BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Dengan berkembangnya teknik pengambilan informasi secara ilegal, banyak orang yang mencoba untuk mengakses informasi yang bukan haknya. Oleh karena itu diperlukan suatu sistem pengamanan data yang bertujuan untuk meningkatkan keamanan data, melindungi suatu data atau pesan agar tidak dibaca oleh pihak yang tidak berwenang, dan mencegah pihak yang tidak berwenang untuk menyisipkan, menghapus, ataupun merubah data. Salah satu ilmu pengamanan data yang terkenal adalah kriptografi dan steganografi. Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan, data, atau informasi dengan cara menyamarkannya menjadi bentuk tersandi yang tidak mempunyai makna. Sedangkan steganografi yaitu penyisipan pesan tersembunyi pada *file cover* yang berfungsi sebagai media penampung sehingga tampak seperti pesan biasa, dimana pesan yang dikirim hanya dapat dibaca oleh penerima yang memiliki hak untuk mengetahui isi pesan tersebut. Berdasarkan hal tersebut pada penelitian ini akan dibangun sebuah aplikasi *mobile* yang bertujuan untuk mengamankan pesan dan menyembunyikannya pada gambar. Sehingga dapat bermanfaat untuk digunakan sebagai media bertukar pesan rahasia yang aman.

III.2. Strategi Pemecahan Masalah

Beberapa strategi pemecahan masalah dalam Perancangan Aplikasi Penyembunyian Pesan Ke Dalam Gambar Menggunakan Algoritma RC4 Dan LSB1 ini adalah sebagai berikut:

1. Aplikasi ini dibangun menggunakan perangkat lunak *eclipse* untuk digunakan pada perangkat *android*.
2. Aplikasi ini dapat digunakan untuk menyembunyikan pesan ke dalam gambar.
3. Aplikasi ini juga dapat digunakan untuk mengekstrak pesan yang terdapat pada gambar yang dihasilkan oleh aplikasi ini.
4. Algoritma yang digunakan dalam aplikasi ini adalah algoritma RC4 dan LSB1.

III.3. Studi Kasus Algoritma RC4 dan LSB1

Algoritma RC4 ini dikembangkan oleh Ronald Rivest untuk RSA *data security* pada tahun 1987 dan baru dipublikasikan untuk umum pada tahun 1994. RC4 merupakan salah satu algoritma kunci simetris yang berbentuk *stream cipher* dimana algoritma ini melakukan proses enkripsi/dekripsi dalam satu *byte* dan menggunakan kunci yang sama. RC4 menggunakan variabel yang panjang kuncinya dari 1 sampai 256 *bit* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *bit*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random bit* dan kemudian untuk menggenerasikan aliran *pseudo random* digunakan operasi *XOR* dengan *plaintext* untuk menghasilkan *ciphertext*. Masing-masing elemen dalam tabel ditukarkan minimal sekali.

Berikut adalah implementasi algoritma RC4 dengan mode 4 *byte* (untuk lebih menyederhanakan dalam perhitungan manual) serta untuk kebutuhan sistem yang sangat terbatas. S-*Box* dengan panjang 4 *byte*, dengan S[0]=0, S[1]=1, S[2]=2 dan S[3]=3 sehingga *array* S menjadi: 0 1 2 3

Inisialisasi 4 *byte* kunci *array*, K. Misalkan kunci Ulang kunci sampai memenuhi seluruh adalah 2 5 7 3, sehingga *array* K berisi 2 5 7 3 dan mencoba untuk mengenkripsikan kata SAYA.

Inisialisasi i dan j dengan 0 kemudian dilakukan KSA (*Key-scheduling algorithm*) agar tercipta *state-array* yang acak. Penjelasan iterasi lebih lanjut dapat dijelaskan sebagai berikut:

Iterasi 1

i = 0

j = (0 + S[0] + K [0 mod 4]) mod 4

= (0 + 0 + 2) mod 4 = 2

Swap (S[0],S[2])

Hasil Array S

2 1 0 3

Iterasi 2

i = 1

j = (2 + S[1] + K [1 mod 4]) mod 4

= (2 + 1 + 5) mod 4 =0

Swap (S[1],S[0])

Hasil Array S

1 2 0 3

Iterasi 3

i = 2

j = (0 + S[2] + K [2 mod 4]) mod 4

= (0 + 0 + 7) mod 4 = 3

Swap (S[2],S[3])

Hasil

1 2 3 0

Iterasi 4

i = 3

j = (3 + S[3] + K [3 mod 4]) mod 4

= (3 + 0 + 3) mod 4 = 2

Swap (S[3],S[2])

Hasil Array S

1 2 0 3

Setelah melakukan KSA, akan dilakukan PRGA (*Pseudo-random generation algorithm*). PRGA akan dilakukan sebanyak 4 kali dikarenakan plainteks yang akan dienkripsi berjumlah 4 karakter. Hal ini disebakan karena dibutuhkan 1 kunci dan 1 kali pengoperasian *XOR* untuk tiap tiap karakter pada plainteks. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

Array S

1 2 0 3

Inisialisasi

i = 0

j = 0

Iterasi 1

i = (0 + 1) mod 4 = 1

j = (0 + S[1]) mod 4 = (0+ 2) mod 4 = 2

swap (S[1],S[2])

1 0 2 3

K1 = S[(S[1]+S[2]) mod 4] = S[2 mod 4] = 2

K1 = 00000010

Iterasi 2

i = (1 + 1) mod 4 = 2

j = (2 + S[2]) mod 4 = (2+ 2) mod 4 = 0

swap (S[2],S[0])

2 0 1 3

K2 = S[(S[2]+S[0]) mod 4] = S[3 mod 4] = 3

K2 = 00000011

Iterasi 3

i = (2 + 1) mod 4 = 3

j = (0 + S[3]) mod 4 = (0+ 3) mod 4 = 3

swap (S[3],S[3])

1 0 2 3

K3 = S[(S[3]+S[3]) mod 4] = S[6 mod 4] = 2

K3 = 00000010

Iterasi 4

i = (3 + 1) mod 4 = 0

j = (3 + S[0]) mod 4 = (3+ 1) mod 4 = 0

swap (S[0],S[0])

1 0 2 3

K4 = S[(S[0]+S[0]) mod 4] = S[2 mod 4] = 2

K4 = 00000010

Setelah menemukan kunci untuk tiap karakter, maka dilakukan operasi *XOR* antara karakter pada *plaintext* dengan kunci yang dihasilkan. Berikut adalah tabel *ASCII* untuk tiap-tiap karakter pada *plaintext* yang digunakan.

Huruf Kode *ASCII* (*Binary* 8 *bit*)

KATA	S	A	Y	A
HEXADESIMAL	53	41	59	41
BINER	01010011	01000001	01011001	01000001

Berikut adalah proses peng*XOR*an dari *plaintext* dengan *key* yang telah didapat:

S A Y A

Plaintext : 01010011 01000001 01011001 01000001

Key : 00000010 00000011 00000010 00000010 ⊕

Ciphertext : 01010001 01000010 01011011 01000011

Q B [C

Setelah di dapat hasil enkripsi menggunakan algoritma RC4 selanjutnya *ciphertext* yang dihasilkan disisipkan ke dalam sebuah gambar menggunakan algoritma LSB1. Gambar yang digunakan adalah berukuran 44x44 *pixel* yang akan di zoom bagian tertentu untuk dilihat nilai RGB-nya sebagai berikut :

Gambar III.1. Contoh Gambar Asli

Gambar III.2. Contoh Gambar yang sudah di Zoom

Setelah proses zoom gambar di dapat nilai dari masing-masing *pixel* yang akan digunakan untuk menyembunyikan pesan. Nilainya adalah sebagai berikut :

R	G	B	R	G	B	R	G	B
255	0	0	255	0	0	255	1	0
255	0	0	255	0	0	255	1	0
255	0	1	255	0	1	255	1	0
255	0	1	255	0	1	255	0	1

Nilai RGB tersebut selanjutnya diubah kedalam bentuk biner :

R	G	B	R	G	B	R	G	B
11111111	00000000	00000000	11111111	00000000	00000000	11111111	00000001	00000000
11111111	00000000	00000000	11111111	00000000	00000000	11111111	00000001	00000000
11111111	00000000	00000001	11111111	00000000	00000001	11111111	00000001	00000000
11111111	00000000	00000001	11111111	00000000	00000001	11111111	00000000	00000001

Dan pesan yang akan disisipkan adalah QB[C, yang merupakan hasil enkripsi dari algoritma RC4 maka sebelum proses penyisipan pesan akan diubah ke dalam bentuk biner sehingga dihasilkan :

Q : **0 1 0 1 0 0 0 1**

B : **0 1 0 0 0 0 1 0**

[: **0 1 0 1 1 0 1 1**

C : **0 1 0 0 0 0 1 1**

Dalam algoritma LSB1 ketentuan penyisipan pesan adalah mengganti *bit* ke 8, 16 dan 24 setiap *pixel* gambar dengan nilai biner dari pesan yang akan disisipkan, sehingga berdasarkan ketentuan tersebut hasil penyisipan dapat dilihat pada *bit* yang ditebalkan (*bold*).

R	G	B	R	G	B	R	G	B
1111111**0**	0000000**1**	0000000**0**	1111111**1**	0000000**0**	0000000**0**	1111111**0**	0000000**1**	0000000**0**
1111111**1**	0000000**0**	0000000**0**	1111111**0**	0000000**0**	0000000**1**	1111111**0**	0000000**0**	0000000**1**
1111111**0**	0000000**1**	0000000**1**	1111111**0**	0000000**1**	0000000**1**	1111111**0**	0000000**1**	0000000**0**
1111111**0**	0000000**0**	0000000**0**	1111111**1**	0000000**1**	00000001	11111111	00000000	00000001

R	G	B	R	G	B	R	G	B
0000000**0**	0000000**1**	0000000**0**	0000000**1**	0000000**0**	0000000**0**	0000000**0**	0000000**1**	0000000**0**
0000000**1**	0000000**0**	0000000**0**	0000000**0**	0000000**0**	0000000**1**	0000000**0**	0000000**0**	0000000**1**
0000000**0**	0000000**1**	0000000**1**	0000000**0**	0000000**1**	0000000**1**	0000000**0**	0000000**1**	0000000**0**
0000000**0**	0000000**0**	0000001**0**	0000001**1**	0000001**1**	00000011	00000011	00000011	00000011

Proses pengambilan pesan menggunakan algoritma LSB1 adalah dengan mengambil nilai tiap-tiap *bit* paling kanan dari nilai biner *pixel* gambar. Sehingga

dari hasil penyembunyian pesan diatas, dari proses pengambilan tiap-tiap *bit* paling kanan akan dihasilkan nilai :

R	G	B	R	G	B	R	G	B
1111111**0**	0000000**1**	0000000**0**	1111111**1**	0000000**0**	0000000**0**	1111111**0**	0000000**1**	0000000**0**
1111111**1**	0000000**0**	0000000**0**	1111111**0**	0000000**0**	0000000**1**	1111111**0**	0000000**0**	0000000**1**
1111111**0**	0000000**1**	0000000**1**	1111111**0**	0000000**1**	0000000**1**	1111111**0**	0000000**1**	0000000**0**
1111111**0**	0000000**0**	0000000**0**	1111111**1**	0000000**1**	00000001	11111111	00000000	00000001

R	G	B	R	G	B	R	G	B
0000000**0**	0000000**1**	0000000**0**	0000000**1**	0000000**0**	0000000**0**	0000000**0**	0000000**1**	0000000**0**
0000000**1**	0000000**0**	0000000**0**	0000000**0**	0000000**0**	0000000**1**	0000000**0**	0000000**0**	0000000**1**
0000000**0**	0000000**1**	0000000**1**	0000000**0**	0000000**1**	0000000**1**	0000000**0**	0000000**1**	0000000**0**
0000000**0**	0000000**0**	0000001**0**	0000001**1**	0000001**1**	00000011	00000011	00000011	00000011

0 1 0 1 0 0 0 1 : Q

0 1 0 0 0 0 1 0 : B

0 1 0 1 1 0 1 1 : [

0 1 0 0 0 0 1 1 : C

Setelah pesan diambil dari gambar selanjutnya dilakukan proses dekripsi *ciphertext* menggunakan algoritma RC4. Proses ini sama untuk proses *key-schedule*-nya. Untuk mendapatkan *plaintext*, *ciphertext* yang diperoleh di *XOR*kan dengan *pseudo random byte* yang didapat sebelumnya. Maka hasilnya adalah *plaintext* atau teks asli.

Proses *XOR pseudo random byte* dengan *ciphertext* pada dekripsi yaitu:

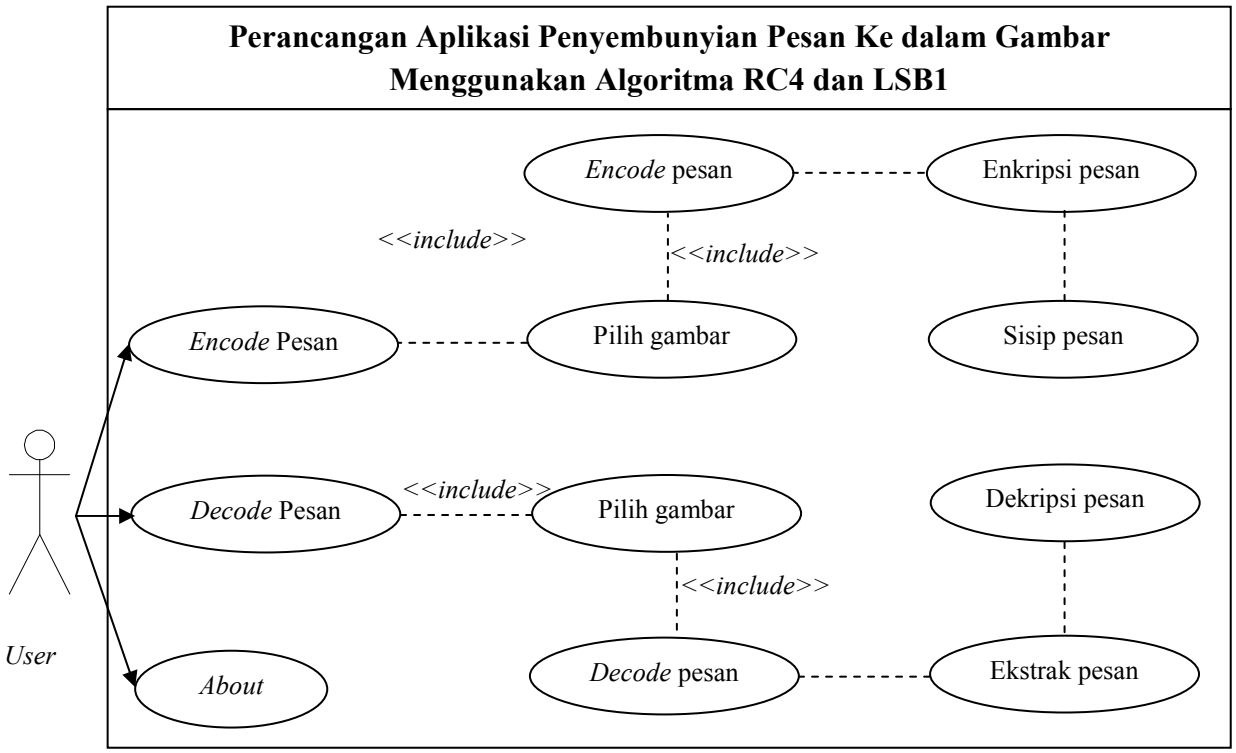
	Q	B	[	C
Ciphertext	: 01010001	01000010	01011011	01000011
Key	: 00000010	00000011	00000010	00000010 ⊕
Plaintext	: 01010011	01000001	01011001	01000001
	S	A	Y	A

III.4. Desain Sistem

Perancangan aplikasi penyembunyian pesan ke dalam gambar menggunakan algoritma RC4 dan LSB1 dirancang dengan menggunakan perangkat lunak *Eclipse*. Perancangan sistem yang dirancang terdiri dari *use case*, *flowchart*, *activity diagram* serta desain dan penjelasan dari sistem yang dirancang. Berikut adalah perancangannya :

III.4.1. *Use Case Diagram*

Use case mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi yang ada didalam sistem informasi tersebut. Berikut adalah *use case diagram* dari sistem yang dirancang :

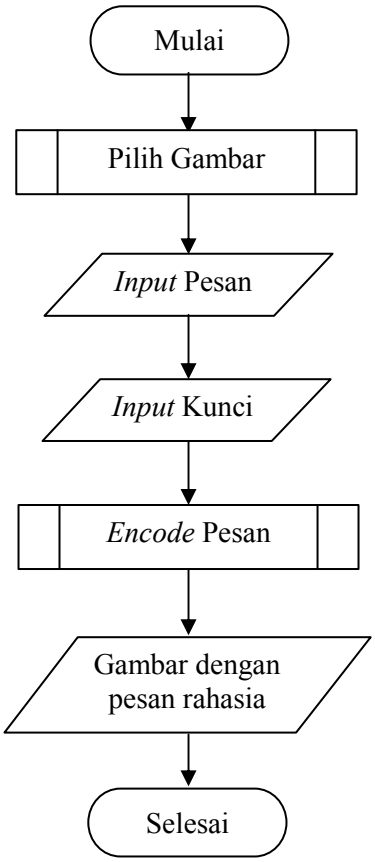


Gambar III.3. *Use Case Diagram* Aplikasi Penyembunyian Pesan

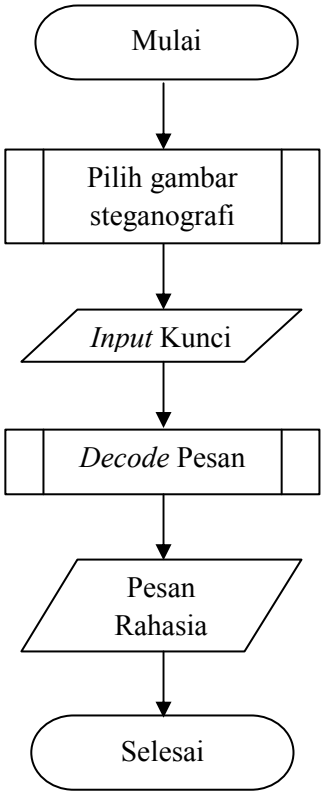
III.4.2. *Flowchart*

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.



Gambar III.4. *Flowchart Encode* Pesan



Gambar III.5. *Flowchart Decode* Pesan

1. Proses *Flowchart Encode* Pesan
    a. Mulai
    b. Pilih gambar
    c. Masukkan pesan yang akan disisipkan
    d. Masukkan kunci
    e. Lakukan proses enkripsi dan penyembunyian pesan
    f. Gambar steganografi
    g. Selesai

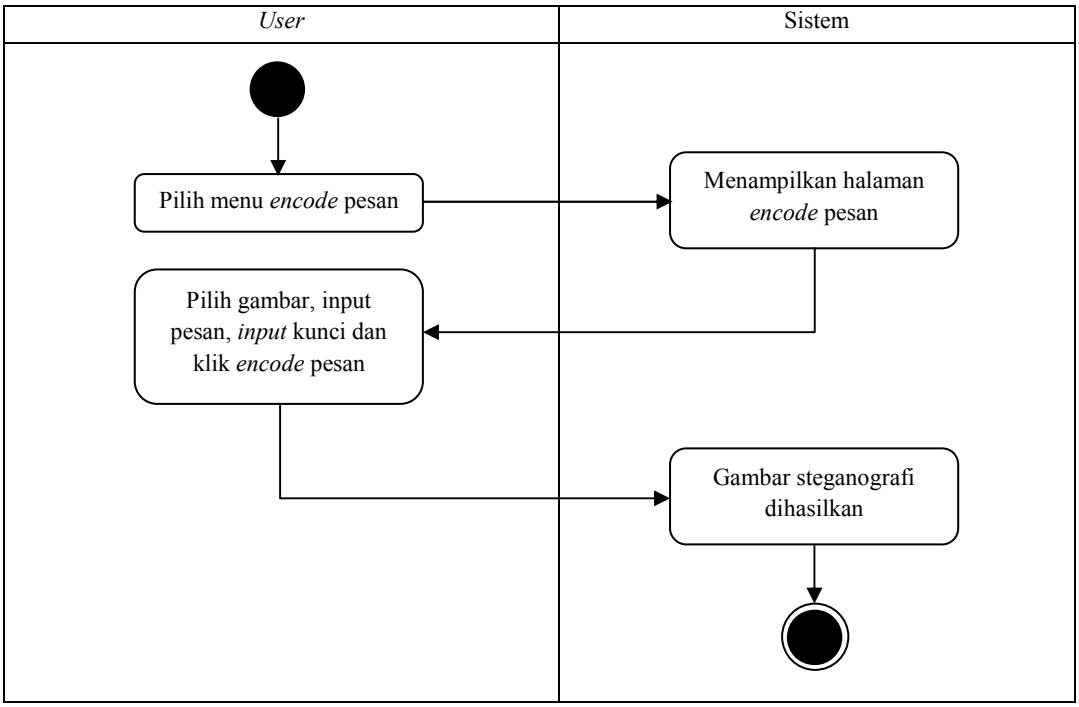
2. Proses *Flowchart Decode* Pesan
    a. Mulai
    b. Pilih gambar steganografi
    c. Masukkan kunci
    d. Ekstrak pesan
    e. Pesan rahasia ditampilkan
    f. Selesai

III.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 yaitu sebagai berikut :

III.4.3.1. *Activity Diagram Encode* Pesan

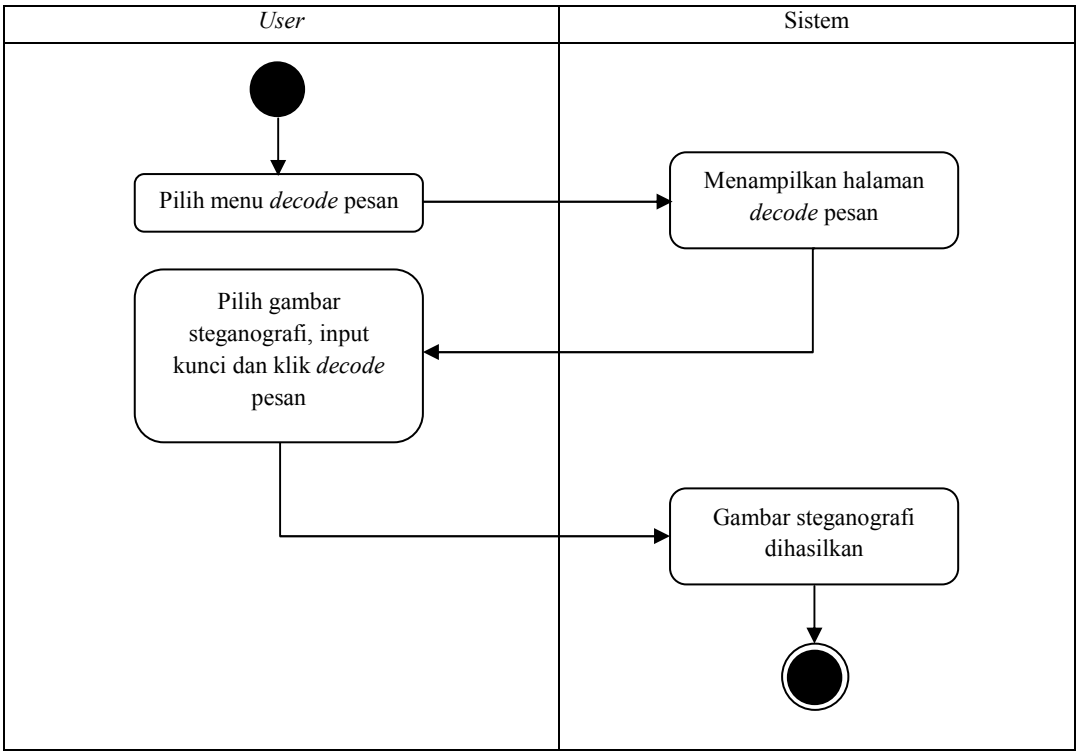
Activity diagram encode pesan menggambarkan alir aktifitas penyembunyian pesan yang dilakukan oleh pengguna dan diproses didalam sistem. *Activity diagram encode* pesan dapat dilihat pada gambar III.6.



Gambar III.6. *Activity Diagram Encode* Pesan

III.4.3.2. *Activity Diagram Decode* Pesan

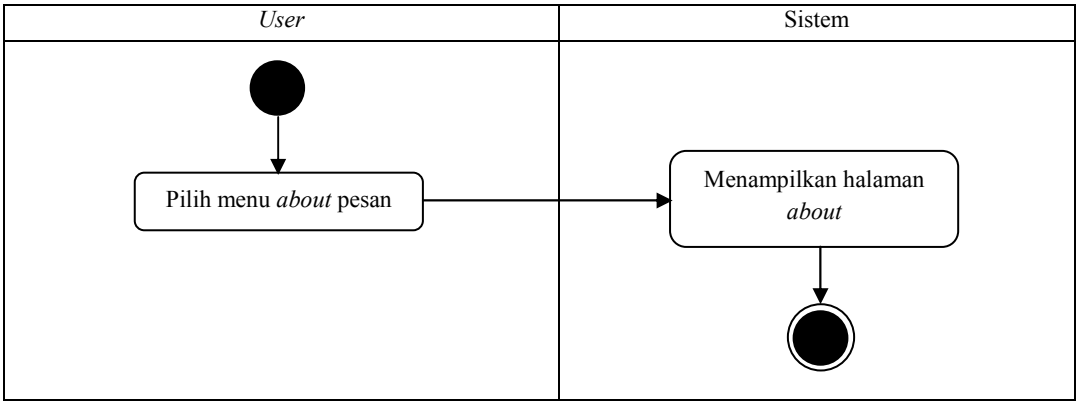
Activity diagram decode pesan menggambarkan alir aktifitas ekstrak pesan yang dilakukan oleh pengguna dan diproses didalam sistem. *Activity diagram decode* pesan dapat dilihat pada gambar III.7.



Gambar III.7. *Activity Diagram Decode* Pesan

III.4.3.3. *Activity Diagram About*

Activity diagram about menggambarkan alir aktifitas untuk menampilkan halaman *about* pada aplikasi yang akan dirancang. *Activity diagram about* dapat dilihat pada gambar III.8.

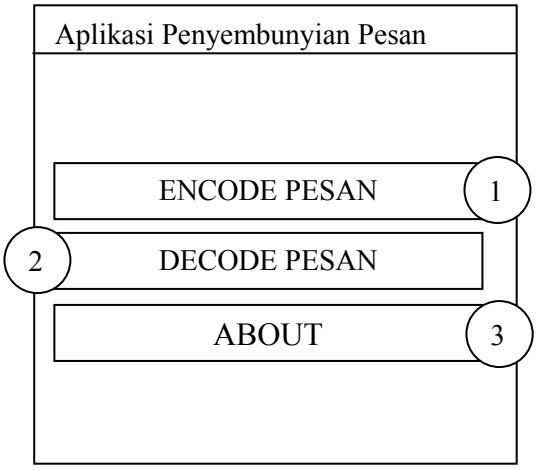


Gambar III.8. *Activity Diagram About* Pesan

III.5. Desain *User Interface*

Antarmuka peamakai (*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 penyembunyian pesan ke dalam gambar menggunakan algoritma RC4 dan LSB1 pada perangkat *mobile*, yaitu :

1. Desain Halaman Utama

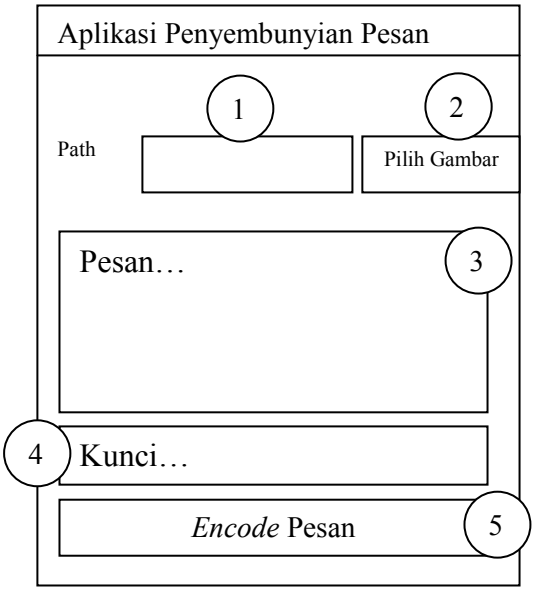


Gambar III.9. Desain *Form Menu* Utama

Keterangan tampilan halaman utama, yaitu :

1) Tombol untuk membuka halaman penyembunyian pesan.
2) Tombol untuk membuka halaman ekstrak pesan.
3) Tombol untuk melihat informasi pembuat aplikasi.

2. Desain Halaman *Menu Encode* Pesan

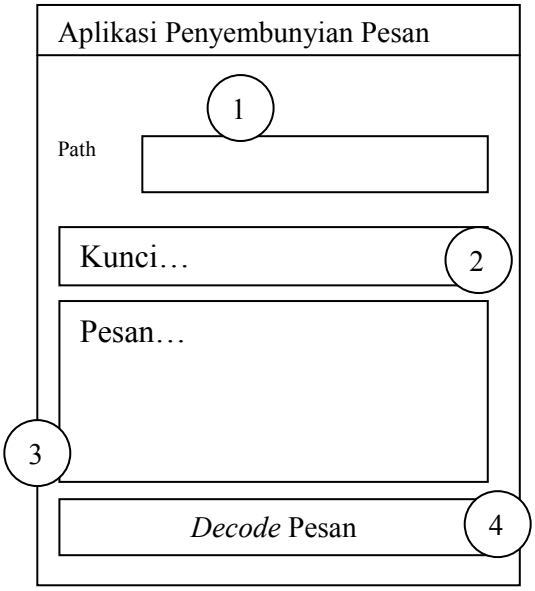


Gambar III.10. Desain Halaman *Menu Encode* Pesan

Tampilan halaman penyembunyian pesan yang berfungsi sebagai tempat berlangsungnya proses penyembunyian pesan teks ke dalam gambar. Adapun keterangannya sebagai berikut :

1) Direktori dari gambar yang dipilih.
2) Tombol untuk memilih gambar.
3) *Textbox* untuk menginputkan pesan.
4) *Textbox* untuk menginputkan kunci
5) Tombol untuk proses penyembunyian pesan ke dalam gambar.

3. Desain Halaman *Menu Decode* Pesan

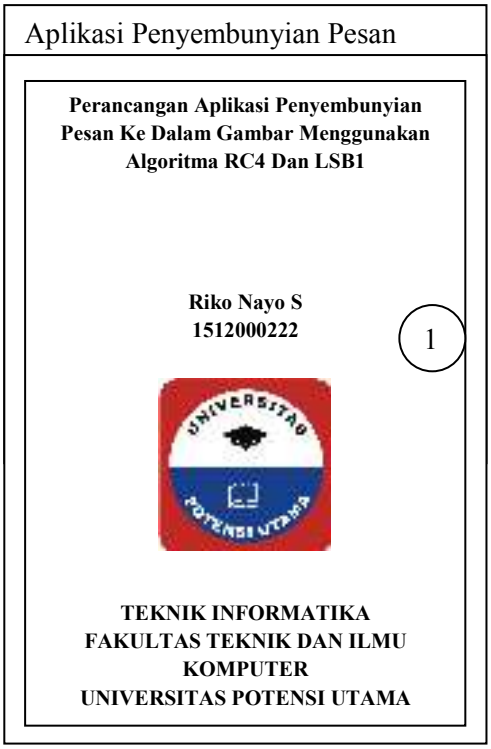


Gambar III.11. Desain *Form Menu Decode* Pesan

Merupakan tampilan dari halaman *menu decode* pesan yang berfungsi sebagai proses pengambilan pesan yang tersembunyi didalam gambar. Adapun keterangan tampilan halaman *menu decode* pesan sabagai berikut :

1) Direktori dari gambar yang dipilih.
2) *Textbox* untuk menginputkan kunci
3) *Textbox* untuk menampilkan pesan.
4) Tombol untuk melakukan ekstrak pesan.

4. Tampilan Halaman *Menu About*



Gambar III.12. Tampilan Halaman *Menu About*

Adapun keterangannya sebagai berikut :

1) Menampilkan judul skripsi dan data *programmer*.