BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Analisis sistem merupakan suatu tahapan yang bertujuan untuk mengetahui dan mengamati apa saja yang terlibat dalam suatu sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis masalah, analisis algoritma, analisis kebutuhan nonfungsional, dan analisis kebutuhan fungsional.

III.1.1 Analisis Masalah

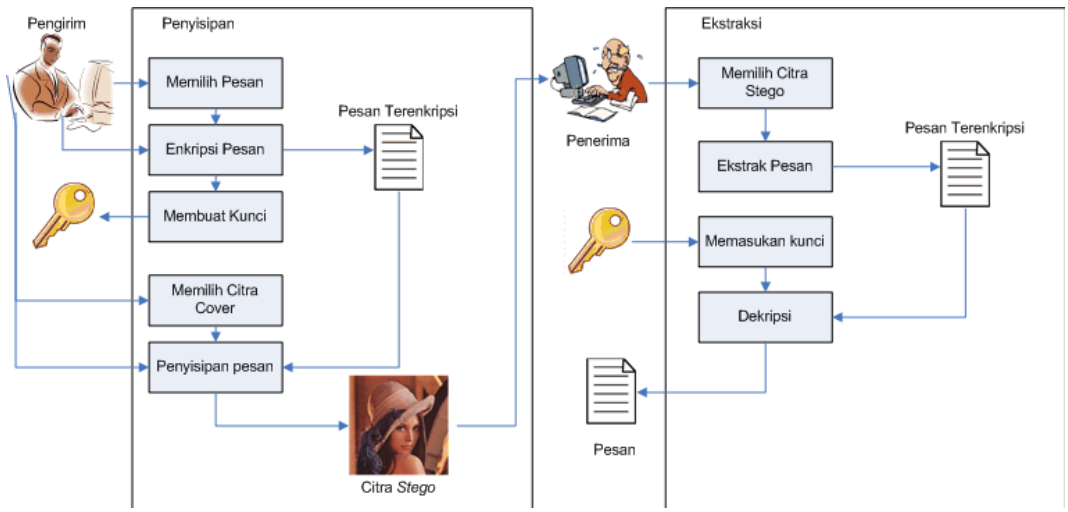
Analisis masalah dilakukan untuk mengetahui masalah-masalah apa saja yang terjadi dalam pembangunan aplikasi steganografi. Masalah yang terjadi ketika pembangunan steganografi pada citra digital yaitu kapasitas citra yang dapat disisipkan informasi kecil, kualitas citra digital yang telah disisipkan informasi, dan keamanan informasi yang disisipkan kedalam citra tidak terjamin, karena saat ini banyak aplikasi yang digunakan untuk menanalisis citra apakah terdapat informasi yang disembunyikan dan letak dari informasi yang disembunyikan. Jika letak dari informasi yang disisipkan pada citra diketahui, maka informasi akan langsung dapat diketahui.

III.1.2 Analisis Algoritma

Analisis algoritma digunakan untuk mengetahui alur proses dari algoritma yang digunakan untuk dapat diterapkan ke dalam aplikasi yang dibangun. Pembangunan aplikasi ini menggunakan algoritma *Rijndael* untuk proses enkripsi dan dekripsi dengan panjang kunci 128*bit* sebagai keamanan terhadap informasi yang akan disisipkan. Sedangkan metode steganografi yang digunakan yaitu metode *Pixel Value Differencing* (PVD) untuk proses penyisipan dan ekstraksi informasi. Tahapan-tahapan yang dilakukan pada setiap prosesnya yaitu sebagai berikut :

1. Proses Penyisipan
   a. Pengirim memilih file pesan yang akan disisipkan
   b. Pengirim melakukan enkripsi terhadap pesan menggunakan algoritma rijndael *128bit.*
   c. Pengirim memilih file citra yang akan digunakan sebagai media penampung dari pesan.
   d. Pengirim melakukan proses penyisipan menggunakan metode *Pixel Value Differencing* (PVD).
2. Proses Ekstraksi
   a. Penerima memilih file citra yang telah disisipkan pesan (*stego object*)
   b. Penerima melakukan proses ekstraksi menggunakan metode *Pixel Value Differencing* (PVD).
   c. Penerima melakukan dekripsi terhadap pesan menggunakan algoritma rijndael *128bit.*
   d. Penerima menyimpan pesan yang telah diekstraksi dari citra.

Tahapan-tahapan proses pada aplikasi steganografi secara umum dapat dilihat pada gambar III.1.



Gambar III.1 Gambaran Umum Aplikasi Steganografi

III.1.2.1 Analisis Algoritma *Rijndael*

Algoritma *rijndael* merupakan algoritma kriptografi yang sifatnya simetris dan *cipher block*. Dengan demikian algoritma ini mengunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah *bit* tertentu. Algoritma *rijndael* yang digunakaan pada aplikasi steganografi yang dibangun menggunakan ukuran blok dan kunci 128*bit.* Algoritma *rijndael* terdapat dua proses yaitu proses penjadwalan kunci dan enkripsi.

III.1.2.1.1 Analisis Penjadwalan Kunci

Proses penjadwalan kunci merupakan proses dimana *cipherkey* di jadwalkan untuk menghasilkan *subkey-subkey* yang digunakan untuk proses enkripsi dan dekripsi pada algoritma *rijndael*. Contoh penjadwalan kunci pada algoritma *rijndael* jika diketahui kunci yang akan digunakan untuk enkripsi dengan panjang 16 byte yaitu:

***Cipherkey* = abcdefghijklmnop**

Tahap awal ubah *cipherkey* kedalam bentuk *hexadecimal* menjadi sebagai berikut:

***Cipherkey* = 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70**

Tahap selanjutnya melakukan operasi-operasi penjadwalan kunci. Operasi-operasi yang dilakukan yaitu *RotWord*, *SubByte*, dan melakukan operasi XOR untuk menghasilkan *subkey*. Operasi-operasi yang dilakukan yaitu sebagai berikut:

1. Masukan cipherkey tersebut kedalam blok 16 byte menjadi.

$$W = \begin{bmatrix} 61 & 65 & 69 & 6d \\ 62 & 66 & 6a & 6e \\ 63 & 67 & 6b & 6f \\ 64 & 68 & 6c & 70 \end{bmatrix}$$

2. Melakukan operasi *RotWord* pada kolom terakhir dari *ciphertext*.

$$\begin{bmatrix} \boldsymbol{6d} \\ 6e \\ 6f \\ 70 \end{bmatrix} = \begin{bmatrix} 6e \\ 6f \\ 70 \\ \boldsymbol{6d} \end{bmatrix}$$

3. Melakukan operasi *SubByte* dengan tabel *s-box* pada tabel II.2.

$$\begin{bmatrix} 6e \\ 6f \\ 70 \\ 6d \end{bmatrix} = \begin{bmatrix} 9f \\ a8 \\ 51 \\ 3c \end{bmatrix}$$

4. Hasil dari operasi *SubByte* dilakukan operasi XOR dengan *rcon* dan W_1 (kolom ke-1 dari W).

$$rcon = \begin{bmatrix} 01 & 02 & 04 & 08 & 10 & 20 & 40 & 80 & 1b & 36 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

$$\begin{bmatrix} 61 \\ 62 \\ 63 \\ 64 \end{bmatrix} \oplus \begin{bmatrix} 9f \\ a8 \\ 51 \\ 3c \end{bmatrix} \oplus \begin{bmatrix} 01 \\ 00 \\ 00 \\ 00 \end{bmatrix} = \begin{bmatrix} ff \\ ca \\ 32 \\ 58 \end{bmatrix}$$

5. Melakukan operasi XOR untuk untuk kolom selanjutnya dengan kolom yang baru.

$$\begin{bmatrix} 65 \\ 66 \\ 67 \\ 68 \end{bmatrix} \oplus \begin{bmatrix} ff \\ ca \\ 32 \\ 58 \end{bmatrix} = \begin{bmatrix} 9a \\ ac \\ 55 \\ 30 \end{bmatrix}$$

$$\begin{bmatrix} 69 \\ 6a \\ 6b \\ 6c \end{bmatrix} \oplus \begin{bmatrix} 9a \\ ac \\ 55 \\ 30 \end{bmatrix} = \begin{bmatrix} f3 \\ c6 \\ 3e \\ 5c \end{bmatrix}$$

$$\begin{bmatrix} 6d \\ 6e \\ 6f \\ 70 \end{bmatrix} \oplus \begin{bmatrix} f3 \\ c6 \\ 3e \\ 5c \end{bmatrix} = \begin{bmatrix} 9e \\ a8 \\ 51 \\ 2c \end{bmatrix}$$

6. Simpan kedalam *subkey*.

$$SubKey = \begin{bmatrix} ff & 9a & f3 & 9e \\ ca & ac & c6 & a8 \\ 32 & 55 & 3e & 51 \\ 58 & 30 & 5c & 2c \end{bmatrix}$$

Subkey ini yang akan digunakan untuk proses enkripsi atau dekripsi pada algoritma rijndael pada *round* ke-1 untuk *round* selanjutnya dilakukan penjadwalan kunci kembali sampai *round* ke-10.

III.1.2.1.2 Analisis Proses Enkripsi Algoritma *Rijndael*

Proses enkripsi pada algoritma *rijndael* terdiri dari empat operasi yaitu *Add Round Key*, *Sub Bytes*, *Shift Rows*, dan *Mix Columns*. Operasi-operasi ini diulang terus-menerus hingga menghasilkan *ciphertext*. Jumlah perulangan yang dilakukan tergantung pada ukuran blok dan kunci yang digunakan, dalam hal ini ukuran blok dan kunci yang digunakan yaitu 128 bit, sehingga berdasarkan pada tabel II.1, maka perulangan yang dilakukan sebanyak 10 kali. Contoh enkripsi pada algoritma rijndael, jika diketahui kunci dan *plaintext* yang akan digunakan untuk enkripsi dengan panjang 16 byte.

***Cipherkey* = abcdefghijklmnop**

***Plaintext* = UNIKOM BANDUNG !**

Tahap awal ubah *cipherkey* dan *plaintext* kedalam bentuk *hexadecimal* menjadi sebagai berikut:

***Cipherkey* = 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70**

***Plaintext* = 55 4e 49 4b 4f 4d 20 42 41 4e 44 55 4e 47 20 21**

Masukan *cipherkey* dan *plaintext* ke dalam blok 16 byte sehingga menjadi

$$State = \begin{bmatrix} 55 & 4f & 41 & 4e \\ 4e & 4d & 4e & 47 \\ 49 & 20 & 44 & 20 \\ 4b & 42 & 55 & 21 \end{bmatrix} \quad Cipherkey = \begin{bmatrix} 61 & 65 & 69 & 6d \\ 62 & 66 & 6a & 6e \\ 63 & 67 & 6b & 6f \\ 64 & 68 & 6c & 70 \end{bmatrix}$$

Cipherkey dan *plaintext* yang telah dimasukan kedalam *block* selanjutnya dapat dilakukan operasi-operasi enkripsi pada algoritma rijndael sebagai berikut:

1. Melakukan operasi *AddRoundKey* dengan melakukan operasi XOR pada setiap kolom di *state* dengan kolom di *ciphertext*, sehingga menghasilkan *state* baru seperti berikut.

$$State = \begin{bmatrix} 34 & 2a & 28 & 23 \\ 2c & 2b & 24 & 28 \\ 2a & 47 & 2f & 4f \\ 2f & 2a & 39 & 41 \end{bmatrix}$$

2. *State* yang telah dilakukan operasi *AddRoundKey* tersebut dilakukan perulangan dengan urutan operasi pertama yaitu operasi *SubByte*. Operasi ini yaitu melakukan subtitusi *state* dengan tabel *s-box* pada tabel II.2 sehingga menghasilkan *state* baru yaitu:

$$State = \begin{bmatrix} d4 & e0 & b8 & 1e \\ 27 & bf & b4 & 41 \\ 11 & 98 & 5d & 52 \\ ae & f1 & e5 & 30 \end{bmatrix}$$

3. Hasil dari operasi *SubByte* dilakukan operasi *ShiftRows* yaitu memutar tiga baris terakhir dari *state* seperti berikut:

$$State = \begin{bmatrix} d4 & e0 & b8 & 1e \\ 27 & bf & b4 & \mathbf{41} \\ 11 & 98 & \mathbf{5d} & \mathbf{52} \\ ae & \mathbf{f1} & \mathbf{e5} & \mathbf{30} \end{bmatrix} \text{ Menjadi } \boldsymbol{State} = \begin{bmatrix} d4 & e0 & b8 & 1e \\ bf & b4 & \mathbf{41} & 27 \\ \mathbf{5d} & \mathbf{52} & 11 & 98 \\ \mathbf{30} & ae & \mathbf{f1} & \mathbf{e5} \end{bmatrix}$$

4. Melakukan operasi *MixColumns* yaitu melakukan perkalian tiap kolom pada state dengan matriks seperti berikut.

$$\begin{bmatrix} d4 \\ bf \\ 5d \\ 30 \end{bmatrix} \bullet \begin{bmatrix} 02 & 01 & 01 & 03 \\ 03 & 02 & 01 & 01 \\ 01 & 03 & 02 & 01 \\ 01 & 01 & 02 & 03 \end{bmatrix} = \begin{bmatrix} 04 \\ 66 \\ 81 \\ e5 \end{bmatrix}$$

$$State = \begin{bmatrix} 04 & e0 & 48 & 28 \\ 66 & cb & f8 & 06 \\ 81 & 19 & d3 & 26 \\ e5 & 9a & 7a & 4c \end{bmatrix}$$

5. Melakukan *AddRoundKey* kembali dengan menggunakan *SubKey* hasil dari penjadwalan kunci *CipherKey*.

$$\begin{bmatrix} 04 & e0 & 48 & 28 \\ 66 & cb & f8 & 06 \\ 81 & 19 & d3 & 26 \\ e5 & 9a & 7a & 4c \end{bmatrix} \oplus \begin{bmatrix} ff & 9a & f3 & 9e \\ ca & ac & c6 & a8 \\ 32 & 55 & 3e & 51 \\ 58 & 30 & 5c & 2c \end{bmatrix}$$

Sehingga menghasilkan *state* baru yaitu

$$State = \begin{bmatrix} fb & 7a & bb & b6 \\ ac & 67 & 35 & ae \\ b3 & 4c & 5d & 77 \\ bd & aa & 26 & 60 \end{bmatrix}$$

Semua operasi tersebut diulang sebanyak 10 kali hingga mendapatkan *ciphertext*. Untuk perulangan 1 sampai 9 dilakukan operasi *SubByte*, *ShiftRow*, *MixColumn*, dan *AddRoundKey*. Sedangkan untuk perulangan terakhir hanya dilakukan operasi *SubByte*, *ShiftRow*, dan *AddRoundKey*.

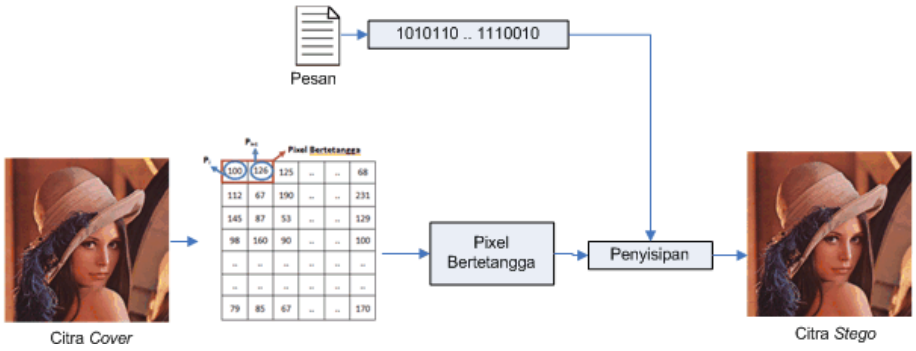
Proses dekripsi menggunakan algoritma rijndael merupakan kebalikan dari proses enkripsi. Operasi-operasi yang dilakukan yaitu *InvSubByte*, *InvShiftRow*, *InvMixColumn*, dan *AddRoundKey*.

III.1.2.2 Analisis Metode Pixel Value Differencing (PVD)

Metode *Pixel Value Differencing* (PVD) merupakan salah satu metode steganografi pada citra digital yang beroperasi pada ranah spasial. Konsep dari metode ini yaitu dengan menyisipkan pesan kedalam dua *pixel* yang bertetangga, dengan memanfaatkan perbedaan intensitas warna dari kedua *pixel* yang bertetangga tersebut. Seperti halnya metode steganografi lainnya, pada metode ini terdapat dua proses yaitu proses penyisipan/*embedding* dan pengungkapan/ ekstraksi.

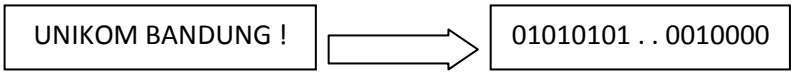
III.1.2.2.1 Analisis Proses Penyisipan/*Embedding*

Proses penyisipan yaitu proses menyembunyikan informasi kedalam media penampung, dalam hal ini media penampung berupa citra digital. Proses ini akan menghasilkan citra yang telah disisipkan pesan (*stego-object*) yang menyerupai dengan citra sebelum disisipkan pesan. Proses penyisipan pada metode *pixel value differencing* terlihat pada gambar III.2.



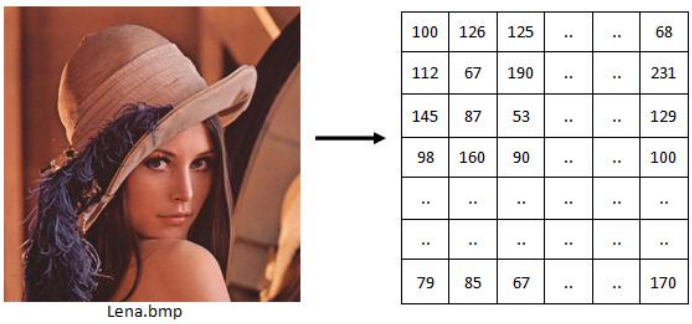
Gambar III.2 Proses Penyisipan Pesan

Contoh proses penyisipan, jika diketahui pesan yang akan disisipkan berupa file unikom.txt dengan isi pesan yaitu “UNIKOM BANDUNG !”. Tahap awal yang dilakukan yaitu merubah isi pesan kedalam bentuk biner seperti pada gambar III.3.



Gambar III.3 Perubahan Pesan Teks ke Bit

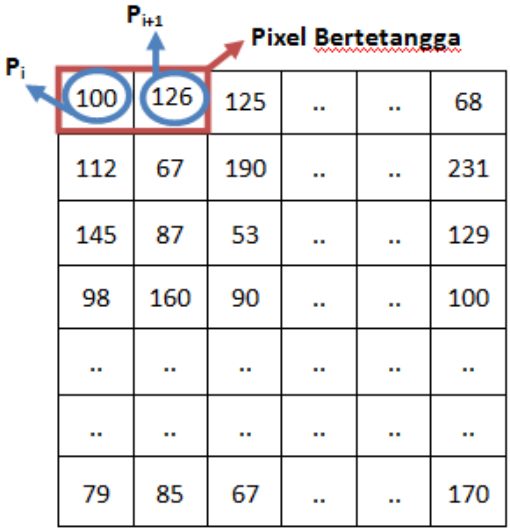
Tahap selanjutnya yaitu mengambil nilai pixel dari citra yang akan disisipkan pesan. Jika diketahui citra yang digunakan yaitu lena.bmp dengan nilai pixel seperti pada gambar III.4.



Gambar III.4 Nilai Pixel Dari Citra Cover

Maka tahap selanjutnya yaitu melakukan proses penyisipan menggunakan metode *pixel value differencing* dengan tahapan-tahapan yaitu sebagai berikut:

1. Mengambil pixel yang bertetangga dari citra. Contoh *pixel* yang bertetangga yaitu *pixel*(0,0) dengan *pixel*(0,1) seperti pada gambar III.5. Nilai dari *pixel* yang bertetangga tersebut diambil untuk dilakukan penyisipan. Jika P_i dan P_{i+1} merupakan *pixel* yang bertetangga, maka $P_i = 100$ dan $P_{i+1} = 126$.



Gambar III.5 Pixel yang Bertetangga Dari Citra Cover

2. Menghitung nilai *differencing value* dari kedua *pixel* tersebut menggunakan persamaan II.4 yaitu $d = |100 - 126|$, sehingga didapat $d = 26$.
3. Mencari letak *continues range* dari nilai *difference value* pada skema wu dan tsai R = {[0,7],[8,15],[16,31],[32,63],[64,127],[128,255]}. Letak *continues range* yang didapat dari $d = 26$ yaitu [16, 31] dimana $lk = 16$, dan $uk = 31$.
4. Menghitung berapa banyak bit dari pesan yang dapat disisipkan kedalam kedua *pixel* yang dibandingkan menggunakan persamaan II.5 yaitu $t = Log2(31 - 16)$ sehingga didapat $t = 4$, maka ambil bit dari pesan sebanyak t yaitu 0101.
5. Mengubah nilai bit sebanyak t kedalam nilai *decimal*. Bit informasi yang disisipkan yaitu 0101, maka nilai *decimal*-nya yaitu 5 atau $b = 5$.
6. Menghitung nilai *differencing value* yang baru menggunakan persamaan II.6, $d' = 16 + 5$ sehingga didapat nilai *differencing value* yang baru yaitu $d' = 21$.
7. Melakukan penyisipan dengan mengubah nilai dari *pixel* yang dibandingkan dengan nilai *pixel* yang baru sesuai dengan aturan – aturan yang ada, dimana m = 5 didapat menggunakan persamaan II.7 yaitu $M = |21 - 26|$. Aturan yang terpenuhi yaitu $d' < d$ dan $P_i' < P_{i+1}'$, maka $P_i' = 100 + \lfloor 5/2 \rfloor$ dan $P_{i+1}' = 126 - \lceil 5/2 \rceil$.
8. Menyimpan nilai *pixel* yang baru yaitu $P_i' = 102$ dan $P_{i+1}' = 123$ kedalam citra. Tahapan ini dilakukan sampai semua pesan tersisipi, sehingga menjadi seperti pada gambar III.6.

102	**123**	123	..	..	69
116	70	192	..	..	232
144	88	55	..	..	128
96	162	91	..	..	99
..	..	..	..	..	..
..	..	..	..	..	..
77	85	68	..	..	172

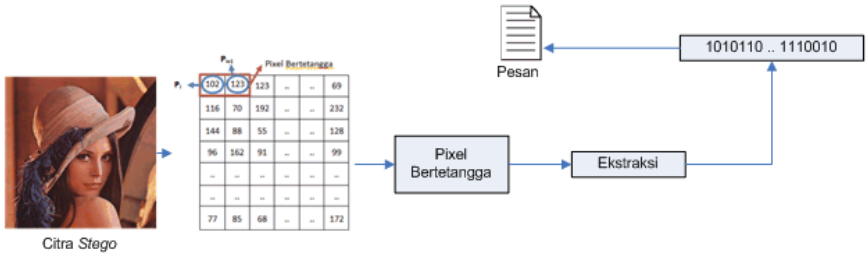


StegoLena.bmp

Gambar III.6 Nilai Pixel Citra Setelah Disisipi Pesan

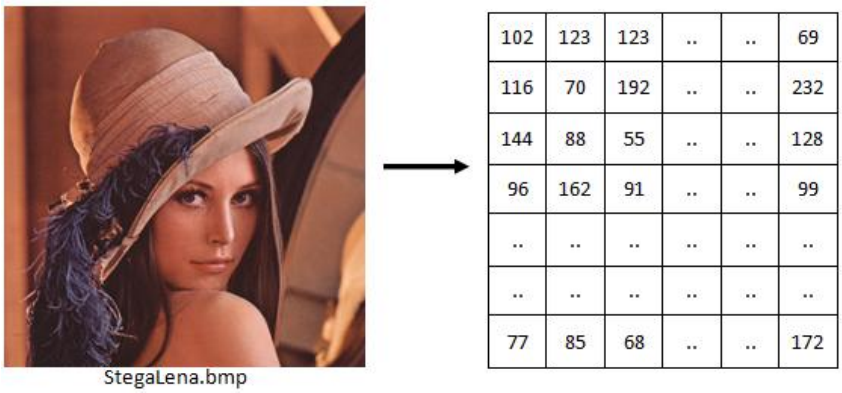
III.1.2.2.2 Analisis Proses Pengungkapan / Ekstraksi.

Proses ekstraksi yaitu proses pengambilan informasi yang tersembunyi pada citra digital. Proses ini akan menghasilkan *file* informasi yang disembunyikan, dengan masukan berupa citra *stego-object.* Proses ekstraksi pada metode *pixel value differencing* terlihat pada gambar III.7.



Gambar III.7 Proses Ekstraksi Pesan

Tahap awal pada proses ekstraksi pesan yaitu mengambil nilai *pixel* dari citra yang telah disisipkan pesan. Jika diketahui citra yang digunakan yaitu StegaLena.bmp dengan nilai *pixel* seperti pada gambar III.8.

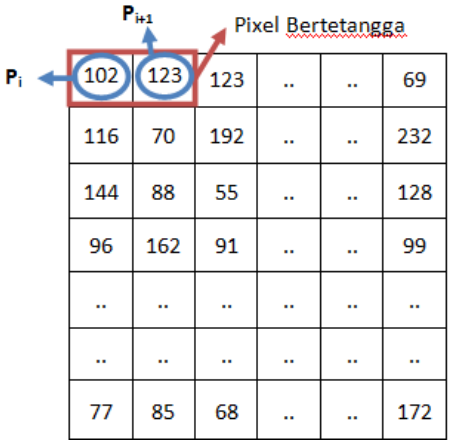


102	123	123	..	..	69
116	70	192	..	..	232
144	88	55	..	..	128
96	162	91	..	..	99
..	..	..	..	..	..
..	..	..	..	..	..
77	85	68	..	..	172

Gambar III.8 Nilai Pixel Dari Citra Stego

Maka tahap selanjutnya yaitu melakukan proses ekstraksi menggunakan metode *pixel value differencing* dengan tahapan-tahapan yaitu sebagai berikut:

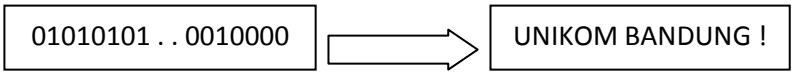
1. Mengambil pixel yang bertetangga dari citra. Contoh *pixel* yang bertetangga yaitu *pixel*(0,0) dengan *pixel*(0,1) seperti pada gambar III.9. Nilai dari *pixel* yang bertetangga tersebut diambil untuk dilakukan penyisipan. Jika P_i dan P_{i+1} merupakan *pixel* yang bertetangga, maka $P_i = 102$ dan $P_{i+1} = 123$.



Gambar III.9 *Pixel* yang Bertetangga Fari Citra Stego

2. Menghitung nilai *differencing value* dari kedua *pixel* tersebut menggunakan persamaan II.4 yaitu $d = |102 - 123|$, sehingga didapat $d = 21$.
3. Mencari letak *continues range* dari nilai *difference value* pada skema wu dan tsai R = {[0,7],[8,15],[16,31],[32,63],[64,127],[128,255]}. Letak *continues range* yang didapat dari $d = 21$ yaitu [16, 31] dimana $lk = 16$, dan $uk = 31$.
4. Menghitung berapa banyak bit dari informasi yang disisipkan kedalam kedua *pixel*. Banyak bit tersebut dihitung menggunakan persamaan II.5 yaitu $t = Log2(31 - 16)$ sehingga didapat $t = 4$, atau terdapat 4 bit pesan yang disisipkan pada kedua *pixel*.
5. Menghitung nilai *b* atau nilai *decimal* dari bit pesan yang disisipkan dengan menggunakan persamaan II.6 yaitu $b = 21 - 16$ sehingga didapat nilai $b = 5$ atau nilai *decimal* dari bit pesan adalah 5.
6. Mengubah nilai *b* atau nilai *decimal* pesan kedalam bentuk bit sebanyak *t*, maka didapat bit pesan 0101.

Tahapan-tahapan pada metode *pixel value differencing* tersebut diulang hingga semua pesan yang terdapat di dalam citra terekstrak. Tahap selanjutnya setelah semua pesan terekstrak yaitu merubah pesan dalam bentuk biner ke bentuk semula. Jika pesan yang disisipkan berupa pesan teks, maka diubah kedalam bentuk teks seperti pada gambar III.10.



Gambar III.10 Perubahan Pesan Bit ke Teks

III.1.3 Analisis Kebutuhan non-Fungsional

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana seorang pembangun aplikasi menganalisis sumber daya yang dibutuhkan untuk menggunakan aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam tiga tahap, yaitu analisis pengguna (*user*), analisis kebutuhan perangkat keras, dan analisis perangkat lunak.

III.1.3.1 Analisis Pengguna

Pengguna yang akan menggunakan aplikasi steganografi ini dibagi menjadi dua bagian yaitu bagian pengirim dan penerima dengan kebutuhan spesifikasi seperti pada tabel III.1.

Tabel III.1 Kebutuhan Pengguna Aplikasi

Tipe Pengguna	Hak Akses	Tingkat Keterampilan	Jenis pelatihan
Pengirim	Dapat melakukan operasi enkripsi pesan dan penyisipan pesan kedalam citra.	Dapat mengoperasikan komputer dan memahami konsep steganografi dan kriptografi.	Tidak diperlukan pelatihan khusus.
Penerima	Dapat melakukan operasi dekripsi pesan dan ekstrak pesan dari citra.	Dapat mengoperasikan komputer dan memahami konsep steganografi dan kriptografi.	Tidak diperlukan pelatihan khusus.

III.1.3.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras merupakan salah satu kebutuhan yang sangat penting bagi pembuatan steganografi. Perangkat keras akan mempengaruhi kinerja dari pembuatan steganografi, semakin tinggi spesifikasi dari perangkat keras yang digunakan maka akan semakin cepat pula pembuatan steganografinya. Perangkat keras yang digunakan pada pembangunan aplikasi steganografi ini yaitu seperti pada tabel III.2.

Tabel III.2 Spesifikasi Perangkat Keras

No	Nama Perangkat	Spesifikasi
1	Prosessor	Intel Core 2.26 GHz
2	Monitor	Monitor 14 inch (1366x768)
3	Memori	RAM 2 GB DDR2
4	Harddisk	320 GB SATA 7200rpm

Sedangkan kebutuhan perangkat keras untuk menjalankan aplikasi yang dibangun, yang harus dipenuhi yaitu seperti pada tabel III.3.

Tabel III.3 Spesifikasi Perangkat Keras

No	Nama Perangkat	Spesifikasi
1	Prosessor	Intel Core 2.26 GHz
2	Monitor	Monitor 14 inch (1366x768)
3	Memori	RAM 1 GB DDR2
4	Harddisk	250 GB SATA 7200rpm

III.1.3.3 Analisis Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun aplikasi steganografi ini yaitu seperti pada tabel III.4.

Tabel III.4 Spesifikasi Perangkat Lunak

No	Nama Perangkat	Spesifikasi
1	Sistem Operasi	Windows 8 Professional
2	JDK (*Java Development Kit*)	JDK 1.7.0
3	JRE (*Java Runtime Environment*)	JRE 7

Sedangkan kebutuhan Perangkat keras untuk menggunakan aplikasi steganografi ini yaitu seperti pada tabel III.5.

Tabel III.5 Kebutuhan Perangkat Lunak

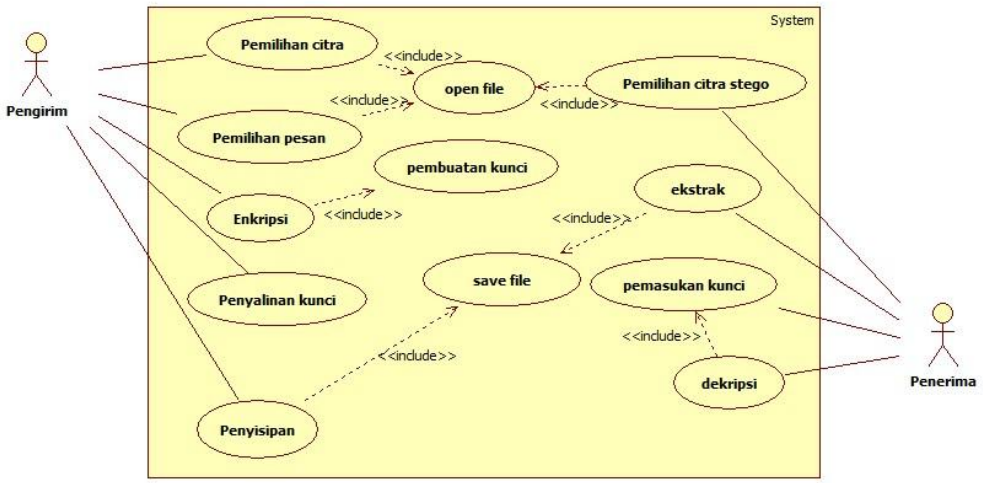
No	Nama Perangkat	Spesifikasi
1	Sistem Operasi	Windows xp
2	JDK (*Java Development Kit*)	JDK 1.7.0
3	JRE (*Java Runtime Environment*)	JRE 7

III.1.4 Analisis Kebutuhan Fungsional

Analisi kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Aplikasi yang dibangun akan dimodelkan menggunakan *Unified Modeling Language* (UML), dan *tools* yang akan digunakan yaitu *use case* diagram, *activity* diagram, *sequence* diagram, dan *class* diagram.

III.1.4.1 *Use Case Diagram*

Use case diagram digunakan untuk mengetahui apa saja yang dapat dilakukan oleh pengguna/aktor terhadap fungsionalitas yang terdapat pada aplikasi yang dibangun. *Use case* diagram pada aplikasi steganografi terlihat pada gambar III.11.



Gambar III.11 *Use Case* Diagram Aplikasi Steganografi

III.1.4.2 Skenario *Use Case*

Skenario *use case* merupakan penjelasan secara rinci setiap *use case* yang terdapat pada diagram *use case*. Skenario *use case* pada aplikasi ini yaitu sebagai berikut:

1. Skenario Pemilihan Citra

Skenario *use case* pemilihan citra dapat dilihat pada tabel III.6.

Tabel III.6 Skenario *Use Case* Pemilihan Citra

Use case name	Pemilihan Citra
Actor	Pengirim
Use case description	Digunakan untuk memilih *file* citra yang akan digunakan sebagai media penampung pesan.
Precondition	Sistem menampilkan menu penyisipan pesan.
Trigger	Pengirim menekan tombol *browse* citra.
Scenario :	
1. Pengirim menekan tombol *browse* citra 2. Sistem menampilkan dialog *open file* 3. Pengirim memilih citra yang akan digunakan 4. Pengirim menekan tombol *open* 5. Sistem Menampilkan informasi citra	
Exception :	
1. Pengirim menekan tombol *cancel*, pemilihan citra dibatalkan.	

2. Skenario Pemilihan Pesan

Skenario *use case* pemilihan pesan dapat dilihat pada tabel III.7.

Tabel III.7 Skenario *Use Case* Pemilihan Pesan

Use case name	Pemilihan Pesan
Actor	Pengirim
Use case description	Digunakan untuk memilih *file* pesan yang akan disisipkan kedalam citra.
Precondition	Sistem menampilkan menu penyisipan pesan.
Trigger	Pengirim menekan tombol *browse* pesan.
Scenario :	

1. Pengirim menekan tombol *browse* pesan 2. Sistem menampilkan dialog *open file* 3. Pengirim memilih pesan yang akan disisipkan 4. Pengirim menekan tombol *open* 5. Sistem menampilkan informasi pesan
Exception :
1. Pengirim menekan tombol *cancel*, pemilihan pesan dibatalkan.

3. Skenario Enkripsi

Skenario *use case* enkripsi dapat dilihat pada tabel III.8.

Tabel III.8 Skenario *Use Case* Enkripsi

Use case name	Enkripsi
Actor	Pengirim
Use case description	Digunakan untuk mengenkripsi *file* pesan yang akan disisipkan ke dalam citra.
Precondition	Sistem menampilkan menu penyisipan pesan, dan pengirim telah memilih pesan.
Trigger	Pengirim menekan tombol enkripsi.
Scenario :	
1. Pengirim menekan tombol enkripsi. 2. Sistem melakukan proses pembuatan kunci. 3. Sistem melakukan proses enkripsi. 4. Sistem menampilkan informasi enkripsi. 5. Sistem menampilkan kunci yang digunakan untuk enkripsi.	
Exception :	
1. Enkripsi gagal. a. Sistem menampilkan informasi pesan gagal dienkripsi. b. Pengirim memilih pesan yang akan disisipkan. c. Pengirim menekan tombol enkripsi.	

4. Skenario Penyalinan Kunci

Skenario *use case* penyalinan kunci dapat dilihat pada tabel III.9.

Tabel III.9 Skenario *Use Case* Penyalinan Kunci

Use case name	Penyalinan Kunci
Actor	Pengirim
Use case description	Digunakan untuk Menyalin kunci yang dihasilkan dari proses enkripsi

Precondition	Sistem menampilkan menu penyisipan pesan, dan pengirim telah melakukan proses enkripsi.
Trigger	Pengirim menekan tombol *copy*.
Scenario :	
1. Pengirim menekan tombol *copy*. 2. Sistem menyalin kunci ke *clipboard* 3. Sistem menampilkan info penyalinan	
Exception :	
1. Kunci belum terisi. a. Sistem menampilkan informasi gagal penyalinan.	

5. Skenario Penyisipan

Skenario *use case* penyisipan dapat dilihat pada tabel III.10.

Tabel III.10 Skenario *Use Case* Penyisipan

Use case name	Penyisipan
Actor	Pengirim
Use case description	Digunakan untuk menyisipkan pesan ke dalam citra.
Precondition	Sistem menampilkan menu penyisipan pesan, pengirim telah memilih citra, pengirim telah memilih pesan, dan pesan telah dienkripsi.
Trigger	Pengirim menekan tombol *embed*.
Scenario :	
1. Pengirim menekan tombol *embed*. 2. Sistem menampilkan *save dialog*. 3. Pengirim memilih letak penyimpanan *citra stego*. 4. Pengirim menekan tombol *save*. 5. Sistem melakukan proses penyisipan 6. Sistem menampilkan info penyisipan	
Exception :	
1. Penyisipan gagal a. Sistem menampilkan informasi penyisipan gagal. b. Pengirim memilih citra, dan pesan c. Pengirim menekan tombol enkripsi. 2. Pengirim menekan tombol cancel, penyisipan pesan dibatalkan.	

6. Skenario Pemilihan Citra *Stego*

Skenario *use case* pemilihan citra *stego* dapat dilihat pada tabel III.11.

Tabel III.11 Skenario *Use Case* Pemilihan Citra *Stego*

Use case name	Pemilihan Citra *Stego*
Actor	Penerima
Use case description	Digunakan untuk memilih *file* citra *stego* yang akan diekstraksi pesan yang ada didalamnya.
Precondition	Sistem menampilkan menu ekstraksi pesan.
Trigger	Penerima menekan tombol *browse* citra.
Scenario :	
1. Penerima menekan tombol *browse* citra. 2. Sistem menampilkan dialog *open file* 3. Penerima memilih citra yang akan diekstrak. 4. Penerima menekan tombol *open*.	
Exception :	
1. Penerima menekan tombol *cancel*, pemilihan citra *stego* dibatalkan.	

7. Skenario Ekstrak

Skenario *use case* ekstrak dapat dilihat pada tabel III.12.

Tabel III.12 Skenario *Use Case* Ekstrak

Use case name	Ekstrak
Actor	Penerima
Use case description	Digunakan untuk ekstraksi pesan yang terdapat pada citra *stego*.
Precondition	Sistem menampilkan menu ekstraksi pesan, Penerima telah memilih citra *stego*.
Trigger	Penerima menekan tombol ekstrak.
Scenario :	
1. Penerima menekan tombol ekstrak. 2. Sistem menampilkan save dialog. 3. Penerima memilih letak penyimpanan pesan. 4. Penerima menekan tombol *save*. 5. Sistem melakukan proses ekstraksi 6. Sistem menampilkan info ekstraksi. 7. Sistem menampilkan letak pesan.	
Exception :	
1. Ekstraksi gagal. a. Sistem menampilkan informasi ekstraksi gagal.	

b. Penerima memilih *file* citra *stego.* c. Penerima menekan tombol ekstraksi 2. Penerima menekan tombol *cancel,* ekstraksi pesan dibatalkan.

8. Skenario Pemasukan Kunci

Skenario *use case* pemasukan kunci dapat dilihat pada tabel III.13.

Tabel III.13 Skenario *Use Case* Pemasukan Kunci

Use case name	Pemasukan Kunci
Actor	Penerima
Use case description	Digunakan untuk memasukan kunci yang akan digunakan untuk dekripsi pesan.
Precondition	Sistem menampilkan menu ekstraksi pesan, pesan telah diekstrak.
Trigger	Penerima memasukan kunci
Scenario :	
1. Sistem menampilkan letak pesan. 2. Penerima mengetikan kunci yang digunakan.	
Exception :	

9. Skenario Dekripsi

Skenario *use case* dekripsi dapat dilihat pada tabel III.14.

Tabel III.14 Skenario *Use Case* Dekripsi

Use case name	Dekripsi
Actor	Penerima
Use case description	Digunakan untuk mendekripsi pesan yang telah diekstraksi.
Precondition	Sistem menampilkan menu ekstraksi pesan, pesan telah diekstraksi, kunci telah dimasukan.
Trigger	Penerima menekan tombol dekripsi.
Scenario :	
1. Penerima menekan tombol dekripsi. 2. Sistem mengecek masukan kunci 3. Sistem melakukan proses dekripsi 4. Sistem menampilkan informasi dekripsi.	
Exception :	

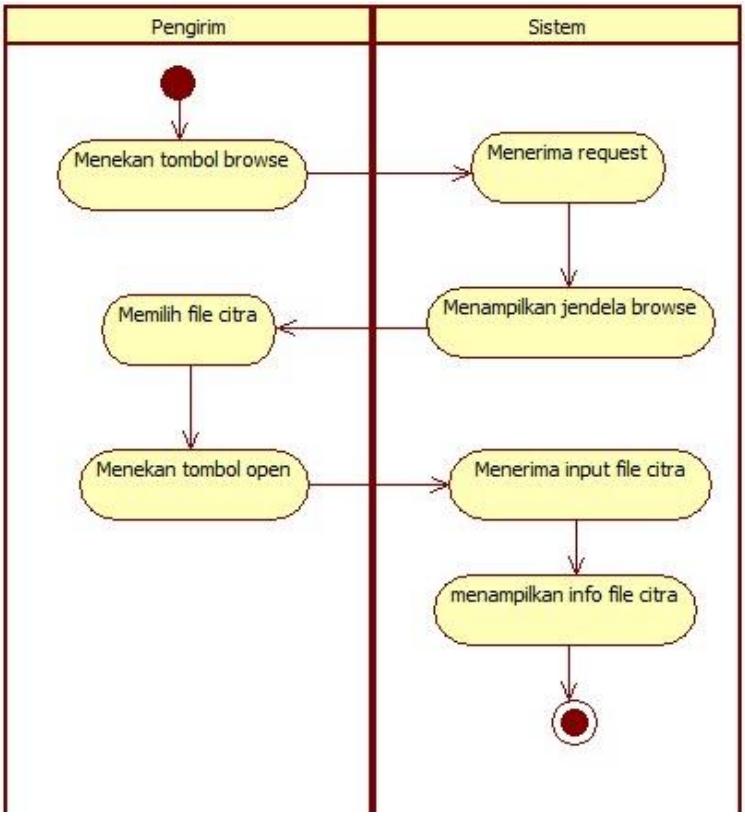
1. Enkripsi gagal. a. Sistem menampilkan informasi pesan gagal didekripsi. b. Penerima memasukan kunci dekripsi. c. Penerima menekan tombol dekripsi.

III.1.4.3 *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram yang terdapat pada aplikasi yang dibangun yaitu sebagai berikut:

1. *Activity* Diagram Pemilihan Citra

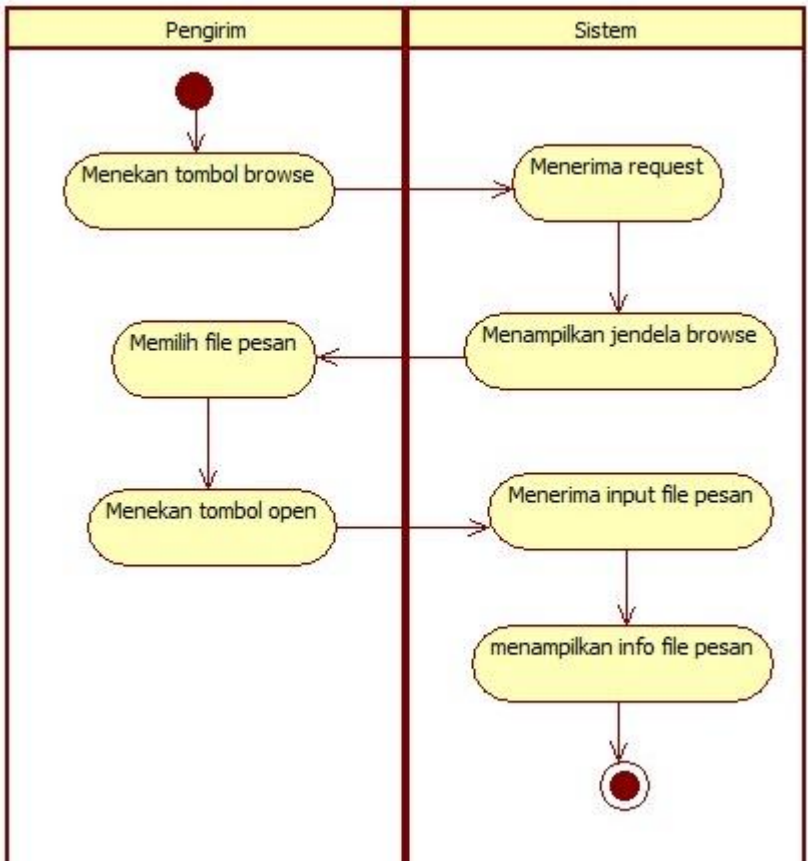
Activity diagram pemilihan citra menggambarkan alir aktivitas pengambilan *file* citra antara pengirim dangan sistem seperti terlihat pada gambar III.12.



Gambar III.12 *Activity* Diagram Pemilihan Citra

2. *Activity* Diagram Pemilihan Pesan

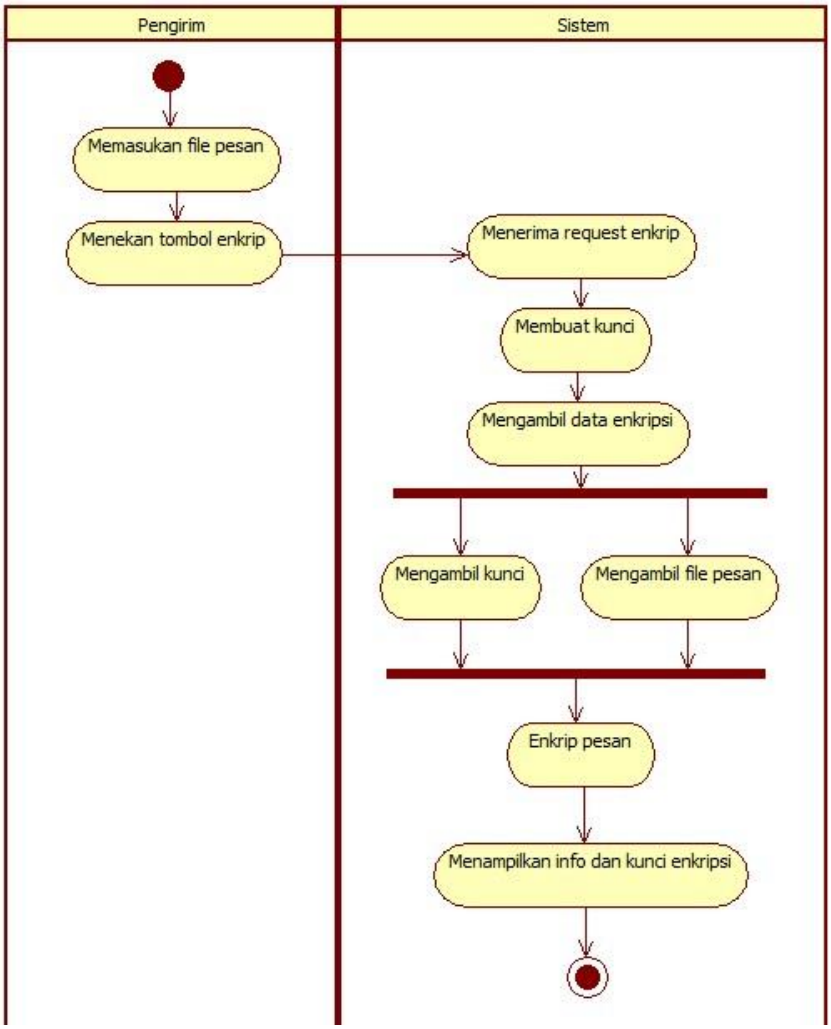
Activity diagram pemilihan pesan menggambarkan alir aktivitas pengambilan *file* pesan antara pengirim dangan sistem seperti terlihat pada gambar III.13.



Gambar III.13 *Activity* Diagram Pemilihan Pesan

3. *Activity* Diagram Enkripsi

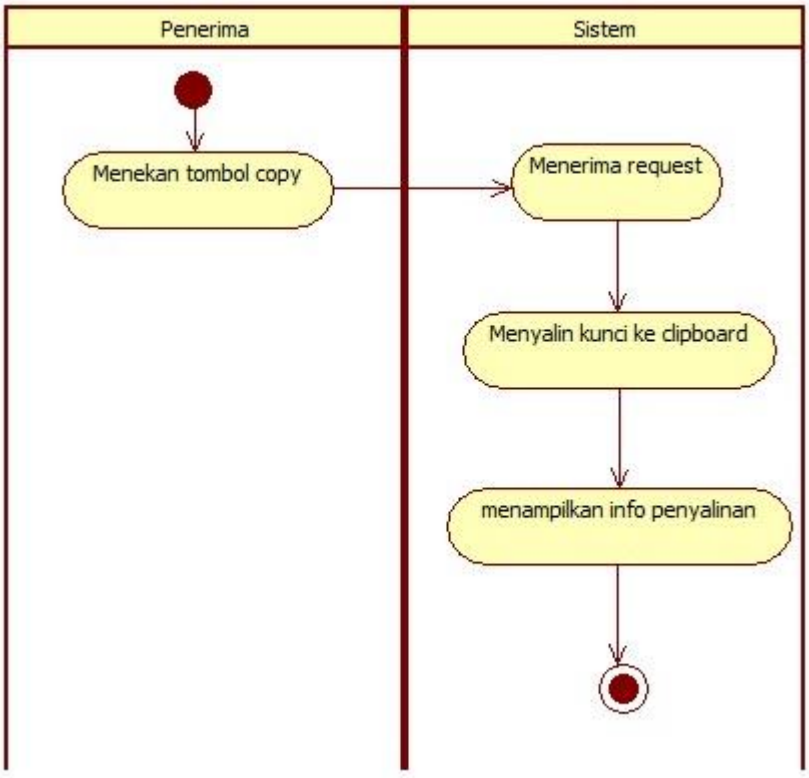
Activity diagram enkripsi menggambarkan alir aktivitas enkripsi yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar III.14.



Gambar III.14 *Activity* Diagram Enkripsi

4. *Activity* Diagram Penyalinan Kunci

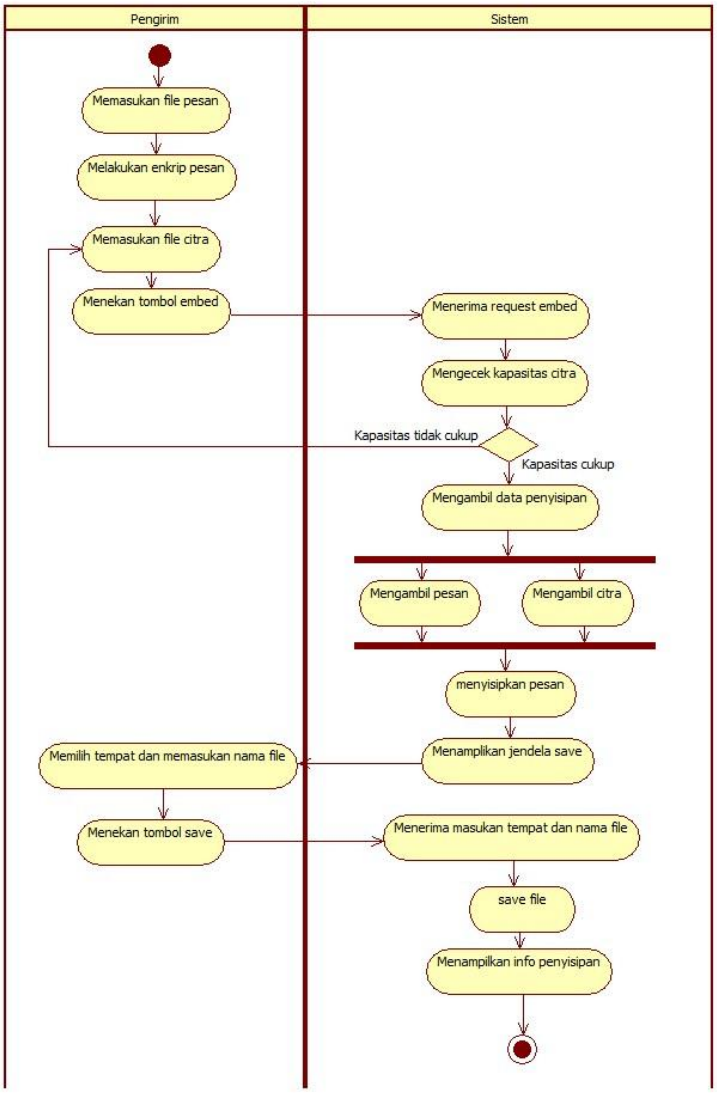
Activity diagram penyisipan menggambarkan alir aktivitas penyalinan kunci yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar III.15.



Gambar III.15 *Activity* Diagram Penyalinan Kunci

5. *Activity* Diagram Penyisipan

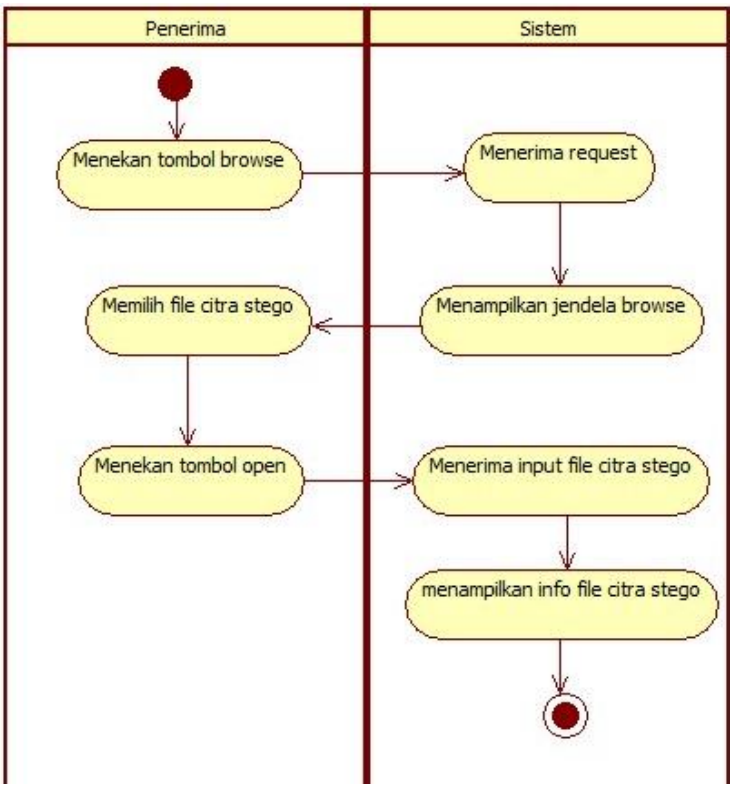
Activity diagram penyisipan menggambarkan alir aktivitas penyisipan yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar III.16.



Gambar III.16 *Activity* Diagram Penyisipan

6. *Activity* Diagram Pemilihan Citra *Stego*

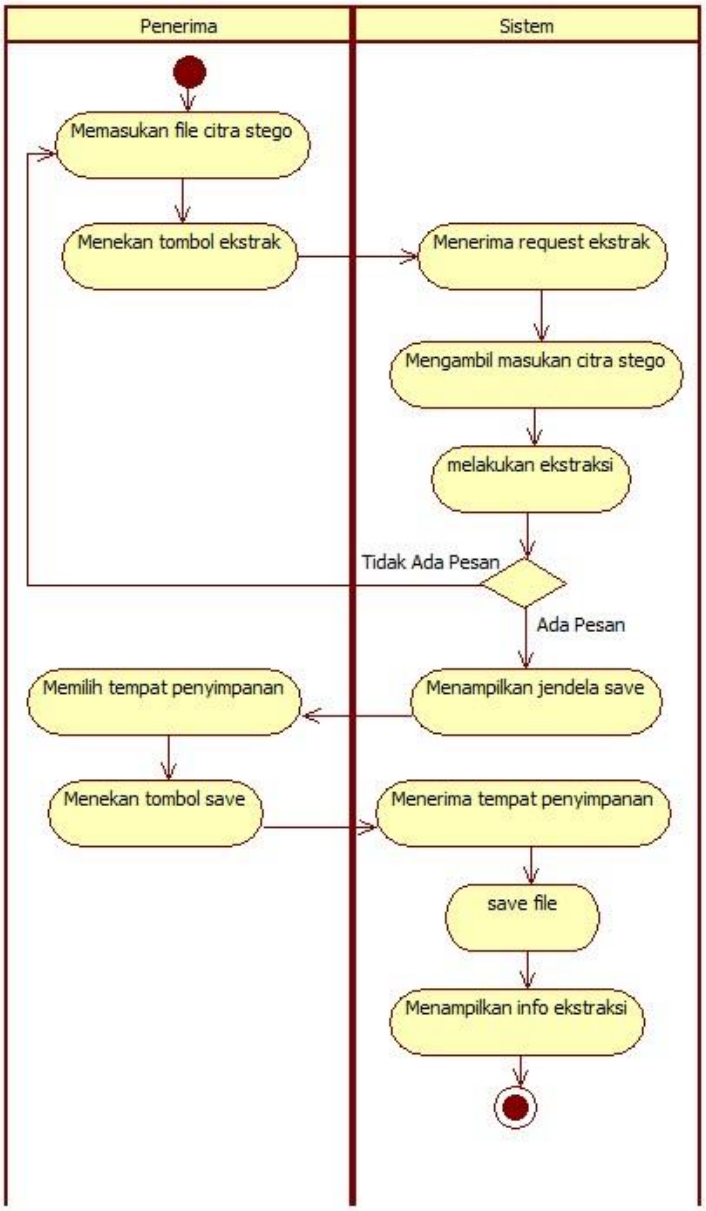
Activity diagram pemilihan citra *stego* menggambarkan alir aktivitas pengambilan *file* citra *stego* antara penerima dangan sistem seperti terlihat pada gambar III.17.



Gambar III.17 *Activity* Diagram Pemilihan Citra *Stego*

7. *Activity* Diagram Ekstrak

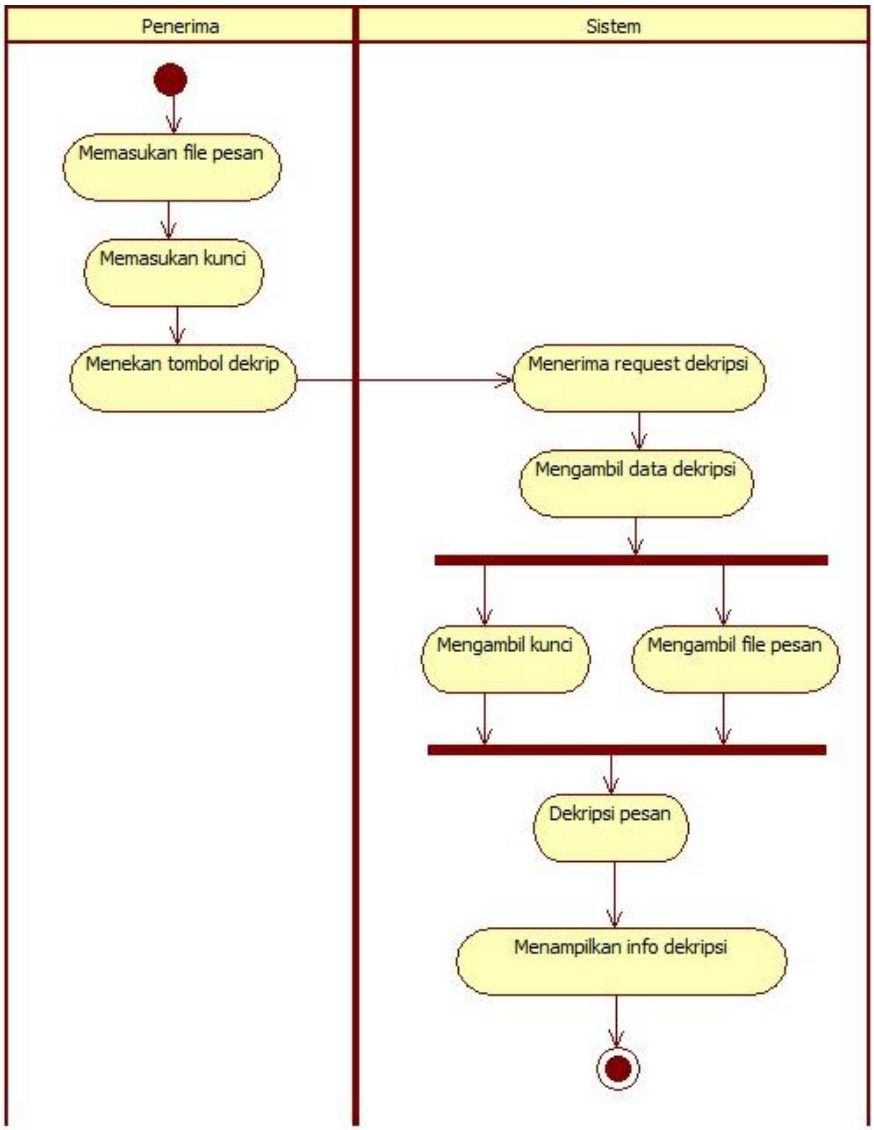
Activity diagram ekstrak menggambarkan alir aktivitas ekstrak yang dilakukan antara penerima dangan sistem seperti terlihat pada gambar III.18.



Gambar III.18 *Activity* Diagram Ekstrak

8. *Activity* Diagram Dekripsi

Activity diagram dekripsi menggambarkan alir aktivitas dekripsi yang dilakukan antara penerima dangan sistem seperti terlihat pada gambar III.19.



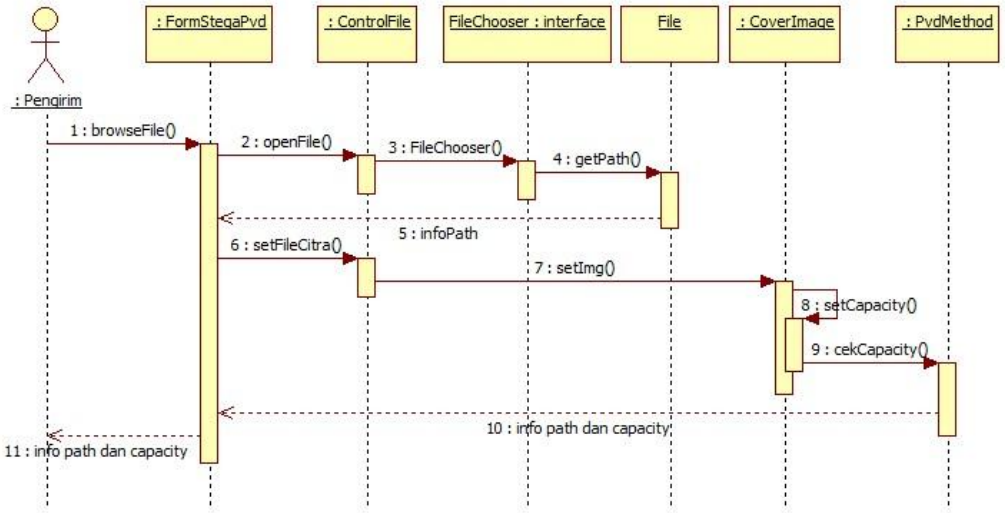
Gambar III.19 *Activity* Diagram Dekripsi

III.1.4.4 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) yang digambarkan terhadap waktu. *Sequence* diagram yang terdapat pada aplikasi steganografi yaitu sebagai berikut:

1. *Sequence* Diagram Pemilihan Citra

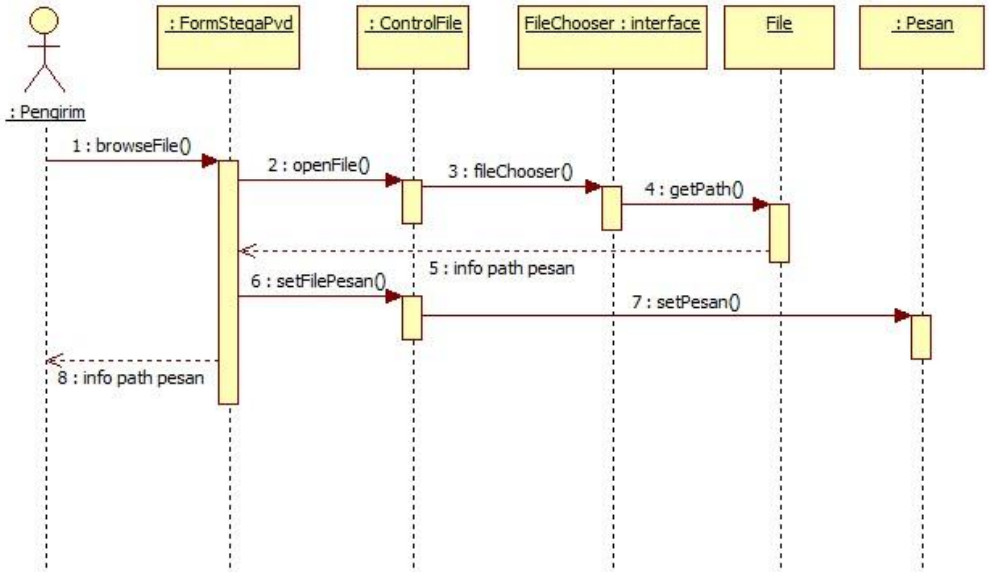
Sequence diagram pemilihan citra merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam pengambilan *file* citra. *Sequence* diagram pemilihan citra aplikasi steganografi terlihat seperti pada gambar III.20.



Gambar III.20 *Sequence* Diagram Pemilihan Citra

2. *Sequence* Diagram Pemilihan Pesan

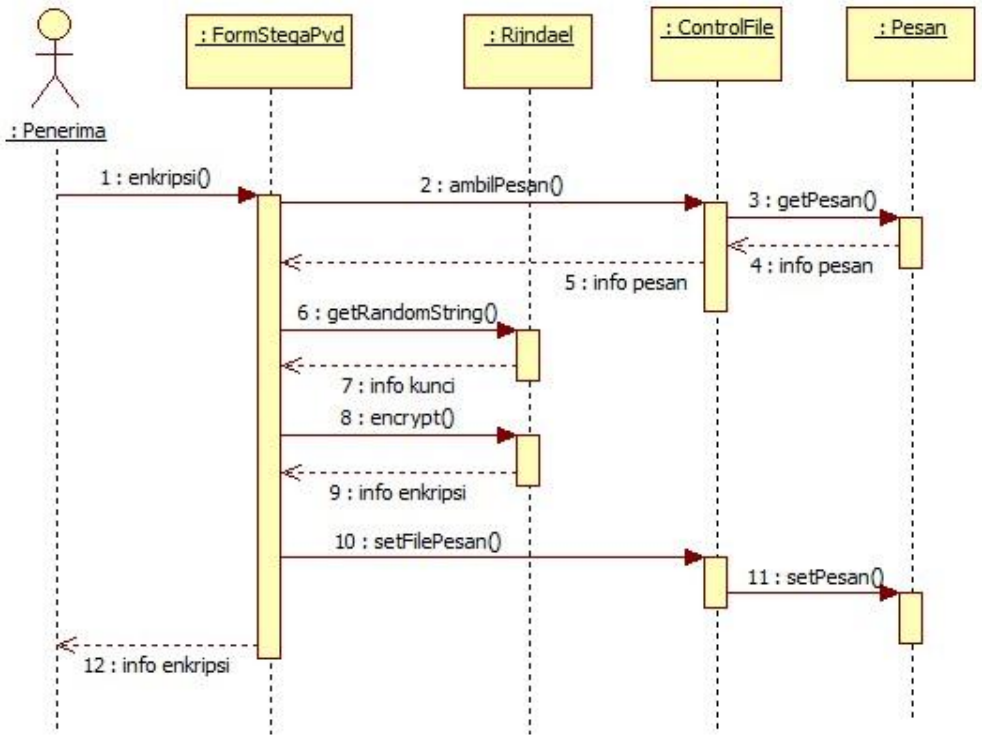
Sequence diagram pemilihan pesan merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam pengambilan *file* pesan. *Sequence* diagram pemilihan pesan aplikasi steganografi terlihat seperti pada gambar III.21.



Gambar III.21 *Sequence* Diagram Pemilihan Pesan

3. *Sequence* Diagram Enkripsi

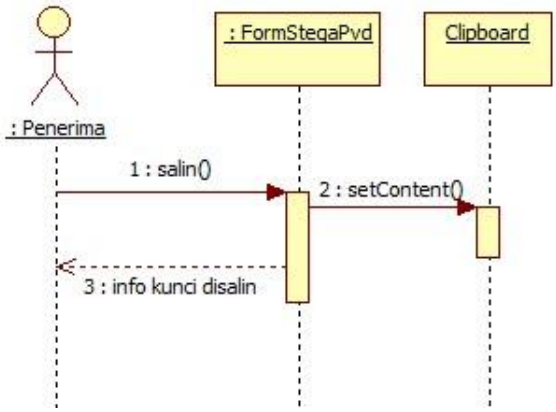
Sequence diagram enkripsi merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam mengenkripsi pesan. *Sequence* diagram enkripsi aplikasi steganografi terlihat seperti pada gambar III.22.



Gambar III.22 *Sequence* Diagram Enkripsi

4. *Sequence* Diagram Penyalinan Kunci

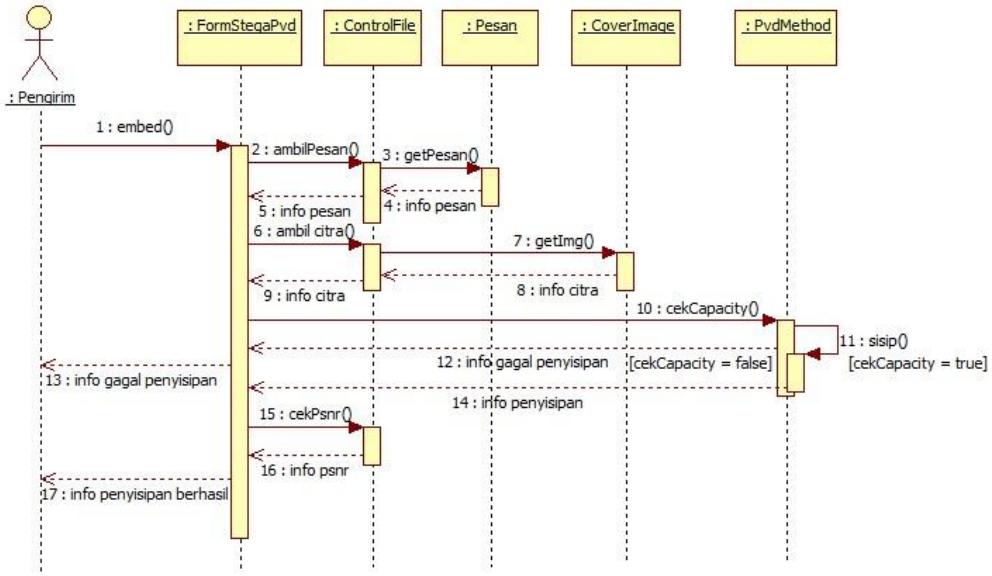
Sequence diagram penyalinan kunci merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam penyalinan kunci. *Sequence* diagram penyalinan kunci aplikasi steganografi terlihat seperti pada gambar III.23.



Gambar III.23 *Sequence* Diagram Penyalinan Kunci

5. *Sequence* Diagram Penyisipan

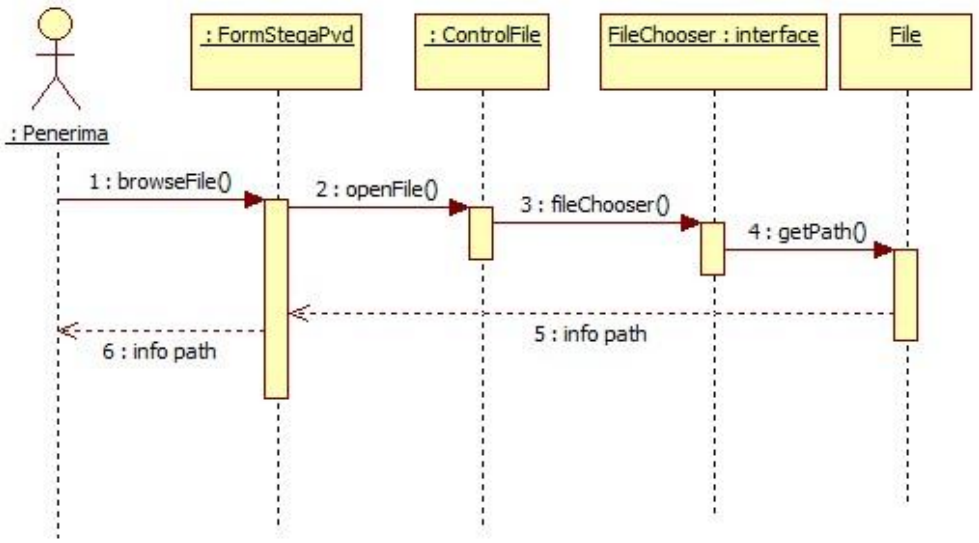
Sequence diagram penyisipan merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam penyisipan pesan kedalam citra. *Sequence* diagram penyisipan aplikasi steganografi terlihat seperti pada gambar III.24.



Gambar III.24 *Sequence* Diagram Penyisipan

6. *Sequence* Diagram Pemilihan Citra *Stego*

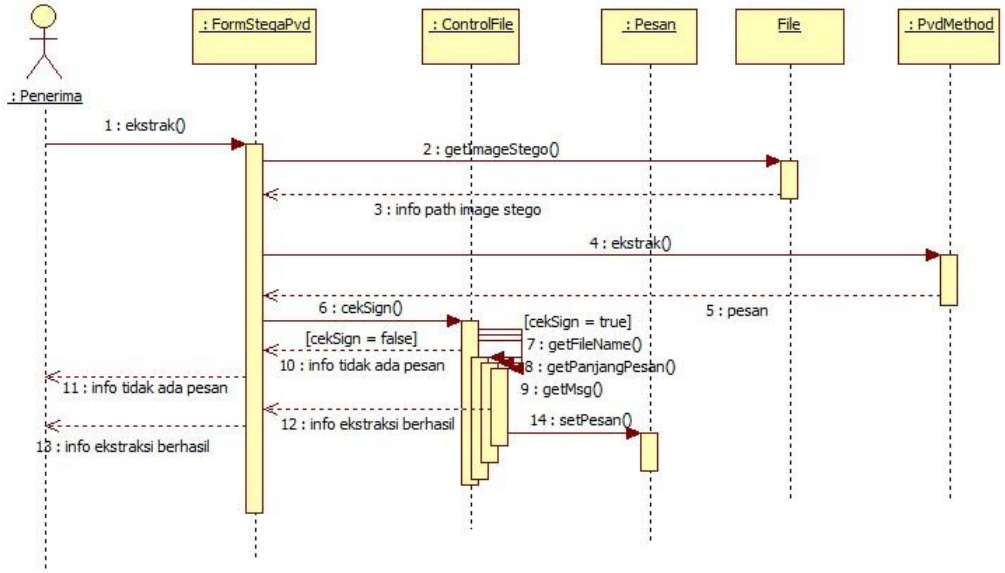
Sequence diagram pemilihan citra *stego* merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam pengambilan *file* citra *stego*. *Sequence* diagram pemilihan citra *stego* aplikasi steganografi terlihat seperti pada gambar III.25.



Gambar III.25 *Sequence* Diagram Pemilihan Citra *Stego*

7. *Sequence* Diagram Ekstrak

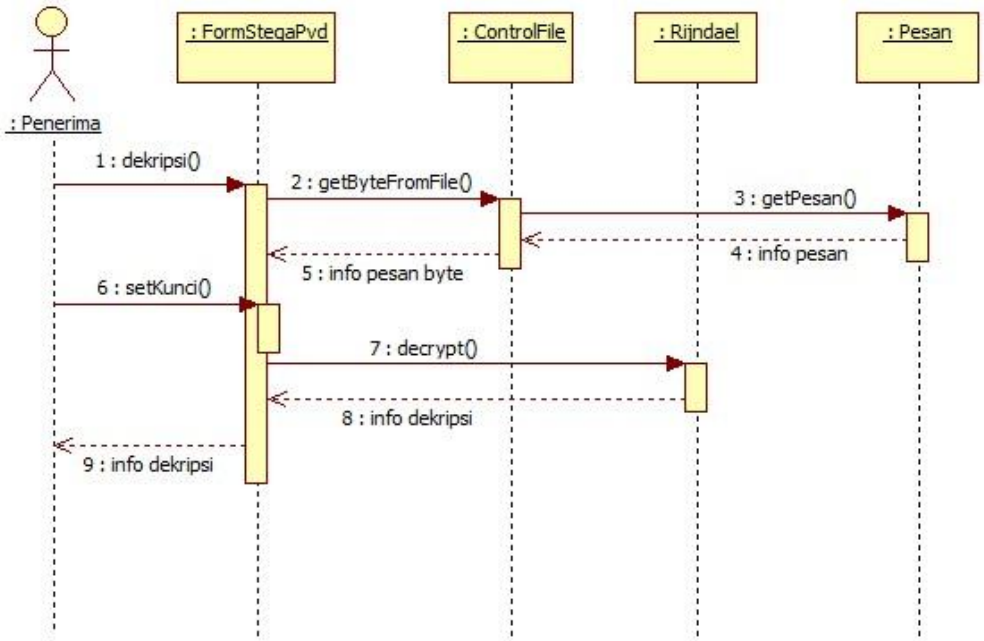
Sequence diagram ekstrak merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam ekstraksi pesan dari citra. *Sequence* diagram ekstrak aplikasi steganografi terlihat seperti pada gambar III.26.



Gambar III.26 *Sequence* Diagram Ekstraksi

8. *Sequence* Diagram Dekripsi

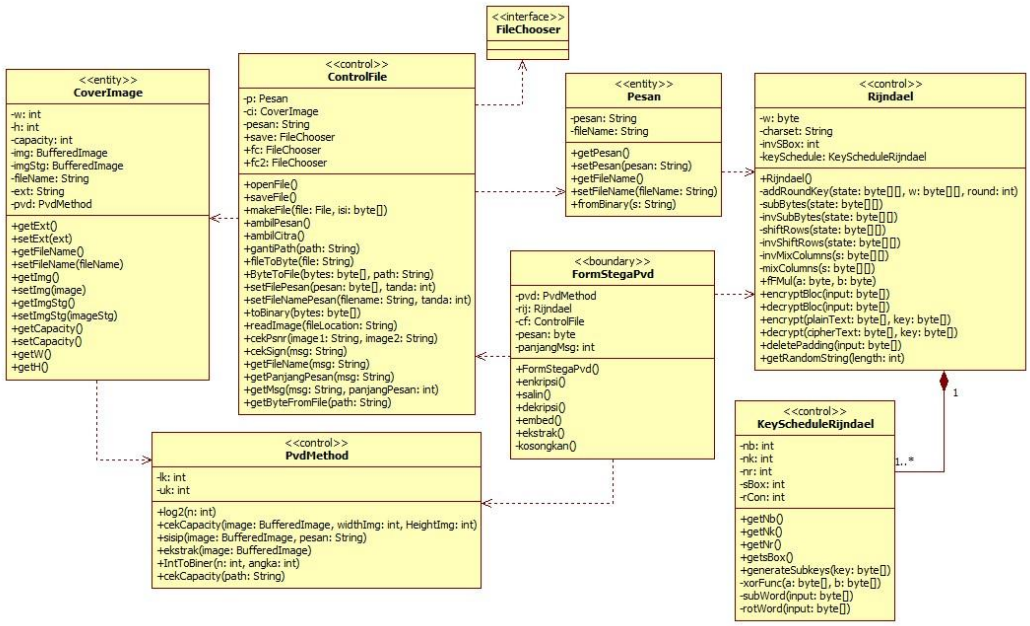
Sequence diagram dekripsi merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam dekripsi pesan. *Sequence* diagram dekripsi aplikasi steganografi terlihat seperti pada gambar III.27.



Gambar III.27 *Sequence* Diagram Dekripsi

III.1.4.5 *Class Diagram*

Class diagram digunakan untuk menggambarkan keadan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem. *Class* diagram yang digunakan pada pembangunan aplikasi steganografi ini yaitu terlihat pada gambar III.28.



Gambar III.28 *Class* Diagram Aplikasi Steganografi

III.2 Perancangan Sistem

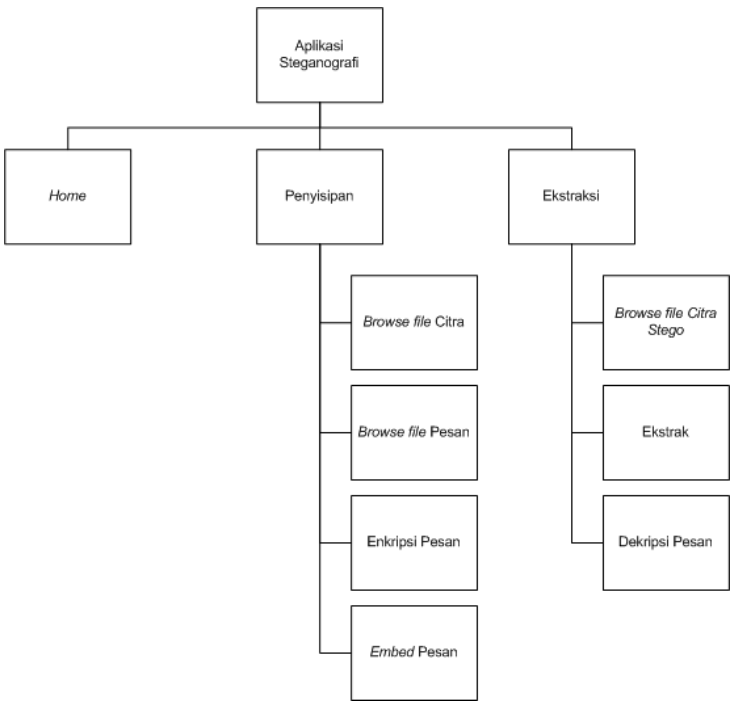
Perancangan sistem merupakan suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Tahap perancangan terdiri dari dua bagian yaitu perancangan arsitektur, dan peracangan *method*.

III.2.1 Perancangan Arsitektur

Perancangan arsitektur terdiri dari perancangan struktur menu, perancangan antarmuka, dan perancangan jaringan semantik.

III.2.1.1 Perancangan Struktur Menu

Perancangan menu dilakukan untuk mempermudah interaksi antara sistem dengan pengguna. Gambar III.29 merupakan struktur menu dari aplikasi steganografi yang dibangun.



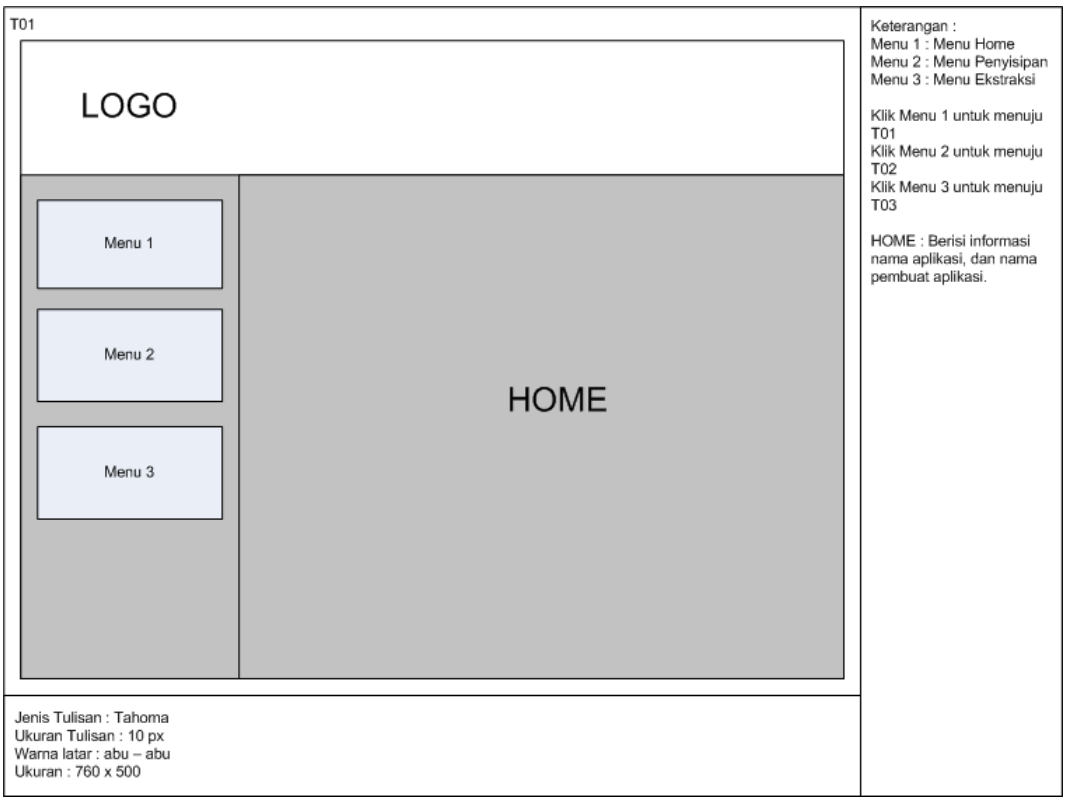
Gambar III.29 Struktur Menu Aplikasi Steganografi

III.2.1.2 Perancangan Antarmuka

Perancangan antarmuka adalah tahapan pembuatan rancangan antarmuka untuk digunakan pada pembangunan aplikasi steganografi yang dibagi menjadi empat bagian yaitu perancangan antarmuka *home*, perancangan antarmuka penyisipan, perancangan antarmuka ekstraksi, dan Perancangan Pesan.

III.2.1.2.1 Perancangan Antarmuka Home (T01)

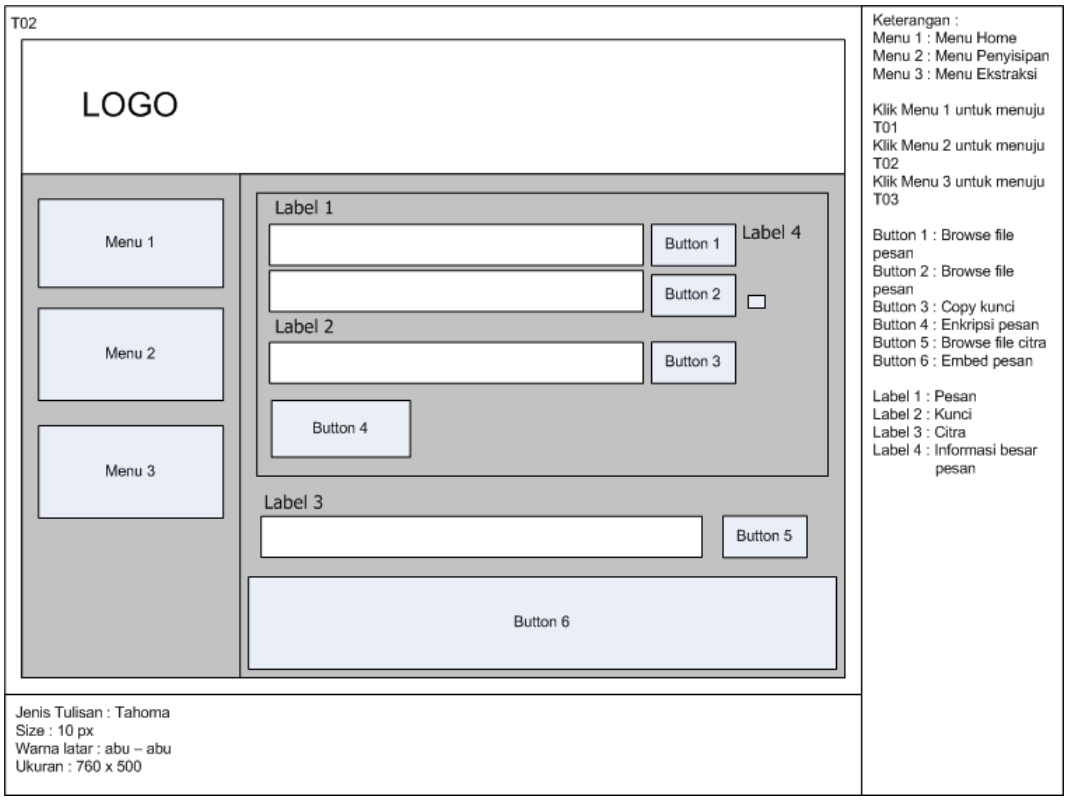
Perancangan antarmuka *home* merupakan tahapan rancangan antarmuka dari awal aplikasi ketika dijalankan. Perancangan antarmuka *home* dari aplikasi steganografi ini terlihat pada gambar III.30.



Gambar III.30 Perancangan Antarmuka *Home*

III.2.1.2.2 Perancangan Antarmuka Penyisipan (T02)

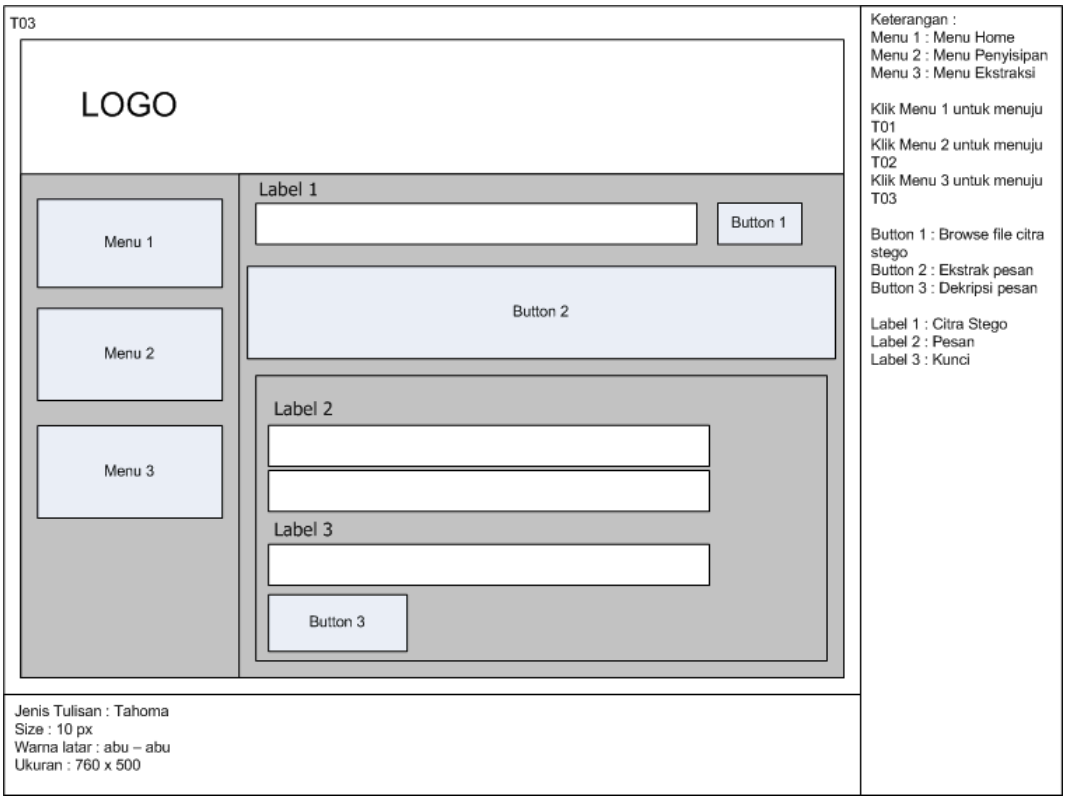
Perancangan antarmuka *home* merupakan tahapan rancangan antarmuka dari awal aplikasi ketika dijalankan. Perancangan antarmuka *home* dari aplikasi steganografi ini terlihat pada gambar III.31.



Gambar III.31 Perancangan Antarmuka Penyisipan

III.2.1.2.3 Perancangan Antarmuka Ekstraksi (T03)

Perancangan antarmuka *home* merupakan tahapan rancangan antarmuka dari awal aplikasi ketika dijalankan. Perancangan antarmuka *home* dari aplikasi steganografi ini terlihat pada gambar III.32.



Gambar III.32 Perancangan Antarmuka Ekstraksi

III.2.1.3 Perancangan Pesan

Pesan merupakan tampilan dari suatu perangkat lunak yang berfungsi untuk menyampaikan notifikasi dan informasi kepada pengguna agar perangkat lunak lebih interaktif. Perancangan pesan pada aplikasi steganografi dibagi menjadi dua bagian yaitu perancangan pesan menu penyisipan dan perancangan pesan menu ekstraksi.

III.2.1.3.1 Perancangan pesan menu penyisipan

Perancangan pesan pada menu penyisipan ini terdapat enam perancangan pesan yaitu sebagai berikut :

1. **Perancangan pesan jika citra belum dipilih (M01)**

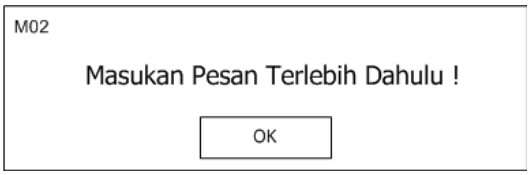
   Pesan ini muncul jika pengguna menekan tombol *embed* sebelum memilih citra. Perancangan pesan ini terlihat pada gambar III.33.



Gambar III.33 Perancangan Pesan Jika Belum Memilih Citra

2. Perancangan pesan jika pesan belum dipilih (M02)

Pesan ini muncul jika pengguna menekan tombol *embed* atau enkripsi sebelum memilih pesan. Perancangan pesan ini terlihat pada gambar III.34.



Gambar III.34 Perancangan Pesan Jika Belum Memilih Pesan

3. Perancangan pesan informasi citra (M03)

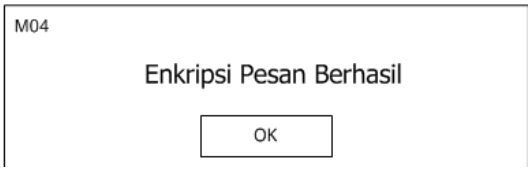
Pesan ini muncul setelah pengguna memilih citra yang digunakan sebagai media penampung. Pesan ini berisi informasi resolusi, dan kapasitas pesan yang dapat disisipkan. Perancangan pesan ini terlihat pada gambar III.35.



Gambar III.35 Perancangan Pesan Informasi Citra

4. Perancangan pesan informasi enkripsi berhasil (M04)

Pesan ini muncul jika enkripsi berhasil dilakukan. Perancangan pesan ini terlihat pada gambar III.36.



Gambar III.36 Perancangan Pesan Enkripsi Berhasil

5. Perancangan pesan jika kapasitas citra tidak mencukupi (M05)

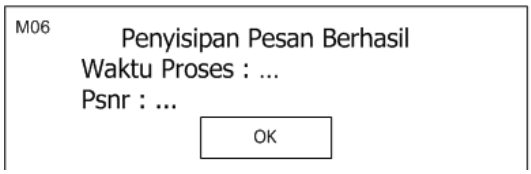
Pesan ini muncul jika kapasitas citra tidak mencukupi untuk menyisipkan pesan pada saat proses penyisipan. Perancangan pesan ini terlihat pada gambar III.37.



Gambar III.37 Perancangan Pesan Kapasitas Tidak Cukup

6. Perancangan pesan informasi penyisipan berhasil (M06)

Pesan ini muncul jika penyisipan telah berhasil dilakukan. Informasi yang ditampilkan yaitu waktu proses penyisipan, dan nilai PSNR dari citra. Perancangan pesan ini terlihat pada gambar III.38.



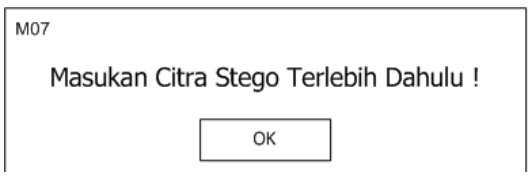
Gambar III.38 Perancangan Pesan Penyisipan Berhasil

III.2.1.3.2 Perancangan pesan menu ekstraksi

Perancangan pesan pada menu ekstraksi ini terdapat empat perancangan pesan yaitu sebagai berikut :

1. Perancangan pesan jika citra *stego* belum dipilih (M07)

Pesan ini muncul jika pengguna menekan tombol ekstrak sebelum memilih citra *stego*. Perancangan pesan ini terlihat pada gambar III.39.



Gambar III.39 Perancangan Pesan Jika Belum Memilih Citra *Stego*

2. Perancangan pesan jika kunci belum dimasukan (M08)

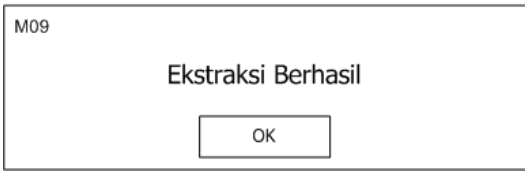
Pesan ini muncul jika pengguna menekan tombol dekripsi sebelum memasukan kunci. Perancangan pesan ini terlihat pada gambar III.40.



Gambar III.40 Perancangan Pesan Jika Belum Memasukan Kunci

3. Perancangan pesan ekstraksi berhasil (M09)

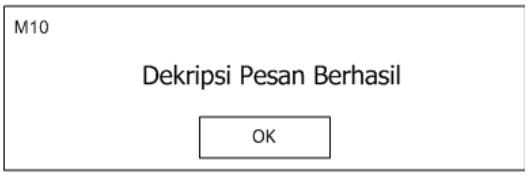
Pesan ini muncul jika ekstraksi pesan berhasil dilakukan. Perancangan pesan ini terlihat pada gambar III.41.



Gambar III.41 Perancangan Pesan Ekstraksi Berhasil

4. Perancangan pesan dekripsi berhasil (M10)

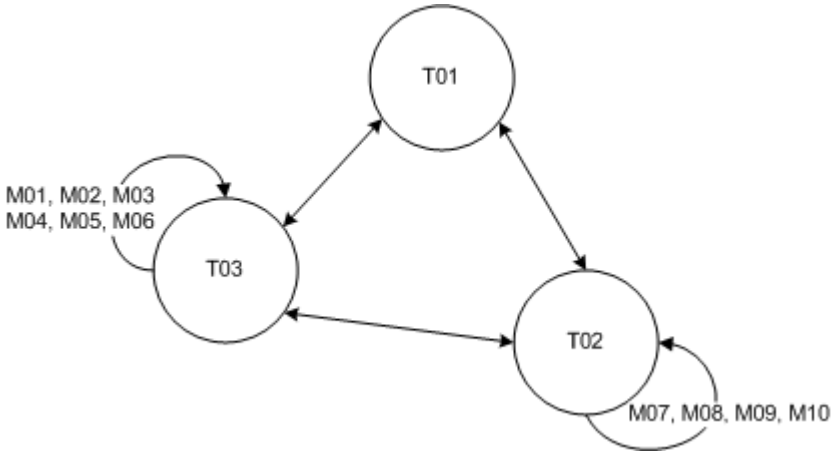
Pesan ini muncul jika dekripsi pesan berhasil dilakukan. Perancangan pesan ini terlihat pada gambar III.42.



Gambar III.42 Perancangan Pesan Dekripsi Berhasil

III.2.1.4 Perancangan Jaringan Semantik

Jaringan semantik adalah diagram yang menggambarkan aliran-aliran menu dan pesan dalam sebuah program. Jaringan semantik dari aplikasi steganografi yang dibangun yaitu seperti pada gambar III.43.



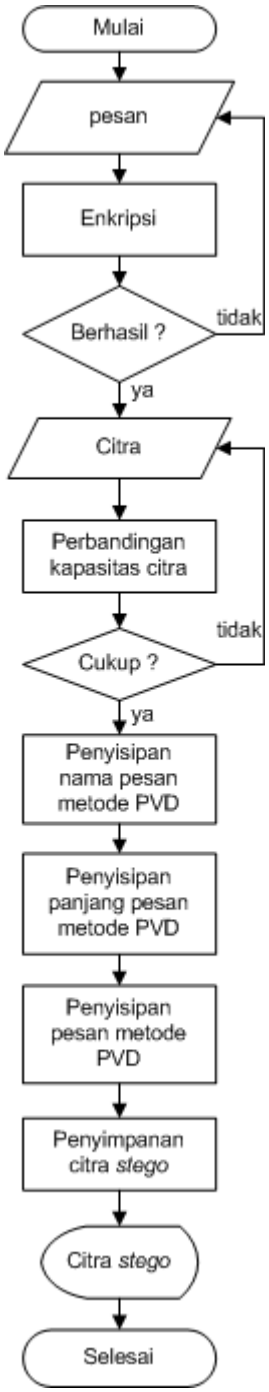
Gambar III.43 Jaringan Semantik Aplikasi Steganografi

III.2.2 Perancangan *Method*

Perancangan *method* merupakan perancangan yang digunakan untuk memberikan gambaran algoritma untuk diimplementasikan kedalam aplikasi yang dibangun. Perancangan *method* pada aplikasi ini dibagi menjadi dua bagian, yaitu perancangan *method* penyisipan, dan perancangan *method* ekstraksi.

III.2.2.1 Perancangan *Method* Penyisipan

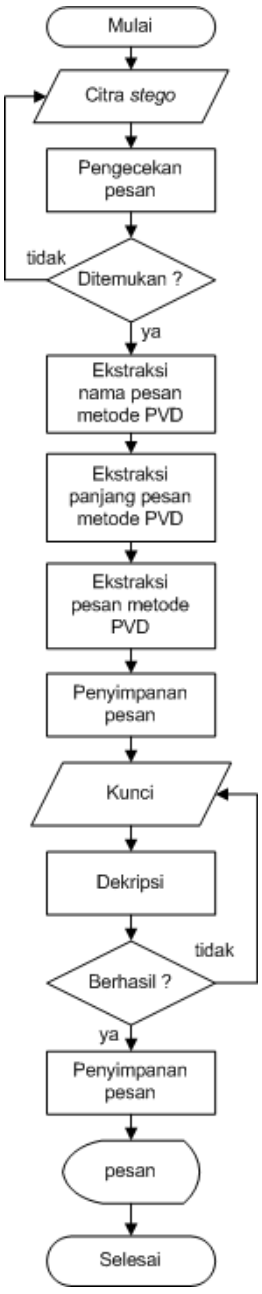
Perancangan *method* penyisipan merupakan gambaran umum dari langkah-langkah proses penyisipan di aplikasi steganografi yang dibangun. Proses penyisipan dimulai dengan mengenkripsi pesan dan melakukan penyisipan terhadap pesan yang telah terenkripsi ke dalam citra. Perancangan *method* penyisipan pada aplikasi steganografi digambarkan dalam *flowchart method* penyisipan seperti pada gambar III.44.



Gambar III.44 *Flowchart Method* Penyisipan Aplikasi Steganografi

III.2.2.2 Perancangan *Method* Ekstraksi

Perancangan *method* ekstraksi merupakan gambaran umum dari langkah-langkah proses ekstraksi di aplikasi steganografi yang dibangun. Perancangan *method* ekstraksi pada aplikasi steganografi digambarkan dalam *flowchart method* ekstraksi seperti pada gambar III.45.



Gambar III.45 *Flowchart Method* Ekstraksi Aplikasi Steganografi

III.2.3 Perancangan Pengujian Performansi Metode Steganografi

Perancangan pengujian performansi metode steganografi digunakan untuk mengetahui rancangan pengujian yang akan dilakukan dalam menguji performansi dari metode steganografi yang digunakan. Pengujian performansi yang akan dilakukan yaitu dengan membandingkan performansi metode steganografi yang digunakan dengan performansi metode steganografi yang sejenis, dalam hal ini metode sejenis yang akan dijadikan pembanding yaitu metode *Least Significant Bits* (LSB). Metode ini digunakan, karena mempunyai kesamaan dari cara penyisipan, yang dilakukan secara berurutan dan tempat/ranah penyisipan yang dilakukan pada citra, yaitu pada ranah spasial. Metode LSB mempunyai tahapan-tahapan dalam penyisipan dan ekstraksinya yaitu sebagai berikut:

1. Penyisipan metode LSB
    a. Mengambil *pixel* secara berurut.
    b. Mengambil nilai bit dari setiap komponen warna (RGB) pada *pixel.*
    c. Mengubah nilai bit terakhir atau nilai bit yang paling tidak dari setiap komponen warna pada setiap *pixel* dengan nilai bit dari pesan.
    d. Menyimpan kembali pixel yang telah dirubah kedalam citra.
2. Ekstraksi metode LSB
    a. Mengambil *pixel* secara berurut.
    b. Mengambil nilai bit dari setiap komponen warna (RGB) pada *pixel.*
    c. Mengambil nilai bit terakhir atau nilai bit yang paling tidak signifikan dari setiap komponen warna pada setiap *pixel* dengan nilai bit dari pesan.
    d. Menyimpan bit pesan.