadi salah satu keunggulan metode *end of file* dibandingkan metode steganografi yang lain karena disisipkan pada akhir *file*, pesan yang disisipkan tidak akan bersinggungan dengan isi *file*, hal ini menyebabkan integritas data dari *file* yang disisipi tetap dapat terjaga (Utami & , 2007). Namun, metode *end of file* akan mengubah besar ukuran *file* sesuai dengan ukuran pesan yang disisipkan kedalam *file* awal namun tidak mengubah citra melalui media yang dipakai sebagai tempat penyisipan pesan tersebut (Sari & Rachmawanto, 2014).

3.8 Mean Square Error (MSE)

Mean Squared Error (MSE) adalah nilai yang diharapkan *(expected value)* dari kuadrat *error*. *Error* yang dimaksud adalah jumlah dari perbedaan antara estimator dan kuantitas yang diestimasikan (Cheddad, et al., 2009). *Error* tersebut dapat menjelaskan bahwa seberapa besar perbedaan hasil dari estimasi dengan nilai yang akan diestimasi. Dalam kasus menggunakan teknik steganografi, MSE adalah nilai *error* kuadrat rata-rata antara citra asli (*cover-image*) dengan citra hasil penyisipan (*stego-image*). Nilai MSE yang besar, menyatakan bahwa penyimpangan atau selisih antara citra hasil modifikasi dengan citra aslinya cukup besar. Perbedaan itu terjadi karena adanya keacakan pada data atau karena estimator tidak mengandung informasi yang dapat menghasilkan estimasi yang lebih akurat (Barni, 2006). Untuk menghitung nilai MSE dapat dilihat pada persamaan (2).

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy} - C_{xy}\right)^2 \quad \ldots\ldots\ldots (2)$$

Dimana :

MSE = Nilai *Mean Square Error* dari citra tersebut

m = panjang citra tersebut (dalam piksel)

n = lebar citra tersebut (dalam piksel)

(x,y) = koordinat masing-masing piksel

S = nilai *bit* citra pada koordinat x,y

C = nilai derajat keabuan citra pada koordinat x,y

3.9 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) adalah perbandingan antara nilai maksimum suatu sinyal yang diukur dengan besarnya nilai *error* atau *derau* (*noise*) yang berpengaruh pada sinyal tersebut. PSNR biasanya diukur dalam satuan desibel (dB) (Hidayat & Udayanti, 2011). Dalam penelitian ini, PSNR digunakan untuk mengetahui perbandingan kualitas citra sebelum dan sesudah disisipkan pesan. Semakin besar nilai PSNR citra hasil kompresi, berarti semakin mirip citra tersebut dengan citra asli, sedangkan nilai MSE akan semakin kecil (Hermawati, 2013). Untuk menentukan PSNR, terlebih dulu harus ditentukan nilai rata-rata kuadrat dari *error* MSE (*Mean Square Error*). Formula untuk menghitung PSNR dapat dilihat pada persamaan (2.3):

$$PSNR = 10 \log_{10} \left(\frac{C_{max}^2}{MSE} \right) \quad \ldots\ldots\ldots\ldots (3)$$

Dimana :

PSNR = nilai PSNR citra (dalam dB)

MAX = nilai maksimum piksel

MSE = nilai MSE

3.10 ASCII (American Standard Code for Information Interchange)

American Standard Code for Information Interchange (ASCII) adalah format standard yang sering 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. Format kode ASCII menggunakan format data tujuh *bit* untuk mewakili semua karakter yang ada termasuk tanda baca dan penanda

kontrol. Dengan menggunakan format tujuh 7 *bit* tersebut, maka ASCII dapat menampung 27 = 128 data (Singh & Garg, 2013). Pada masa tersebut hanya memiliki lebar 7-bit saja karena komputer pada awalnya memiliki ukuran memori yang sangat terbatas, dan 128 karakter dianggap memadai untuk menampung semua huruf Latin dengan tanda bacanya, dan beberapa karakter kontrol. ASCII telah dibakukan oleh ANSI (American National Standards Institute) menjadi standar ANSI X3.41986 (Yuwono, et al., 2008).