BAB 2
LANDASAN TEORI

Landasan teori menjelaskan dasar-dasar teori yang digunakan dalam penelitian yang dilakukan pada tugas akhir. Teori-teori yang digunakan meliputi keamanan informasi, citra digital, Pengukuran *error* pada citra, steganografi, kriptografi, *Unified Modelling Language* (UML), dan teknologi java.

2.1. Keamanan Informasi

Keamanan informasi merupakan salah satu aspek yang penting dalam berkomunikasi, namun keamanan seringkali dilupakan ketika melakukan komunikasi. Informasi yang jatuh ke tangan yang salah dapat menimbulkan masalah yang besar seperti penipuan, pencurian, pemerasan dan masih banyak lagi masalah yang akan ditimbulkan.

Keamanan informasi mempunyai beberapa aspek yang harus dipenuhi, agar informasi dapat terjamin keaslian dan keamanannya. Aspek-aspek umum tersebut meliputi *confidentiality, integrity, authentication*, *availability* dan *non repudiation*[2].

1. *Confidentiality*

Confidentiality atau kerahasiaan yaitu keamanan informasi harus dapat menjamin kerahasiaan dari informasi. Salah satu cara yang dapat dilakukan yaitu dengan membatasi hanya orang yang mempunyai hak saja yang dapat membaca atau mengubah suatu informasi.

2. *Integrity*

Integrity berhubungan dengan keaslian informasi, *integrity* yaitu menjamin keutuhan dan keaslian informasi yang dikirimkan, agar informasi tidak dirusak atau diubah oleh orang yang tidak berhak. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya

3. *Authentication*

Authentication yaitu usaha atau metode untuk mengetahui keaslian dari informasi, dan memastikan bahwa informasi diterima oleh orang yang benar. Untuk menjaga otentikasi terhadap informasi, dapat digunakan digital *signature* untuk memastikan keaslian informasi.

4. *Availability*

Availability atau ketersediaan data yaitu keamanan informasi harus dapat menjamin bahwa data atau informasi harus tersedia ketika akan digunakan. Untuk menjamin ketersediaan data, sistem harus mempunyai cadangan data jika data tersebut hilang.

5. *Non Repudiation*

Non repudiation yaitu menjamin bahwa seorang pengirim informasi tidak dapat menyangkal keaslian dari informasi yang dikirimnya. Sehingga penerima informasi dapat memastikan bahwa informasi yang diterima merupakan informasi yang asli.

2.2. Citra digital

Citra secara harafiah, adalah gambar pada bidang dua dimensi (dwimatra). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang dwimatra. Sedangkan citra digital adalah citra yang dapat diolah oleh komputer[4].

Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut *pixel*, yaitu elemen terkecil dari sebuah citra. Sebuah citra digital dapat ditulis dalam bentuk fungsi 2.1 berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & .. & f(0,M-1) \\ f(1,0) & .. & .. & f(1,M-1) \\ .. & .. & .. & .. \\ f(N-1,0) & f(N-1,1) & .. & f(N-1,M-1) \end{bmatrix} \quad \text{..... (2.1)}$$

Terdapat banyak format *file* dari citra digital, format *file* citra ini terdiri dari beberapa jenis dan setiap jenisnya memiliki karakteristik masing- masing. Beberapa contoh format umum, pada citra digital yaitu Bitmap (BMP), *Joint Photographic*

Group Experts (JPEG), *Graphics Interchange Format* (GIF), dan *Portable Network Graphics* (PNG).

1. Bitmap (BMP)

Bitmap merupakan format baku citra pada sistem operasi windows dan IBM OS/2. Citra berformat BMP merupakan citra yang tidak terkompresi, sehingga pada umumnya citra berformat BMP mempunyai ukuran yang relatif lebih besar dibandingkan dengan forman citra lainnya. Intensitas *pixel* dari citra berformat BMP dipetakan ke sejumlah bit tertentu. Panjang setiap *pixel* pada bitmap yaitu 4 bit, 8 bit, sampai 24 bit yang merepresentasikan nilai intensitas *pixel*. Dengan demikian ada sebanyak 28 = 256 derajat keabuan, mulai dari 0 sampai 255[4].

2. Joint Photographic Group *Experts* (JPEG)

Joint Photographic Group Experts (JPEG) merupakan standar kompresi *file* yang dikembangkan oleh *Group Joint Photographic Experts* menggunakan kombinasi DCT dan pengkodean *Huffman* untuk mengkompresikan citra. Citra JPEG merupakan citra terkompresi yang bersifat *lossy*, artinya citra tidak bias dikembalikan ke bentuk aslinya. Citra ini memiliki ukuran yang relatif lebih kecil dibandingkan dengan citra berformat BMP karena telah terkompresi.

3. *Portable Network Graphics* (PNG)

Portable Network Graphics (PNG) adalah salah satu format penyimpanan citra yang menggunakan metode kompresi yang tidak menghilangkan bagian dari citra tersebut (*lossless compression*). Citra berformat PNG merupakan salah satu format yang baik untuk digunakan pengolahan citra, karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah.

2.3. Pengukuran Error Citra

Pengukuran *error* citra dilakukan untuk mengetahui tingkat kesamaan antara citra asli dengan citra yang telah dimanipulasi. Pengukuran error dilakukan dengan menghitung *Mean Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR) dari citra.

Mean Square Error (MSE) adalah tingkat kesalahan *pixel* - *pixel* citra hasil dari pemrosesan terhadap citra aslinya. Rumus untuk menghitung MSE pada citra digital yaitu menggunakan persamaan II.2. Sedangkan *Peak Signal to Noise Ratio*

(PSNR) merupakan nilai (rasio) yang menunjukan tingkat toleransi *noise* tertentu terhadap banyaknya *noise* pada suatu citra. Semakin tinggi nilai PSNR dari suatu citra, maka semakin kecil tingkat kesalahan yang dimiliki citra tersebut. Untuk menghitung nilai PSNR digunakan persamaan 2.3

$$MSE = \frac{1}{MN} \sum_{y=1}^{m} \sum_{x=1}^{n} [I(x,y) - I'(x,y)]^2 \quad \ldots\ldots\ldots(2.2)$$

Dimana: I(x,y) nilai *pixel* dari citra asli.

I'(x,y) nilai *pixel* pada citra stegano.

M = Panjang citra stegano (dalam *pixel*)

N = Lebar citra stegano (dalam *pixel*)

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

$$PSNR = 10.LOG\left(\frac{MAXt^2}{MSE}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2.3)$$

Semakin besar nilai PSNR dari citra, maka citra yang telah dimanipulasi semakin mendekati citra aslinya, dengan kata lain semakin bagus kualitas citra hasil manipulasi tersebut. Sebaliknya, semakin kecil nilai PSNR semakin berkurang kualitas citra hasil manipulasi. Nilai PSNR pada umumnya berada pada rentang 20 - 40 db[6].

2.4. Steganografi

Steganografi adalah suatu ilmu, teknik dan seni tentang bagaimana menyembunyikan data rahasia didalam wadah (media) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang lain. Steganografi membutuhkan dua properti, yaitu media penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya

citra (gambar), suara *(audio*), teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video, Steganografi dapat dipandang sebagai kelanjutan kriptografi. Jika pada kriptografi data yang telah disandikan (*ciphertext*) tetap tersedia, maka dengan steganografi *ciphertext* dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya. Data

rahasia yang disembunyikan dapat diekstraksi kembali persis sama seperti keadaan aslinya.

Pada umumnya, pesan steganografi muncul dengan bentuk lain seperti gambar, artikel, daftar belanjaan, atau pesan-pesan lainnya. Pesan yang tertulis ini merupakan tulisan yang menyelubungi atau menutupi.

Contohnya, suatu pesan bisa disembunyikan dengan menggunakan tinta yang tidak terlihat diantara garis-garis yang kelihatan. Teknik steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau gambar) didalam berkas-berkas lain yang mengandung teks, *image*, bahkan audio tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari berkas semula.

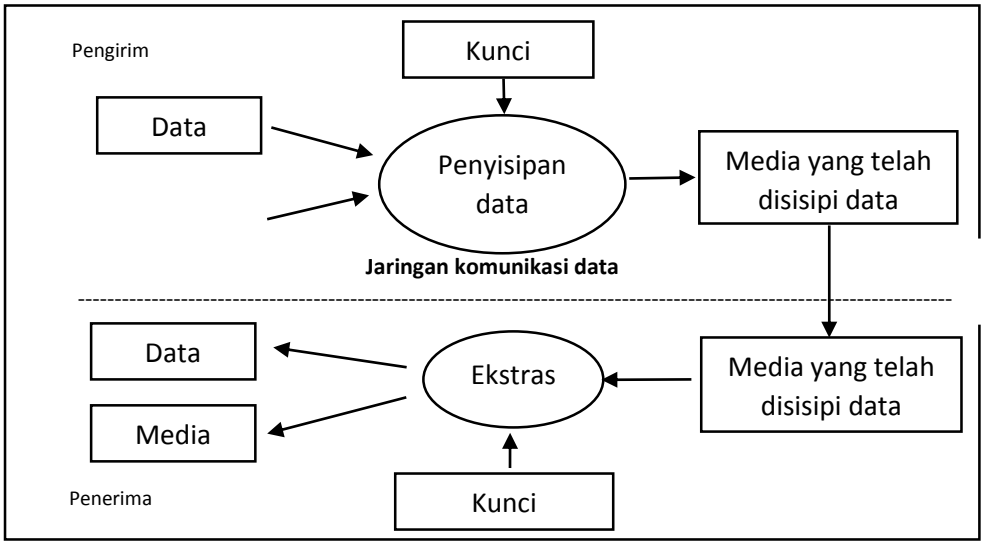
Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya, kebanyakan pesan disembunyikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya. Perubahan ini bergantung pada kunci (sama pada kriptografi) dan pesan untuk disembunyikan. Orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan. Pada metode steganografi cara ini sangat berguna jika digunakan pada cara steganografi komputer karena banyak format berkas digital yang dapat dijadikan media untuk menyembunyikan pesan. Format yang biasa digunakan di antaranya:

1. Format *image* : *bitmap (bmp), gif, png, jpeg*, dll.
2. Format audio : *wav, voc, mp3*, dll.
3. Format lain : teks *file, html, pdf*, dll.

Kelebihan steganografi jika dibandingkan dengan kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan. Seringkali, steganografi dan kriptografi digunakan secara bersamaan untuk menjamin keamanan pesan rahasianya. Sebuah pesan

steganografi *(plaintext)*, biasanya pertama-tama dienkripsikan dengan beberapa arti tradisional, yang menghasilkan *ciphertext.* Kemudian, *covertext* dimodifikasi dalam beberapa cara sehingga berisi ciphertext, yang menghasilkan *stegotext.* Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik *covertext* lainnya dapat dimanipulasi untuk membawa pesan tersembunyi hanya penerima (yang harus mengetahui teknik yang digunakan) dapat membuka pesan dan mendekripsikannya.

Steganografi mempunyai dua proses utama yaitu *embed*/penyisipan dan *exstrak*(pengungkapan) terlihat pada gambar 2.1. Proses penyisipan merupakan proses menyisipkan *hidden object* atau informasi pesan yang akan disisipkan, ke dalam sebuah *cover object* atau media penampung, sehingga menghasilkan *file* baru yang telah tersisipi pesan didalamnya yang disebut dengan stego *file*. Sedangkan proses ekstrak merupakan proses pengembalian *hidden object* secara utuh setelah disisipkan ke dalam *cover object*.



Gambar 2.1 Proses Steganografi

Ada beberapa Kriteria-kriteria yang harus dipenuhi dalam pembuatan steganografi. Kriteria - kriteria tersebut yaitu[3] :

1. *Impercepbility*, yaitu keberadaan pesan tidak dapat dipersepsi oleh indrawi. Jika pesan disisipkan ke dalam sebuah citra, citra yang telah disisipi pesan harus tidak dapat dibedakan dengan citra asli oleh mata. Begitu pula dengan suara,telinga harus mendapati perbedaan antara suara asli dan suara yang telah disisipi pesan.

2. *Fidelity*, yaitu mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan yang terjadi harus tidak dapat dipersepsi oleh indra.
3. *Recovery*, yaitu pesan yang disembunyikan harus dapat diungkap kembali. Tujuan steganografi adalah menyembunyikan informasi, maka sewaktu-waktu informasi yang disembunyikan harus dapat diambil kembali untuk dapat digunakan lebih lanjut sesuai keperluan.

2.4.1. Sejarah Steganografi

Sejarah steganografi cukup panjang. Awalnya adalah penggunaan *hieroglyphic* oleh bangsa Mesir, yakni menulis menggunakan karakter-karakter dalam wujud gambar. Tulisan Mesir kuno tersebut menjadi ide untuk membuat pesan rahasia saat ini. Oleh karena itulah, tulisan mesir kuno yang menggunakan gambar dianggap sebagai steganografi pertama di dunia.[5]

Menurut penelitian para ahli, Yunani termasuk bangsa yang menggunakan steganografi setelah bangsa Mesir. Herodotus mendokumentasikan konflik antara Persia dan Yunani pada abad ke-50 sebelum masehi. Dokumentasi pada masa Raja Xerxes, raja dari Persia, disimpan di Yunani menggunakan steganografi. Berikut adalah beberapa contoh penggunaan teknik steganografi klasik[6]:

1. Abad ke-15 orang Italia menggunakan tawas dan cuka untuk menulis pesan rahasia diatas kulit telur. Kemudian telur tersebut direbus hingga "tinta" yang ada meresap dan tidak terlihat pada kulit telur. Penerima pesan cukup mengupas kulit telur tersebut untuk membaca pesan.
2. Selama terjadinya Perang Dunia ke-2, tinta yang tidak tampak (invisible ink) telah digunakan untuk menulis informasi pada lembaran kertas sehingga saat kertas tersebut jatuh di tangan pihak lain hanya akan tampak seperti lembaran kertas kosong biasa.
3. Pada sejarah Yunani kuno, masyarakatnya biasa menggunakan seorang pembawa pesan sebagai perantara pengiriman pesan. Pengirim pesan tersebut akan dicukur rambutnya, untuk kemudian dituliskan suatu pesan pada kepalanya yang sudah botak. Setelah pesan dituliskan, pembawa pesan harus menunggu hingga rambutnya tumbuh kembali sebelum dapat mengirimkan

pesan kepada pihak penerima. Pihak penerima kemudian akan mencukur rambut pembawa pesan tersebut untuk melihat pesan yang tersembunyi.

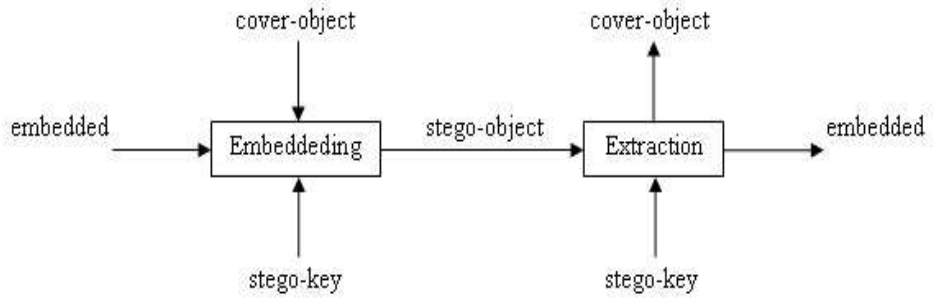
4. Metode lain yang digunakan oleh masyarakat Yunani kuno adalah dengan menggunakan lilin sebagai media penyembunyi pesan mereka. Pesan dituliskan pada suatu lembaran, dan lembaran tersebut akan ditutup dengan lilin untuk menyembunyikan pesan yang telah tertulis. Pihak penerima kemudian akan menghilangkan lilin dari lembaran tersebut untuk melihat pesan yang disampaikan oleh pihak pengirim.

2.4.2. Konsep dan Terminologi Steganografi

Terdapat beberapa istilah yang berkaitan dengan steganografi [7], yaitu:

1. *Hiddentext* atau *embedded message*: pesan yang disembunyikan.
2. *Cover-object*: pesan yang digunakan untuk menyembunyikan *embedded message*.
3. *Stego-object*: pesan yang sudah berisi *embedded message*.

Di dalam steganografi digital, baik *embedded message* maupun *cover-object* dapat berupa teks, citra, *audio*, maupun *video*. Penyisipan pesan ke dalam media *cover-object* dinamakan *encoding*, sedangkan ekstraksi pesan dari *stego-object* dinamakan *decoding*. Kedua proses ini mungkin memerlukan kunci rahasia (*stegokey*) agar hanya pihak yang berhak saja yang dapat melakukan penyisipan pesan dan ekstraksi pesan sehingga menambah tingkat keamanan data. Proses umum penyisipan pesan dan ekstraksi pesan dapat dilihat pada Gambar 2.2



Gambar 2.2 Proses Umum Penyisipan dan Ekstrasi Pesan

2.4.3. Teknik Steganografi

Pada dasarnya, terdapat tujuh teknik yang digunakan dalam steganografi [5], yaitu:

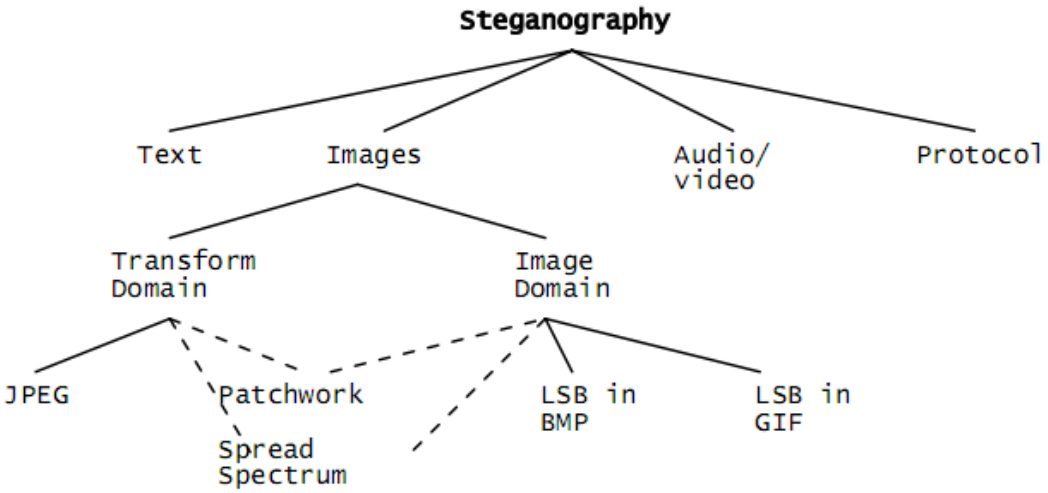
1. *Injection*, merupakan suatu teknik menanamkan pesan rahasia secara langsung ke suatu media. Salah satu masalah dari teknik ini adalah ukuran media yang diinjeksi enjadi lebih besar dari ukuran normalnya sehingga mudah dideteksi.
2. Teknik substitusi (*Substitution Techniques*), pada teknik ini data asli digantikan dengan data rahasia. Biasanya, hasil teknik ini tidak terlalu mengubah ukuran data asli, tetapi tergantung pada *file* media dan data yang akan disembunyikan. Teknik substitusi ini bisa menurunkan kualitas media penampung.
3. Teknik Domain Transformasi (*Domain Transform Techniques*), yaitu dengan cara menyimpan informasi rahasia pada transformasi ruang (misalnya domain frekuensi) dari media penampung (*cover*). Akan lebih efektif jika teknik ini diterapkan pada *file* berekstensi Jpeg (gambar).
4. Teknik *Spread Spectrum* (*Spread Spectrum Techniques*), merupakan sebuah teknik pentransmisian menggunakan *pseudo-noise code,* yang independen terhadap data informasi sebagai modulataor bentuk gelombang untuk menyebarkan energi sinyal dalam sebuah jalur komunikasi (*bandwidth*) yang lebih besar dari pada sinyal jalur komunikasi informasi. Penerima mengumpulkan kembali sinyal dengan menggunakan *replica pseudo-noise code* tersinkronisasi.
5. Teknik Statistik (*Statistical Techniques*), dengan teknik ini data di-encoding melalui pengubahan beberapa informasi statistik dari media penampung (*cover*). Media penampung di bagi dalam blok-blok dimana setiap blok tersebut menyimpan satu *pixel* informasi rahasia yang disembunyikan. Perubahan statistik ditunjukkan dengan indikasi 1 dan jika tidak ada perubahan, terlihat indikasi 0. Sistem ini bekerja berdasarkan kemampuan penerima dalam membedakan antara informasi yang dimodifikasi dan yang belum.
6. Teknik Distorsi (*Distortion Techniques*), informasi yang hendak disembunyikan disimpan berdasarkan distorsi sinyal. Teknik ini menciptakan perubahan atas benda yang ditumpangi oleh data rahasia.

7. Teknik Pembangkitan Wadah (*Cover Generation Techniques*), teknik ini menyembunyikan informasi rahasia sejalan dengan pembangkitan *cover*.

2.4.4. Steganografi Pada Citra Digital

Citra digital merupakan media penampung yang banyak digunakan dalam steganografi. Penggunaan citra digital sebagai media penampung mempunyai kelebihan karena indra penglihatan manusia memiliki keterbatasan terhadap warna, sehingga dengan keterbatasan tersebut manusia sulit membedakan citra digital yang asli dengan citra digital yang telah disisipi pesan rahasia.

Banyak metode yang dapat digunakan dalam pembuatan steganografi pada citra digital seperti *Least Significant Bits (LSB*), *Bit Plane Complexity Segmentation (BPCS), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Spread Spectrum* dan metode-metode lainnya. Dari banyaknya metode tersebut, metode steganografi pada citra digital dapat dibagi menjadi dua bagian yaitu metode pada *transform domain* dan *image domain* seperti pada gambar 2.3.



Gambar 2.3 Kategori Metode Steganografi Pada Citra Digital[3]

2.5. Random Pixel Positioning (RPP) Steganografi

Random Pixel Positioning merupakan salah satu metode yang dapat digunakan dalam steganografi. Metode ini beroperasi pada ranah spasial dari citra. Berdasarkan analisis terhadap sistem penglihatan manusia yang menyatakan bahwa, mata manusia tidak sensitif terhadap perubahan pada *pixel* yang memiliki kekontrasan tinggi melainkan sensitif terhadap perubahan pada *pixel* yang memiliki

kekontrasan rendah. Melalui sifat tersebut maka lebih banyak bit data rahasia yang dapat disisipkan pada *pixel* yang memiliki nilai kekontrasan tinggi, dan sedikit bit yang dapat disisipkan pada *pixel* dengan kekontrasan rendah.dengan proses penyisipan yang secara acak. Hal tersebut yang menjadi dasar pemikiran metode *Random Pixel Positioning (RPP)* pada steganografi.

Penyisipan pesan dapat dilakukan dengan mengambil sebanyak t bit dari pesan yang akan disisipkan. Selanjutnya dihitung nilai *positioning random* yang baru untuk penyisipan kedalam citra menggunakan persamaan 2.4.

$$d'_i = l_i + b \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.4)$$

Dimana : b : Nilai desimal dari jumlah bit disisipkan.

d_i : Nilai terkecil dari *range* selisih perbandingan dua *pixel*.

Untuk menyisipkan pesan ada beberapa aturan yang harus dipenuhi yaitu :

1. Jika $P_i \geq P_{i+1}$ dan $d'i > d_i$, maka (P_i + [m/2] , P_{i+1} - [m/2])
2. Jika $P_i < P_{i+1}$ dan $d'i > d_i$, maka (P_i - [m/2] , P_{i+1} + [m/2])
3. Jika $P_i \geq P_{i+1}$ dan $d'i \leq d_i$, maka (P_i - [m/2] , P_{i+1} + [m/2])
4. Jika $P_i < P_{i+1}$ dan $d'i \leq d_i$, maka (P_i + [m/2] P_{i+1} - [m/2])

Dimana m didapat dari selisih d'i dengan di menggunakan persamaan 2.5.

$$M = |d'_i - d_i| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.5)$$

Proses tersebut dilakukan terus hingga bit pesan tersisipi semuanya kedalam citra.

Proses ekstraksi pesan dari citra stego menggunakan metode ini dimulai dengan menghitung nilai *positioning random* (d_i) antara dua *pixel* secara acak. Nilai *positioning random* tersebut digunakan untuk mengetahui nilai *continuous ranges* (R).

Berdasarkan informasi tersebut dapat diketahui ukuran data rahasia yang disisipkan pada *pixel*, sehingga pesan rahasia yang telah disisipkan didapatkan kembali. Proses ekstraksi ini dilakukan sampai semua data rahasia yang telah disisipkan didapatkan kembali.

2.6. Kriptografi

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, otentikasi, dan autentifikasi data[2]. Kriptografi bukan satu-satunya cara untuk menyediakan keamanan informasi, melainkan satu set teknik yang dapat digunakan untuk mengamankan informasi.

Secara umum, kriptografi terdiri dari dua buah bagian utama yaitu bagian enkripsi dan bagian dekripsi. Enkripsi adalah proses transformasi informasi menjadi bentuk lain sehingga isi pesan yang sebenarnya tidak dapat dipahami, hal ini dimaksudkan agar informasi tetap terlindung dari pihak yang tidak berhak menerima. Sedangkan dekripsi adalah proses kebalikan enkripsi, yaitu transformasi data terenkripsi ke data bentuk semula. Proses transformasi dari plainteks menjadi *cipherteks* akan dikontrol oleh kunci. Peran kunci sangatlah penting, kunci bersama-sama dengan algoritma matematisnya akan memproses *plainteks* menjadi *cipherteks* dan sebaliknya.

Kriptografi tidak memenuhi semua aspek dari keamanan informasi. Kriptografi hanya memenuhi empat aspek dalam keamanan informasi yang merupakan tujuan dari kriptografi. Ke empat aspek tersebut yaitu kerahasiaan (*confidentiality*), integritas data (*integrity*), otentikasi data (*authentication*), dan *non-repudiation*[5].

1. **Kerahasiaan**

Kerahasiaan adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah disandi. Kriptografi memenuhi aspek kerahasiaan karena informasi tidak dapat secara langsung diketahui.

2. **Integritas data**

Integritas data adalah layanan yang berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. Otentikasi

Otentikasi adalah layanan yang berhubungan dengan identifikasi/ pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui jaringan harus diotentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain. Untuk alasan ini aspek kriptografi biasanya dibagi menjadi dua kelas utama yaitu otentikasi entitas dan otentikasi data asal.

4. Non-repudiation

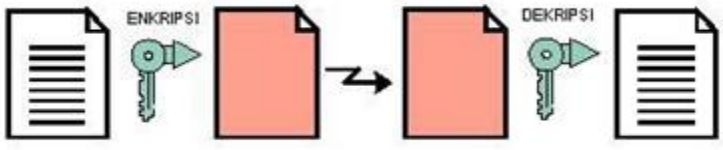
Non-repudiation adalah layanan yang mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan atau membuat. Sebagai contoh, satu entitas dapat mengijinkan pembelian properti oleh entitas lain dan kemudian berusaha menyangkal otorisasi tersebut diberikan. Sebuah prosedur yang melibatkan pihak ketiga yang terpercaya diperlukan untuk menyelesaikan sengketa tersebut.

2.6.1. Algoritma Kriptografi

Algoritma kriptografi yang handal adalah algoritma kriptografi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Berdasarkan jenis kuncinya, algoritma kriptografi dibagi menjadi dua jenis yaitu algoritma simetris dan algoritma asimetris.

2.6.1.1. Algoritma Simetris

Algoritma simetris (*symmetric algorithm*) adalah suatu algoritma dimana kunci enkripsi yang digunakan sama dengan kunci dekripsi sehingga algoritma ini disebut juga sebagai *single-key algorithm*[5]. Ilustrasi penggunaan algoritma kriptografi dengan kunci simetris dapat terlihat pada gambar 2.3 berikut.



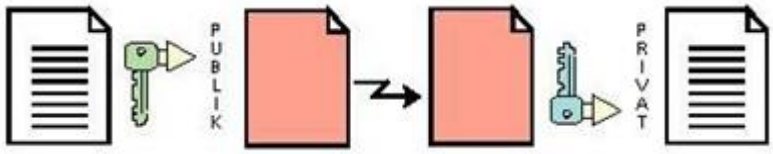
Gambar 2.3 Ilustrasi Kriptografi Dengan Kunci Simetris

Algoritma simetris banyak digunakan karena lebih cepat dan lebih simpel, namun penggunaan kunci simetris juga mempunyai kekurangan, karena jika kunci

dapat diketahui, maka informasi pun dapat diketahui. Beberapa algoritma kriptografi yang termasuk pada algoritma simetris yaitu DES, AES, *Rijndael*, *Blowfish*, dan IDEA.

2.6.1.2. Algoritma Asimetris

Algoritma asimetris (*asymmetric algorithm*) dalah suatu algoritma dimana kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi[5]. Algoritma ini menggunakan dua kunci yakni kunci publik (*public key*) dan kunci privat(*private key*). Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna. Ilustrasi penggunaan algoritma kriptografi dengan kunci asimetris dapat terlihat pada gambar 2.4 berikut.



Gambar 2.4. Ilustrasi Kriptografi Dengan Kunci Asimetris.

Algoritma asimetris mempunyai keamanan yang lebih baik, karena jika *public key* diketahui, informasi belum tentu dapat diketahui karena *private key* kemungkinan berbeda. Namun akan menjadi sulit dan lama ketika implementasinya. Beberapa algoritma kriptografi yang termasuk pada algoritma asimetris yaitu Diffie - Hellman, RSA, ElGamal, dan DSA.

2.6.2. Algoritma advanced encryption standard (AES)

Advanced Encryption Standard termasuk dalam jenis algoritma simetris dan *cipher block*. Dengan demikian algoritma ini menggunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu.

Advanced Encryption Standard mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan, namun *Advanced Encryption Standard* mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses

enkripsi dan dekripsi. Tabel 2.1 adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

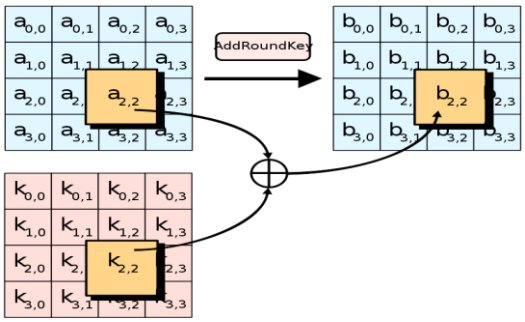
Tabel 2.1 Jumlah Proses Berdasarkan Ukuran Bit Blok dan Kunci

Ukuran blok dan kunci	Panjang Kunci (Nk) Dalam words	Ukuran Blok Data (Nb) Dalam words	Jumlah Proses (Nr)
128 bit	4	4	10
192 bit	6	4	12
256 bit	8	4	14

Blok-blok data masukan dan kunci dioperasikan dalam bentuk array. Setiap anggota array sebelum menghasilkan keluaran *cipher text* dinamakan dengan *state*. Setiap *state* akan mengalami proses yang terdiri dari empat tahap yaitu, *Add Round Key*, *Sub Bytes*, *Shift Rows*, dan *Mix Columns*. Kecuali pada tahap *Mix Columns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap *Mix Columns* tidak akan dilakukan pada tahap terakhir[8].

2.6.2.1. Add Round Key

Proses *Add Round Key* yaitu proses menggabungkan subkey dengan state menggunakan operasi XOR untuk setiap *byte* dari *subkey* dengan *byte* yang dari *state*[8]. Untuk setiap tahap, *subkey* dibangkitkan dari kunci utama dengan menggunakan proses *key schedule*. Proses *Add Round Key* dapat dilihat pada Gambar 2.5.



Gambar 2.5 *Sub Bytes* Proses Add Round Key Algoritma *AES*

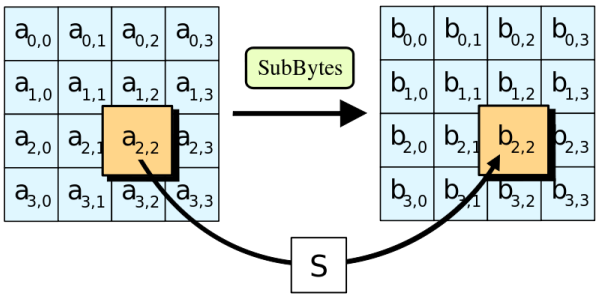
2.6.2.2. Sub Bytes

Proses adalah proses substitusi dengan cara mengganti setiap *byte* state dengan *byte* pada sebuah tabel yang dinamakan tabel *S-Box* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 *S-Box*

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

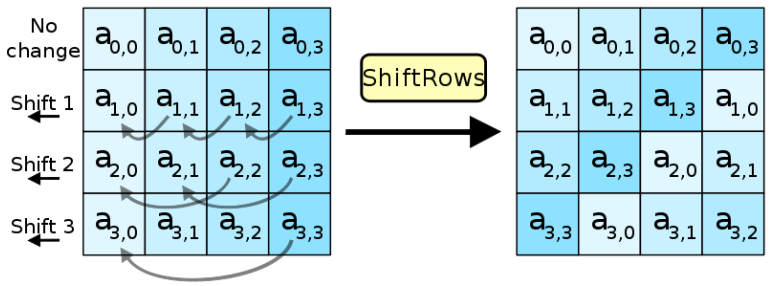
Sebuah tabel *S-Box* terdiri dari 16 baris dan 16 kolom dengan masing-masing berukuran 1 *byte*. Proses *Sub Bytes* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Proses *Sub Bytes Algoritma AES*

2.6.2.3. Shift Rows

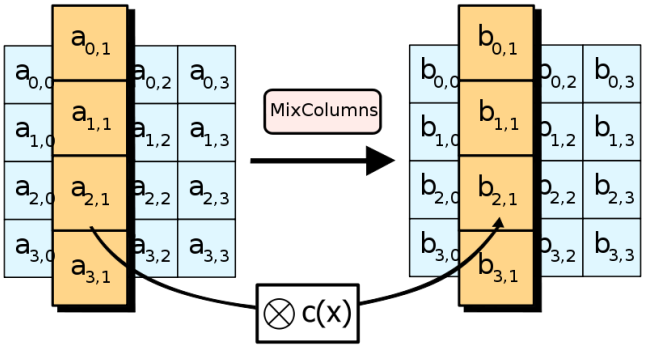
Proses *Shift Rows* adalah proses perputaran pada tiga baris terakhir dari *state* dengan jumlah putaran yang berbeda-beda. Baris ke-1 akan diputar sebanyak 1 kali, baris ke-2 akan diputar sebanyak 2 kali, dan baris ke-3 akan diputar sebanyak 3 kali, sedangkan baris ke-0 tidak akan diputar[8]. Proses *Shift Rows* diperlihatkan pada Gambar 2.7.



Gambar 2.7 Proses *Shift Rows Algoritma AES*

2.6.2.4. Mix Columns

Operasi *Mix Columns* adalah proses menggabungkan 4 *bytes* dari setiap kolom dari tabel *state* dengan menggunakan transformasi *linier*[8]. Operasi *Mix Columns* memperlakukan setiap kolom sebagai polinomial 4 suku dalam *Galois field* dan kemudian dikalikan dengan c(x) modulo (x4+1),dimana c(x)=3x3+x2+x+2. Kebalikkan dari polinomial ini adalah (x)=11x3+13x2+9x+14. Operasi *Mix Columns* juga dapat dipandang sebagai perkalian matrix seperti terlihat pada gambar 2.8.



Gambar 2.8 Proses *Mix Columns Algoritma AES*

2.7. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa pemodelan standar yang memiliki sintaks dan semantic. Pemodelan ini sangat cocok digunakan untuk merancang dan memodelkan sistem berorientasi objek[9].

Diagram pada UML dibagi menjadi dua bagian yaitu *structural* diagram dan *behavior* diagram. *Structural* diagram digunakan untuk mendeskripsikan relasi antar kelas. *Tools* yang digunakan pada bagian ini yaitu *class* diagram. Sedangkan *behavior* diagram digunakan untuk mendeskripsikan interaksi antara aktor dan

sebuah *use case* (bagaimana seorang aktor menggunakan sistem). *Tools* yang digunakan pada bagian ini yaitu *use case* diagram, s*equence* diagram, *collaboration* diagram, s*tatechart* diagram, dan *activity* diagram.

1. Diagram *Class*

Diagram *class* digunakan untuk menggambarkan keadaan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem[9]. Sebuah class terdiri dari nama, atribut dan *method.* Atribut dan *method* dari sebuah *class* mempunyai *visibility*, Ada tiga jenis *visibility* yang digunakan yaitu *private*, *public*, dan *protected*. Setiap *class* pada *class* diagram mempunyai hubungan dengan *class* lainnya, ada beberapa jenis hubungan *class*, yaitu *dependency*, asosiasi, agregasi, komposisi dan generalisasi.

a. *Dependency*

Dependency merupakan hubungan terlemah antar *class*. *Dependency* bermakna satu *class* menggunakan atau memiliki pengetahuan terhadap *class* lain, namun hubungannya hanya sementara dan tidak ada batas waktu yang jelas.

b. Asosiasi

Asosiasi mempunyai hubungan yang lebih kuat dari hubungan *dependency*, dimana suatu *class* tetap berhubungan dengan *class* lain seterusnya. Hubungan asosiasi dibagi menjadi dua jenis yaitu *directional* dan *bidirectional*.

c. Agregasi

Agregasi merupakan bentuk hubungan yang mengimplikasikan kepemilikan suatu *class*. Agregasi juga merupakan bentuk yang lebih kuat dari asosiasi. Hubungan agregasi dinyatakan dengan simbol diamond pada pemilik kelas dan garis utuh berpanah kekelas yang dimiliki.

d. Komposisi

Komposisi merupakan bentuk hubungan antar *class* yang paling kuat. Komposisi digunakan untuk mengambil seluruh bagian dari *class* yang berhubungan. Aturan dari komposisi yaitu hanya boleh ada satu komposisi dalam satu waktu.

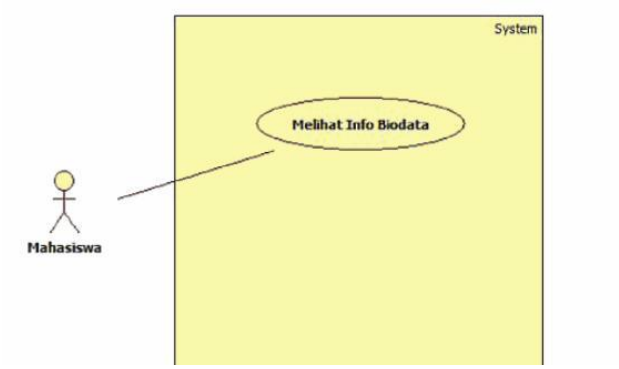
e. Generalisasi

Generalisasi merupakan bentuk hubungan antar *class*, dari *class* yang umum dengan *class* yang lebih khusus. Contoh *class* hewan memiliki hubungan generalisasi dengan *class* kucing, karena kucing merupakan *class* khusus dari *class* hewan yang lebih umum.

2. Diagram *Use Case*

Diagram use case digunakan untuk mendeskripsikan kejadian-kejadian apa saja yang dapat dilakukan oleh *user* aktor dan fungsionalitas-fungsionalitas apa saja yang diharapkan dari sistem yang akan dibangun, tanpa mendeskripsikan bagaimana sistem menyelesaikannya[9]. Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam *fase requirements*, model *use case* mengambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk *naratif*, yang terdiri dari *input user* dan respon-respon sistem.

Setiap *use case* menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model *use case* secara pararel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap *use case*. Contoh *use case* diagram diperlihatkan pada gambar 2.9.

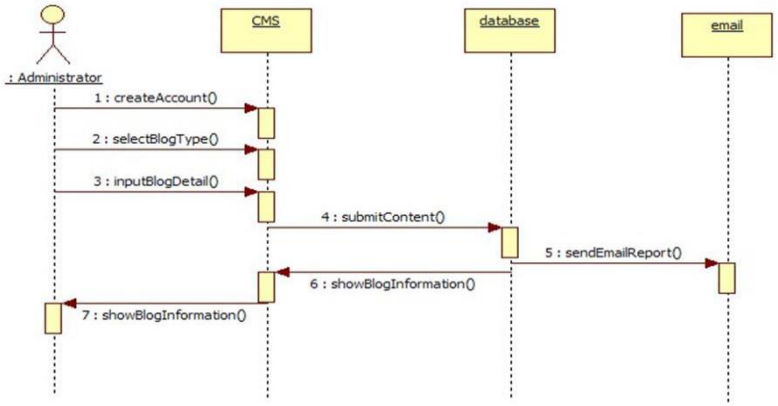


Gambar 2.9 Contoh *Use Case* Diagram

3. Diagram *Sequence*

Diagram *sequence* menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang menekankan pada pengiriman pesan dalam suatu waktu tertentu[9]. *Sequence* diagram biasa digunakan untuk menggambarkan *scenario*

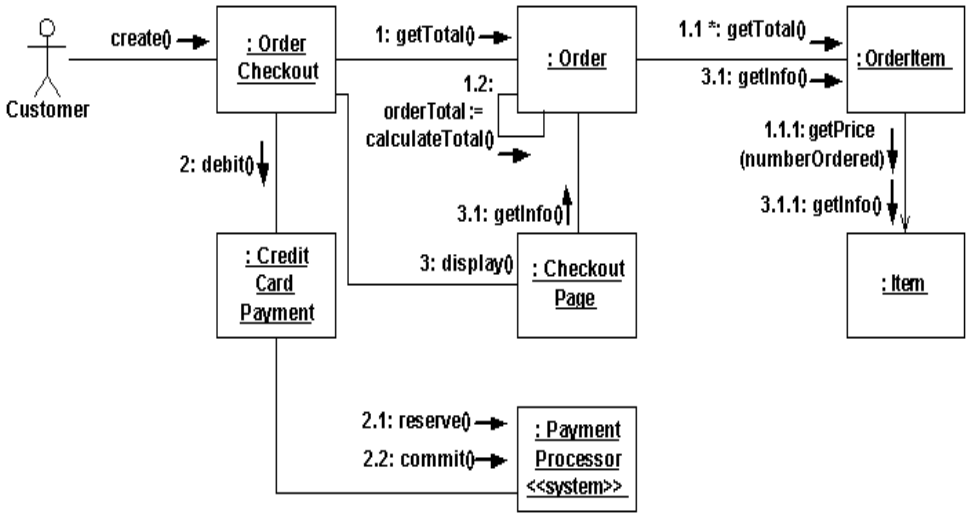
atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Contoh *sequence* diagram diperlihatkan pada gambar 2.10.



Gambar 2.10 Contoh *Sequence* Diagram

4. Diagram *Collaboration*

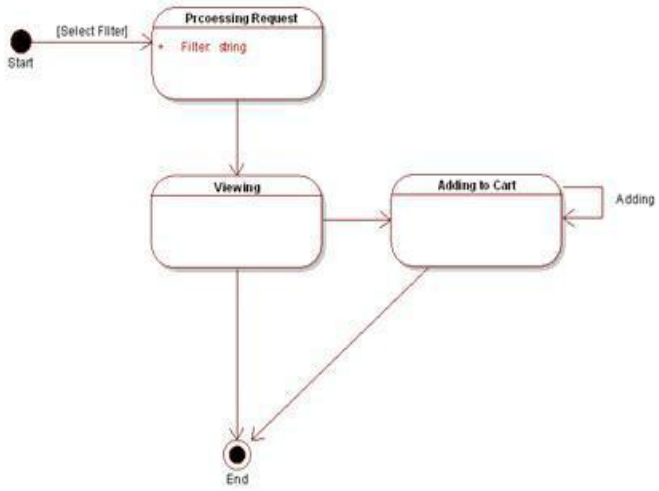
Diagram *collaboration* menunjukan informasi yang sama seperti dalam *sequence* diagram. Perbedaannya adalah dari cara mengelompokan objek *sequence* berdasarkan urutan nomor dari pesan[9]. Contoh *collaboration* diagram diperlihatkan pada gambar 2.11.



Gambar 2.11 Contoh *Collaboration* Diagram

5. Diagram *Statechart*

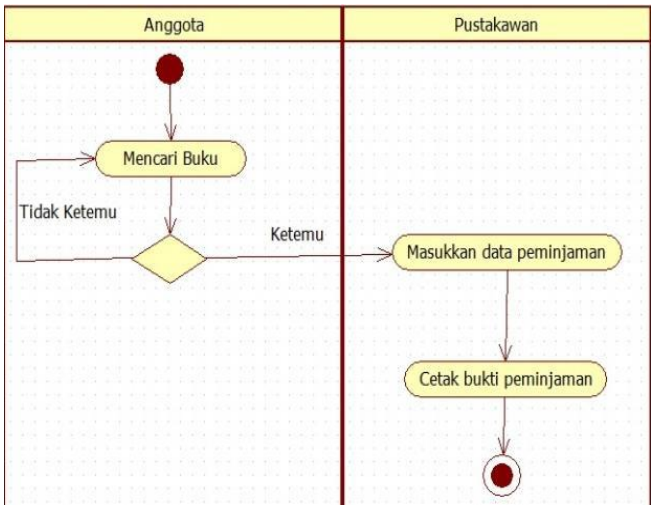
Diagram statechart menggambarkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas[9]. Contoh *Statechart* diagram diperlihatkan pada gambar 2.12.



Gambar 2.12 Contoh *Statechart* Diagram

6. Diagram *Activity*

Diagram activity menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir[9]. *Activity* diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (internal *processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur. aktivitas dari *level* atas secara umum. Contoh *Activity* diagram diperlihatkan pada gambar 2.13.



Gambar 2.13 Contoh *Activity* Diagram

2.8. Teknologi Java

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *stand alone* ataupun pada lingkungan jaringan. Java berdiri di atas sebuah mesin *interpreter* yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca *bytecode* dalam *file* c*lass* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin.

Oleh karena itu, bahasa *Java* disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. *Platform* Java terdiri dari kumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah *compiler*, *debuger*, dan perangkat lain yang dipaket dalam *Java Development Kit* (JDK)[1].

Java merupakan salah satu bahasa pemrograman yang menggunakan paradigma pemrograman berbasis objek. Paradigma ini yaitu menggunakan objek untuk membungkus atribut dan operasi yang mungkin pada objek tersebut. Java mempunyai kelebihan dan kekurangan yaitu:

A. Kelebihan Java

Beberapa kelebihan dari java yaitu sebagai berikut:

1. *Multiplatform*

Kelebihan utama dari java ialah dapat dijalankan di beberapa *platform*/*system* operasi komputer, sesuai dengan prinsip tulis sekali, jalankan dimana saja. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas sistem operasi linux tetapi dijalankan dengan baik di atas *Microsoft Windows.*

2. OOP *(Object Oriented Programming)*

Java merupakan salah satu bahasan pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut objek. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan, dan mengalokasi kesalahan sebuah program dengan basis java secara cepat, tepat, mudah dan terorganisir.

3. *Library* yang lengkap

Java terkenal dengan kelengkapan *library* (perpustakaan) (kumpulan program-program yang disertakan dalam pemrogram java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

4. Bergaya C++

Java memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke java. Saat ini pengguna java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke java.

5. Pengumpulan sampah otomatis

Java memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas) [1].

B. Kekurangan Java

Beberapa kekurangan Java antara lain :

1. Mudah didekompilasi

   Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini memungkinkan karena kode jadi java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi seperti nama-nama kelas, metode dan tipe data.

2. Penggunaan memori yang banyak

   Penggunaan memori untuk program berbasis java jauh lebih besar dari pada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi delphi dan Objek Pascal) [1].