BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 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 non-fungsional, dan analisis kebutuhan fungsional.

3.1.1 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.

3.1.2 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 *Advanced Encryption Standard (AES)* untuk proses enkripsi dan dekripsi dengan panjang kunci 32*bit* sebagai keamanan terhadap informasi yang akan disisipkan. Sedangkan metode steganografi yang digunakan yaitu metode *Random Pixel positioning (RPP)* untuk proses penyisipan dan ekstraksi informasi. Tahapan-tahapan yang dilakukan pada setiap prosesnya yaitu sebagai berikut :

1. Proses Penyisipan
    a. Pengirim memilih *cover* citra yang akan disisipkan
    b. Pengirim memilih *file* pesan yang akan disisipkan

c. Pengirim melakukan enkripsi terhadap pesan menggunakan algoritma *Advanced Encryption Standard (AES)256bit.*

d. Pengirim memilih *file* citra yang akan digunakan sebagai media penampung dari pesan.

e. Pengirim melakukan proses penyisipan menggunakan metode *Random Pixel Positioning(RPP).*

2. Proses Ekstraksi

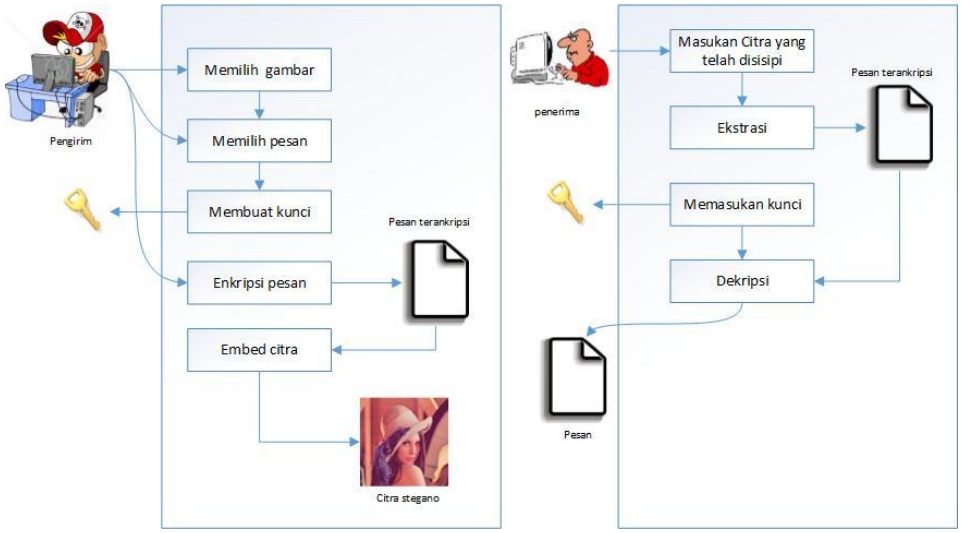
a. Penerima memilih *file* citra yang telah disisipkan pesan (*stego object*).

b. Penerima melakukan proses ekstraksi menggunakan metode *Random Pixel positioning (RPP).*

c. Penerima melakukan dekripsi terhadap pesan menggunakan algoritma *Advanced Encryption Standard (AES).*

d. Penerima menyimpan pesan yang telah diekstraksi dari citra.

Tahapan-tahapan proses pada aplikasi steganografi secara umum dapat dilihat pada gambar 3.1.



Gambar 3. 1 Gambaran Umum Aplikasi Steganografi

3.1.2.1 Analisis Algoritma Advanced Encryption Standard(AES)

Algoritma *Advanced Encryption Standard (AES)* 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 *Advanced*

Encryption Standard (AES) yang digunakaan pada aplikasi steganografi yang dibangun menggunakan ukuran blok dan kunci 32*bit.* Algoritma *Advanced Encryption Standard (AES)* terdapat dua proses yaitu proses penjadwalan kunci dan enkripsi.

3.1.2.1.1 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 *Advanced Encryption Standard (AES)*. Contoh penjadwalan kunci pada algoritma *Advanced Encryption Standard (AES)* jika diketahui kunci yang akan digunakan untuk enkripsi dengan panjang 32 byte yaitu:

***Cipherkey* = abcdefghijklmnopqrstuvwxyz123456**

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 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f 80**

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 32 byte menjadi.

$$W = \begin{bmatrix} 61 & 65 & 69 & 6d & 71 & 75 & 79 & 7d \\ 62 & 66 & 6a & 6e & 72 & 76 & 7a & 7e \\ 63 & 67 & 6b & 6f & 73 & 77 & 7b & 7f \\ 64 & 68 & 6c & 70 & 74 & 78 & 7c & 80 \end{bmatrix}$$

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

$$\begin{bmatrix} \mathbf{7d} \\ 7e \\ 7f \\ 80 \end{bmatrix} = \begin{bmatrix} 7e \\ 7f \\ 80 \\ \mathbf{7d} \end{bmatrix}$$

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

$$\begin{bmatrix} 7e \\ 7f \\ 80 \\ 7d \end{bmatrix} = \begin{bmatrix} f3 \\ d2 \\ cd \\ ff \end{bmatrix}$$

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

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

$$\begin{bmatrix} 61 \\ 62 \\ 63 \\ 64 \end{bmatrix} \oplus \begin{bmatrix} f3 \\ d2 \\ cd \\ ff \end{bmatrix} \oplus \begin{bmatrix} 01 \\ 00 \\ 00 \\ 00 \end{bmatrix} = \begin{bmatrix} 93 \\ b0 \\ ae \\ 9b \end{bmatrix}$$

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

$$\begin{bmatrix} 65 \\ 66 \\ 67 \\ 68 \end{bmatrix} \oplus \begin{bmatrix} 93 \\ b0 \\ ae \\ 9b \end{bmatrix} = \begin{bmatrix} f6 \\ d6 \\ c9 \\ f3 \end{bmatrix} \qquad \begin{bmatrix} 75 \\ 76 \\ 77 \\ 78 \end{bmatrix} \oplus \begin{bmatrix} 83 \\ a0 \\ be \\ 9b \end{bmatrix} = \begin{bmatrix} f6 \\ d6 \\ c9 \\ e3 \end{bmatrix}$$

$$\begin{bmatrix} 69 \\ 6a \\ 6b \\ 6c \end{bmatrix} \oplus \begin{bmatrix} f6 \\ d6 \\ c9 \\ f3 \end{bmatrix} = \begin{bmatrix} 9f \\ bc \\ a2 \\ 9f \end{bmatrix} \qquad \begin{bmatrix} 79 \\ 7a \\ 7b \\ 7c \end{bmatrix} \oplus \begin{bmatrix} f6 \\ d6 \\ c9 \\ e3 \end{bmatrix} = \begin{bmatrix} 8f \\ ac \\ b2 \\ 9f \end{bmatrix}$$

$$\begin{bmatrix} 6d \\ 6e \\ 6f \\ 70 \end{bmatrix} \oplus \begin{bmatrix} 9f \\ bc \\ a2 \\ 9f \end{bmatrix} = \begin{bmatrix} f2 \\ d2 \\ cd \\ ef \end{bmatrix} \qquad \begin{bmatrix} 7d \\ 7e \\ 7f \\ 80 \end{bmatrix} \oplus \begin{bmatrix} 8f \\ ac \\ b2 \\ 9f \end{bmatrix} = \begin{bmatrix} f2 \\ d2 \\ cd \\ 1f \end{bmatrix}$$

$$\begin{bmatrix} 71 \\ 72 \\ 73 \\ 74 \end{bmatrix} \oplus \begin{bmatrix} f2 \\ d2 \\ cd \\ ef \end{bmatrix} = \begin{bmatrix} 83 \\ a0 \\ be \\ 9b \end{bmatrix}$$

6. Simpan kedalam *subkey*.

$$SubKey = \begin{bmatrix} 93 & f6 & 9f & f2 & 83 & f6 & 8f & f2 \\ b0 & d6 & bc & d2 & a0 & d6 & ac & d2 \\ ae & c9 & a2 & cd & be & c9 & b2 & cd \\ 9b & f3 & 9f & ef & 9b & e3 & 9f & 1f \end{bmatrix}$$

Subkey ini yang akan digunakan untuk proses enkripsi atau dekripsi pada algoritma *Advanced Encryption Standard (AES)* pada *round* ke-1 untuk *round* selanjutnya dilakukan penjadwalan kunci kembali sampai *round* ke-14.

3.1.2.1.2 Analisis Proses Enkripsi Algoritma (AES)

Proses enkripsi pada algoritma *Advanced Encryption Standard (AES)* 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 256 bit, sehingga berdasarkan pada tabel 2.1, maka perulangan yang dilakukan sebanyak 14 kali. Contoh enkripsi pada algoritma *advanced encryption standard,* jika diketahui kunci dan *plaintext* yang akan digunakan untuk enkripsi dengan panjang 32 byte.

***Cipherkey* = abcdefghijklmnopqrstuvwxyx123456**

***Plaintext* = UNIVERSITAS KOMPUTER INDONESIA !**

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 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f 80**

***Plaintext* = 55 4E 49 56 45 52 53 49 54 41 53 20 4b 4F 4D 50 55 54 45 52 20 49 4E 44 4F 4E 45 49 53 41 20 21**

Masukan *cipherkey* dan *plaintext* ke dalam blok 32 byte sehingga menjadi

$$State = \begin{bmatrix} 55 & 45 & 54 & 4b & 55 & 20 & 4f & 53 \\ 4e & 52 & 41 & 4f & 54 & 49 & 4e & 41 \\ 49 & 53 & 53 & 4d & 45 & 4e & 45 & 20 \\ 56 & 49 & 20 & 50 & 52 & 44 & 49 & 21 \end{bmatrix} Cipherkey$$

$$= \begin{bmatrix} 61 & 65 & 69 & 6d & 71 & 75 & 79 & 7d \\ 62 & 66 & 6a & 6e & 72 & 76 & 7a & 7e \\ 63 & 67 & 6b & 6f & 73 & 77 & 7b & 7f \\ 64 & 68 & 6c & 70 & 74 & 78 & 7c & 80 \end{bmatrix}$$

Cipherkey dan *plaintext* yang telah dimasukan kedalam *block* selanjutnya dapat dilakukan operasi-operasi enkripsi pada algoritma *advanced encryption standard,* 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.

$$State = \begin{bmatrix} 34 & 20 & 3d & 26 & 24 & 55 & 36 & 2e \\ 2c & 34 & 2b & 21 & 26 & 3f & 34 & 3f \\ 2a & 34 & 38 & 22 & 36 & 39 & 3e & 5f \\ 32 & 21 & 4c & 20 & 26 & 3c & 35 & a1 \end{bmatrix}$$

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 2.2 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*.

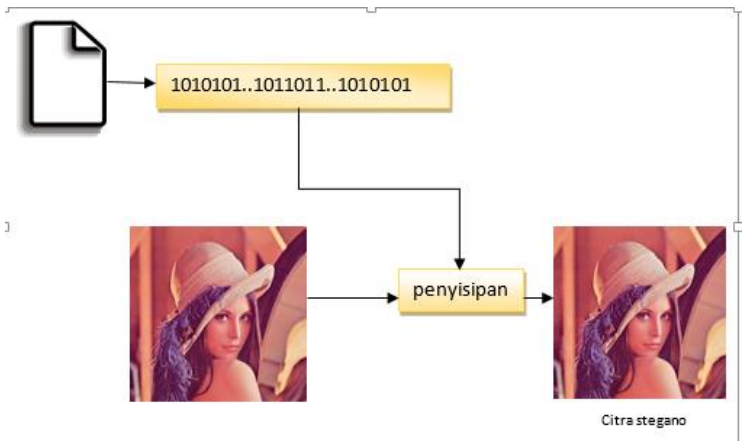
Proses dekripsi menggunakan algoritma *Advanced Encryption Standard (AES)* merupakan kebalikan dari proses enkripsi. Operasi-operasi yang dilakukan yaitu *InvSubByte*, *InvShiftRow*, *InvMixColumn*, dan *AddRoundKey*.

3.1.2.2 Analisis Metode Random Pixel Positioning (RPP)

Metode *Random Pixel Positioning (RPP)* merupakan salah satu metode steganografi pada citra digital yang beroperasi merubah file gambar maupun file sisipan ke dalam bentuk bit dan menyisipkan bit-bit dari file sisipan tersebut ke dalam bit-bit pada file gambar. Konsep dari metode ini yaitu mengacak posisi bit-bit ke dalam bit-bit gambar sehingga tidak mudah di lacak. Dengan memanfaatkan perbedaan intensitas warna RGB pada *pixel*. Seperti halnya metode steganografi lainnya, pada metode ini terdapat dua proses yaitu proses penyisipan/*embedding* dan pengungkapan atau ekstraksi.

3.1.2.2.1 Analisis Prose 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 *Random Pixel Positioning (RPP)* terlihat pada gambar 3.2.



Gambar 3. 2 proses penyisipan

Dari gambar diatas dapat dijelaskan proses penyisipan file pesan dimulai dengan memilih gambar, proses penyisipan jika diketahui pesan yang akan disisipkan berupa file unikom.txt dengan ini yaitu “UNIVERSITAS KOMPUTER INDONESIA !”. tahapan awal yang akan dilakukan yaitu merubah bit pesan kedalam bentuk biner seperti pada gambar 3.3

UNIVERSITAS KOMPUTER INDONESIA	1010101101..11101010..010010111101

Gambar 3. 3 Perubahan Pesan Teks Ke Bit

Tahap selanjutnya yaitu pengungkapan nilai pixel dari citra yang akan disisipkan pesan. Jika diketahui citra yang digunakan yaitu lena.bmp dengan nilai pixel seperti pada gambar 3.4.



01010101 01001110 01001001 01010110 01000101
01010010 01010011 01001001 01010100 01000001
01010011 00100000 01001011 01001111 01001101
01010000 01010101 01010100 01000101 01010010
00100000 01001001 01001110 01000100 01001111
01001110 01000101 01010011 01001001 01000001
00100000 00100001

Gambar 3.4 Penyisipan Dari Citra Cover

Bit ciphertext tersisipkan per 1 (bit) di setiap bit gambar, disisipkan terbalik pada bit terakhir gambar, ditandai dengan cetak tebal, penyisipan dilakukan terbalik agar mempersulit proses pemecahan analis frekuensi pada gambar/citra stega

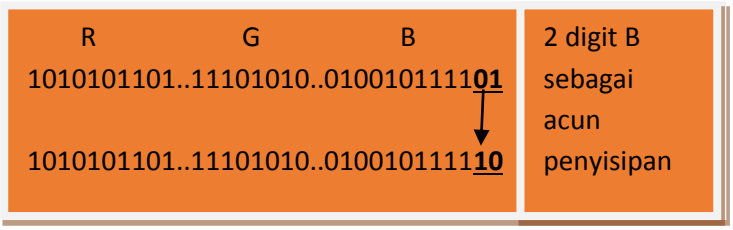
Bit bit rahasia tidak mengunakan menganti pixel yang berurutan, namun di pilih susunan pixel secara acak, misalnya jika terdapat 20 pixel dari 8 bit data yang akan disembunyikan, maka pixel yang diganti bit LSB-nya dipilih secara acak, misalnya pixel nomor 1,3,6,7,10,14 parameter yang digunakan sebagai acuan untuk menuju pixel berikutnya adalah 2 bit lsb dari komponen warna blue dari pixel yang sedang ditempati, pengunakan metode ini akan mengurangi daya tampung gambar, karna tidak semua pixel dipakai untuk menyembunyikan pesan prosedure penyembunyian data secara *random pixel* sebagai berikut :

a. Tentukan f(x,y). f(x,y) pixel dengan posisi koordinat x,y

b. Dapat nilai bit dari komponen warna blue(B) dari f(x,y)

c. Dapatkan 2 lsb dari B dengan n=B AND $(11)_2$

d. tambah nilai n x,f'(x,y)=f'(x+n,y)

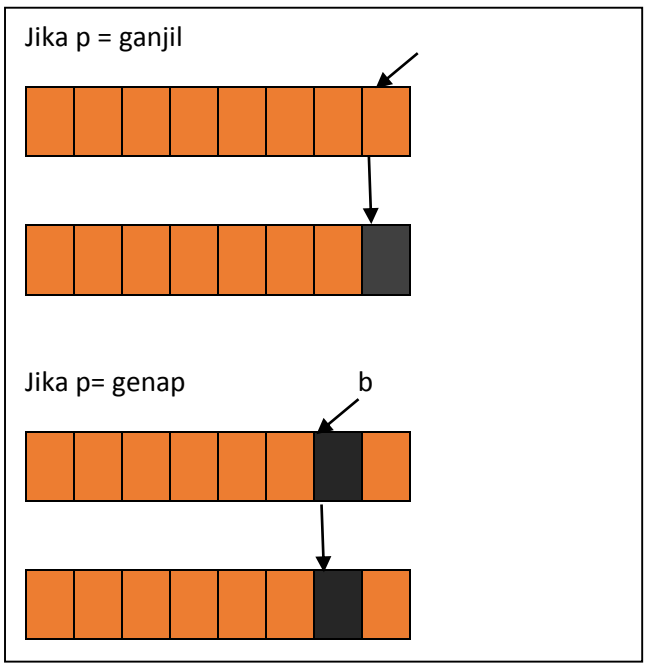


prosedur penyembunyian data mengunakan RPP yang berbeda(*random)* adalah sebagai berikut

a. Bangkitkan p (pseudo random number)

b. Lakukan proses penyembunyian dengan cara menyisipkan 1 bit data dengan aturan

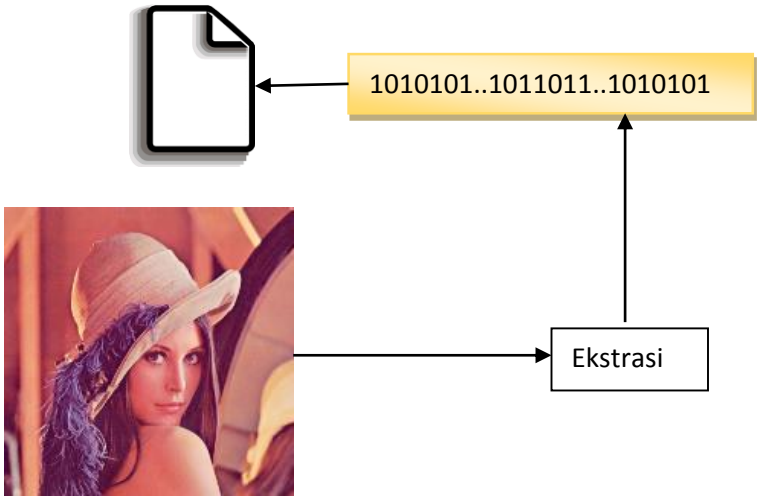
- Jika p adalah bilangan ganjil, sisipkan 1 bit data b pada RPP (bit pertama)
- Jika p adalah bilangan genap, sisipkan 1 bit data pada RPP (bit kedua)



3.1.2.2.2 Analisis Proses Pengungkapan / Ekstrasi

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 *random pixel positioning* terlihat pada gambar 3.5.



Gambar 3. 4 Proses Ekstraksi Pesan

3.1.3 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.

3.1.3.1 Analisis Pengguna

Pengguna yang akan menggunakan aplikasi steganografi ini dibagi menjadi dua bagian yaitu bagian pengirim dan penerima dengan kebutuhan spesifikasi seperti pada tabel 3.1.

Tabel 3. 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.

3.1.3.2 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 3.2.

Tabel 3. 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 3.3.

Tabel 3. 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

3.1.3.3 Analisis Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun aplikasi steganografi ini yaitu seperti pada tabel 3.4.

Tabel 3. 4 Spesifikasi Perangkat Lunak

No	Nama Perangkat	Spesifikasi
1	Sistem Operasi	Windows 7 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 3.5.

Tabel 3. 5 Kebutuhan Perangkat Lunak

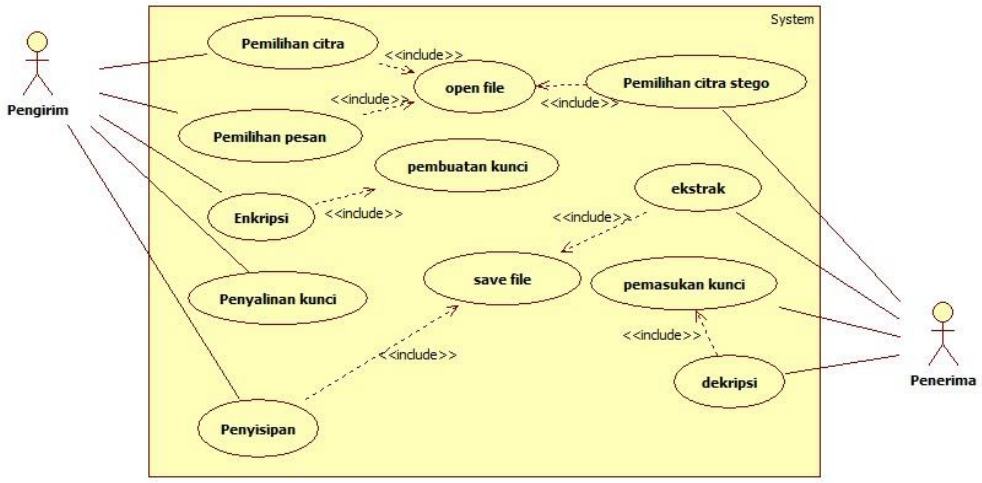
No	Nama Perangkat	Spesifikasi
1	Sistem Operasi	Windows xp
2	JDK (*Java Development Kit*)	JDK 1.7.0
3	JRE (*Java Runtime Environment*)	JRE 7

3.1.4 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.

3.1.4.1 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 3.6.



Gambar 3. 5 *Use Case* Diagram Aplikasi Steganografi

3.1.4.2 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 3.6.

Tabel 3. 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 3.7.

Tabel 3. 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 3.8.

Tabel 3. 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 3.9.

Tabel 3. 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 3.10.

Tabel 3. 10 Skenario Use Case Penyisipan

Use case name	Penyisipan acak
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 3.11.

Tabel 3. 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 3.12.

Tabel 3. 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 3.13.

Tabel 3. 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 3.15.

Tabel 3. 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.	

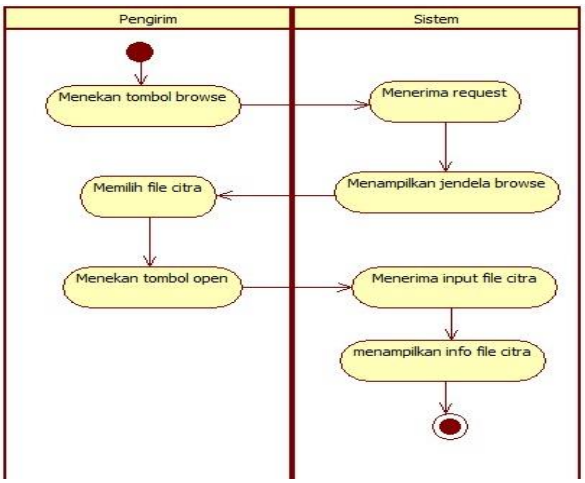
c. Penerima menekan tombol dekripsi.

3.1.4.3 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

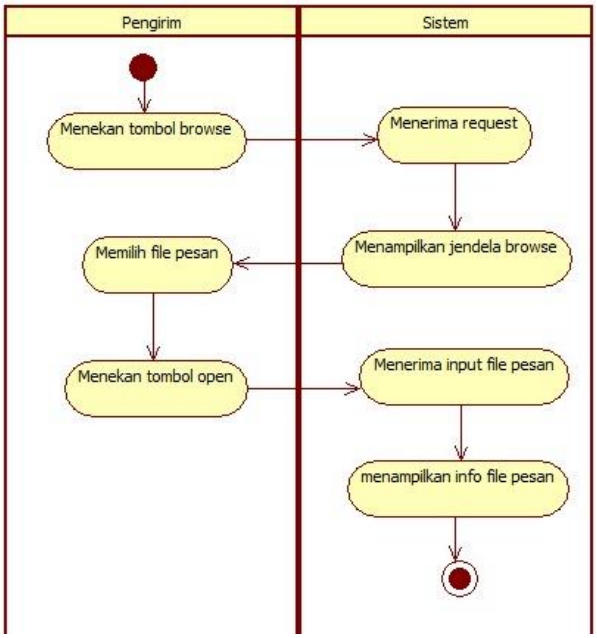
Activity diagram pemilihan citra menggambarkan alir aktivitas pengambilan *file* citra antara pengirim dangan sistem seperti terlihat pada gambar 3.5.



Gambar 3. 6 *Activity* Diagram Pemilihan Citra

***2.* *Activity* Diagram Pemilihan Pesan**

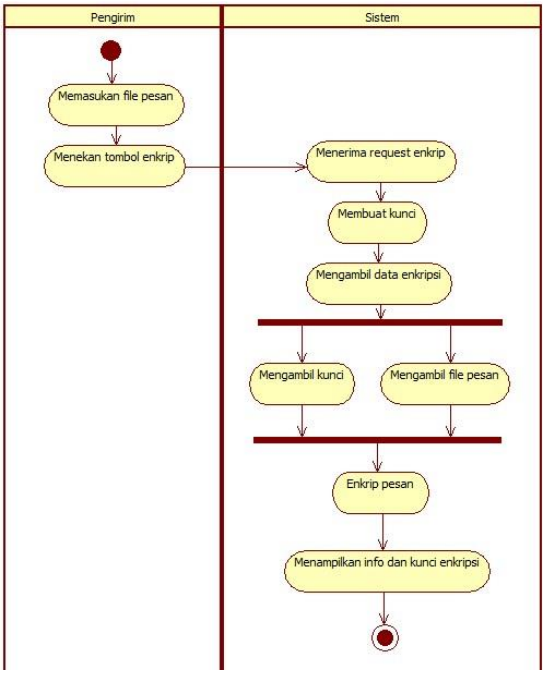
Activity diagram pemilihan pesan menggambarkan alir aktivitas pengambilan *file* pesan antara pengirim dangan sistem seperti terlihat pada gambar 3.8.



Gambar 3. 7 *Activity* Diagram Pemilihan Pesan

3. *Activity* Diagram Enkripsi

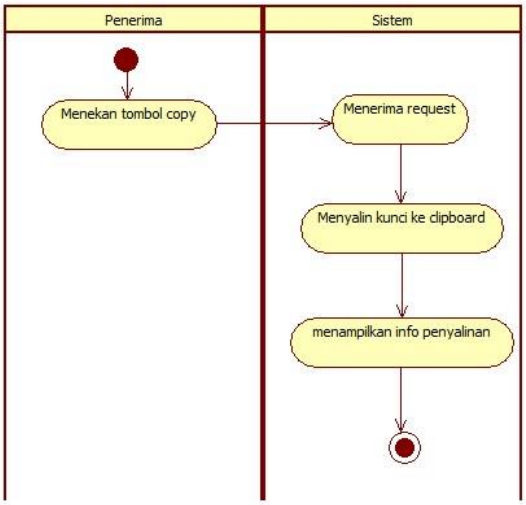
Activity diagram enkripsi menggambarkan alir aktivitas enkripsi yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar 3.7.



Gambar 3. 8 *Activity* Diagram Enkripsi

4. *Activity* Diagram Penyalinan Kunci

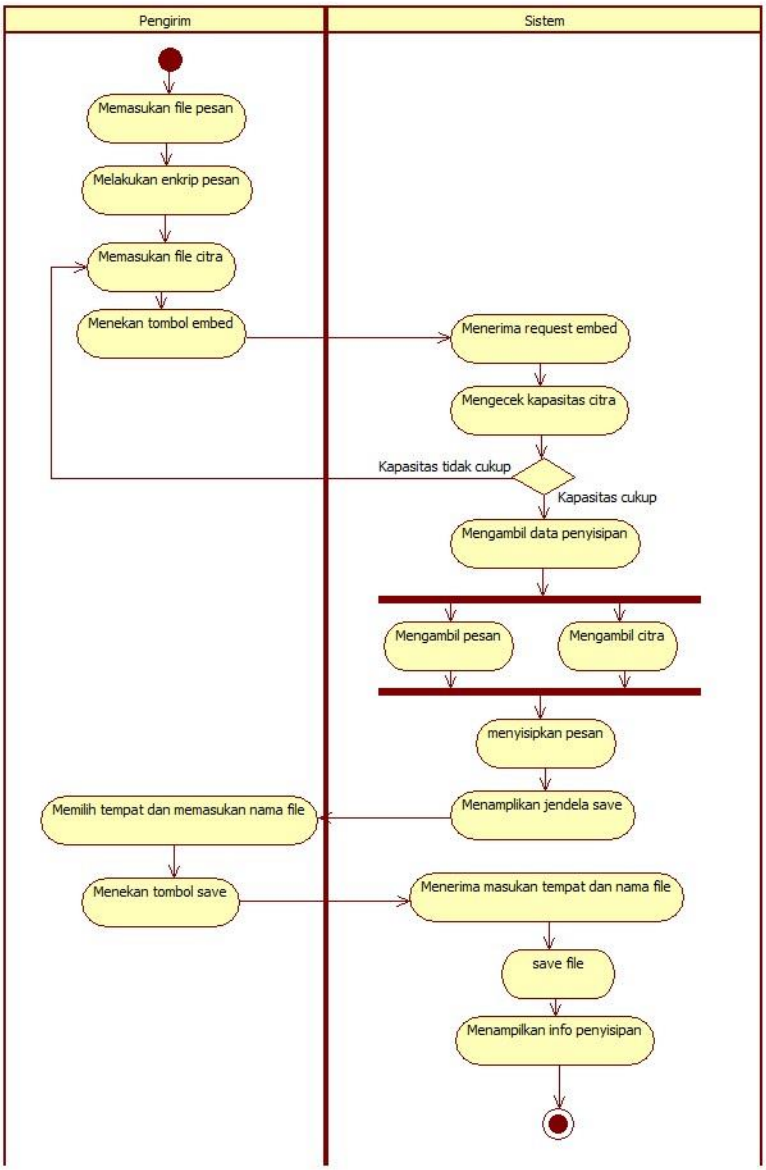
Activity diagram penyisipan menggambarkan alir aktivitas penyalinan kunci yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar 3.10.



Gambar 3. 9 *Activity* Diagram Penyalinan Kunci

5. *Activity* Diagram Penyisipan

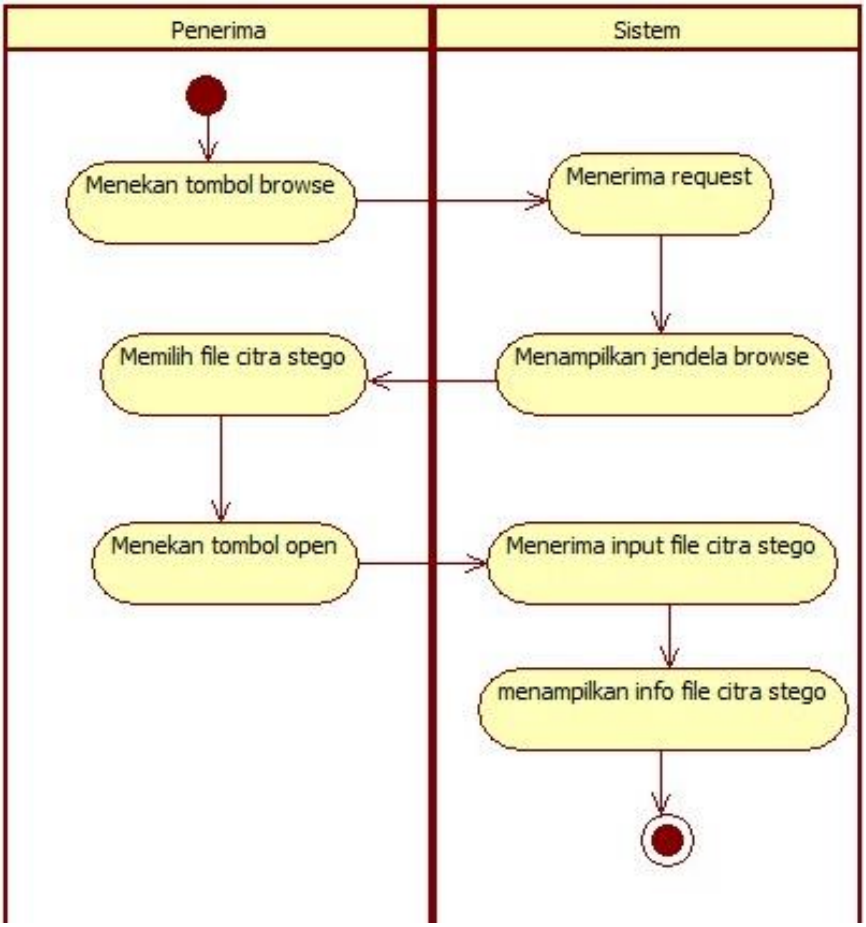
Activity diagram penyisipan menggambarkan alir aktivitas penyisipan yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar 3.11.



Gambar 3. 10 *Activity* Diagram Penyisipan

6. *Activity* Diagram Pemilihan Citra *Stego*

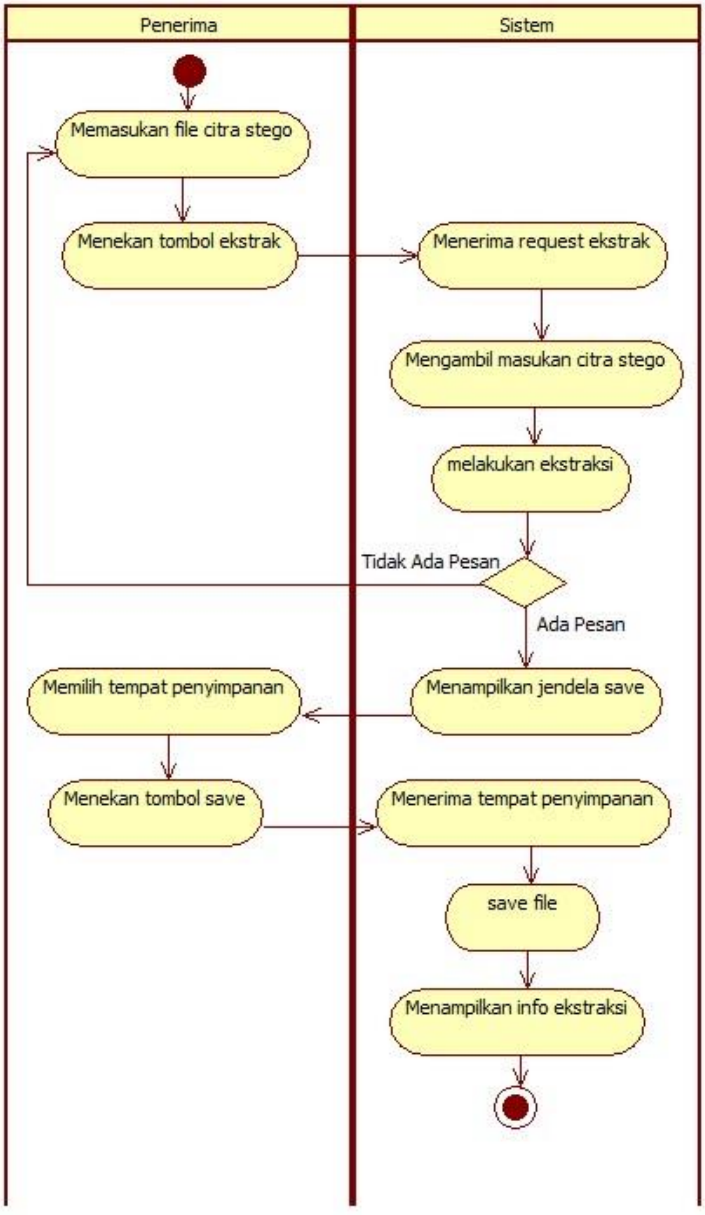
Activity diagram pemilihan citra *stego* menggambarkan alur aktivitas pengambilan *file* citra *stego* antara penerima dangan sistem seperti terlihat pada gambar 3.12.



Gambar 3. 11 *Activity* Diagram Pemilihan Citra *Stego*

7. *Activity* Diagram Ekstrak

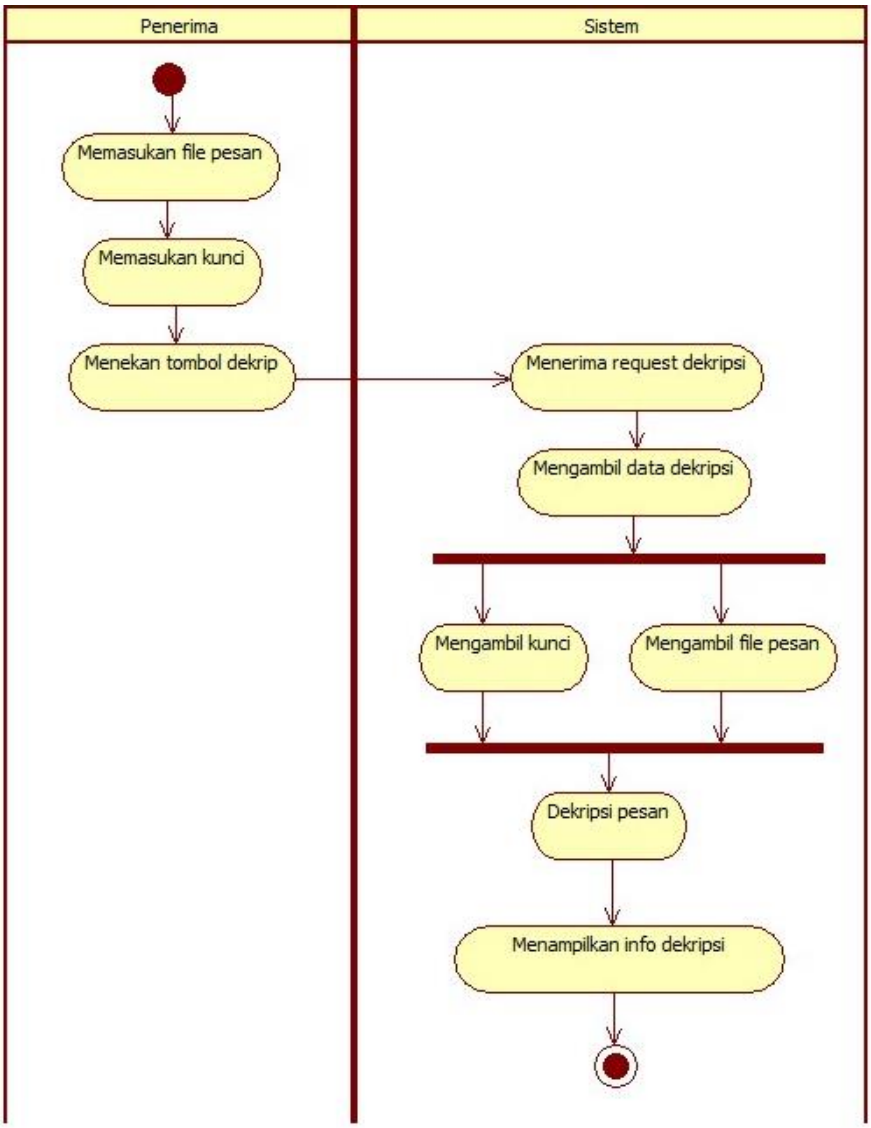
Activity diagram ekstrak menggambarkan alir aktivitas ekstrak yang dilakukan antara penerima dangan sistem seperti terlihat pada gambar 3.13.



Gambar 3. 12 *Activity* Diagram Ekstrak

8. *Activity* Diagram Dekripsi

Activity diagram dekripsi menggambarkan alur aktivitas dekripsi yang dilakukan antara penerima dangan sistem seperti terlihat pada gambar 3.14.



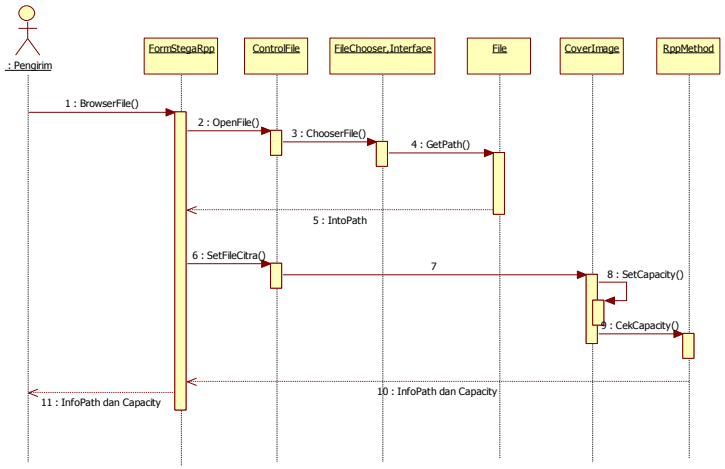
Gambar 3. 13 *Activity* Diagram Dekripsi

3.1.4.4 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

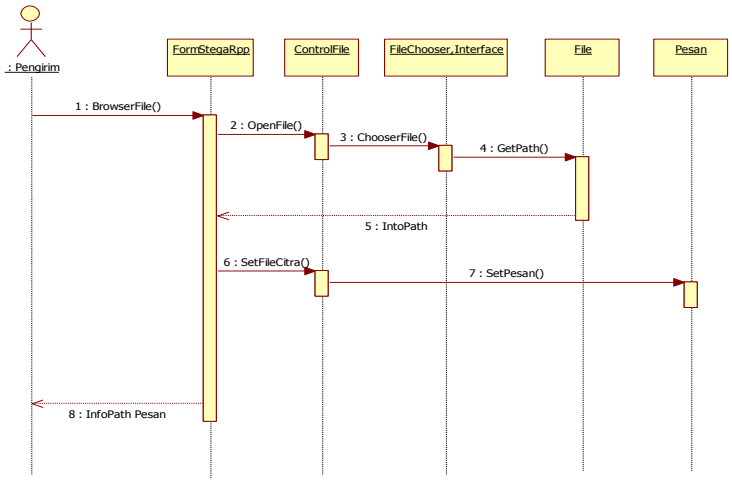
Sequence diagram pemilihan citra merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam pengambilan *file* citra. *Sequence* diagram pemilihan citra aplikasi steganografi terlihat seperti pada gambar 3.13.



Gambar 3. 14 Sequence Diagram Pemilihan Citra

2. *Sequence* Diagram Pemilihan Pesan

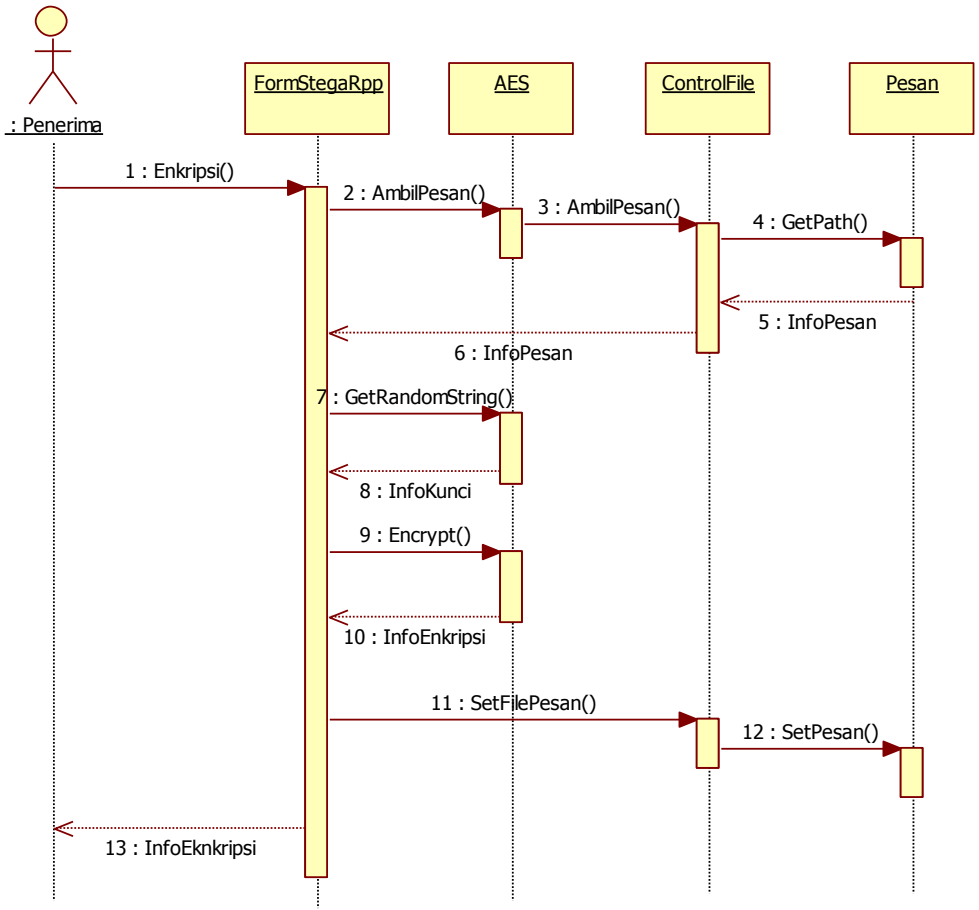
Sequence diagram pemilihan pesan merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam pengambilan *file* pesan. *Sequence* diagram pemilihan pesan aplikasi steganografi terlihat seperti pada gambar 3.14.



Gambar 3. 15 *Sequence* Diagram Pemilihan Pesan

3. *Sequence* Diagram Enkripsi

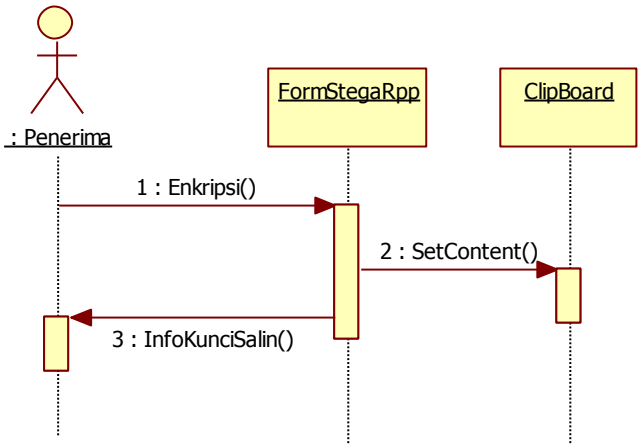
Sequence diagram enkripsi merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam mengenkripsi pesan. *Sequence* diagram enkripsi aplikasi steganografi terlihat seperti pada gambar 3.17.



Gambar 3. 16 Sequence Diagram Enkripsi

4. *Sequence* Diagram Penyalinan Kunci

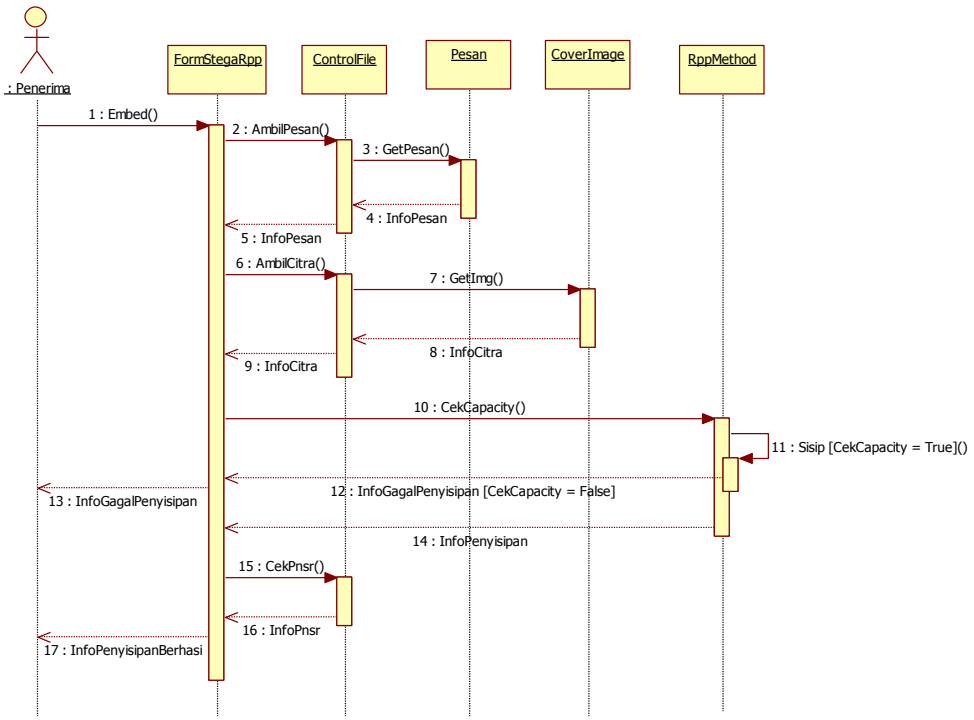
Sequence diagram penyalinan kunci merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam penyalinan kunci. *Sequence* diagram penyalinan kunci aplikasi steganografi terlihat seperti pada gambar 3.18.



Gambar 3. 17 *Sequence* Diagram Penyalinan Kunci

5. *Sequence* Diagram Penyisipan

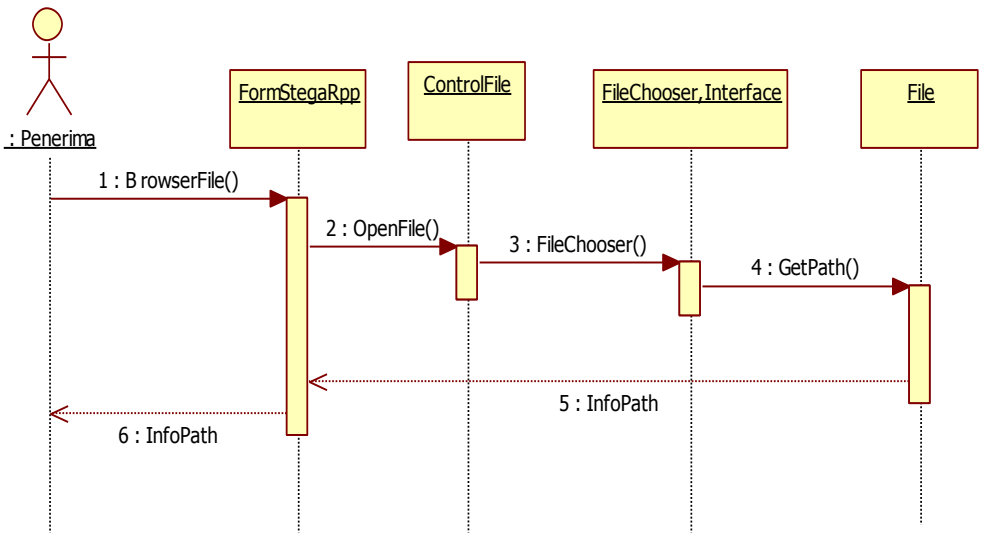
Sequence diagram penyisipan merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam penyisipan pesan kedalam citra. *Sequence* diagram penyisipan aplikasi steganografi terlihat seperti pada gambar 3.17.



Gambar 3. 18 *Sequence* Diagram Penyisipan

6. *Sequence* Diagram Pemilihan Citra *Stego*

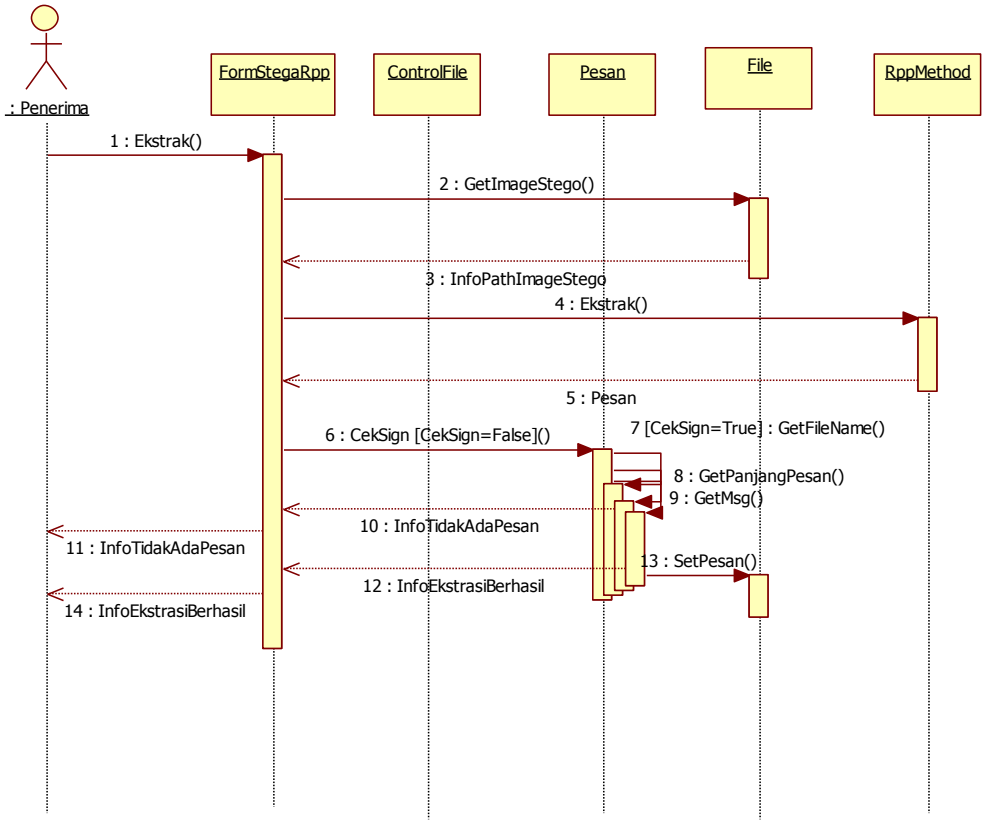
Sequence diagram pemilihan citra *stego* merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam pengambilan *file* citra *stego*. *Sequence* diagram pemilihan citra *stego* aplikasi steganografi terlihat seperti pada gambar 3.18.



Gambar 3. 19 *Sequence* Diagram Pemilihan Citra *Stego*

7. *Sequence* Diagram Ekstrak

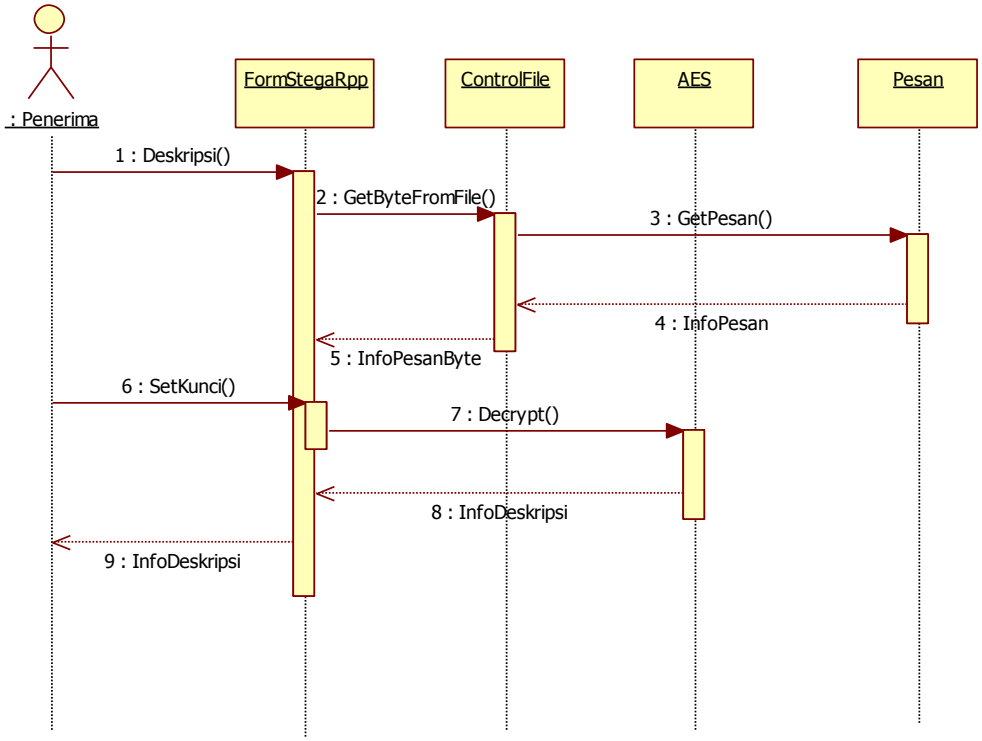
Sequence diagram ekstrak merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam ekstraksi pesan dari citra. *Sequence* diagram ekstrak aplikasi steganografi terlihat seperti pada gambar 3.19.



Gambar 3. 20 *Sequence* Diagram Ekstraksi

8. *Sequence* Diagram Dekripsi

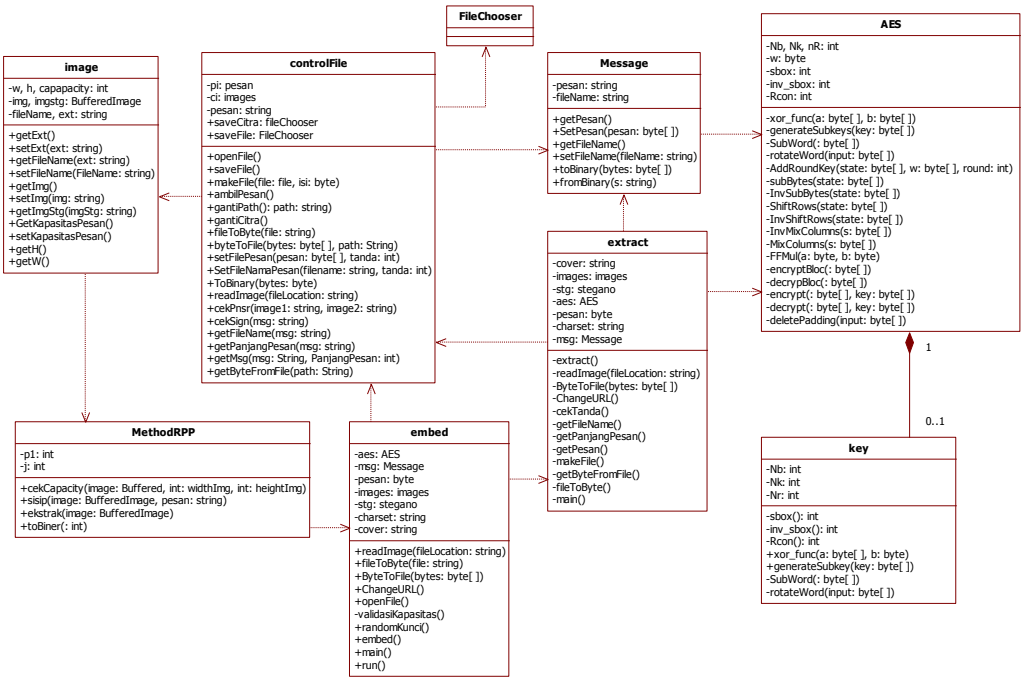
Sequence diagram dekripsi merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam dekripsi pesan. *Sequence* diagram dekripsi aplikasi steganografi terlihat seperti pada gambar 3.20.



Gambar 3. 21 *Sequence* Diagram Dekripsi

3.1.4.5 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 3.23.



Gambar 3. 22 *Class* Diagram Aplikasi Steganografi

3.2 Perancangan Sistem

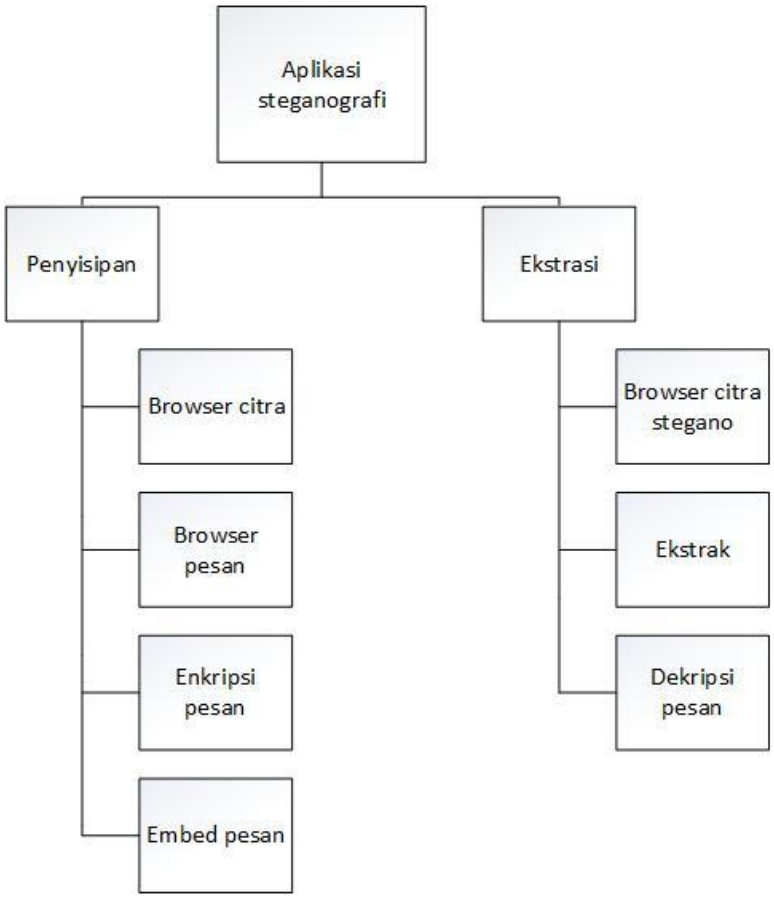
Perancangan sistem merupakan suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Tahap perancangan terdiri dari dua bagian yaitu perancangan arsitektur, dan peracangan *method*.

3.2.1 Perancangan Arsitektur

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

3.2.1.1 Perancangan Struktur Menu

Perancangan menu dilakukan untuk mempermudah interaksi antara sistem dengan pengguna. Gambar 3.22 merupakan struktur menu dari aplikasi steganografi yang dibangun.



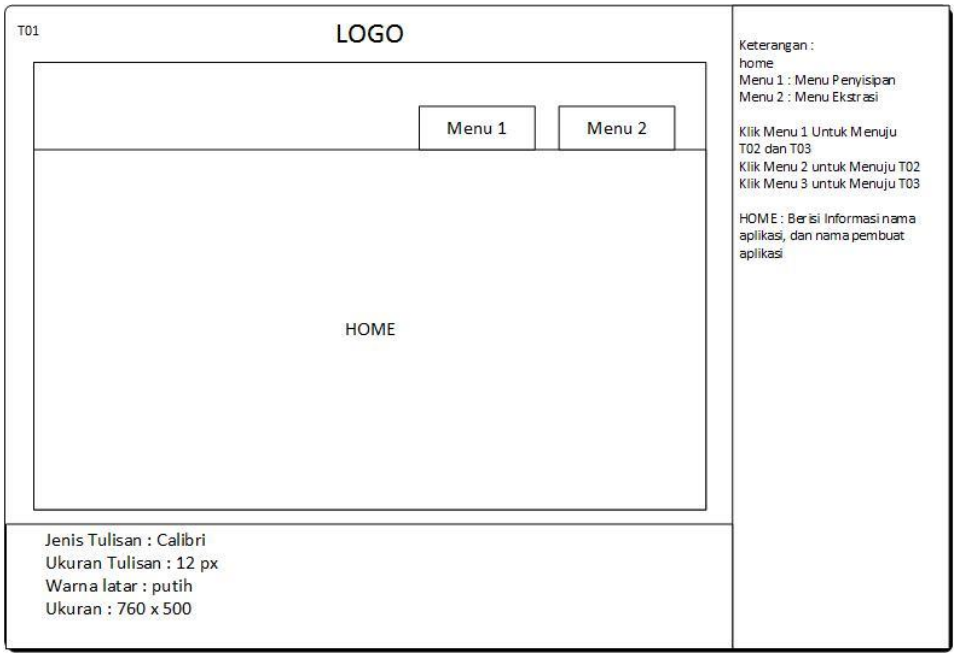
Gambar 3. 23 Struktur Menu Aplikasi Steganografi

3.2.1.2 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.

3.2.1.1.1 Perancangan Antarmuka Home (T01)

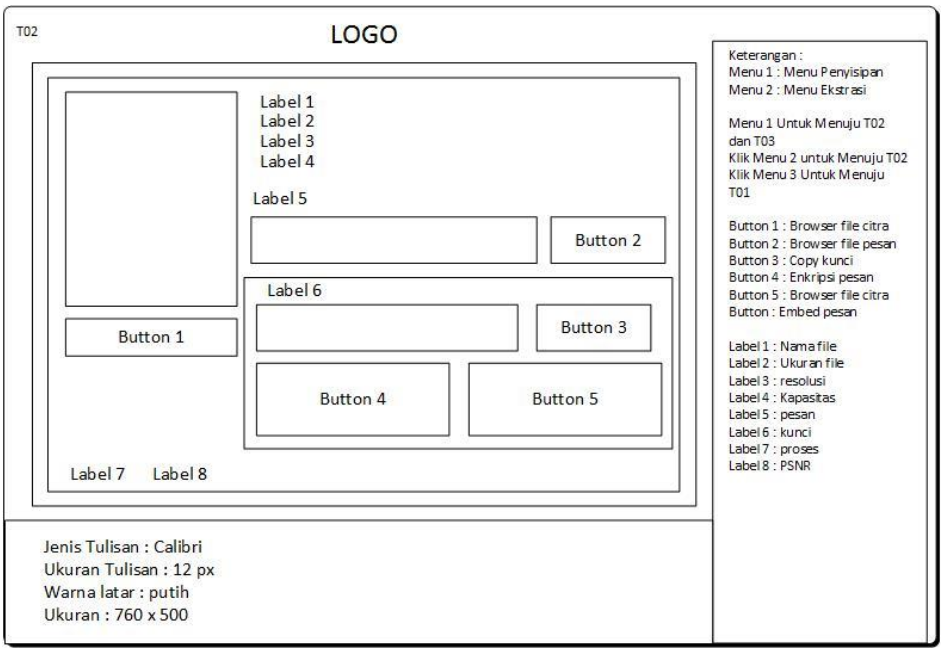
Perancangan antarmuka *home* merupakan tahapan rancangan antarmuka dari awal aplikasi ketika dijalankan. Perancangan antarmuka *home* dari aplikasi steganografi ini terlihat pada gambar 3.25.



Gambar 3. 24 Perancangan Antarmuka *Home*

3.2.1.1.2 Perancangan Antarmuka Penyisipan (T02)

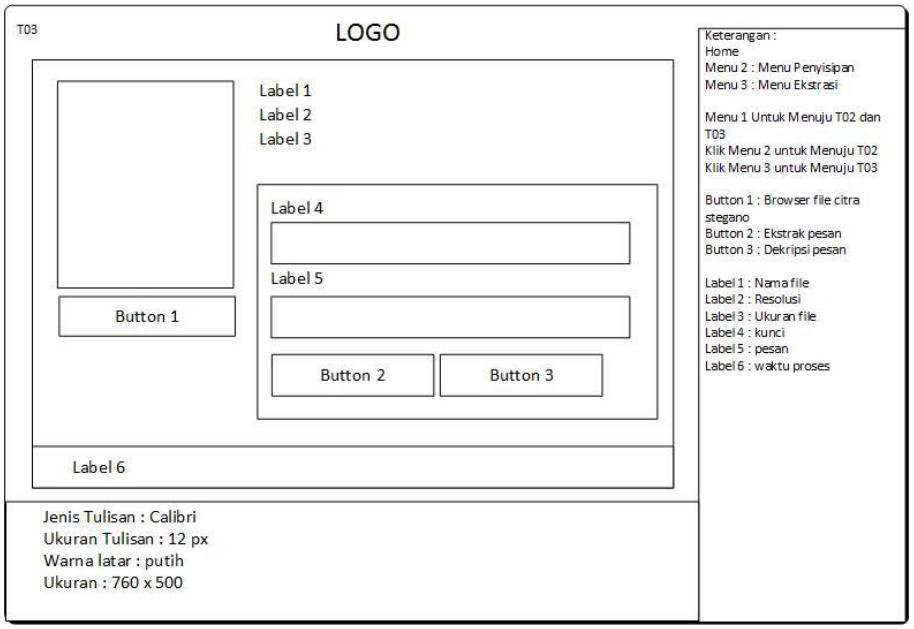
Perancangan antarmuka *home* merupakan tahapan rancangan antarmuka dari awal aplikasi ketika dijalankan. Perancangan antarmuka *home* dari aplikasi steganografi ini terlihat pada gambar 3.26.



Gambar 3. 25 Perancangan Antarmuka Penyisipan

3.2.1.1.3 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 3.27.



Gambar 3. 26 Perancangan Antarmuka Ekstraksi

3.2.1.1 Perancangan Pesan

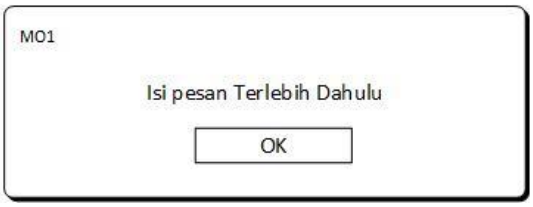
Pesan merupakan tampilan dari suatu perangkat lunak yang berfungsi untuk menyampaikan notifikasi dan informasi kepada pengguna agar perangkat lunak lebih interaktif. Perancangan pesan pada aplikasi steganografi dibagi menjadi dua bagian yaitu perancangan pesan menu penyisipan dan perancangan pesan menu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

3.2.1.1.1 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)

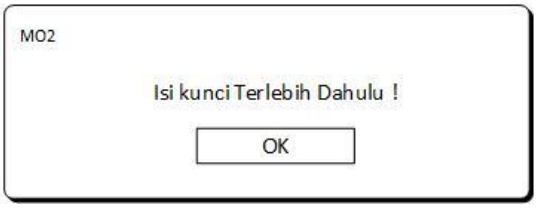
Pesan ini muncul jika pengguna menekan tombol enkripsi sebelum memilih citra. Perancangan pesan ini terlihat pada gambar 3.28.



Gambar 3. 27 Perancangan Pesan Jika Belum Memilih Citra

2. Perancangan pesan jika pesan belum dipilih (M02)

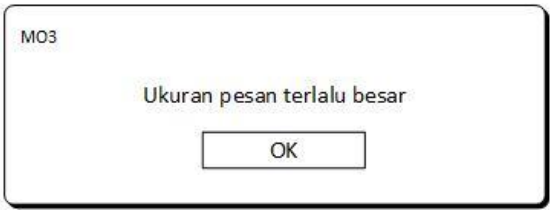
Pesan ini muncul jika pengguna menekan tombol enkripsi atau enkripsi sebelum memilih pesan. Perancangan pesan ini terlihat pada gambar 3.29.



Gambar 3. 28 Perancangan Pesan Jika Belum Memilih Pesan

3. Perancangan pesan jika kapasitas citra tidak mencukupi (M03)

Pesan ini muncul jika ukuran pesan terlalu besar untuk menyisipkan pesan pada saat proses penyisipan. Perancangan pesan ini terlihat pada gambar 3.32.



Gambar 3. 30 Perancangan Pesan Kapasitas Tidak Cukup

4. Perancangan pesan informasi penyisipan berhasil (M04)

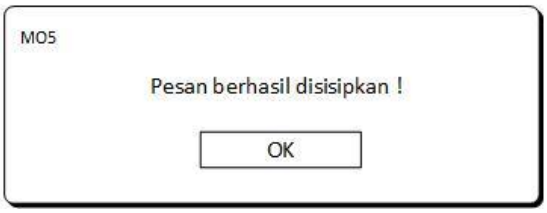
Pesan ini muncul jika penyisipan telah berhasil dilakukan. Informasi yang ditampilkan yaitu waktu proses penyisipan, dan nilai PSNR dari citra. Perancangan pesan ini terlihat pada gambar 3.33.



Gambar 3. 31 Perancangan Pesan Berhasil Dienkripsi

5. Perancangan pesan informasi enkripsi berhasil (M05)

Pesan ini muncul jika pesan berhasil disisipkan. Pada saat proses *embed*. Perancangan ini terlihat pada gambar 3.33



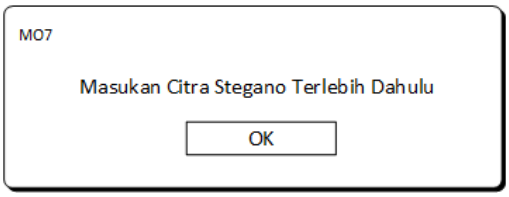
Gambar 3. 29 Perancangan Pesan Berhasil Disisipkan

3.2.1.1.2 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 (M06)

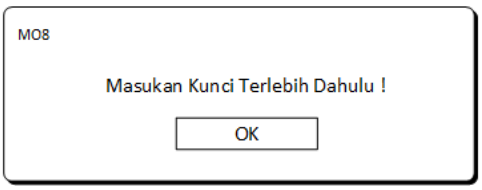
Pesan ini muncul jika pengguna menekan tombol ekstrak sebelum memilih citra *stego*. Perancangan pesan ini terlihat pada gambar 3.34.



Gambar 3. 30 Perancangan Pesan Jika Belum Memilih Citra *Stego*

2. Perancangan pesan jika kunci belum dimasukan (M07)

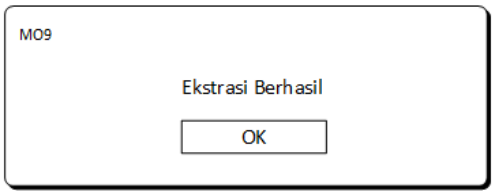
Pesan ini muncul jika pengguna menekan tombol dekripsi sebelum memasukan kunci. Perancangan pesan ini terlihat pada gambar 3.35.



Gambar 3. 31 Perancangan Pesan Jika Belum Memasukan Kunci

3. Perancangan pesan ekstraksi berhasil (M08)

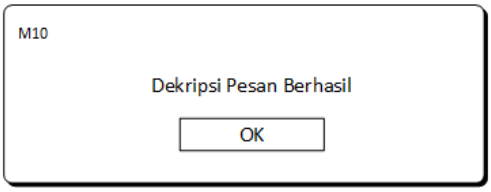
Pesan ini muncul jika ekstraksi pesan berhasil dilakukan. Perancangan pesan ini terlihat pada gambar 3.36.



Gambar 3. 32 Perancangan Pesan Ekstraksi Berhasil

4. Perancangan pesan dekripsi berhasil (M09)

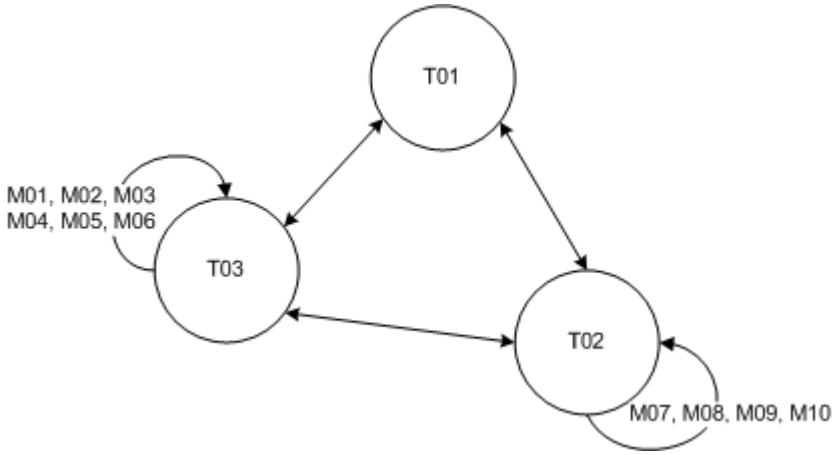
Pesan ini muncul jika dekripsi pesan berhasil dilakukan. Perancangan pesan ini terlihat pada gambar 3.37.



Gambar 3. 33 Perancangan Pesan Dekripsi Berhasil

3.2.1.2 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 3.38.



Gambar 3. 34 Jaringan Semantik Aplikasi Steganografi

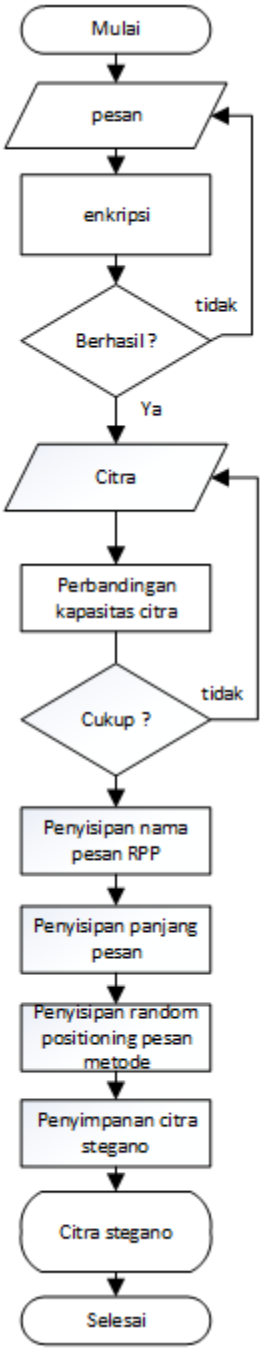
3.2.2 Perancangan Method

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.

3.2.2.1 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 3.39.

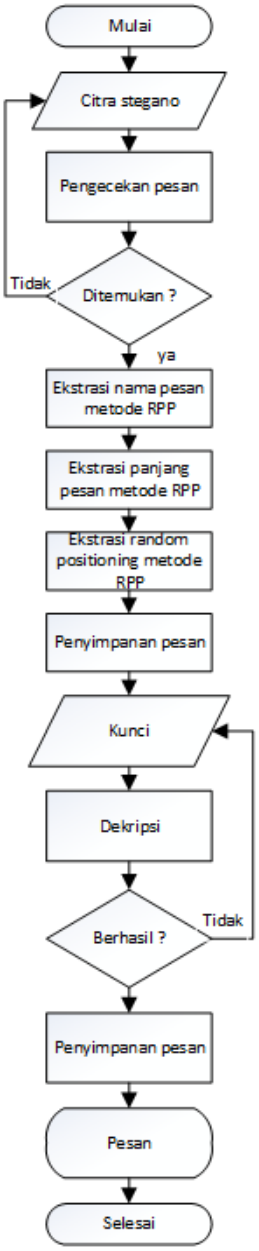


Gambar 3. 35 *Flowchart Method* Penyisipan Aplikasi Steganografi

3.2.2.2 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 3.40.



Gambar 3. 36 *Flowchart Method* Ekstraksi Aplikasi Steganografi