BAB III

ANALISA DAN DESAIN SISTEM

III.1. Analisis Masalah

Sebuah pesan yang dikelola oleh pengelola informasi umumnya berbentuk sebuah tulisan, sehingga pesan tersebut dapat dibaca dan dimengerti dengan mudah. Namun pesan yang tersimpan dapat dibaca oleh siapa saja walaupun tidak berhak mengetahui pesan tersebut termasuk pencuri pesan. Sehingga pencuri pesan dapat menggunakan pesan yang dicuri untuk dikelola menjadi informasi yang baru dan lebih bermanfaat dan merugikan pemilik pesan. Oleh karena itu dibutuhkan sebuah cara yang dapat mengamankan pesan ataupun informasi yang bersifat rahasia sehingga tidak dapat dimiliki oleh pihak yang tidak diinginkan. Pada bidang ilmu komputer, untuk mengamankan informasi dalam bentuk tulisan dikenal sebagai teknik kriptografi. Namun peneliti ingin memperkuat keamanan sebuah pesan dengan menyisipkan pesan yang telah dirahasiakan ke dalam sebuah gambar dengan teknik steganografi, sehingga keamanan pesan menjadi lebih baik. Namun untuk menggunakan teknik kriptografi dibutuhkan sebuah metode yang baik dalam penyandian pesan. Peneliti menggunakan metode RSA untuk menyandikan pesan. Dengan menggunakan teknik kriptografi menggunakan metode RSA akan dapat menyandikan sebuah pesan kemudian pesan yang telah disandikan disisipkan pada sebuah gambar menggunakan teknik steganografi. Namun teknik steganografi membutuhkan sebuah metode, oleh karena itu peneliti menggunakan metode *Least Significant Bit* (LSB+2). Pada penelitian ini peneliti menambahkan nilai dua pada hasil penyisipan pesan ke dalam gambar pada metode LSB+2,

sehingga keamanan menjadi lebih baik. Dengan adanya penerapan dari dua teknik keamanan pesan yaitu kriptografi dan steganografi maka pesan yang memliki informasi yang bersifat pribadi memiliki keamanan yang lebih baik.

III.2. Penerapan Metode

Untuk dapat membuktikan keberhasilan pada suatu metode yang diterapkan ke dalam sebuah aplikasi, maka diperlukan sebuah perhitungan manual. Adapun perhitungan manual metode RSA dan LSB+2 adalah sebagai berikut :

III.2.1. Metode RSA

Langkah-langkah metode RSA untuk enkrip dapat dilihat sebagai berikut :

1. Enkrip Metode RSA

Berikut ini adalah enkrip dari metode RSA, enkrip metode RSA menggunakan fungsi eksponensial dalam modular n sebagai berikut :

$C_i = P_i^e \bmod n$

Keterangan :

C_i = *Ciphertext* hasil enkrip

P_i = *Plaintext* yang akan dienkrip

e = Fungsi eksponensial

mod = Sisa Bagi/Modulus

n = Hasil perkalian dua buah bilangan prima

Contoh Kasus :

a. Pembentukan Kunci :

1. Menentukan dua buah bilangan prima

Tentukan dua buah bilangan prima besar dengan ketentuan kedua bilangan prima tidak boleh sama.

$P_1 = 31$

$P_2 = 37$

2. Mencari nilai n

Untuk mendapatkan nilai n, maka gunakan rumus berikut :

$n = P_1 \times P_2$

$= 31 \times 37$

$= 1147$

3. Mencari nilai θn

Untuk mencari nilai θn gunakan rumus berikut :

$\theta n = (P_1 - 1) \times (P_2 - 1)$

$\theta n = (31 - 1) \times (37 - 1)$

$\theta n = 30 \times 36$

$\theta n = 1080$

4. Mencari nilai e

Untuk menentukan nilai e, gunakan algoritma berikut :

$e = 2$

While $\theta n \bmod e \neq 0$

$e = e + 1$

End While

Artinya :

Sampai $\theta n \bmod e \neq 0$, lakukan e = e + 1. Proses berhenti ketika nilai θn dibagi dengan nilai e memiliki sisa bagi tidak sama dengan nilai 0, maka akan didapat nilai e.

e = 3

Iterasi Pertama :

$\theta n \bmod e = 1080 \bmod 3$

$= 0$

e = 3 + 1

e = 4

Iterasi Kedua :

$\theta n \bmod e = 1080 \bmod 4$

$= 0$

e = 4 + 1

e = 5

Iterasi Ketiga :

$\theta n \bmod e = 1080 \bmod 5$

$= 0$

e = 5 + 1

e = 6

Iterasi Keempat :

$\theta n \bmod e = 1080 \bmod 6$

= 0

e = 6 + 1

e = 7

Iterasi Kelima :

θn mod e = 1080 mod 7

= 2

Proses berhenti pada iterasi kelima, maka telah didapat nilai e = 7.

5. Mencari nilai d

Untuk mencari nilai d, maka dapat digunakan teorema *extended euclid* sebagai berikut :

$U_1 = 1$

$U_2 = 0$

$U_3 = \theta n$

$V_1 = 0$

$V_2 = 1$

$V_3 = e$

While $V_3 = 0$

$Q = Int(U_3 / V_3)$

$N_1 = U_1 - (Q \times V_1)$

$N_2 = U_2 - (Q \times V_2)$

$N_3 = U_3 - (Q \times V_3)$

$U_1 = V_1$

$U_2 = V_2$

$U_3 = V_3$

$V_1 = N_1$

$V_2 = N_2$

$V_3 = N_3$

End While

Artinya :

Sampai $V_3 = 0$, lakukan teorema *extended euclid*. Proses berhenti ketika nilai V_3 sama dengan nilai 0, maka akan didapat nilai d.

$U_1 = 1$

$U_2 = 0$

$U_3 = \theta n$

$V_1 = 0$

$V_2 = 1$

$V_3 = e$

Iterasi Pertama :

$Q = Int(U_3 / V_3)$

$= Int(1080 / 7)$

$= 154$

$N_1 = U_1 - (Q \times V_1)$

$= 1 - (154 \times 0)$

$= 1$

$N_2 = U_2 - (Q \times V_2)$

$= 0 - (154 \times 1)$

= -154

$N_3 = U_3 - (Q \times V_3)$

= 1080 - (154 x 7)

= 1080 - 1078

= 2

$U_1 = 0$

$U_2 = 1$

$U_3 = 7$

$V_1 = 1$

$V_2 = -154$

$V_3 = 2$

Iterasi Kedua :

$Q = Int(U_3 / V_3)$

= Int(7 /2)

= 3

$N_1 = U_1 - (Q \times V_1)$

= 0 - (3 x 1)

= -3

$N_2 = U_2 – (Q \times V_2)$

= 1 – (3 x -154)

= 463

$N_3 = U_3 - (Q \times V_3)$

= 7 - (3 x 2)

= 7 - 6

= 1

$U_1 = 1$

$U_2 = -154$

$U_3 = 2$

$V_1 = -3$

$V_2 = 463$

$V_3 = 1$

Iterasi Ketiga :

$Q = Int(U_3 / V_3)$

$= Int(2 / 1)$

$= 2$

$N_1 = U_1 - (Q \times V_1)$

$= 1 - (2 \times -3)$

$= 7$

$N_2 = U_2 - (Q \times V_2)$

$= -154 - (2 \times 463)$

$= -154 - 926$

$= - 1080$

$N_3 = U_3 - (Q \times V_3)$

$= 2 - (2 \times 1)$

$= 2 - 2$

$= 0$

$U_1 = -3$

$U_2 = 463$

$U_3 = 1$

$V_1 = 7$

$V_2 = -1080$

$V_3 = 0$

Proses berhenti ketika $V_3 = 0$, maka telah didapat nilai d = 463.

Maka telah diperoleh kunci *private* untuk enkrip sebagai berikut :

e = 7

n = 1147

Dan diperoleh kunci *public* untuk dekrip sebagai berikut :

d = 463

n = 1147

b. Enkrip *Plaintext*

Contoh Proses Enkrip :

Plaintext : POTENSI UTAMA

Enkrip Pertama :

P = 80

$C_i = P_i^e \bmod n$

$= 80^7 \bmod 1147$

$= 20971520000000 \bmod 1147$

$= 660$

Enkrip Kedua :

O = 79

$C_i = P_i^e \bmod n$

$= 79^7 \bmod 1147$

$= 19203908986159 \bmod 1147$

$= 1128$

Enkrip Ketiga :

T = 84

$C_i = P_i^e \bmod n$

$= 84^7 \bmod 1147$

$= 29509034655744 \bmod 1147$

$= 269$

Enkrip Keempat :

E = 69

$C_i = P_i^e \bmod n$

$= 69^7 \bmod 1147$

$= 7446353252589 \bmod 1147$

$= 648$

Enkrip Kelima :

N = 78

$C_i = P_i^e \bmod n$

$= 78^7 \bmod 1147$

$= 17565568854912 \bmod 1147$

= 659

Enkrip Keenam :

S = 83

$C_i = P_i^e \bmod n$

$= 83^7 \bmod 1147$

= 27136050989627 mod 1147

= 941

Enkrip Ketujuh :

I = 73

$C_i = P_i^e \bmod n$

$= 73^7 \bmod 1147$

= 11047398519097 mod 1147

= 850

Enkrip Kedelapan :

= 32

$C_i = P_i^e \bmod n$

$= 32^7 \bmod 1147$

= 34359738368 mod 1147

= 1055

Enkrip Kesembilan :

U = 85

$C_i = P_i^e \bmod n$

$= 85^7 \bmod 1147$

= 32057708828125 mod 1147

= 122

Enkrip Kesepuluh :

T = 84

$C_i = P_i^e \bmod n$

$= 84^7 \bmod 1147$

= 29509034655744 mod 1147

= 269

Enkrip Kesebelas :

A = 65

$C_i = P_i^e \bmod n$

$= 65^7 \bmod 1147$

= 4902227890625 mod 1147

= 761

Enkrip Kedua belas :

M = 77

$C_i = P_i^e \bmod n$

$= 77^7 \bmod 1147$

= 16048523266853 mod 1147

= 1077

Enkrip Ketiga belas :

A = 65

$C_i = P_i^e \bmod n$

$= 65^7 \bmod 1147$

$= 4902227890625 \bmod 1147$

$= 761$

2. Dekrip Metode RSA

Berikut ini adalah dekrip dari metode RSA, dekrip metode RSA merupakan fungsi eksponensial dalam modular n dengan menggunakan kunci *private* sebagai berikut :

$$P_i = C_i^d \bmod n$$

Keterangan :

P_i = *Plaintext* hasil dekrip

C_i = *Ciphertext* yang akan didekrip

d = Fungsi eksponensial kunci *public*

mod = Sisa Bagi/Modulus

n = Fungsi perkalian dua bilangan prima

1. Terima kunci

d = 463

n = 1147

2. Dekrip *Ciphertext*

Contoh Proses Dekrip :

Dekrip Pertama :

C = 660

$P_i = C_i^d \bmod n$

$= 660^{463} \bmod 1147$

= 2,810879032129157273168565612080 5e+1305 mod 1147

= 80 = P

Dekrip Kedua :

C = 1128

$P_i = C_i^d \bmod n$

$= 1128^{463} \bmod 1147$

= 1,656201359164289833094731671496 3e+1413 mod 1147

= 79 = O

Dekrip Ketiga :

C = 269

$P_i = C_i^d \bmod n$

$= 269^{463} \bmod 1147$

= 9,447255068259840601634893182 32e+1124 mod 1147

= 84 = T

Dekrip Keempat :

C = 648

$P_i = C_i^d \bmod n$

$= 648^{463} \bmod 1147$

= 5,744175735220589484373442905 2717e+1301 mod 1147

= 69 = E

Dekrip Kelima :

C = 659

$P_i = C_i^d \bmod n$

$= 659^{463} \bmod 1147$

$= 1{,}392986658792430704663252958671 1e+1305 \bmod 1147$

$= 78 = N$

Dekrip Keenam :

$C = 941$

$P_i = C_i^d \bmod n$

$= 941^{463} \bmod 1147$

$= 5{,}915557046472247595367728069406e+1376 \bmod 1147$

$= 83 = S$

Dekrip Ketujuh :

$C = 850$

$P_i = C_i^d \bmod n$

$= 850^{463} \bmod 1147$

$= 2{,}093932153165143283083059356539 8e+1356 \bmod 1147$

$= 73 = I$

Dekrip Kedelapan :

$C = 1055$

$P_i = C_i^d \bmod n$

$= 1055^{463} \bmod 1147$

$= 5{,}832957477385789945130796932846 9e+1399 \bmod 1147$

$= 32 =$

Dekrip Kesembilan :

C = 122

$P_i = C_i^d \bmod n$

$= 122^{463} \bmod 1147$

= 9,651650840205358246697375194232 1e+965 mod 1147

= 85 = U

Dekrip Kesepuluh :

C = 269

$P_i = C_i^d \bmod n$

$= 269^{463} \bmod 1147$

= 9,447255068259840601634893182 32e+1124 mod 1147

= 84 = T

Dekrip Kesebelas :

C = 761

$P_i = C_i^d \bmod n$

$= 648^{463} \bmod 1147$

= 5,744175735220589484373442905 2717e+1301 mod 1147

= 65 = A

Dekrip Kedua belas :

C = 1077

$P_i = C_i^d \bmod n$

$= 1077^{463} \bmod 1147$

= 8,238926470782347878069839749 4152e+1403 mod 1147

= 77 = M

Dekrip Ketiga belas :

C = 761

$P_i = C_i^d \bmod n$

$= 761^{463} \bmod 1147$

= 1,20530257608012423327223965893 04e+1334 mod 1147

= 65 = A

Plaintext : POTENSI UTAMA

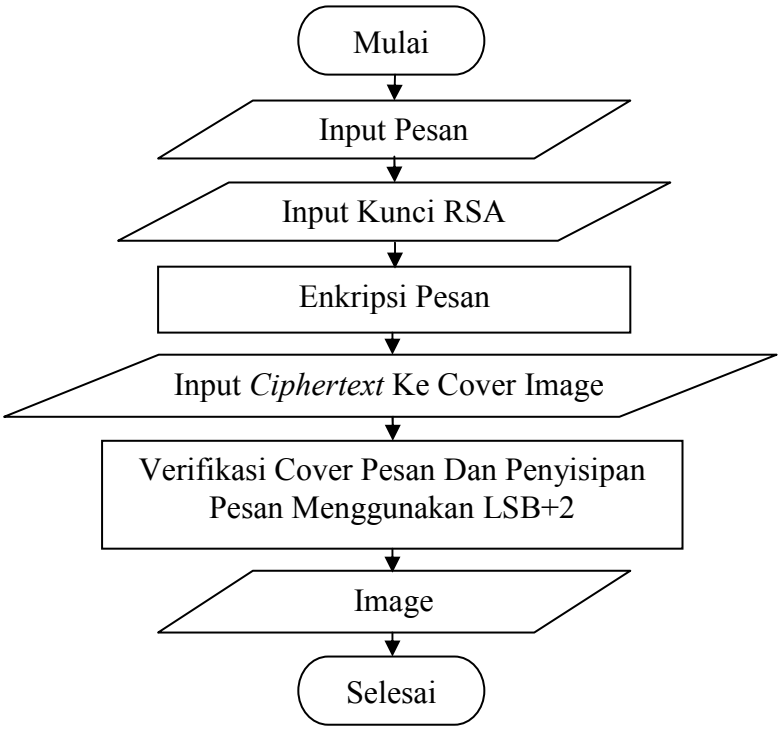
III.2.2. Metode LSB+2

Langkah-langkah metode LSB+2 untuk penyisipan dan pengembalian pesan dapat dilihat sebagai berikut :

1. Tahapan dalam proses penyisipan pesan yakni :
   a. Input pesan rahasia.
   b. Verifikasi pesan, harus diinputkan.
   c. *Input cover image*.
   d. Verifikasi *cover image*.
   e. Penyisipan pesan dengan metode *Least Significant Bit* yaitu dengan cara data disisipkan pada akhir file dengan diberi tanda khusus sebagai pengenal start dari data tersebut dan pengenal akhir dari data tersebut.
2. Tahapan dalam proses pengembalian pesan yakni :
   a. *Input stego image*.
   b. Verifikasi *stego image*.
   c. Verifikasi *stego image*, harus terdapat pesan.

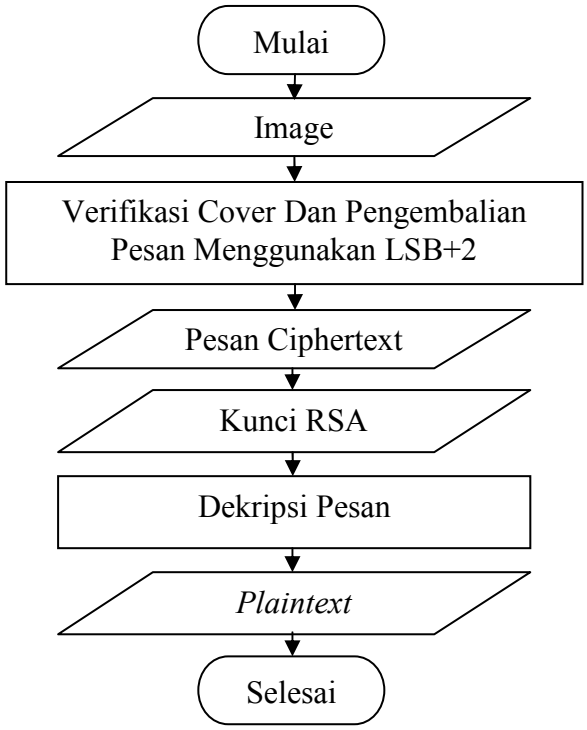
d. Pengambilan pesan dengan metode *Least Significant Bit* yaitu dengan cara data yang telah diisip pesan pada akhir file dibuka dari tanda khusus sebagai pengenal start dari data tersebut dan pengenal akhir dari data tersebut.

Flowchart penyisipan metode *Least Significant Bit* :



Gambar III.1. *Flowchart* Penyisipan *Ciphertext* RSA Dengan LSB+2

Flowchart ekstraksi pesan metode *Least Significant Bit* :



Gambar III.2. *Flowchart* Ekstraksi *Plaintext* RSA Dengan LSB+2

Studi kasus :

Pada sebuah citra grayscale 3x3 pixel disisipkan pesan yang tertulis “n”.

1. Proses Penyisipan Pesan

Untuk menyisipkan pesan ke dalam sebuah gambar, maka terlebih dahulu pesan diubah ke dalam kode *ascii* kemudian diubah lagi ke dalam bilangan biner sebagai berikut :

Kode ASCII dari pesan adalah sebagai berikut :

n = 110 = n = 0110 1110

Misalkan matriks tingkat derajat keabuan citra sebagai berikut :

Tabel III.1. RGB Matriks Gambar Awal

Red (R)	Green (G)	Blue (B)	Blue Diberikan LSB+2
1111 1111	0000 0000	0000 0000	0000 0010
1111 1111	0000 0000	0000 0000	0000 0010

1111 1111	0000 0000	0000 0000	0000 0010
0000 0000	1111 1111	0000 0000	0000 0010
0000 0000	1111 1111	0000 0000	0000 0010
0000 0000	1111 1111	0000 0000	0000 0010
0000 0000	0000 0000	1111 1111	1111 1111
0000 0000	0000 0000	1111 1111	1111 1111
0000 0000	0000 0000	1111 1111	1111 1111

Kemudian pesan n = 0110 1110 disisipkan sehingga menjadi :

Tabel III.2. RGB Matriks Gambar Akhir

Red (R)	Green (G)	Blue (B)
1111 1111	0000 0000	0000 001<u>0</u>
1111 1111	0000 0000	0000 001<u>1</u>
1111 1111	0000 0000	0000 001<u>1</u>
0000 0000	1111 1111	0000 001<u>0</u>
0000 0000	1111 1111	0000 001<u>1</u>
0000 0000	1111 1111	0000 001<u>1</u>
0000 0000	0000 0000	1111 111<u>1</u>
0000 0000	0000 0000	1111 111<u>0</u>
0000 0000	0000 0000	1111 111<u>1</u>

Keterangan :

Angka yang digaris bawahi adalah pesan yang disisipkan menggunakan metode LSB+2.

2. Proses Ekstraksi Pesan

Untuk mengekstrak pesan dari sebuah citra gambar yang disisipkan sebuah pesan, maka terlebih dahulu sebuah gambar diubah dalam bentuk nilai warna ke dalam kode *ascii* kemudian diubah lagi menggunakan bilangan biner sebagai berikut :

Tabel III.3. Citra Gambar Tersisip Pesan

Red (R)	Green (G)	Blue (B)
1111 1111	0000 0000	0000 0000
1111 1111	0000 0000	0000 0001
1111 1111	0000 0000	0000 0001
0000 0000	1111 1111	0000 0000
0000 0000	1111 1111	0000 0001
0000 0000	1111 1111	0000 0001
0000 0000	0000 0000	1111 1111
0000 0000	0000 0000	1111 1110
0000 0000	0000 0000	1111 1111

Pada tabel III.2 merupakan citra gambar yang tersisip sebuah pesan, untuk mengekstrak sebuah pesan menggunakan metode LSB+2 maka dapat diambil bit dari bilangan biner paling kanan dan kemudian disusun menjadi 8 bit bilangan biner sebagai berikut :

0110 1110

Setelah bit tersusun menjadi 8 blok, kemudian 8 bit tersebut diubah ke dalam bentuk bilangan desimal yang menjadi kode *ascii* sebagai berikut :

0110 1110 = 110

Kode *ascii* tersebut diubah ke dalam karakter, sehingga pesan yang tadinya tersisip sudah dapat diketahui :

110 = n

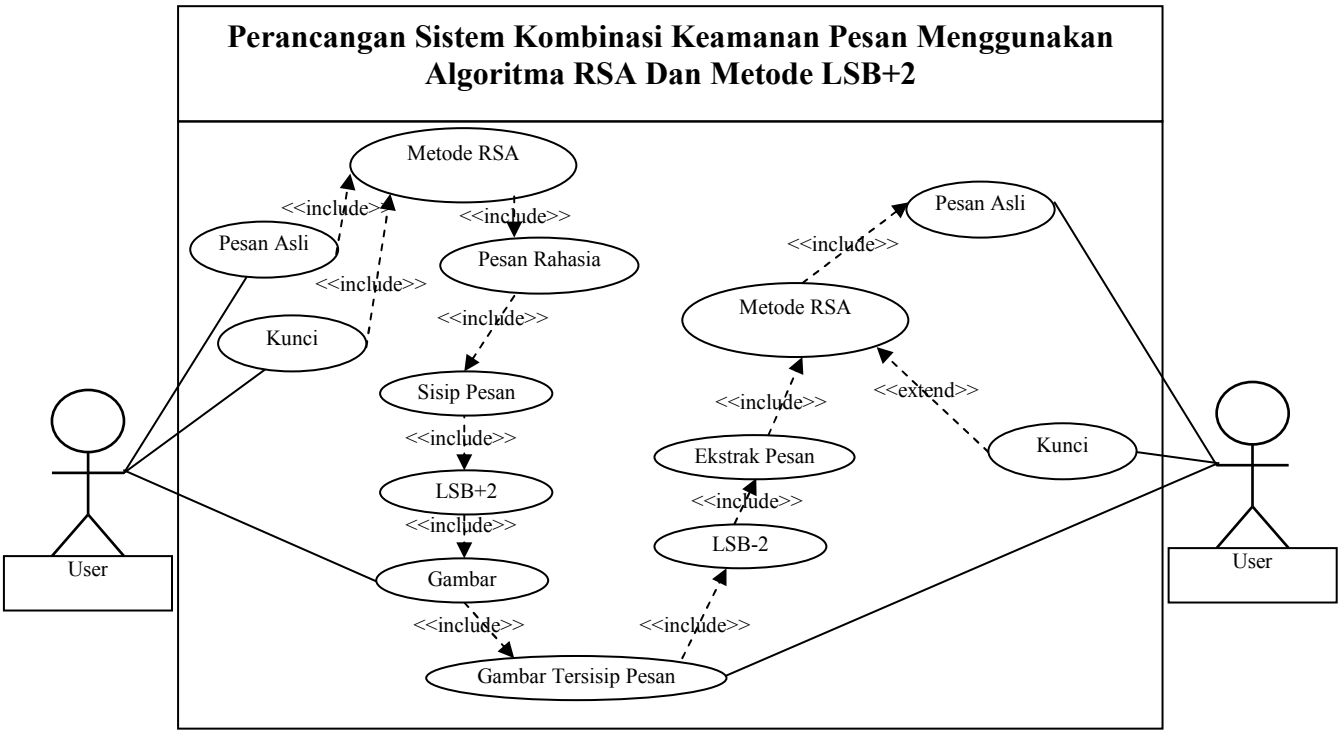
III.3. Desain Sistem

III.3.1. Desain Sistem Secara Global

Desain sistem atau perancangan sistem adalah proses pengembangan spesifikasi baru berdasarkan hasil rekomendasi analisis sistem. Dalam tahap perancangan, diharuskan merancang spesifikasi yang dibutuhkan.Bentuk rancangan sistem yang penulis buat menggunakan beberapa bentuk diagram dari *Unified Modeling Language* (UML) yaitu *Use Cace Diagram*, *Class Diagram* dan *Activity Diagram*.

III.3.1.1 *Use Case Diagram*

Perancangan dimulai dari identifikasi aktor dan bagaimana hubungan antara aktor dan *use case* didalam sistem. Perancangan *Use Case* Diagram dapat dilihat pada gambar III.3.



Gambar III.3. *Use Case* Perancangan Sistem Kombinasi Keamanan Pesan Menggunakan Algoritma RSA Dan Metode LSB+2

Keterangan :

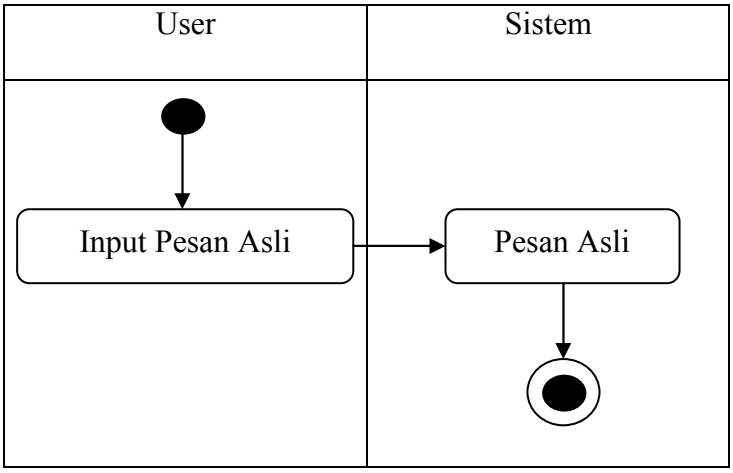
Mula-mula penyisip memasukkan gambar, pesan dan kunci. Pesan dan kunci diproses menggunakan metode RSA sehingga menghasilkan pesan rahasia. Kemudian pesan rahasia disisipkan menggunakan metode LSB+2 kedalam gambar. Untuk mengekstrak pesan, pengekstrak mengambil gambar tersisip pesan dan mengekstrak menggunakan metode LSB+2 dan kemudian memasukkan kunci untuk melakukan dekripsi pesan menggunakan metode RSA.

III.3.1.2. *Activity Diagram*

Pada proses ini kita akan membuat alur dari sistem yang dirancang yaitu *activity diagram*. Berikut adalah *activity diagram* sistem yang dirancang.

1. *Activity Diagram Input* Pesan Asli

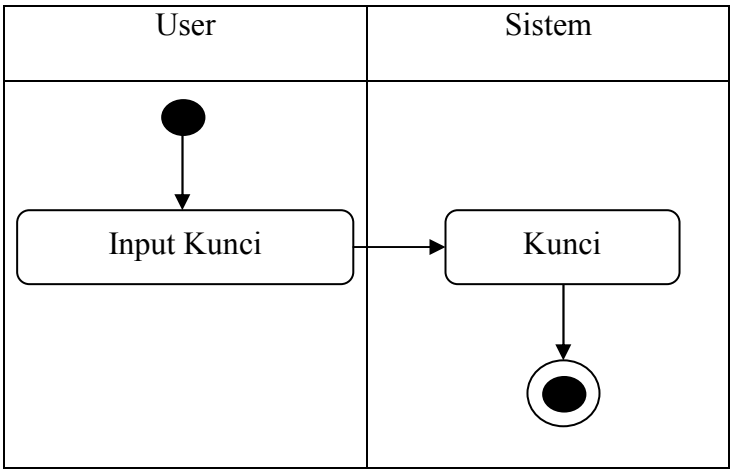
Aktivitas yang dilakukan untuk melakukan *input* pesan asli dapat dilihat seperti pada gambar III.4 berikut :



Gambar III.4. *Activity Diagram Input* Pesan Asli

2. *Activity Diagram Input* Kunci

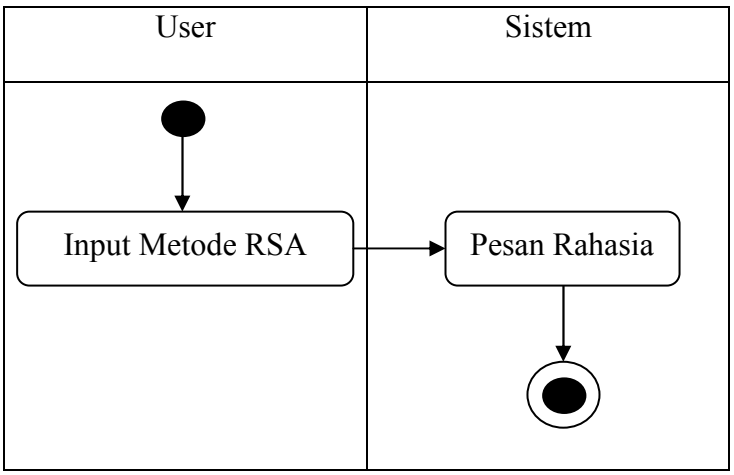
Aktivitas yang dilakukan untuk melakukan *input* kunci dapat dilihat seperti pada gambar III.5 berikut :



Gambar III.5. *Activity Diagram Input* Kunci

3. *Activity Diagram Input* Metode RSA

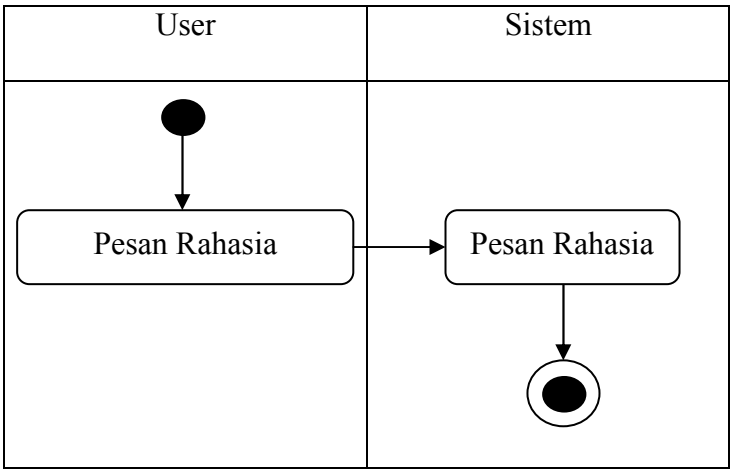
Aktivitas yang dilakukan untuk melakukan *input* Metode RSA dapat dilihat seperti pada gambar III.6 berikut :



Gambar III.6. *Activity Diagram Input* Metode RSA

4. *Activity Diagram* Pesan Rahasia

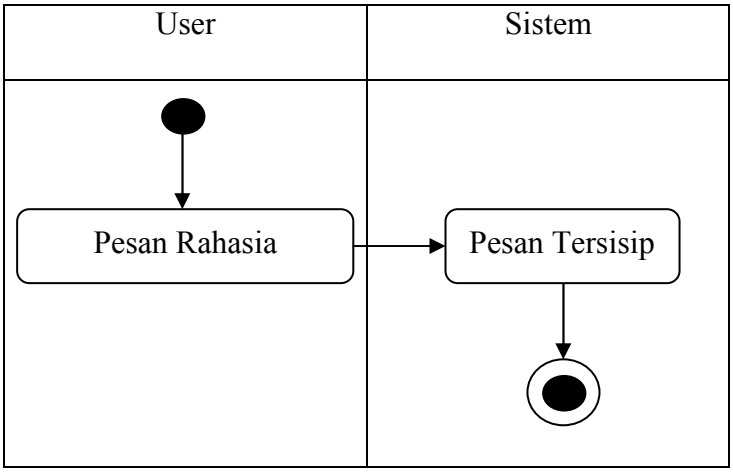
Aktivitas yang dilakukan untuk melakukan Pesan Rahasia dapat dilihat seperti pada gambar III.7 berikut :



Gambar III.7. *Activity Diagram* Pesan Rahasia

5. *Activity Diagram* Sisip Pesan

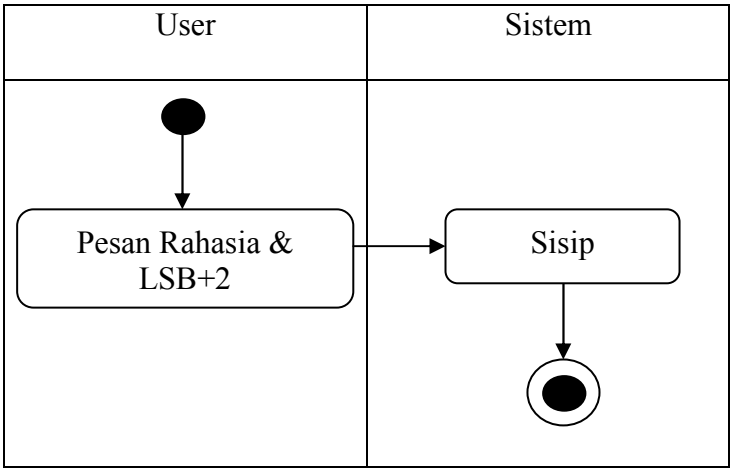
Aktivitas yang dilakukan untuk melakukan Sisip Pesan dapat dilihat seperti pada gambar III.8 berikut :



Gambar III.8. *Activity Diagram* Sisip Pesan

6. *Activity Diagram* LSB+2

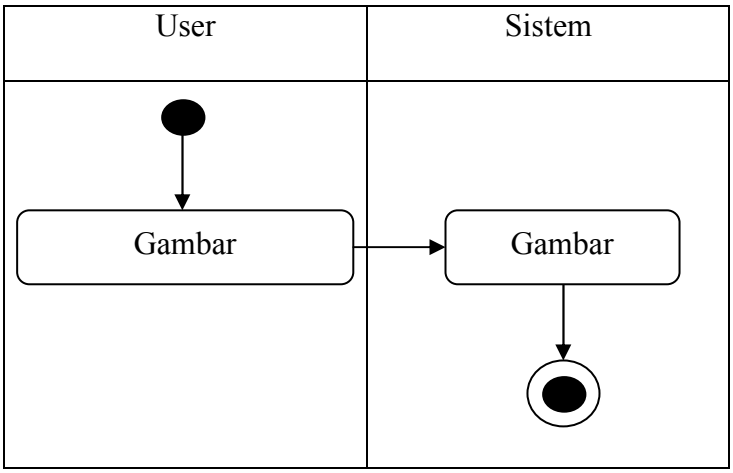
Aktivitas yang dilakukan untuk melakukan LSB+2 dapat dilihat seperti pada gambar III.9 berikut :



Gambar III.9. *Activity Diagram* LSB+2

6. *Activity Diagram* Gambar

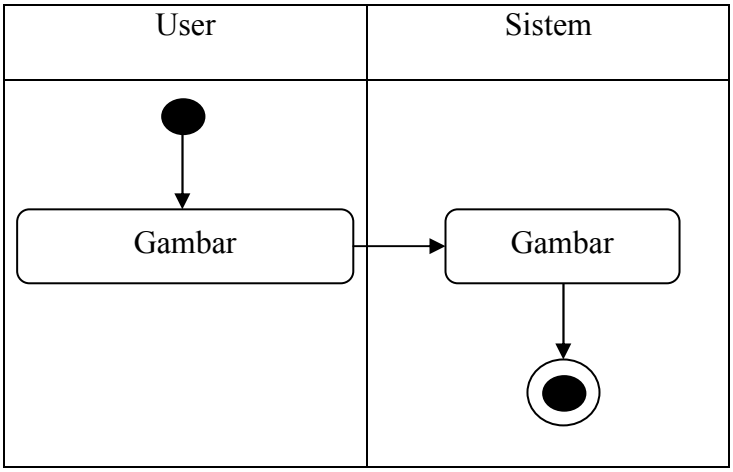
Aktivitas yang dilakukan untuk melakukan Gambar dapat dilihat seperti pada gambar III.10 berikut :



Gambar III.10. *Activity Diagram* Gambar

7. *Activity Diagram* Gambar Tersisip Pesan

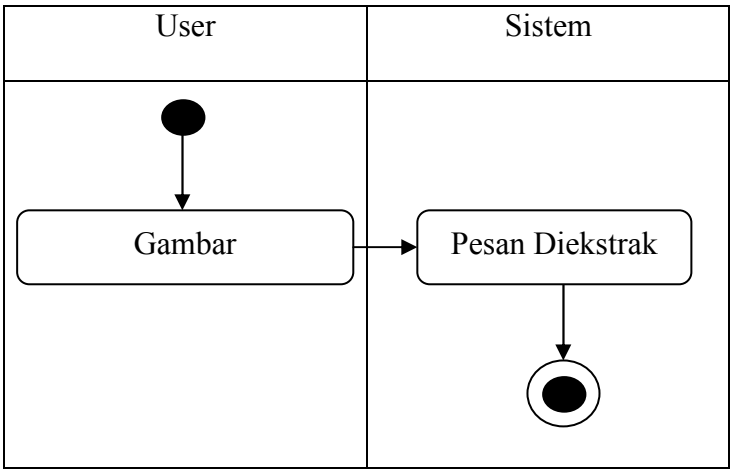
Aktivitas yang dilakukan untuk Gambar Tersisip Pesan dapat dilihat seperti pada gambar III.11 berikut :



Gambar III.11. *Activity Diagram* Gambar Tersisip Pesan

8. *Activity Diagram* Ekstrak Pesan

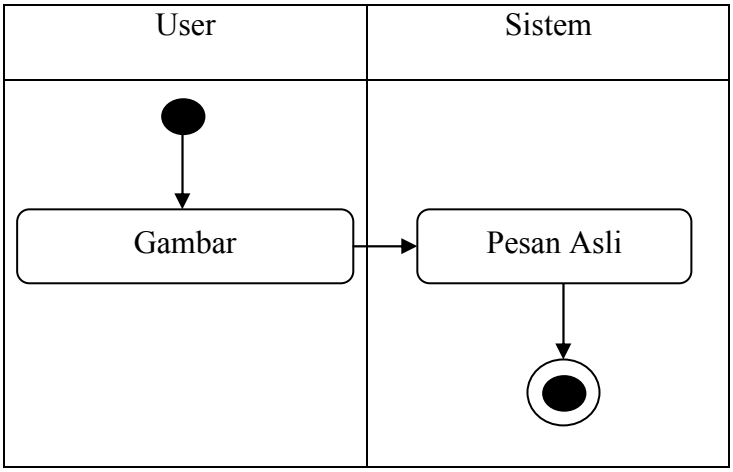
Aktivitas yang dilakukan untuk Elstrak Pesan dapat dilihat seperti pada gambar III.12 berikut :



Gambar III.12. *Activity Diagram* Ekstrak Pesan

9. *Activity Diagram* Pesan Asli

Aktivitas yang dilakukan untuk Pesan Asli dapat dilihat seperti pada gambar III.13 berikut :



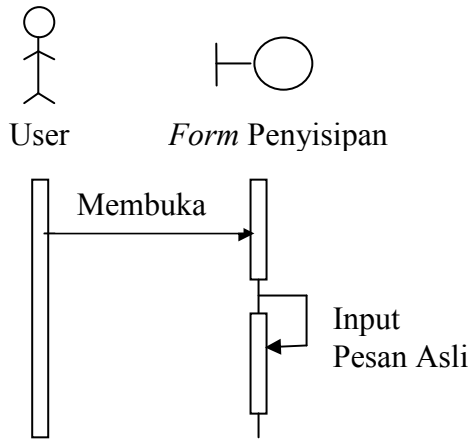
Gambar III.13. *Activity Diagram* Pesan Asli

III.3.1.3. *Sequence Diagram*

Rangkaian kegiatan pada setiap terjadi *event* sistem digambarkan pada *sequence* diagram berikut :

1. *Sequence Diagram Input* Pesan Asli

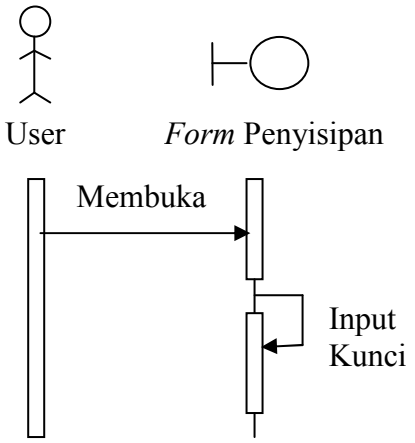
Serangkaian kerja melakukan *input* pesan asli dapat terlihat seperti pada gambar III.14 berikut :



Gambar III.14. *Sequence Diagram Input* Pesan Asli

2. *Sequence Diagram Input* Kunci

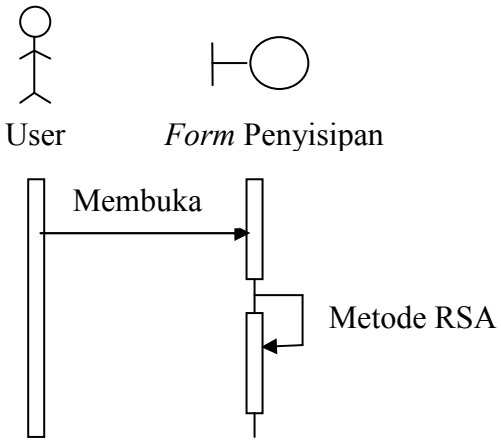
Serangkaian kerja melakukan *input* kunci dapat terlihat seperti pada gambar III.15 berikut :



Gambar III.15. *Sequence Diagram Input* Kunci

3. *Sequence Diagram* Metode RSA

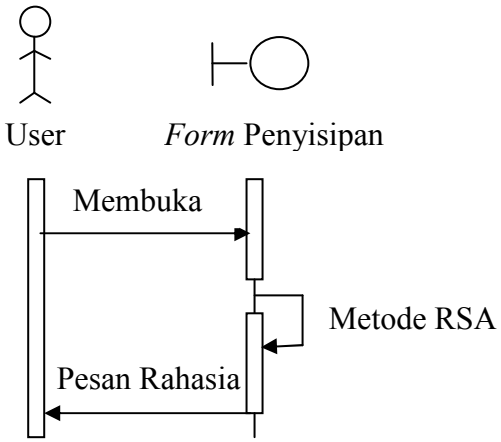
Serangkaian kerja melakukan Metode RSA dapat terlihat seperti pada gambar III.16 berikut :



Gambar III.16. *Sequence Diagram* Metode RSA

4. *Sequence Diagram* Pesan Rahasia

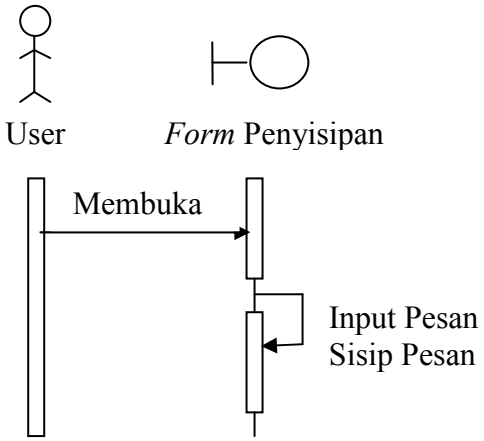
Serangkaian kerja melakukan penerimaan pesan rahasia dapat terlihat seperti pada gambar III.17 berikut :



Gambar III.17. *Sequence Diagram* Metode RSA

5. *Sequence Diagram* Sisip Pesan

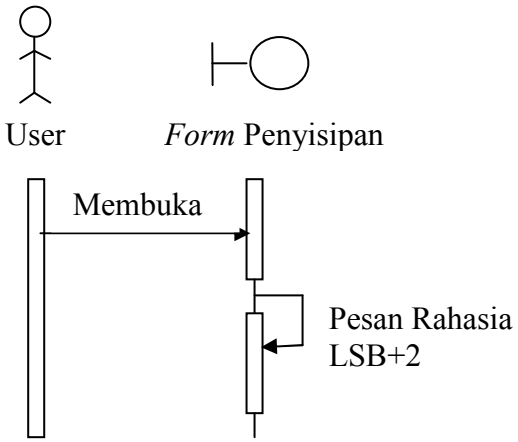
Serangkaian kerja melakukan proses sisip pesan dapat terlihat seperti pada gambar III.18 berikut :



Gambar III.18. *Sequence Diagram* Sisip Pesan

6. *Sequence Diagram* LSB+2

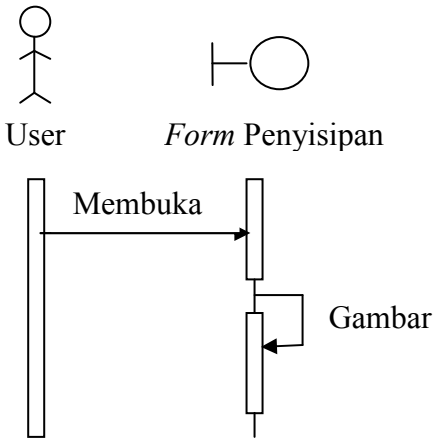
Serangkaian kerja melakukan proses LSB+2 dapat terlihat seperti pada gambar III.19 berikut :



Gambar III.19. *Sequence Diagram* LSB+2

7. *Sequence Diagram* Gambar

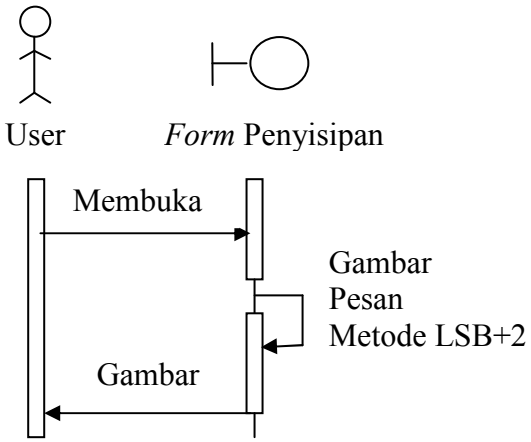
Serangkaian kerja melakukan proses gambar dapat terlihat seperti pada gambar III.20 berikut :



Gambar III.20. *Sequence Diagram* Gambar

8. *Sequence Diagram* Gambar Tersisip Pesan

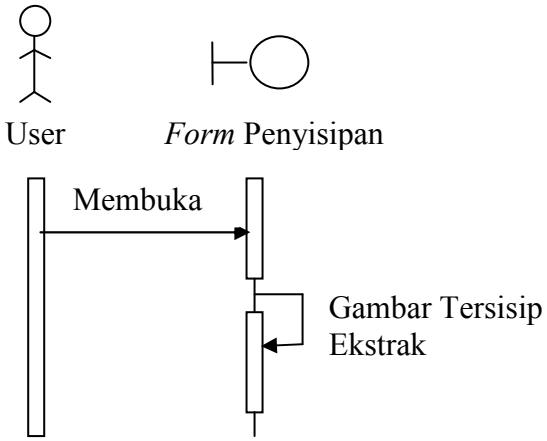
Serangkaian kerja melakukan proses gambar tersisip pesan dapat terlihat seperti pada gambar III.21 berikut :



Gambar III.21. *Sequence Diagram* Gambar Tersisip Pesan

9. *Sequence Diagram* Ekstrak Pesan

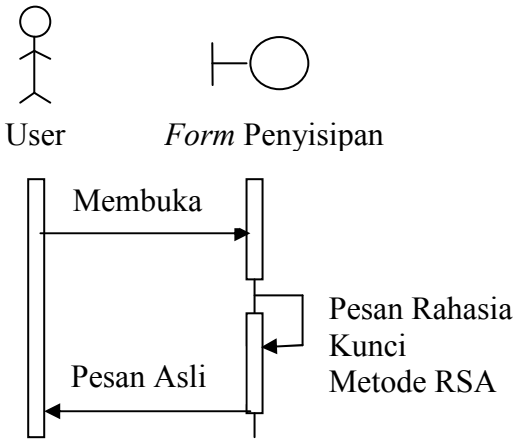
Serangkaian kerja melakukan ekstrak pesan dapat terlihat seperti pada gambar III.22 berikut :



Gambar III.22. *Sequence Diagram* Ekstrak Pesan

10. *Sequence Diagram* Pesan Asli

Serangkaian kerja melakukan pembacaan pesan asli dapat terlihat seperti pada gambar III.23 berikut :



Gambar III.23. *Sequence Diagram* Pesan Asli

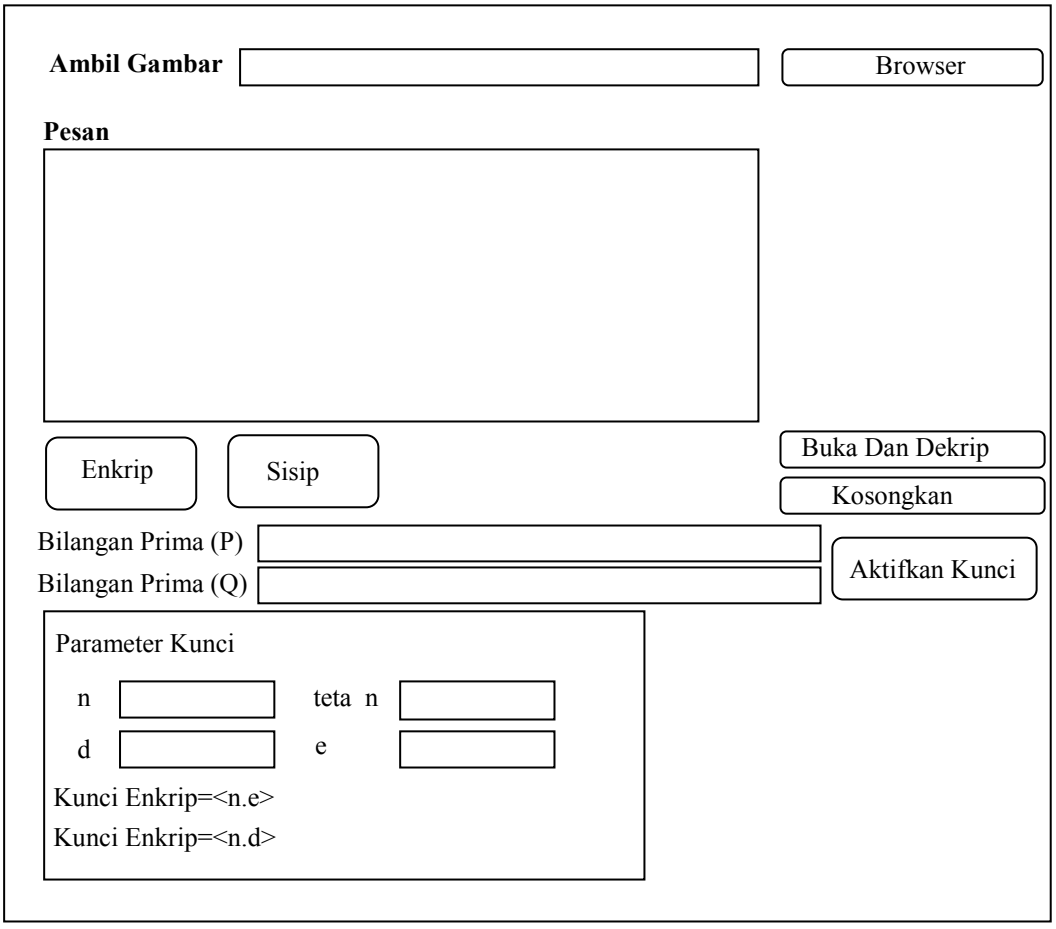
III.4. Desain Sistem Secara Detail

III.4.1. Desain *Input*

Perancangan *Input* merupakan masukan yang penulis rancang guna lebih memudahkan dalam *entry* data. *Entry* data yang dirancang akan lebih mudah dan cepat dan meminimalisir kesalahan penulisan dan memudahkan perubahan. Perancangan tampilan Sistem Kombinasi Keamanan Pesan Menggunakan Algoritma RSA Dan Metode LSB+2 :

1. Perancangan *Form* Enkrip dan Sisip

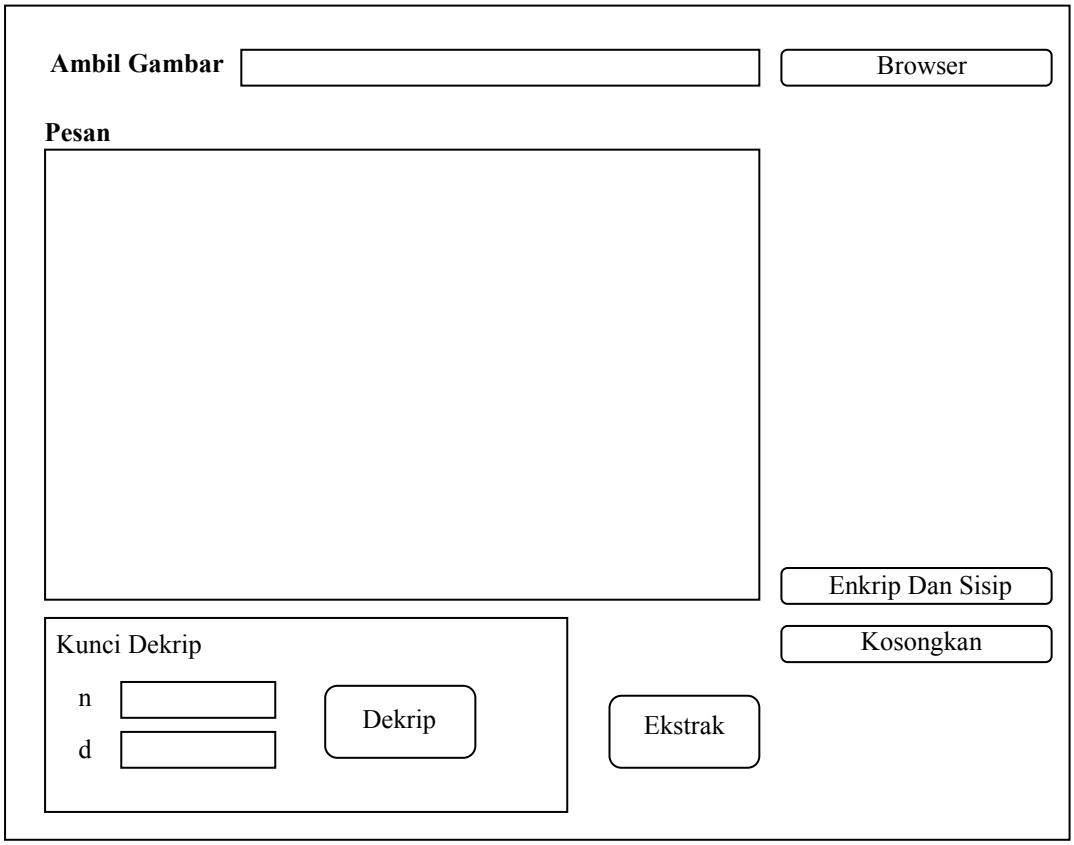
Perancangan *Form* Penyisipan berfungsi untuk menyisipkan pesan teks ke dalam gambar. Adapun rancangan *form* penyisipan dapat dilihat pada gambar III.24. sebagai berikut :



Gambar III.24. Rancangan *Form* Enkrip dan Sisip

2. Perancangan *Form* Buka dan Dekrip

Perancangan *Form* Ekstraksi berfungsi untuk mengekstraksi pesan teks di dalam gambar. Adapun rancangan *form* ekstraksi dapat dilihat pada gambar III.25. sebagai berikut :



Gambar III.25. Rancangan *Form* Buka dan Dekrip