

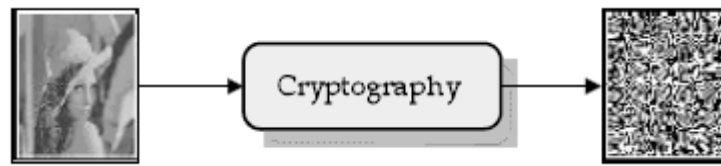
## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Kriptografi**

*Kriptografi* yaitu teknik pengenkripsian pesan. Namun teknik ini dapat menimbulkan kecurigaan karena pesan acak tidak memiliki makna secara kasat mata, sehingga mudah dicurigai (Iza, 2013).

Menurut Rinaldi Munir dalam bukunya yang berjudul *Kriptografi* pada tahun 2006 menjelaskan bahwa *kriptografi* adalah ilmu dan seni untuk menjaga keamanan pesan (*message*). *Kriptografer* (*cryptographer*) adalah praktisinya/pengguna kriptografi. Beberapa istilah yang dipakai didalam teknik *kriptografi* yaitu pengirim pesan (*sender*) dan penerima pesan (*receiver*). Algoritma teknik *kriptografi* adalah aturan untuk enkripsi dan deskripsi dimana enkripsi adalah proses penyandian *plainteks* atau pesan menjadi *cipherteks* (*enchiphering*) dan deskripsi adalah proses pengembalian *cipherteks* menjadi *plaintext* (*deciphering*) (Munir, 2006). Ilustrasi kriptografi disajikan pada Gambar 2.1.



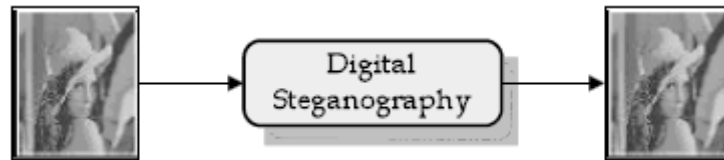
Gambar 2.1 *Kriptografi* (Prayudi dan Kuncoro, 2005).

## 2.2 *Steganografi*

*Steganography* (*steganografi*) merupakan seni untuk menyembunyikan pesan rahasia kedalam pesan lainnya sedemikian rupa sehingga membuat orang lain tidak menyadari adanya sesuatu di dalam pesan tersebut. Kata *Steganography* berasal dari bahasa Yunani, yaitu gabungan dari kata *steganos* (tersembunyi atau terselubung) dan *graphein* (tulisan atau menulis), sehingga makna *Steganography* kurang lebih bisa diartikan sebagai menulis tulisan yang tersembunyi (Sellars, 2006).

Sejalan dengan perkembangan maka konsep awal steganografi diimplementasikan pula dalam dunia komputer, yang kemudian dikenal dengan istilah steganografi digital. Dalam hal ini, steganografi digital memiliki dua properti dasar yaitu media penampung (*cover data* atau *data carrier*) dan data digital yang akan disisipkan (*secret data*), dimana media penampung dan data digital yang akan disisipkan dapat berupa file multimedia (teks/dokumen, citra, audio maupun video). Terdapat dua tahapan umum dalam steganografi digital, yaitu proses *embedding* atau *encoding* (penyisipan) dan proses *extracting* atau *decoding* (pemekaran atau pengungkapan kembali (*reveal*)). Hasil yang didapat setelah proses *embedding* atau *encoding* disebut *stego object* (apabila media penampung hanya berupa data citra maka

disebut *stego image*) (Prayudi dan Kuncoro, 2005). Ilustrasi *steganografi* disajikan pada Gambar 2.2.



Gambar 2.2 *Steganografi* (Prayudi dan Kuncoro, 2005).

Provos dan Honeyman (2003) mendefinisikan *steganografi* adalah ilmu dan seni menyembunyikan dalam komunikasi. Sistem *steganografi* ini menyisipkan konten pada suatu media tanpa menimbulkan kecurigaan. Di masa lalu, orang-orang menggunakan tinta transparan untuk menggunakan teknik *steganografi*. Saat ini, komputer dan jaringan teknologi menyediakan saluran komunikasi yang mudah digunakan untuk steganografi. Pada dasarnya, proses informasi-bersembunyi dalam sistem steganografi dimulai dengan mengidentifikasi *bit* yang berlebihan pada media penutup (dapat dimodifikasi tanpa merusak integritas medium). Proses penyisipan menciptakan media stego dengan mengganti bit-bit yang berlebihan dengan data dari disembunyikan yaitu pesan. Tujuan modern steganografi adalah untuk menjaga data rahasia tidak terdeteksi dengan media penutup (*cover*). Media tersebut dapat terlihat perbedaannya dengan menemukan distorsi pada media, proses menemukan distorsi ini disebut *steganalysis statistik*. (Provos dan Honeyman, 2003).

### 2.2.1 Kriteria *Steganografi*

Kriteria *steganografi* menurut Rinaldi Munir dalam bukunya yang berjudul Pengolahan Citra Digital dengan Pendekatan Algoritmik pada tahun 2004 adalah :

1. *fidelity*. Mutu media penampung tidak jauh berubah. Setelah penambahan data rahasia, *stego object* dalam kondisi yang masih terlihat baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2. *robustness*. Data rahasia yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi atau *editing* pada media penampung. Apabila pada media penampung dilakukan operasi manipulasi atau *editing*, maka data yang disembunyikan seharusnya tidak rusak atau tetap valid.
3. *recovery*. Data yang disembunyikan harus dapat di ungkapkan kembali (*reveal*), karena dikaitkan dengan tujuan dari steganografi digital itu sendiri yaitu *data* sewaktu-waktu data rahasia di dalam media penampung harus dapat diambil kembali untuk digunakan lebih lanjut (Munir, 2004).

### 2.3 Citra *Digital*

Citra atau gambar dapat didefinisikan sebagai sebuah fungsi yang terdiri dari dua variabel  $f(x,y)$ , dengan  $x$  dan  $y$  adalah koordinat bidang datar, dan harga fungsi  $f$  disetiap pasangan koordinat  $(x,y)$  disebut intensitas atau *level* keabuan (*grey level*) dari gambar di titik itu. Jika  $x,y$  dan  $f$  semuanya berhingga (*finite*), dan nilainya diskrit, maka gambarnya disebut citra *digital* (gambar *digital*). Sebuah citra *digital* terdiri dari sejumlah elemen berhingga, dimana masing-masing mempunyai lokasi dan nilai tertentu. Elemen-elemen ini disebut sebagai *picture element*, *image element*, *pels* atau *pixels* (Hermawati, 2013).

Purnomo dan Muntasa (2010) menjelaskan bahwa nilai dari intensitas bentuknya atau *level* keabuan adalah diskrit mulai dari 0 sampai 255. Citra yang ditangkap oleh kamera dan telah dikuantisasi dalam bentuk nilai diskrit disebut dengan citra *digital* (*digital image*) (Purnomo dan Muntasa, 2010).

### **2.3.1 Jenis File Citra Digital**

Bermacam-macam jenis file citra *digital* yaitu jpg/jpeg, png, gif, tiff dan lain sebagainya mempunyai ciri-ciri, karakteristik, keunggulan dan kelemahan dibagiannya. Disetiap jenis *file digital* terdapat kompresi atau dapat diartikan sebagai proses reduksi jumlah data yang diperlukan untuk menyatakan suatu jumlah informasi yang diberikan (Hermawati, 2013).

Teknik-teknik kompresi data dapat dibagi menjadi 2 yaitu (Widhiartha, 2008) :

#### *1. lossy compression.*

*Lossy compression* menyebabkan adanya perubahan data dibandingkan sebelum dilakukan proses kompresi. Sebagai gantinya *lossy compression* memberikan derajat kompresi lebih tinggi. Tipe ini cocok untuk kompresi file suara digital dan gambar digital. File suara dan gambar secara alamiah masih bisa digunakan walaupun tidak berada pada kondisi yang sama sebelum dilakukan kompresi.

#### *2. lossless compression.*

Sebaliknya *Lossless Compression* memiliki derajat kompresi yang lebih rendah tetapi dengan akurasi data yang terjaga antara sebelum dan sesudah proses kompresi. Kompresi ini cocok untuk basis data, dokumen atau *spreadsheet*. Pada

*lossless compression* ini tidak diijinkan ada bit yang hilang dari data pada proses kompresi.

Jenis-jenis file citra *digital* (Ichsan, 2011).

#### 1. JPG/JPEG (*Joint Photographic Experts Group*).

*Joint Photographic Experts Group* (JPEG) adalah format gambar yang banyak digunakan untuk menyimpan gambar-gambar dengan ukuran lebih kecil.

Beberapa karakteristik gambar JPEG adalah sebagai berikut :

- Memiliki ekstensi .jpg atau .jpeg.
- Mampu menayangkan warna dengan kedalaman 24-bit *true color*.
- Mengkompresi gambar dengan sifat *lossy*.
- Umumnya digunakan untuk menyimpan gambar-gambar hasil foto.

#### 2. PNG (*Portable Network Graphics*).

PNG (*Portable Network Graphics*) adalah salah satu format penyimpanan citra yang menggunakan metode pemadatan yang tidak menghilangkan bagian dari citra tersebut (*lossless compression*). PNG dibaca "ping", namun biasanya dieja apa adanya untuk menghindari kerancuan dengan istilah "ping" pada jaringan komputer. Format PNG ini diperkenalkan untuk menggantikan format penyimpanan citra GIF. Secara umum PNG dipakai untuk Citra Web.

#### 2.3.2 Konsep Citra Warna (*True Color*)

Setiap *pixel* pada citra warna yang merupakan kombinasi dari tiga warna dasar (RGB=*Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti

setiap *pixel* mempunyai kombinasi warna sebanyak 16 juta warna lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

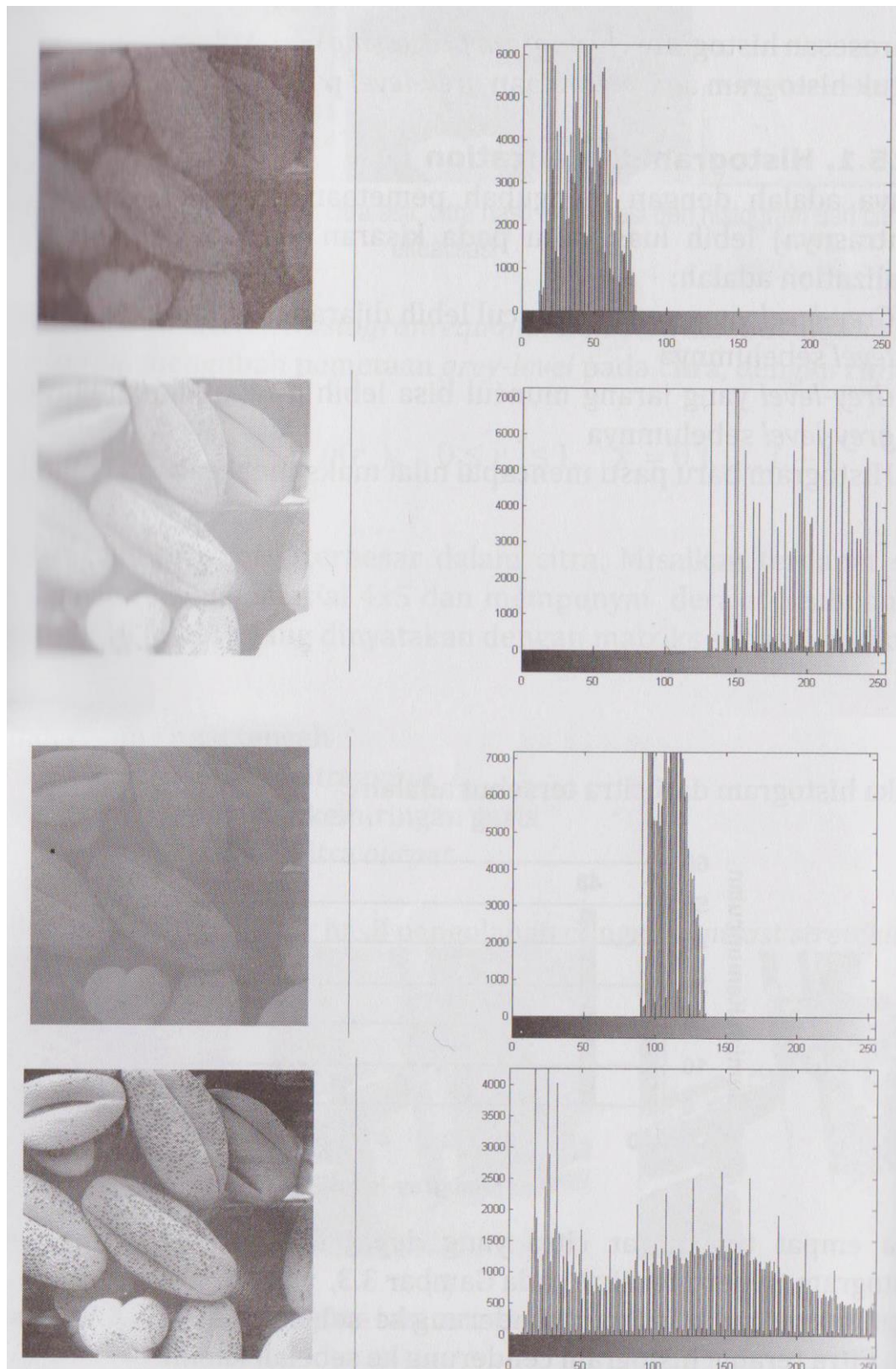
Penyimpanan citra *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap *pixel* dari citra *grayscale* 256 gradasi warna diwakili oleh 1 *byte*. Sedangkan 1 *pixel* citra *true color* diwakili oleh 3 *byte*, dimana masing-masing *byte* mempresentasikan warna merah (*Red*), hijau (*Green*), dan biru (*Blue*) (Widhiartha, 2008).

### 2.3.3 Histogram

Histogram adalah diagram yang menunjukkan jumlah kemunculan nilai *grey level* (Kecerahan) pada suatu citra, dimana sumbu-x dari diagram ini menggambarkan nilai *grey level* (Kecerahan) dan sumbu-y mewakili jumlah kemunculan *grey level* (Kecerahan) tertentu. Ada 4 tipe dasar citra yang dapat digambarkan dengan histogram yaitu (Hermawati, 2013) :

- Citra gelap: histogram cenderung ke sebelah kiri
- Citra terang: histogram cenderung ke sebelah kanan
- Citra *low contrast*: histogram mengumpul di suatu tempat
- Citra *high contrast*: histogram merata di semua tempat

Histogram pada citra digital disajikan pada Gambar 2.3.



Gambar 2.3 4 Tipe Dasar Histogram (Hermawati, 2013).



## 2.4 Teks Sederhana (*Plain Text*)

Format data teks (.txt) merupakan contoh format teks yang paling populer. Saat ini, perangkat lunak yang paling banyak digunakan untuk memanipulasi format data ini adalah Notepad. Format data teks (.txt) adalah format teks yang digunakan untuk menyimpan huruf, angka, karakter kontrol (tabulasi, pindah baris, dan sebagainya) atau simbol-simbol lain yang biasa digunakan dalam tulisan seperti titik, koma, tanda petik, dan sebagainya. Satu huruf, angka, karakter kontrol atau simbol pada arsip teks memakan tempat satu byte. Berbeda dengan jenis teks terformat (.doc) yang satu huruf saja dapat memakan tempat beberapa byte untuk menyimpan format dari huruf tersebut seperti font, ukuran, tebal atau tidak dan sebagainya. Kelebihan dari format data teks ini adalah ukuran datanya yang kecil karena tidak adanya fitur untuk memformat tampilan teks (Purnomo dan Zacharias, 2005).

## 2.5 Metode AMELSB

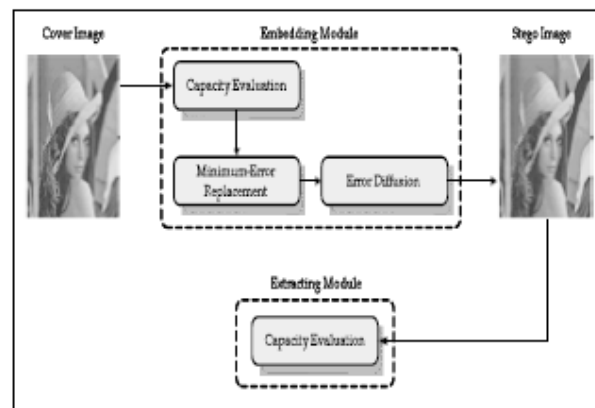
Metode ini pertama kali diperkenalkan oleh Yeuan-Kuen Lee dan Ling-Hwei Chen pada tahun 1999 dalam dua makalahnya “*An Adaptive Image Steganographic Model Based on Minimum- Error LSB Replacement*” dan “*High Capacity Image Steganographic Model*” (Lee dan Chen, 1999). Di dalam kedua makalahnya, Lee dan Chen menerapkan citra hitam-putih (*grayscale image*) sebagai media penampung (*cover image*) dan kemudian pada tahun 2003, Mark David Gan mengimplementasikan metode ini dengan citra berwarna *24 bit (true colors image)* sebagai media penampungnya (Gan, 2003).

Dari hasil penelitian tersebut ternyata metode ini menawarkan beberapa kelebihan dibandingkan dengan metode LSB, yaitu *bit* data rahasia yang akan disisipkan lebih banyak (pada metode LSB umumnya hanya 1 *bit*) tanpa menimbulkan banyak perubahan pada media penampung (dalam hal ini adalah data citra). Dengan metode ini, setiap *pixel* memiliki kapasitas penyembunyian yang berbeda-beda tergantung dari nilai toleransi *pixel* tersebut terhadap proses modifikasi atau penyisipan. Suatu *pixel* pada data citra bisa dikatakan dapat ditoleransi apabila dilakukan proses modifikasi (penyisipan) dengan skala yang tinggi terhadap nilainya adalah memungkinkan tanpa merubah tampak asli dari data citra tersebut, atau dengan kata lain area yang halus dan solid pada suatu data citra memiliki kadar toleransi yang rendah (*less tolerant*) terhadap proses modifikasi dibandingkan dengan area yang memiliki tekstur yang kompleks (Gan, 2003).

Metode AMELSBP yang diterapkan pada citra berwarna (*jpg/jpeg 24-bit*) memiliki beberapa langkah atau tahapan utama untuk melakukan proses penyisipan, antara lain *Capacity Evaluation*, *Minimum Error Replacement* dan *Error Diffusion* (Gan, 2003). Untuk proses pengungkapan, tahapan yang dilakukan yaitu *Capacity Evaluation* (Lee dan Chen, 1999).

Sebelum dilakukan proses penyisipan, maka langkah pertama yang harus dilakukan adalah mengevaluasi kapasitas penyisipan (*capacity evaluation*) dan mencari nilai *color variation*. Kemudian setelah mendapatkan nilai *color variation*, nilai tersebut diproses kembali untuk mendapatkan kapasitas penyisipan sejumlah *K-bit*. Setelah itu, untuk beradaptasi dengan karakteristik lokal *pixel*, maka sejumlah *K-bit*

tersebut ditangani dengan proses evaluasi kapasitas (*capacity evaluation*). Proses selanjutnya adalah mencari *MER*, dimana proses ini akan menentukan apakah *bit* ke  $K+1$  akan dilakukan perubahan atau tidak, dan yang akan menentukan itu adalah berdasarkan pada nilai *embedding error* ( $E_r$ ). Proses tersebut disajikan pada Gambar 2.4.



Gambar 2.4 Gambaran Umum Metode AMELSB (Gan, 2003).

Proses penyisipan (*embedding*) di dalam metode AMELSB, prosesnya tidak sama dengan metode LSB. Apabila proses penyisipan di dalam metode LSB dilakukan langsung per *pixel* pada *byte*-nya, dimana 1 *bit* terakhir (LSB) per *byte*-nya diganti dengan 1 *bit* data rahasia yang akan disisipkan, tetapi tidak dengan metode AMELSB. Di dalam metode ini, citra penampung (*cover image*) akan dibagi dulu menjadi beberapa blok. Setiap blok akan berukuran  $3 \times 3$  *pixel* atau sama dengan 9 *pixel* (Bailey, et al; 2004).

Ketiga tahapan utama akan diterapkan per bloknya atau per operasi penyisipannya, dimana *bit-bit* data rahasia hanya akan disisipkan pada salah satu komponen warna di *pixel P*, dan disajikan pada Gambar 2.5.

|                |              |                |
|----------------|--------------|----------------|
| B<br>(x-1,y-1) | C<br>(x-1,y) | D<br>(x-1,y+1) |
| A<br>(x,y-1)   | P<br>(x,y)   | E<br>(x,y+1)   |
| H<br>(x+1,y-1) | G<br>(x+1,y) | F<br>(x+1,y+1) |

Gambar 2.5 *Pixel* Tetangga dari *Pixel* P (Lee dan Chen, 1999).

*Capacity evaluation*, merupakan tahap pertama dan yang paling krusial dari metode penyisipan AMELSB. Tahap ini mengacu pada karakteristik *human visual system* (HVS) yang tidak sensitif terhadap *noise* dan perubahan warna yang terdapat di dalam citra (Lee dan Chen, 1999). Langkah pertama yang akan dilakukan pada evaluasi kapasitas adalah mencari nilai *color variation* ( $V$ ) atau variasi warna yang melibatkan *pixel* A, B, C dan D. Adapun rumus dari  $V$  adalah sebagai berikut (Gan, 2003)

$$V = \text{round} \{(|C-A| + |A-B| + |B-C| + |C-D|)/4\}$$

dimana :

$V$  = variasi warna (*color variation*)

*Round* = fungsi matematika untuk pembulatan

Rumus di atas akan menghasilkan ketentuan toleransi modifikasi yang akurat di setiap *pixel* P. Langkah ke-dua adalah mencari kapasitas penyisipan ( $K$ ) pada *pixel* P dan dapat diterapkan rumus sebagai berikut (Gan, 2003)

$$K = \text{round} (|\log_2 V|)$$

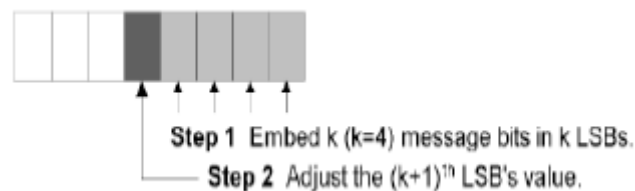
dimana :

$K$  = kapasitas penyisipan pada *pixel* P dalam *bit*.

$V$  = variasi warna

$Round$  = fungsi matematika untuk pembulatan

Tahap selanjutnya adalah mencari *Minimum-Error Replacement* (MER). Tahap ini berfungsi untuk meminimalkan terjadinya perubahan *pixel* pada citra penampung akibat dari proses penyisipan. Proses MER dilakukan dengan mengubah nilai *bit* ke  $K+1$  pada *pixel*  $P$ . Perubahan ini akan terjadi pada salah satu dari ke-tiga komponen warna (R, G atau B) yang terpilih (Lee dan Chen, 1999). Proses ini disajikan pada Gambar 2.6.



Gambar 2.6 Proses MER (Lee dan Chen, 1999).

Bila pada langkah sebelumnya (evaluasi kapasitas) didapat  $K = 4$ , maka *bit* yang ke-lima akan diubah nilainya, misal nilai awal adalah 1, maka akan diubah menjadi 0, begitu juga sebaliknya. Namun demikian pengubahan *bit* ke  $K+1$  belum tentu dilakukan, karena pada tahap MER juga dilakukan proses pengecekan nilai *embedding error*. *Embedding error* ( $Er$ ) adalah selisih nilai (dalam desimal) pada komponen warna yang terpilih di *pixel*  $P$ , sebelum (original) dan sesudah dilakukan proses penyisipan, atau dengan rumus seperti di bawah ini

$$Er = Abs [P(x,y) - P'(x,y)]$$

dimana :

$Abs$  = Nilai absolut

$Er$  = Nilai *embedding error*

$P(x,y)$  = *Pixel P* asli

$P'(x,y)$  = *Pixel P* yang telah dimodifikasi

Pengubahan pada *bit* ke  $K+1$  akan dilakukan apabila nilai *embedding error* memenuhi syarat pada saat pengecekan, uraiannya bisa dijelaskan sebagai berikut.

Asumsi  $P(x,y)$  adalah *pixel P* original,  $P'(x,y)$  adalah *pixel P* yang telah disisipkan sejumlah  $K$ -bit tanpa mengubah *bit* ke  $K+1$  dan  $P''(x,y)$  adalah *pixel P* yang telah disisipkan sejumlah  $K$ -bit sekaligus mengubah *bit* ke  $K+1$ . *Minimum error* yang dapat terjadi di *pixel P* haruslah  $P'(x,y)$  atau  $P''(x,y)$  (Lee dan Chen, 1999).

Kemudian proses pengecekan nilai *embedding error* dilakukan lewat rumus sebagai berikut

$$Er1 = Abs [P(x,y) - P'(x,y)]$$

$$Er2 = Abs [P(x,y) - P''(x,y)]$$

Apabila  $Er1 < Er2$ , maka  $P'(x,y)$  yang akan menggantikan  $P(x,y)$ . Jika sebaliknya maka  $P''(x,y)$  yang akan menggantikan  $P(x,y)$  (Lee dan Chen, 1999).

## 2.6 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini disebabkan karena UML menyediakan Bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

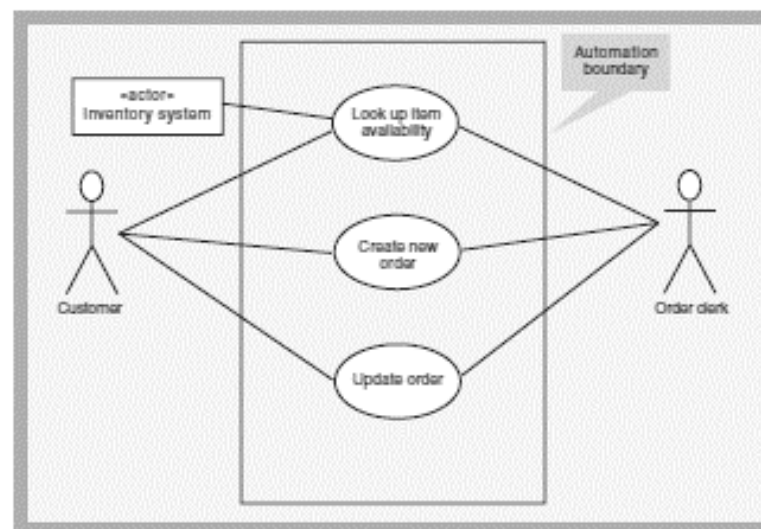
UML adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerja bersama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. Dengan UML dapat diceritakan apa yang seharusnya dilakukan oleh suatu sistem bukan bagaimana yang seharusnya dilakukan oleh suatu sistem (Munawar, 2005).

*Unified Modeling Language* (UML) dideskripsikan oleh beberapa diagram yaitu :

1. *Use case diagram*.

Diagram *use case* menyajikan interaksi antara *use case* dan actor. Dimana, actor berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai (Sholih, 2006).

Berikut contoh diagram *use case* yang disajikan pada Gambar 2.7.

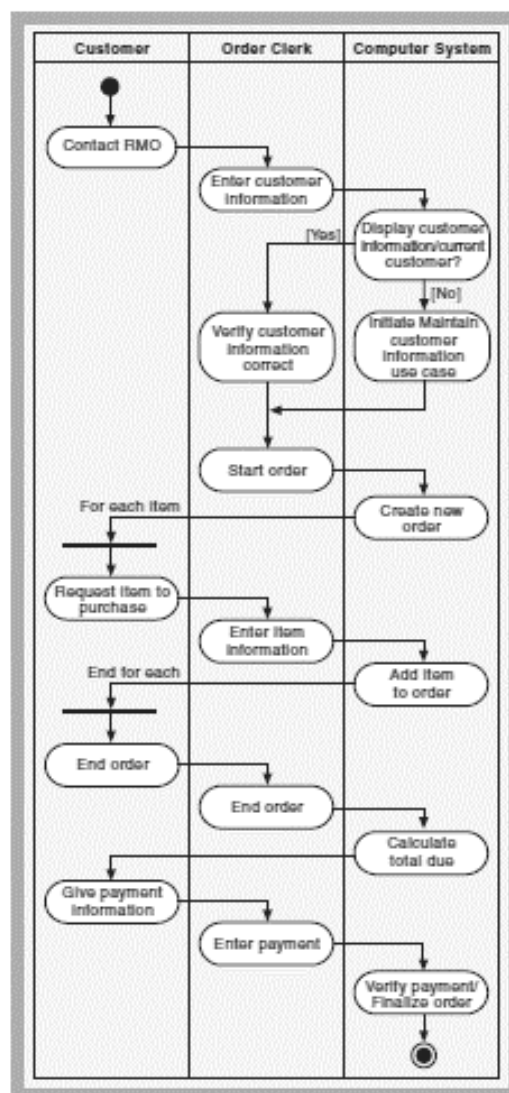


Gambar 2.7 *Use Case Diagram* (Satzinger, 2009).

2. *Diagram aktivitas*.

Diagram aktivitas atau *Activity Diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow* serta dapat digunakan untuk

menggambarkan aliran kejadian (*flow of events*) dalam *use case*. Aktivitas dalam diagram dipresentasikan dengan bentuk bujur sangkar bersudut tidak lancip, yang didalamnya berisi langkah-langkah apa saja yang terjadi dalam aliran kerja. Ada keadaan mulai (*start state*) yang menunjukkan dimulainya aliran kerja, dan keadaan selesai (*end state*) yang menunjukkan akhir diagram, dan titik keputusan dipresentasikan dengan *diamond* (Sholih, 2006). Contoh diagram aktivitas disajikan pada Gambar 2.8.

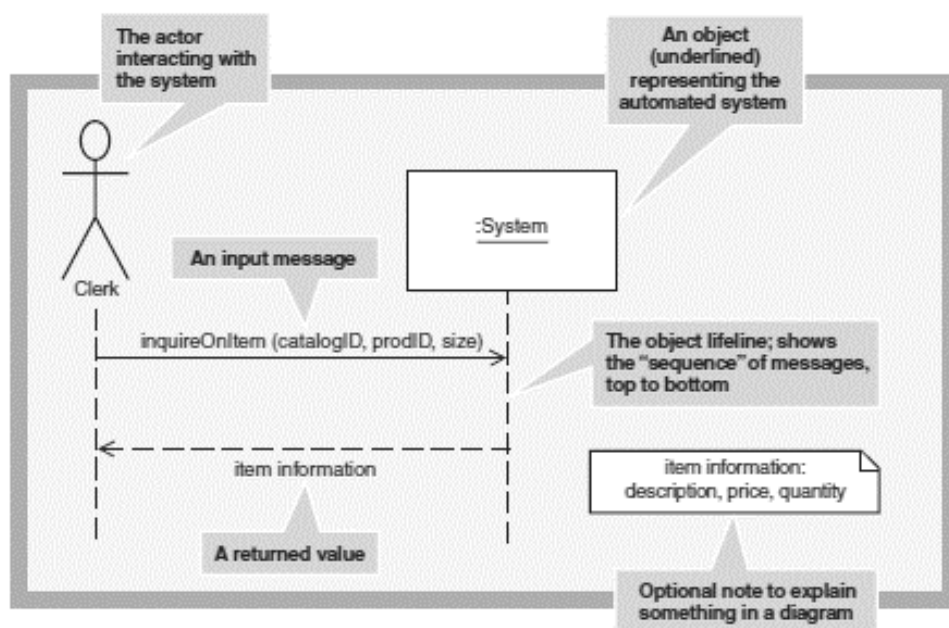


Gambar 2.8 Diagram Aktivitas (Satzinger, 2009).



### 3. Diagram sekuensial.

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Misalkan, pada *use case* “menarik uang” mempunyai beberapa kemungkinan, seperti penarikan uang secara normal, percobaan penarikan uang tanpa kecukupan ketersediaan dana, penarikan dengan penggunaan PIN yang salah, dan lainnya (Sholih, 2006). Contoh diagram sekuensial disajikan pada Gambar 2.9.

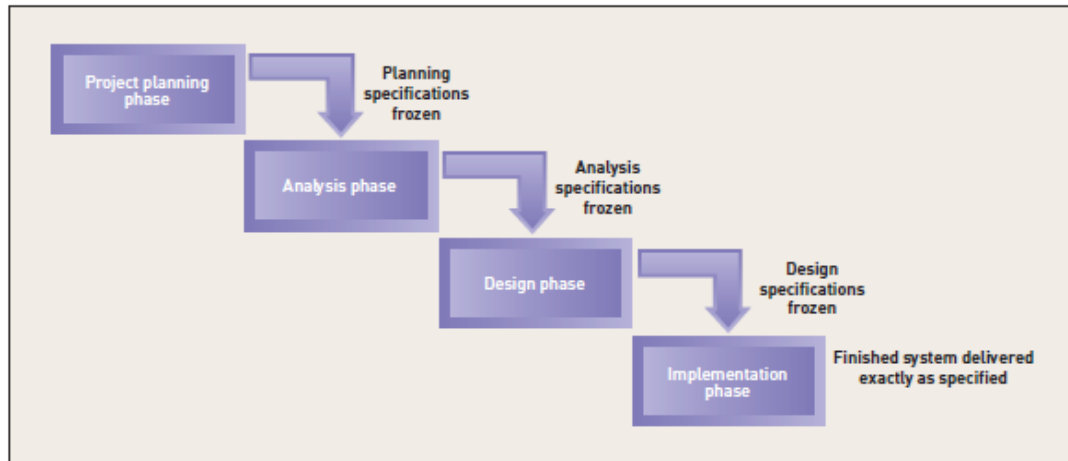


Gambar 2.9 Diagram Sekuensial (Satzinger, 2009).

## 2.7 Metode Pengembangan Sistem *Waterfall*

Metode pengembangan sistem *waterfall* adalah pendekatan SDLC yang penyelesaian proyeknya diselesaikan dengan tahapan-tahapan yang berurutan. Tahap-tahap pada metode *waterfall* adalah perencanaan sistem, analisis kebutuhan, desain dan implementasi (Satzinger, et al; 2010).

Tahapan-tahapan dalam metode pengembangan sistem *waterfall* disajikan pada Gambar 2.10.



Gambar 2.10 *Waterfall* (Satzinger, et al; 2010).