BAB III

ANALISIS DAN DESAIN SISTEM

III.1 Analisis Masalah

Kemajuan cara berpikir manusia membuat masyarakat menyadari bahwa teknologi informasi merupakan salah satu alat bantu penting dalam peradaban manusia untuk mengatasi sebagian masalah derasnya arus informasi. Agar informasi tidak jatuh ketangan orang yang tidak diinginkan, maka dari itu perlu adanya pengamanan informasi. Salah satu cara yang tepat untuk mengamankan informasi adalah dengan steganografi dan kriptografi. Proses penyisipan informasi pada steganografi membutuhkan dua buah masukan , yaitu pesan/informasi yang ingin disembunyikan, dan media pennyisipan yaitu gambar. Gambar yang dimaksudkan disini adalah sebagai media penampung dari pesan rahasia yang akan kita sisipkan/sembunyikan. Steganografi merupakan suatu teknik berkomunikasi dimana informasi disembunyikan pada media pembawa seperti citra, suara, atau video tanpa memberikan perubahan yang berarti pada media tersebut. Jadi untuk mengimplementasikan hal tersebut diperlukan perancangan perangkat lunak.

Perangkat lunak yang dirancang disini menggunakan teknik kriptografi dan steganografi dengan metode *Vigenere Cipher* dan LSB (*Least significant Bit*) yang membutuhkan *file* gambar berformat JPEG (*Joint Photographic Expert Group*) atau PNG (*Portable Network Graphic*) digunakan sebagai wadah penampung pesan yang akan dirahasiakan dan pesan yang akan dirahasiakan

kedalam *file* gambar berupa teks. Proses modifikasi perubahan yang terjadi antara media penampung dengan hasil modifikasi media penampung tersebut secara kasat mata tidak ada perubahan yang mencolok.

III.2 Analisis *Vigenere Cipher*

Sandi *vigenere* merupakan sistem sandi poli-alfabetik yang sederhana. Sistem sandi poli-alfabetik mengenkripsi sekaligus sebuah teks yang terdiri dari beberapa huruf. Sandi *vigenere* menggunakan teknik subtitusi dengan fungsi *shift* seperti pada sandi *Caesar*. Sandi *Vigenere* terdiri dari 26 * 26 *matriks* dimana abjad yang ditulis 26 kali dalam baris yang berbeda mewakili pergeseran yang berbeda pula. Tabel digunakan dan substitusi dibuat sesuai dengan pergeseran variasi nilai-nilai yang berasal dari kunci (Md. Khalid Imam Rahmani, et al, 2012).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
W	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar III.1 Tabel *Vigenere Cipher*

(Sumber : Md. Khalid Imam Rahmani, et al, 2012)

Jika huruf A – Z yang diambil untuk menjadi nomor 0 – 25 kemudian, enkripsi *vigenere* dilambangkan dengan E dan kunci dilambangkan dengan K maka dapat ditulis sebagai berikut:

Ci = EK (Pi) = (Pi + Ki) *mod* 26

Atau

Ci = (Pi + Ki) – 26

dan dekripsi D menggunakan kunci K,

Pi = DK (Ci) = (Ci - Ki) *mod* 26

Atau

Pi = (Ci – Ki) + 26

dimana,

Ci = Nilai desimal karakter *ciphertext* ke-i

Pi = Nilai desimal karakter *plaintext* ke-i

Ki = Nilai desimal karakter kunci ke-i

Nilai desimal karakter : A=0, B=1, C=2, … Z=25

Sebagai contoh jika *plaintext* adalah UNIVERSITAS POTENSI UTAMA dan kunci adalah UTAMA maka proses enkripsi yang terjadi adalah sebagai berikut :

Plaintext : UNIVERSITAS POTENSI UTAMA

Kunci : UTAMAUTAMA UTAMAUT AMAUT

Ciphertext : OGIHELLIFAM IOFEHLI GTUFA

Pada contoh diatas kata kunci UTAMA diulang sedemikian rupa hingga panjang kunci sama dengan panjang *plaintext.* Jika dihitung dengan rumus enkripsi *vigenere, plaintext* huruf pertama U (yang memiliki nilai Pi = 20) akan dilakukan pergeseran dengan huruf U (yang memiliki nilai Ki = 20) maka prosesnya sebagai berikut :

Ci = (Pi + Ki) *mod* 26

= (20 + 20) *mod* 26

= 40 *mod* 26

= 14

Hasilnya Ci = 14, maka huruf *ciphertext* dengan nilai 14 adalah O. Begitu seterusnya dilakukan pergeseran sesuai kunci pada setiap huruf hingga semua *plaintext* telah terenkripsi menjadi *ciphertext*. Setelah semua huruf terenkripsi maka proses dekripsinya dapat dihitung sebagai berikut :

Pi = (Ci – Ki) + 26

= (14 – 20) + 26

= 20

Hasilnya Pi = 20, maka huruf *plaintext* dengan nilai 20 adalah U. Begitu seterusnya dilakukan pergeseran sesuai kunci pada setiap huruf hingga semua *ciphertext* telah terdekripsi menjadi *plaintext*.

Namun saat ini muncul metode baru yaitu metode *Kerckhoffs*, mencocokkan setiap kolom frekuensi pesan ke frekuensi *plaintext* yang bergeser untuk menemukan huruf kunci untuk kolom itu. Setelah setiap huruf kunci diketahui, kriptanalis dengan mudah dapat mendekripsi *ciphertext* dan

mengungkap *plaintext*. Tetapi metode *Kerckhoffs* tidak berlaku ketika tabel *Vigenere* telah diubah. Oleh karena itu disini penulis merancang 2 model tabel yang telah diubah, diantaranya :

1. *Vigenere Chiper Alpha Extended*, merupakan tabel *vigenere cipher* berisi 94 karakter yang terdapat pada keyboard dan disusun berurutan dari A – Z, a – z, 0 – 9, dan diikuti dengan simbol-simbol yang terdapat pada *keyboard.*
2. *Vigenere Cipher Alpha-Qwerty Extended*, merupakan tabel *vigenere cipher* yang berisi 94 karakter yang terdapat pada *keyboard* dan disusun sesuai urutan *alphabet keyboard* atau yang sering disebut dengan *Qwerty*, yaitu dengan mengurutkan Q – M, q – m, 0 – 9, dan diikuti dengan simbol-simbol yang terdapat pada *keyboard.*

III.3 Analisis *Least Significant Bit*

Metode penyisipan LSB (*Least Significant Bit*) ini adalah menyisipi pesan dengan cara mengganti *bit* ke 8, 16 dan 24 pada representasi *biner file* gambar dengan representasi *biner* pesan rahasia yang akan disembunyikan. Dengan demikian pada setiap *pixel file* gambar terdapat 24 *bit* yang dapat disisipkan 3 *bit* pesan. Pada *file* gambar nantinya terdapat 3 *bit* yang dapat disisipi dalam 1 *pixel*. Hal ini dikarenakan dalam 1 *pixel* warna tersusun dari 3 komponen warna, yaitu *Red, Green,* dan *Blue* yang masing-masing disusun oleh 8 digit bilangan *biner* dari rentang nilai 0 sampai dengan 255 dalam desimal atau 00000000 sampai 11111111 dalam representasi *biner* (Tri Prasetyo Utomo, 2011).

Contohnya bilangan *biner* dari 255 adalah 11111111 (kadang-kadang diberi huruf b pada akhir bilangan menjadi 1111 1111b). Bilangan tersebut dapat berarti $1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$. Dari barisan angka 1 tadi, angka 1 paling kanan bernilai 1, dan itu adalah yang paling kecil. Bagian tersebut disebut dengan *least significant bit* (*bit* yang paling tidak berarti), sedangkan bagian paling kiri bernilai 128 dan disebut dengan *most significant bit* (*bit* yang paling berarti).

III.4 Analisis *File* Gambar

Pada sistem yang akan dibangun ini penulis hanya menggunakan dua tipe *file* gambar yaitu JPEG (*Joint Photographic Expert Group*) dan PNG (*Portable Network Graphic*). Penulis memilih tipe *file* gambar tersebut dikarenakan kedua *file* gambar tersebeut memiliki karakteristik yang sama yaitu sama-sama memiliki 24 *bit* dalam setiap *pixel*nya. Yang membedakan kedua *file* gambar tersebut adalah dari sisi kompresi *file* tersebut. *File* gambar JPEG mampu mengkompres objek dengan tingkat kualitas sesuai dengan pilihan yang disediakan, maka dari itu ukuran *file* gambar JPEG cenderung lebih kecil dari pada *file* gambar PNG.

III.5 Analisis Langkah – Langkah Steganografi

Dalam proses penyisipan pesan pada steganografi dibutuhkan langkah-langkah sebagai berikut :

1. Langkah 1, baca *cover image* dan pesan rahasia yang akan disisipkan dalam *cover image*.
2. Langkah 2, konversi *cover image* setiap *pixel* nya dari representasi desimal menjadi *biner*.
3. Langkah 3, konversi pesan rahasia menjadi *biner*, jika pesan berupa *file* teks maka setiap karakter dapat diketahui kode ASCII nya berupa bilangan desimal untuk kemudian dikonversi menjadi *biner*.
4. Langkah 4, hitung LSB dari setiap *pixel cover image*.
5. Langkah 5, ganti LSB dari *cover image* dengan setiap *bit* dari pesan rahasia satu per satu.
6. Langkah 6, tulis *stego image* yang merupakan hasil akhir dari proses penyisipan, biasanya dengan cara menyimpan ulang *file* gambar yang telah disisipi pesan rahasia.

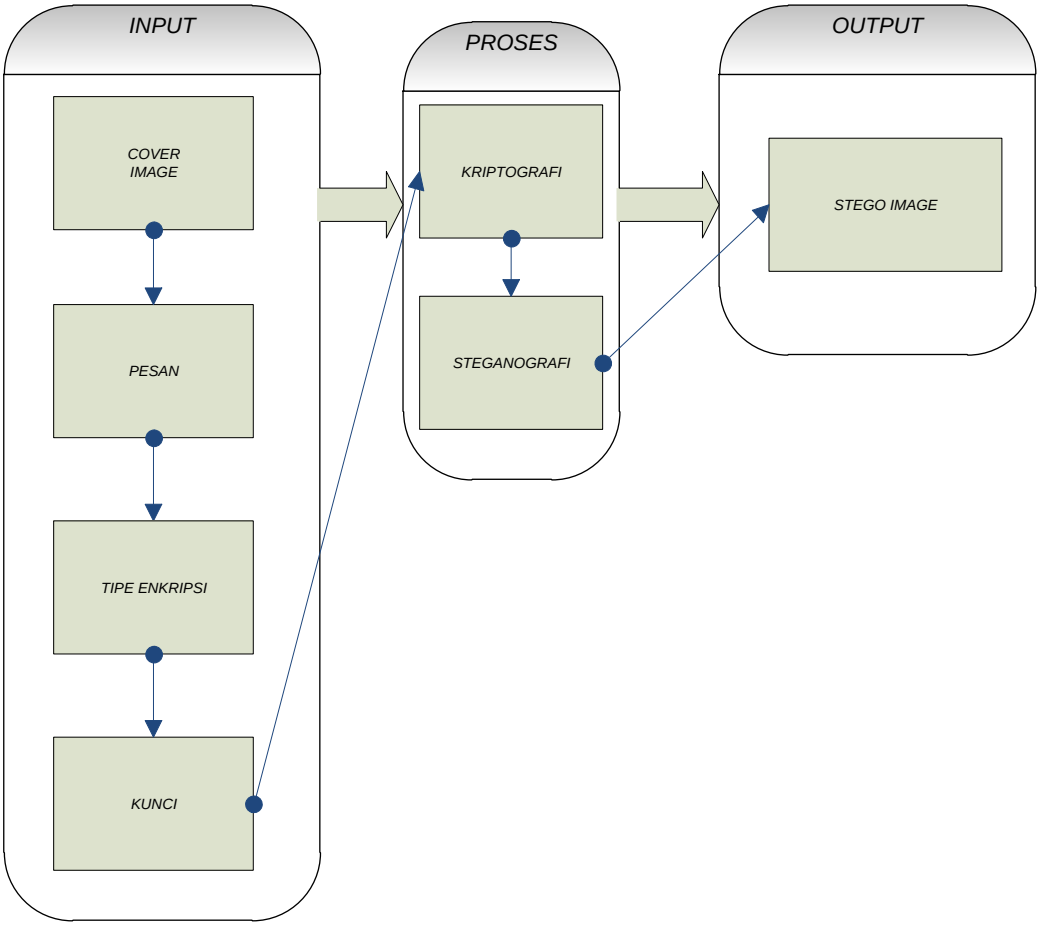
Sedangkan algoritma untuk membaca pesan rahasia langkah-langkahnya sebagai berikut:

1. Langkah 1, baca *stego image*.
2. Langkah 2, hitung LSB dari setiap *pixel* dalam *stego image*.
3. Langkah 3, ambil *bit-bit* yang diperoleh pada step 2 untuk selanjutnya dikonversi menjadi karakter.

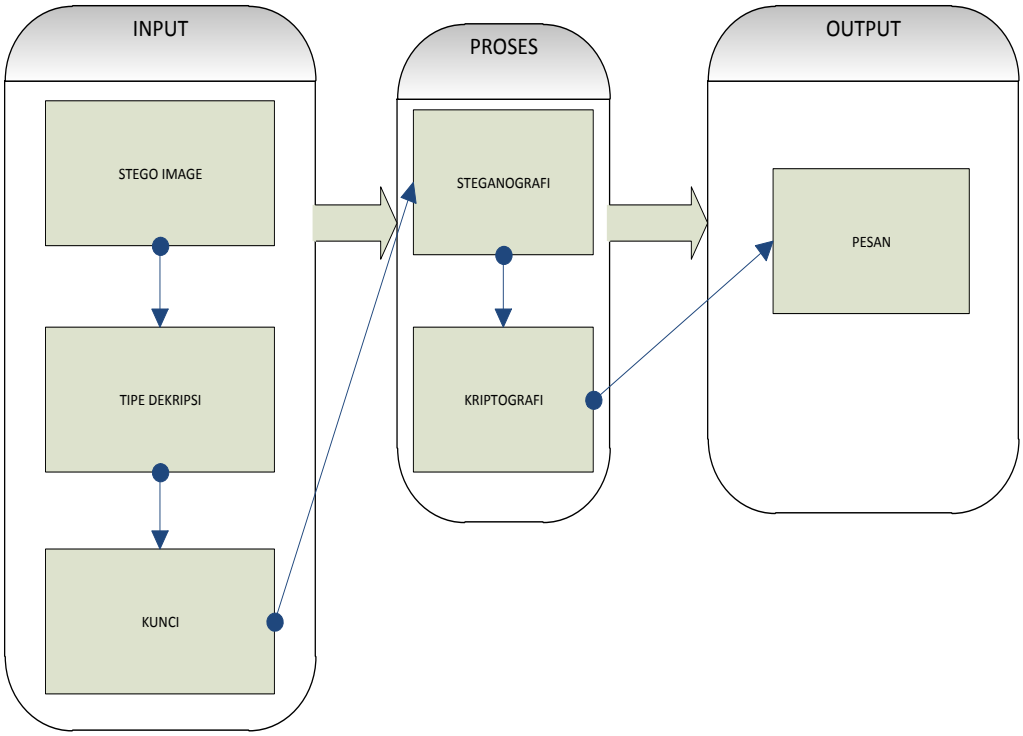
III.6 Desain Sistem

III.6.1 Blok Diagram

Blok diagram adalah diagram dari sebuah sistem , di mana bagian utama atau fungsi yang diwakili oleh blok dihubungkan dengan garis, yang menunjukkan hubungan dari blok. Blok diagram banyak digunakan dalam dunia rekayasa dalam desain *hardware* , desain elektronik , *software* desain , dan proses aliran diagram. Berikut adalah blok diagram dari perancangan perangkat lunak yang akan dibuat.



Gambar III.2 Blok Diagram Penyisipan Pesan

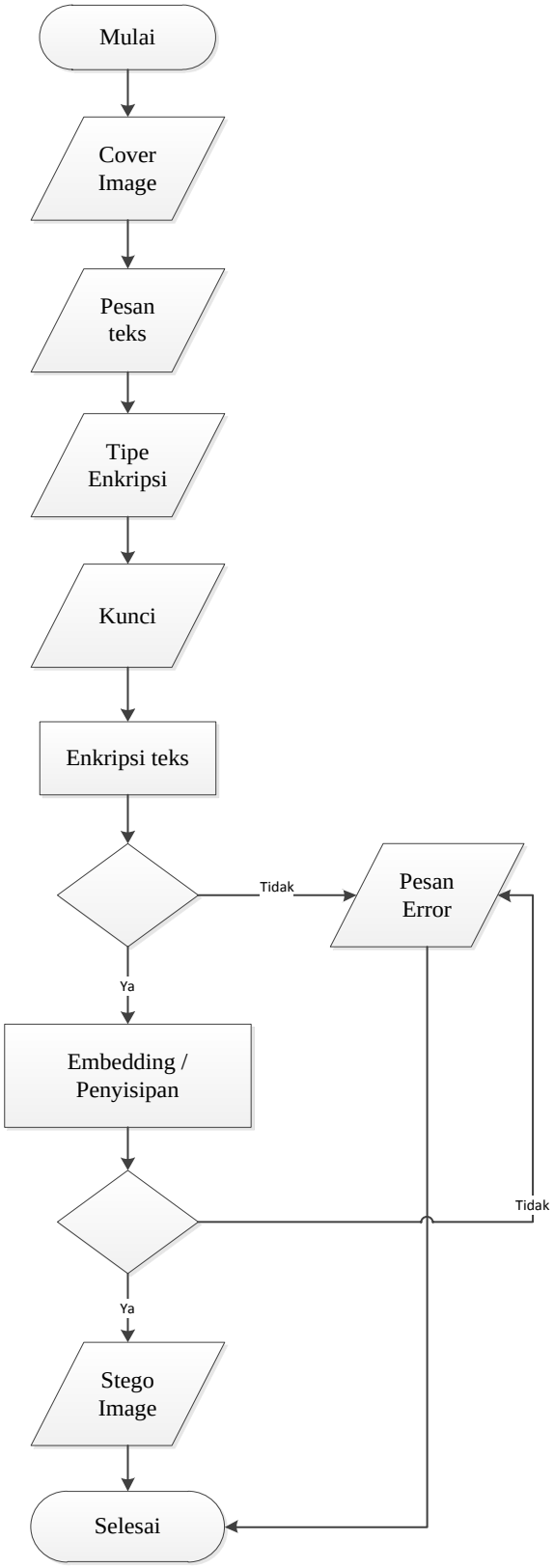


Gambar III.3 Blok Diagram Pennguraian Pesan

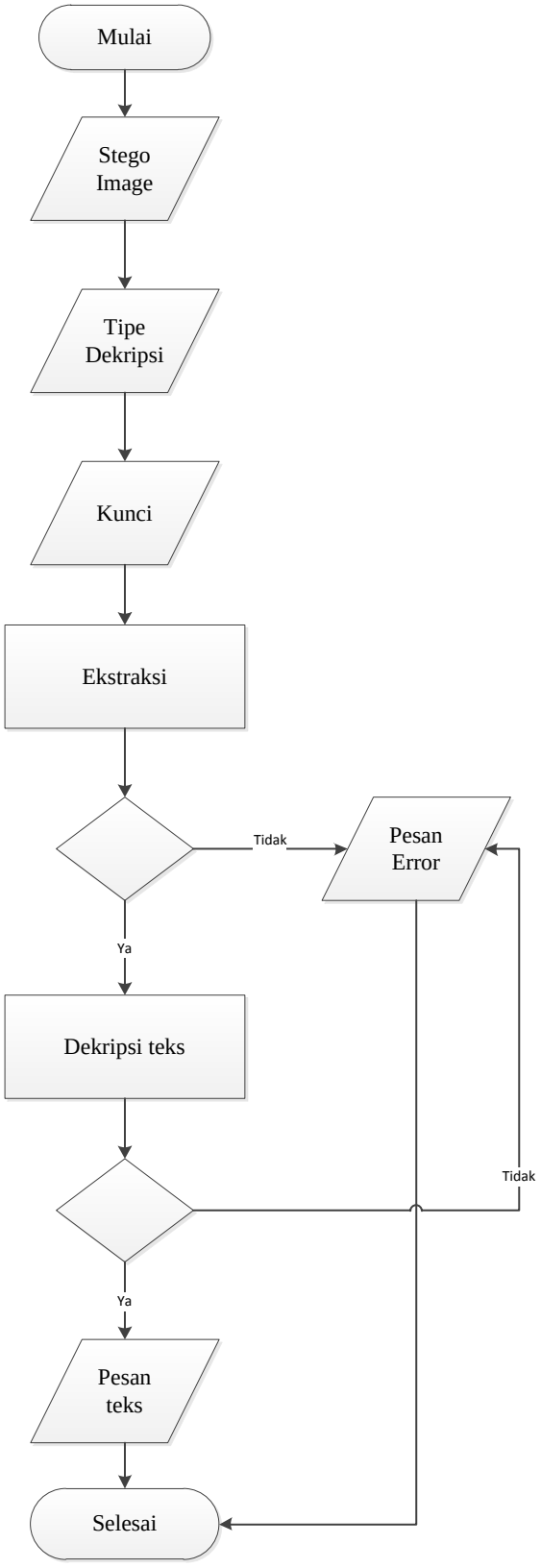
III.6.2 *Flowchart*

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Tujuan utama dari penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol yang standar. Dalam perancangan aplikasi ini digunakan bagan alir (*flowchart*) untuk menjelaskan proses kerja dari perangkat lunak yang dirancang.

Untuk melihat kedua proses penyisipan dan proses ekstraksi lebih jelasnya lagi bisa dilihat pada gambar III.3 dan III.4 dibawah ini.



Gambar III.4 Diagram Alir Proses Penyisipan

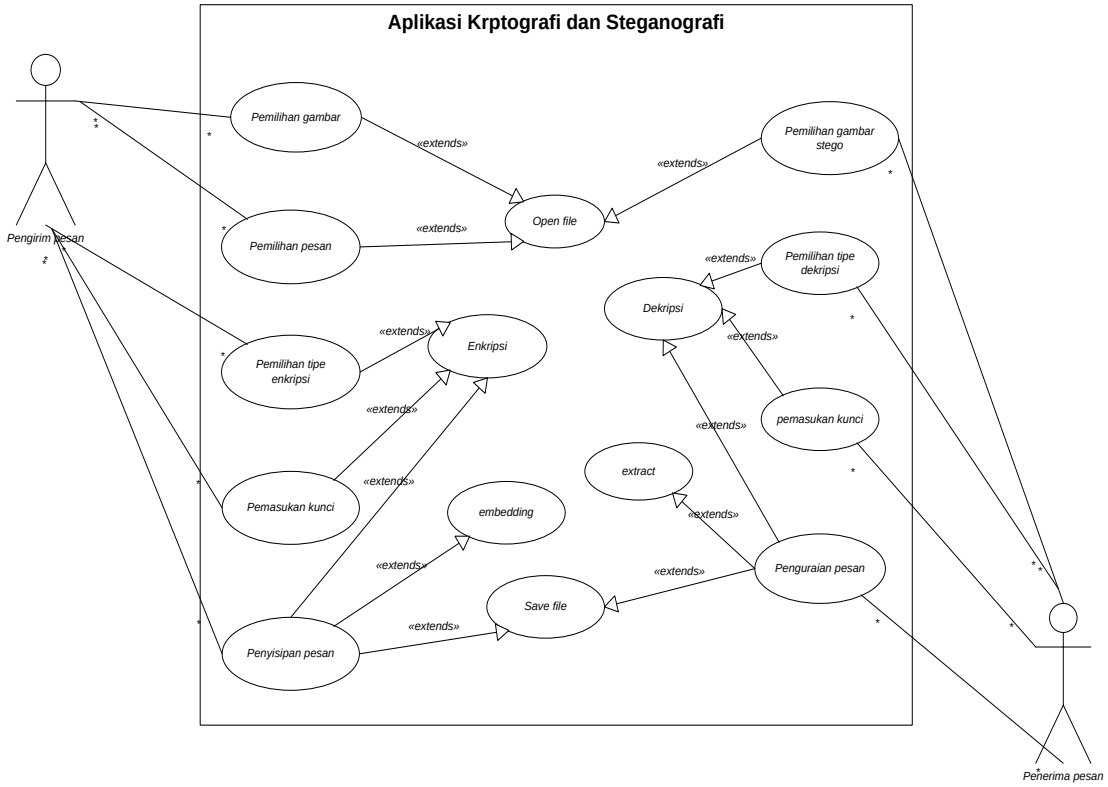


Gambar III.5 Diagram Alir Proses Ekstraksi

1. Algoritma Flowchart Penyisipan Pesan
    a. *Start*
    b. Membuka *Cover Image*
    c. Membuka pesan yang akan disisipkan
    d. Memilih tipe enkripsi atau tabel *vigenere cipher* yang dikehendaki
    e. Masukan kunci untuk proses enkripsi
    f. Melakukan proses enkripsi, bila gagal akan muncul pesan error dan proses selesai, jika berhasil akan lanjut ke proses steganografi.
    g. Pada proses steganografi pesan yang telah dienkripsi akan disisipkan ke gambar, bila gagal proses selesai, bila berhasil akan membentuk *stego image.*
    h. *Finish.*
2. Algoritma Flowchart Penguraian Pesan
    a. *Start*.
    b. Membuka *stego image*.
    c. Memilih tipe deskripsi atau tabel *vigenere cipher* yang sebelumnya digunakan pada proses enkrip.
    d. Memasukan kunci untuk proses deskripsi pesan.
    e. Melakukan proses ekstraksi jika ada kesalahan, maka akan muncul pesan *error* dan proses selesai, jika berhasil pesan akan muncul berserta *ciphertext*.
    f. Pesan dapat disimpan.
    g. *Finish.*

III.6.3 *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 kriptografi dan steganografi terlihat pada gambar III.6**.**



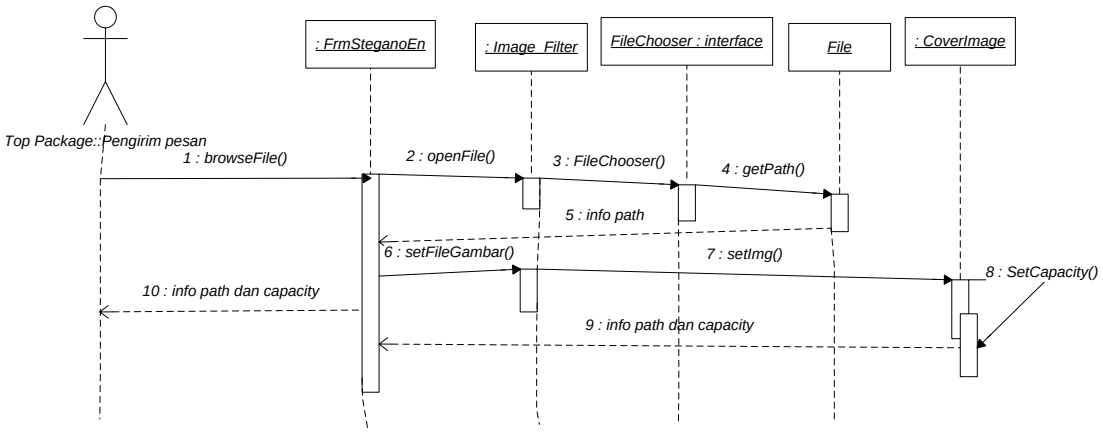
Gambar III.6 *Use Case Diagram* Aplikasi Kriptografi dan Steganografi

III.6.4 *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 kriptografi dan steganografi yaitu sebagai berikut:

1. *Sequence Diagram* Pemilihan Gambar

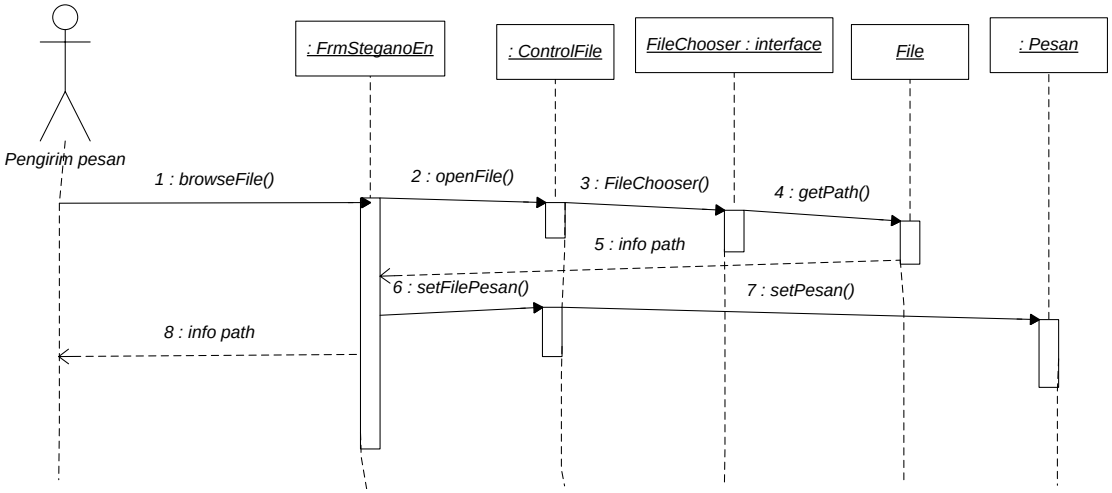
   Sequence diagram pemilihan gambar merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam pengambilan *file* gambar. *Sequence diagram* pemilihan gambar aplikasi kriptografi dan steganografi terlihat seperti pada gambar III.7.



Gambar III.7 *Sequence Diagram* Pemilihan Gambar

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 kriptografi dan steganografi terlihat seperti pada gambar III.8.

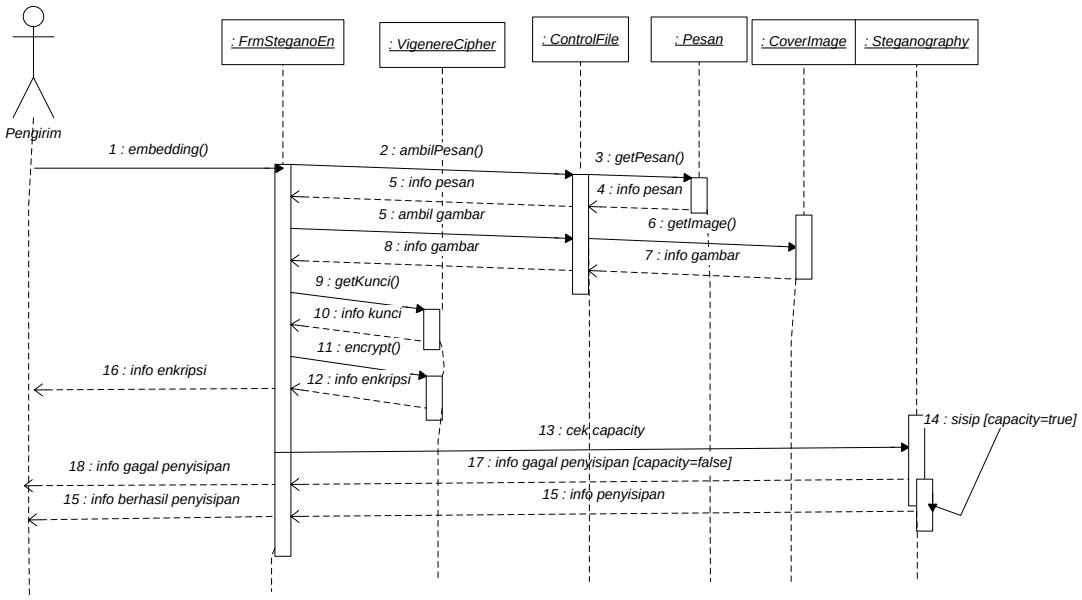


Gambar III.8 *Sequence Diagram* Pemilihan Pesan

3. *Sequence Diagram* Penyisipan

Sequence diagram penyisipan merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara pengirim dengan sistem dalam penyisipan pesan kedalam gambar. *Sequence diagram* penyisipan aplikasi kriptografi dan steganografi terlihat seperti pada gambar III.9

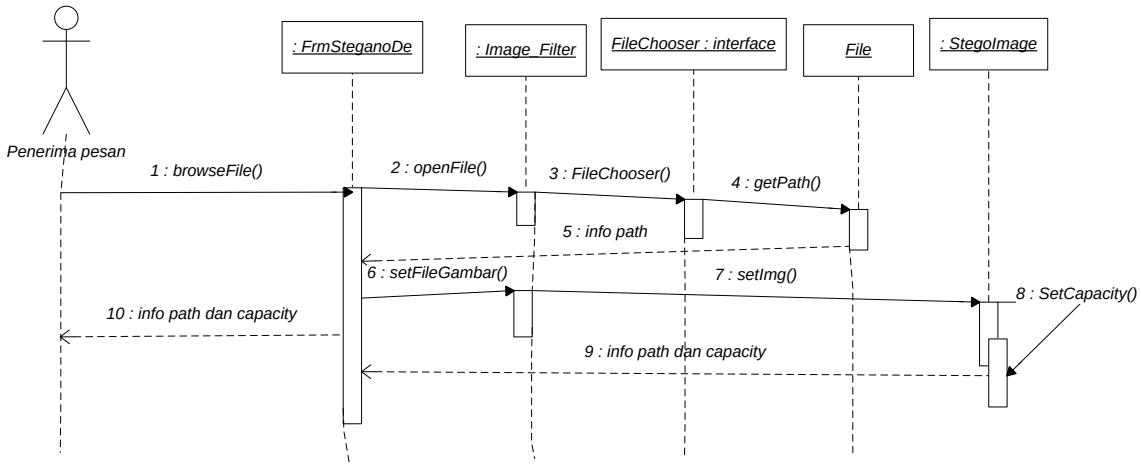
.



Gambar III.9 *Sequence Diagram* Penyisipan

4. *Sequence Diagram* Pemilihan Gambar *Stego*

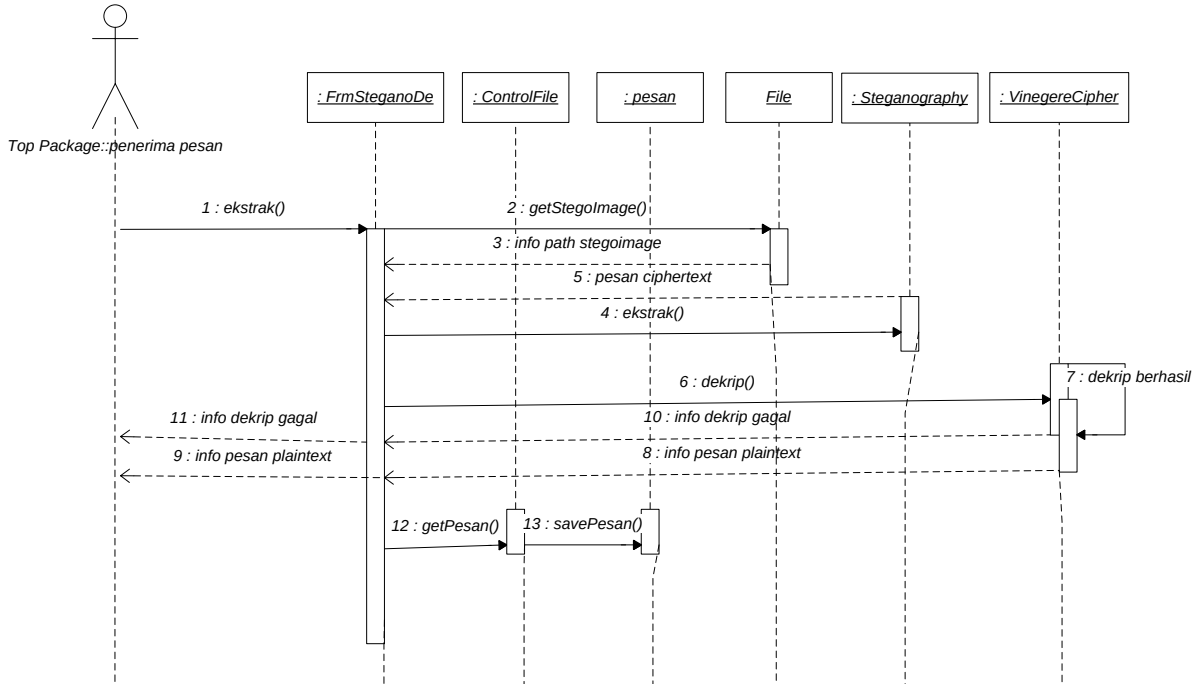
Sequence diagram pemilihan gambar *stego* merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam pengambilan *file* gambar *stego*. *Sequence diagram* pemilihan gambar *stego* aplikasi kriptografi dan steganografi terlihat seperti pada gambar III.10.



Gambar III.10 *Sequence Diagram* Pemilihan Gambar Stego

5. *Sequence Diagram* Ekstrak

Sequence diagram ekstrak merupakan diagram yang mengambarkan interaksi yang terjadi didalam sistem antara penerima dengan sistem dalam ekstraksi pesan dari gambar. *Sequence diagram* ekstrak aplikasi kriptografi dan steganografi terlihat seperti pada gambar III.11.



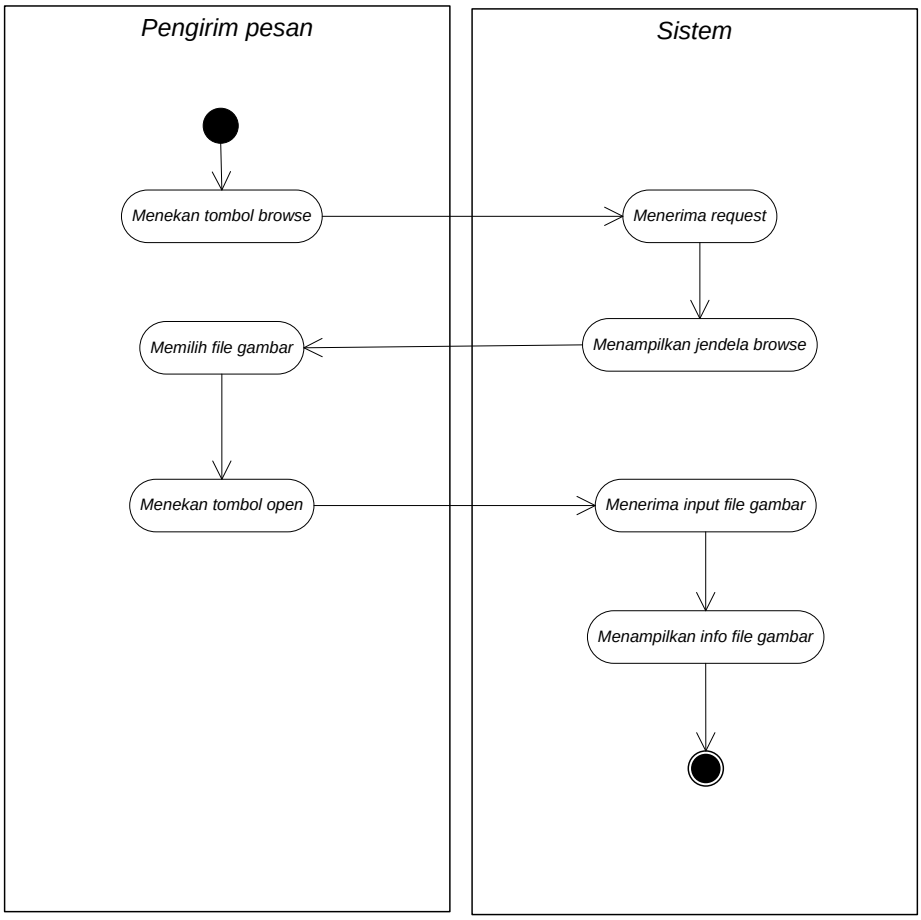
Gambar III.11 *Sequence Diagram* Ekstrak

III.6.5 *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 Gambar

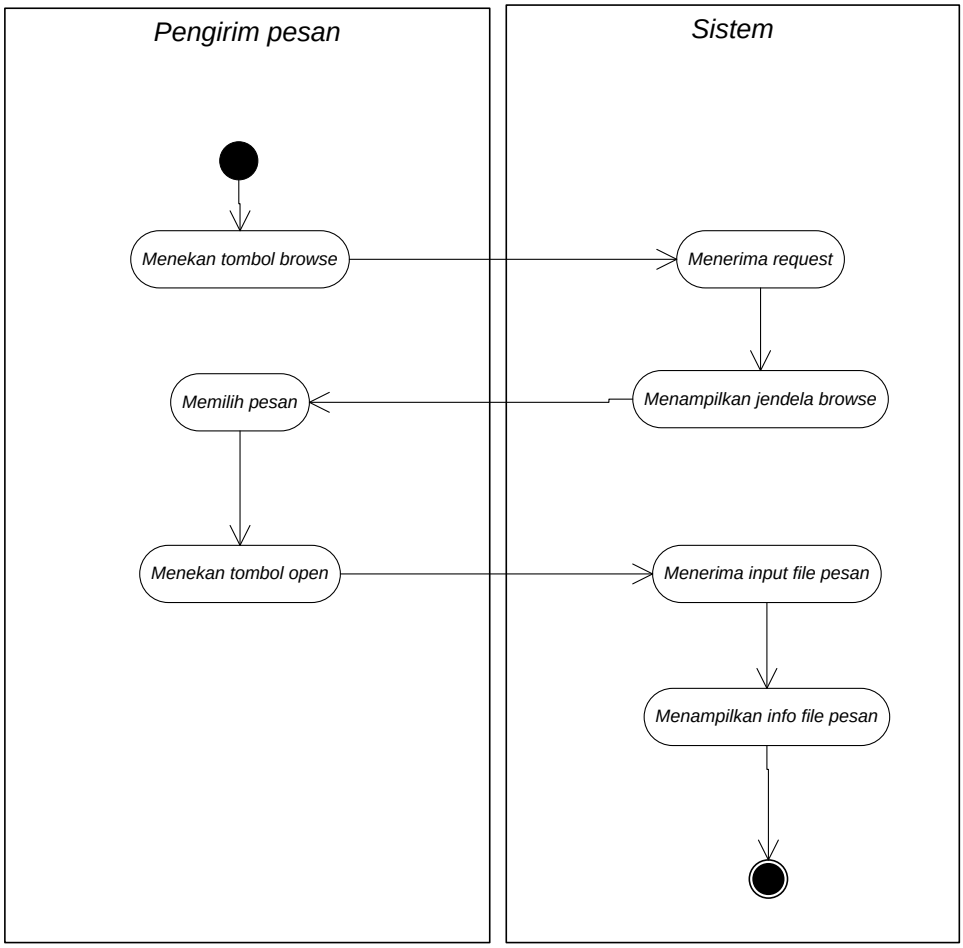
   Activity diagram pemilihan gambar menggambarkan alir aktivitas pengambilan *file* gambar antara pengirim dangan sistem seperti terlihat pada gambar III.12.



Gambar III.12 *Activity Diagram* Pemilihan Gambar

2. *Activity Diagram* Pemilihan Pesan

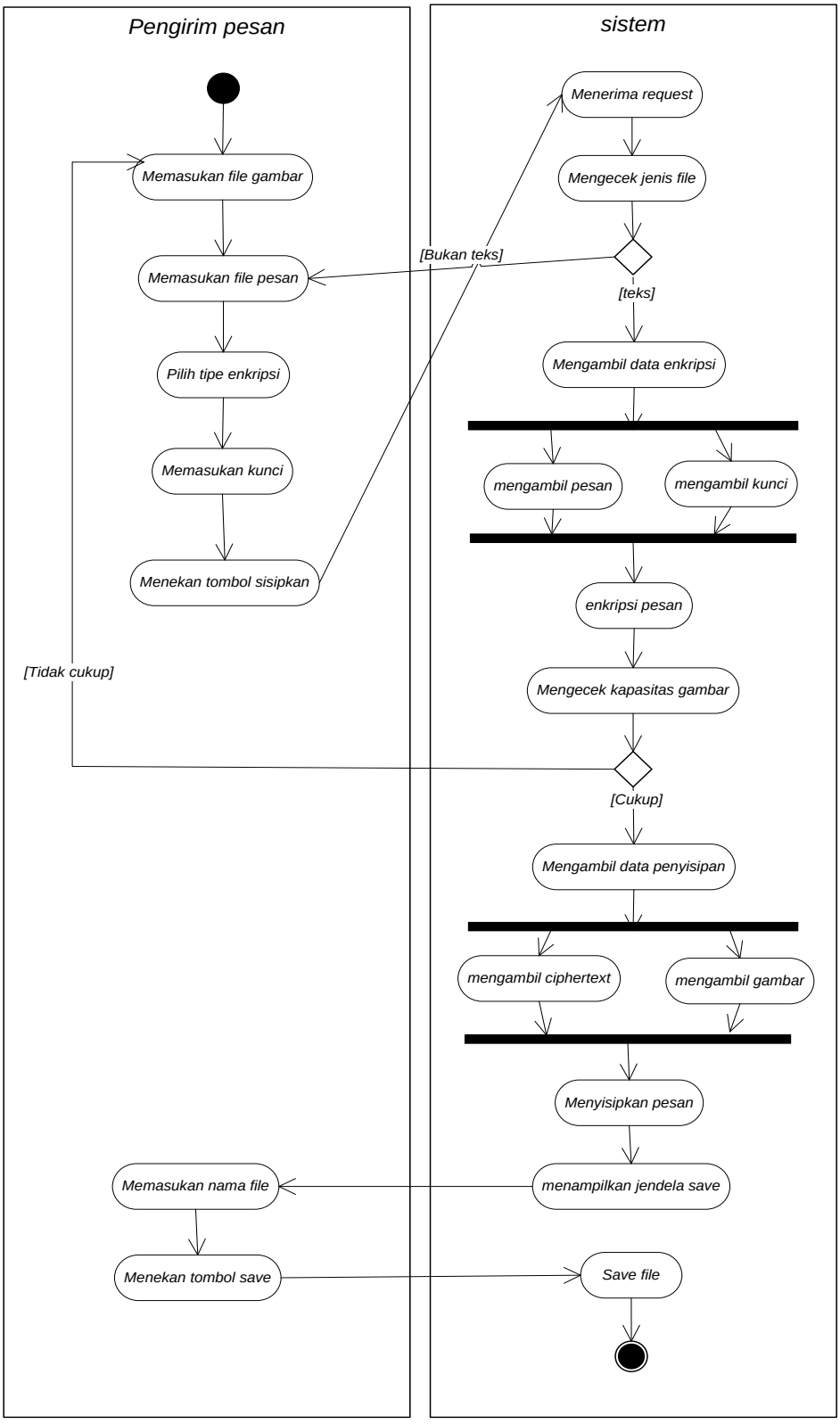
Activity diagram pemilihan pesan menggambarkan alir aktivitas pengambilan *file* pesan antara pengirim dangan sistem seperti terlihat pada gambar III.13.



Gambar III.13 *Activity Diagram* Pemilihan Pesan

3. *Activity Diagram* Penyisipan

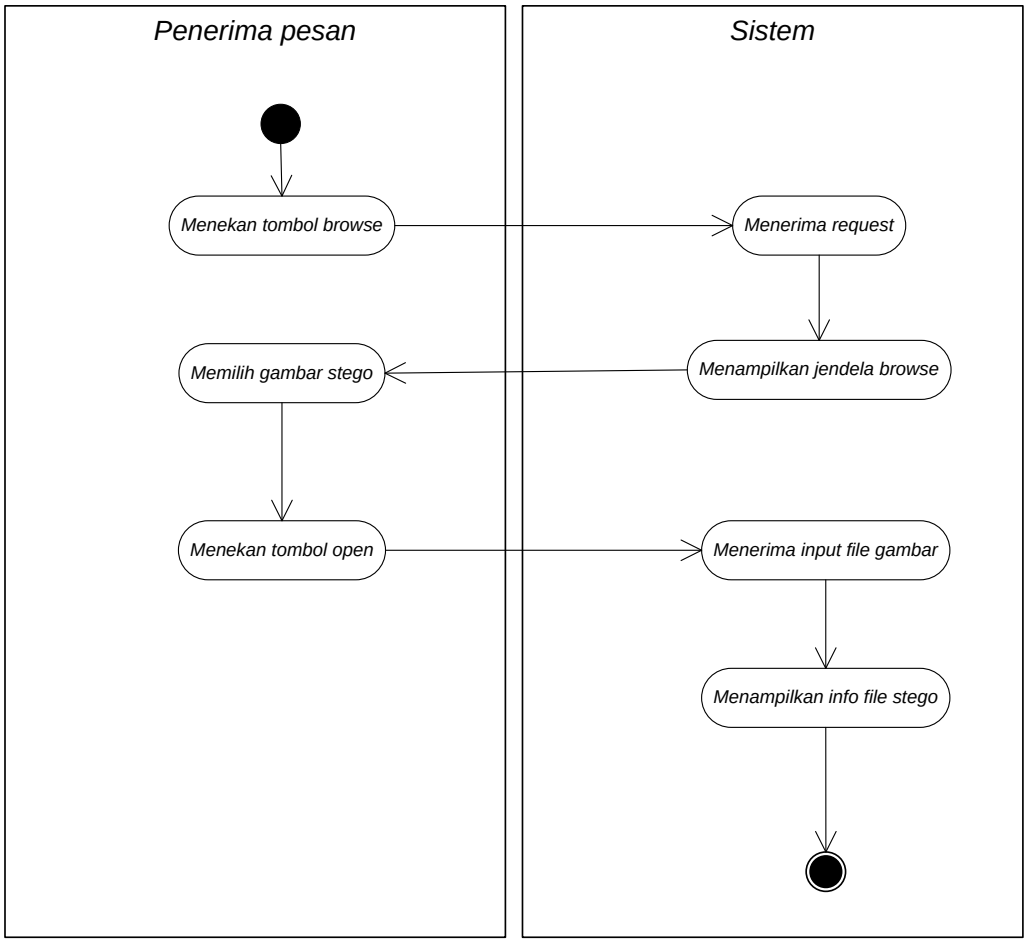
Activity diagram penyisipan menggambarkan alir aktivitas penyisipan yang dilakukan antara pengirim dangan sistem seperti terlihat pada gambar III.14.



Gambar III.14 *Activity Diagram* Penyisipan

4. *Activity Diagram* Pemilihan Gambar *Stego*

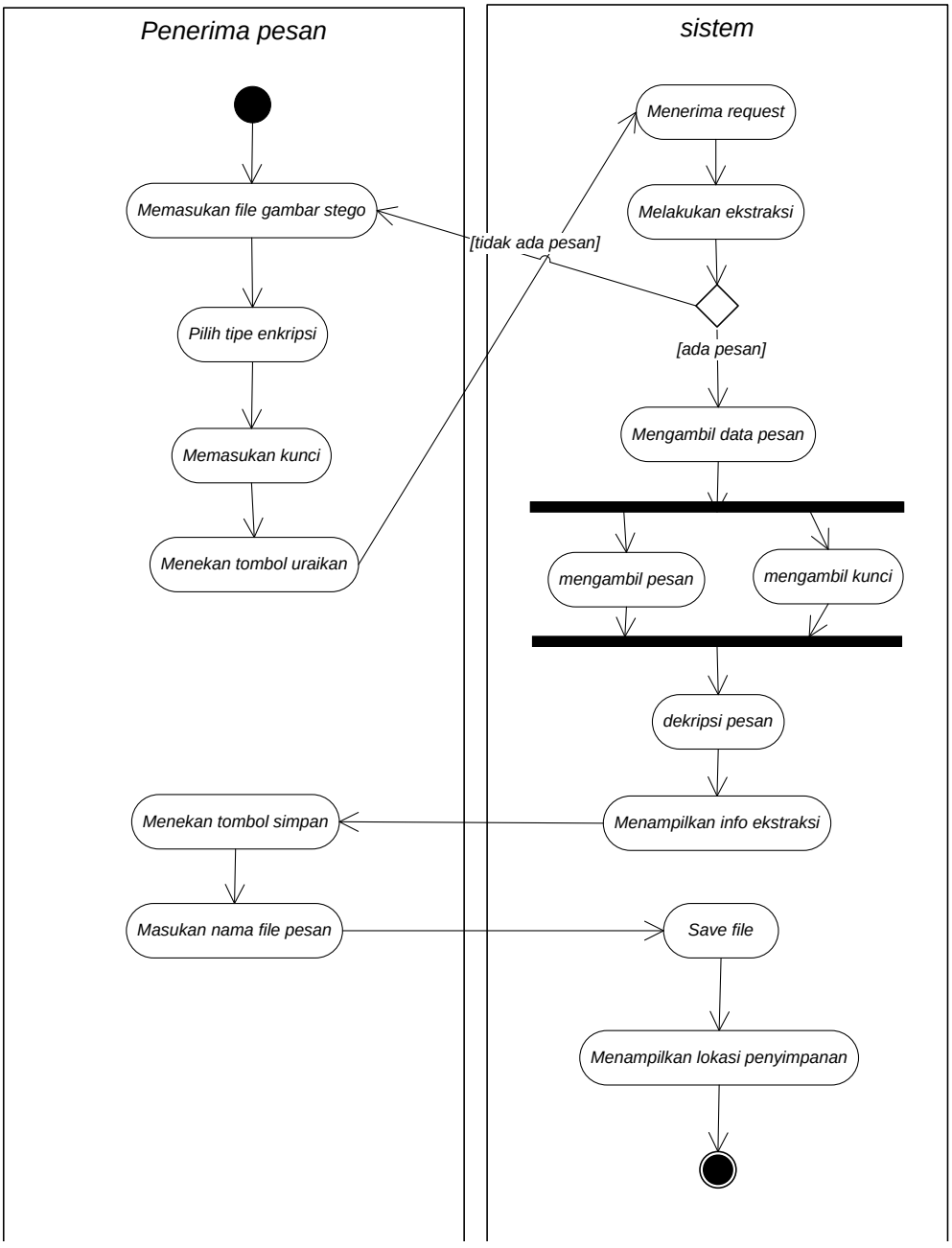
Activity diagram pemilihan gambar *stego* menggambarkan alir aktivitas pengambilan *file* gambar *stego* antara penerima dangan sistem seperti terlihat pada gambar III.15.



Gambar III.15 *Activity Diagram* Penyisipan Pemilihan Gambar *Stego*

5. *Activity Diagram* Ekstrak

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



Gambar III.16 *Activity Diagram* Ekstrak

III.7 Desain *User Interface*

Antar muka pemakai (*user interface*) adalah tampilan program yang dapat dilihat, didengar atau dipersepsikan oleh pengguna dan perintah-perintah atau mekanisme yang digunakan pemakai untuk mengendalikan operasi dan memasukkan data. Berikut ini merupakan perancangan antarmuka aplikasi steganografi dan kriptografi yang dirancang dengan 5 (lima) buah antarmuka, yaitu:

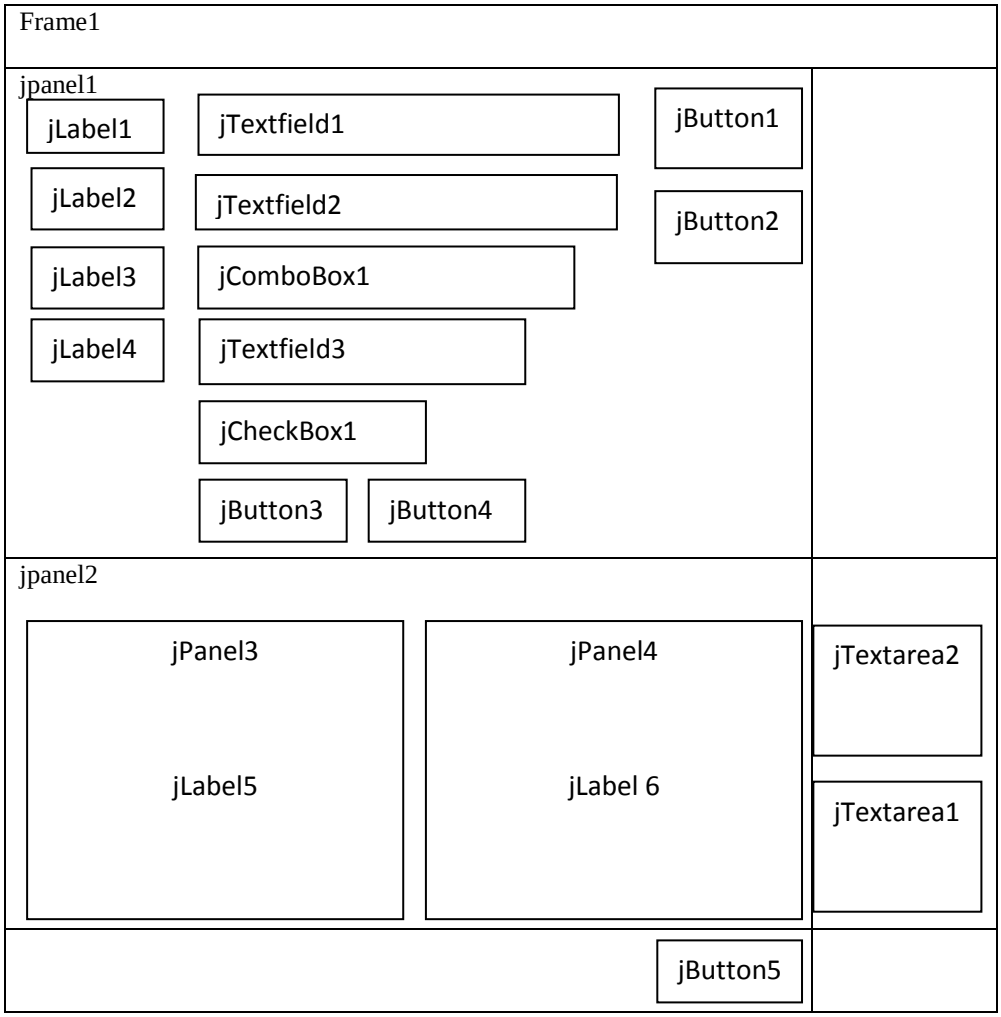
1. Desain *Form* Utama

Gambar III.17 Desain *Form* Utama

Gambar III.17 merupakan tampilan rancangan *form* utama. *Form* utama adalah *form* pembuka dari aplikasi ini. Terdiri atas 3 bagian yaitu bagian atas sebagai informasi nama dari aplikasi. Kemudian bagian utama/tengah

terdapat 5 buah tombol untuk menentukan proses yang akan dilakukan. Tombol “Sisipkan” merupakan *link* ke halaman proses *embedding*, tombol “Ekstraksi” adalah *link* ke halaman proses ekstraksi, tombol “Tentang” adalah *link* ke halaman penjelasan tentang program, tombol “Bantuan” adalah *link* ke halaman *troubleshooting*, tombol “Keluar” yang merupakan *link* untuk keluar jika tidak ingin melakukan proses apapun. Sedangkan pada bagian bawah terdapat informasi pembuat aplikasi.

2. Desain *Form* Penyisipan Pesan



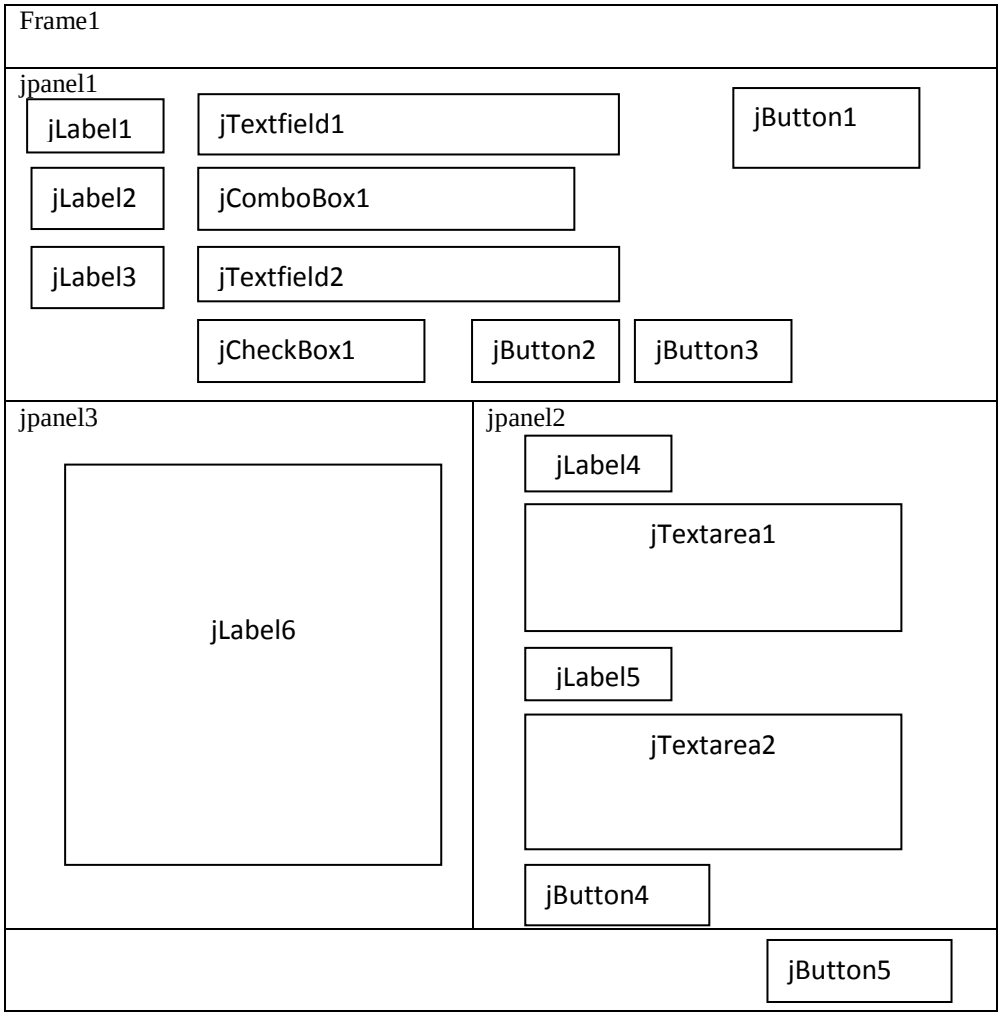
Gambar III.18 Desain *Form* Penyisipan Pesan

Form Penyisipan pesan ini berfungsi sebagai tempat berlangsungnya proses penyembunyian teks ke gambar. Dimana pengguna akan memanggil pesan yang akan disisipkan atau disembunyikan dan kemudian memilih gambar yang akan digunakan sebagai tempat penyisipan data pesan. Dalam *form* ini terdapat 1 buah *check box*, 1 buah *combo box*, 6 buah *label*, 3 buah *text field*, 5 buah *button*, 4 buah *panel*, 2 buah *text area*, 1 buah *frame*.

Tabel III.1 Fungsi *Properties Form* Penyisipan Pesan

No.	Objek	Nama	Fungsi
1	*jCheckbox1*	*jCheckbox1*	Menampilkan dan menyembunyikan karakter kunci pada *jTextfield3*
2	*jLabel1*	*jLabel1*	Keterangan *jTextfield1*
3	*jLabel2*	*jLabel2*	Keterangan *jTextfield2*
4	*jLabel3*	*jLabel3*	Keterangan *jCombobox1*
5	*jLabel4*	*jLabel4*	Keterangan *jTextfield3*
6	*jLabel5*	*jLabel5*	Menampilkan gambar asli
7	*jLabel6*	*jLabel6*	Menampilkan gambar hasil
8	*jTextfield1*	*jTextfield1*	Menampilkan lokasi *file* gambar
9	*jTextfield2*	*jTextfield2*	Menampilkan lokasi *file* pesan
10	*jTextfield3*	*jTextfield3*	Inputan kunci
11	*jButton1*	*jButton1*	Sebagai tombol buka untuk membuka gambar
12	*jButton2*	*jButton2*	Sebagai tombol buka untuk membuka pesan
13	*jButton3*	*jButton3*	Sebagai tombol proses penyisipan pesan
14	*jButton4*	*jButton4*	Menghapus nilai yang dimasukan
15	*jButton5*	*jButton5*	Kembali ke *Form* Utama
16	*jpanel1*	*jpanel1*	Membuat properti tetap tertata rapi
17	*Jpanel2*	*Jpanel2*	Membuat properti tetap tertata rapi
18	*Jpanel3*	*Jpanel3*	Membuat properti tetap tertata rapi
19	*Jpanel4*	*Jpanel4*	Membuat properti tetap tertata rapi
20	*jFrame1*	*jFrame1*	Bingkai *Form* penyisipan pesan
21	*jCombobox1*	*jCombobox1*	Untuk pemilihan tipe enkripsi
22	*jTextarea1*	*jTextarea1*	Untuk menampilkan hasil enkripsi
23	*jTextarea2*	*jTextarea2*	Untuk menampilkan pesan teks

3. Desain *Form* Penguraian Pesan



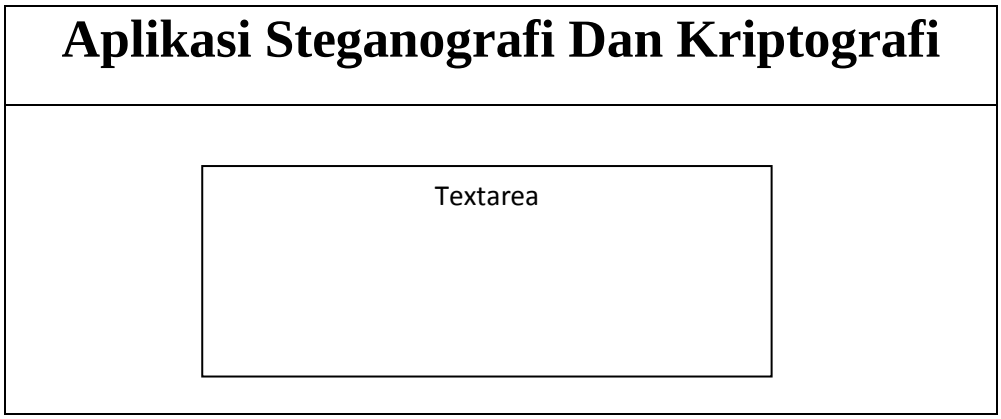
Gambar III.19 Desain *Form* Penguraian Pesan

Form Penguraian Pesan ini berfungsi sebagai tempat berlangsungnya proses pengeluaran pesan. Dimana pengguna akan memanggil gambar yang berisi pesan kemudian melakukan ekstraksi untuk mengeluarkan pesan dan pesan ditampilkan dalam bentuk *ciphertext* dan *plaintext*, yang selanjutnya pesan dapat disimpan.

Tabel III.2 Fungsi *Properties Form* Penguraian Pesan

No.	Objek	Nama	Fungsi
1	*jCheckbox1*	*jCheckbox1*	Menampilkan dan menyembunyikan karakter kunci pada *jTextfield2*
2	*jLabel1*	*jLabel1*	Keterangan *jTextfield1*
3	*jLabel2*	*jLabel2*	Keterangan *jCombobox1*
4	*jLabel3*	*jLabel3*	Keterangan *jTextfield2*
5	*jLabel4*	*jLabel4*	Keterangan *jTextarea1*
6	*jLabel5*	*jLabel5*	Keterangan *jTextarea2*
7	*jLabel6*	*jLabel6*	Menampilkan gambar *stego image*
8	*jTextfield1*	*jTextfield1*	Menampilkan lokasi *file* gambar
9	*jTextfield2*	*jTextfield2*	Inputan kunci
10	*jCombobox1*	*jCombobox1*	Untuk pemilihan tipe dekripsi
11	*jButton1*	*jButton1*	Sebagai tombol buka untuk membuka gambar
12	*jButton2*	*jButton2*	Sebagai tombol proses ekstraksi pesan
13	*jButton3*	*jButton3*	Menghapus nilai yang dimasukan
14	*jButton4*	*jButton4*	Menyimpan pesan yang telah diekstraksi
15	*jButton5*	*jButton5*	Kembali ke *Form* Utama
16	*jpanel1*	*jpanel1*	Membuat properti tetap tertata rapi
17	*Jpanel2*	*Jpanel2*	Membuat properti tetap tertata rapi
18	*Jpanel3*	*Jpanel3*	Membuat properti tetap tertata rapi
19	*jFrame1*	*jFrame1*	Bingkai Form Penguraian pesan

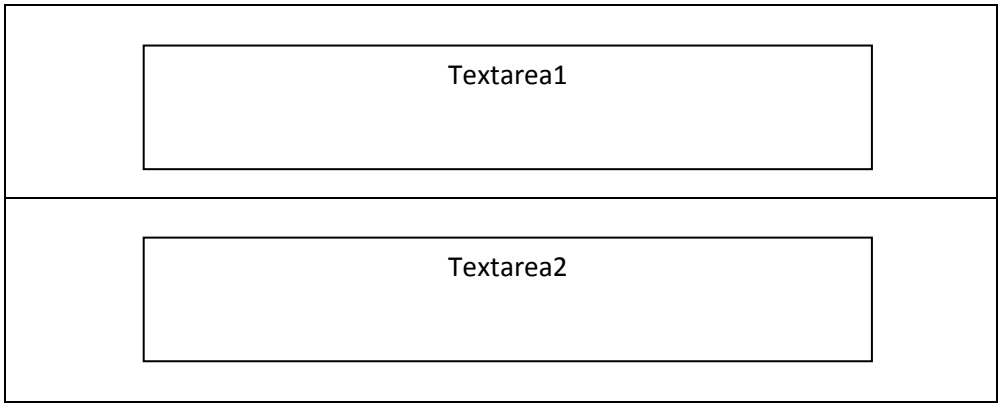
4. Desain *Form* Tentang



Gambar III.20 Desain *Form* Tentang

Gambar III.7 merupakan tampilan rancangan *form* tentang. *Form* tentang adalah *form* yang berisi penjelasan tentang aplikasi ini. Terdiri atas dua bagian yaitu bagian atas sebagai informasi nama dari aplikasi. Kemudian bagian bawah terdapat 1 buah *textarea* untuk menjelaskan kegunaan dan metode apa yang digunakan pada aplikasi.

5. Desain *Form* Bantuan



Gambar III.21 Desain *Form* Bantuan

Gambar III.8 merupakan tampilan rancangan *form* bantuan. *Form* bantuan adalah *form* yang berisi penjelasan untuk menggunakan aplikasi dan menangani masalah *error*. Terdiri atas dua bagian yaitu bagian atas terdapat 1 buah *textarea* sebagai informasi cara menggunakan aplikasi. Kemudian bagian bawah terdapat 1 buah *textarea* untuk menjelaskan masalah *error* yang terdapat pada aplikasi.