BAB III

ANALISA DAN DESAIN SISTEM

III.1. Analisis Masalah

Umumnya para pencuri mengambil pesan penting berupa teks yang terdapat pada sebuah data. Dengan mendapatkan informasi dari pesan yang telah dicuri, maka hal ini dapat menguntungkan pihak pencuri pesan dan merugikan pihak pemilik pesan. Masalah keamanan ini sering kali kurang mendapat perhatian dari para pemilik pesan. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting. Apabila menggangu performansi dari sistem, sering kali keamanan dikurangi atau ditiadakan. Oleh karena itu dibutuhkan sebuah cara agar pesan yang bersifat penting dapat terlindungi dari tindakan pencurian. Cara yang dapat digunakan yaitu merahasiakan isi pesan dengan merubah pesan asli menjadi pesan rahasia dan cara isi dapat menggunakan kriptografi. Kriptografi adalah merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi.Namun peneliti ingin memperkuat kerahasiaan sebuah pesan dengan menyisipkan pesan yang telah dirahasiakan ke dalam sebuah gambar dengan teknik steganografi, sehingga kerahasiaan pesan menjadi lebih baik. Namun untuk menggunakan teknik kriptografi dibutuhkan sebuah metode yang baik dalam penyandian pesan. Peneliti menggunakan metode *Vernam Cipher* dan ROT13 untuk merahasiakan pesan. Dengan menggunakan teknik kriptografi menggunakan metode ROT13 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) untuk menyisipkan sebuah pesan ke dalam gambar. Dengan adanya teknik kriptografi dan steganografi maka pesan yang bersifat penting mendapat keamanan yang baik.

III.2. Penerapan Metode

Metode yang digunakan untuk mendapatkan pesan rahasia serta penyisipan pesan rahasia kedalam gambar adalah algoritma *Vernam Cipher*, ROT13 dan LSB.

III.2.1. Algoritma *Vernam Cipher*

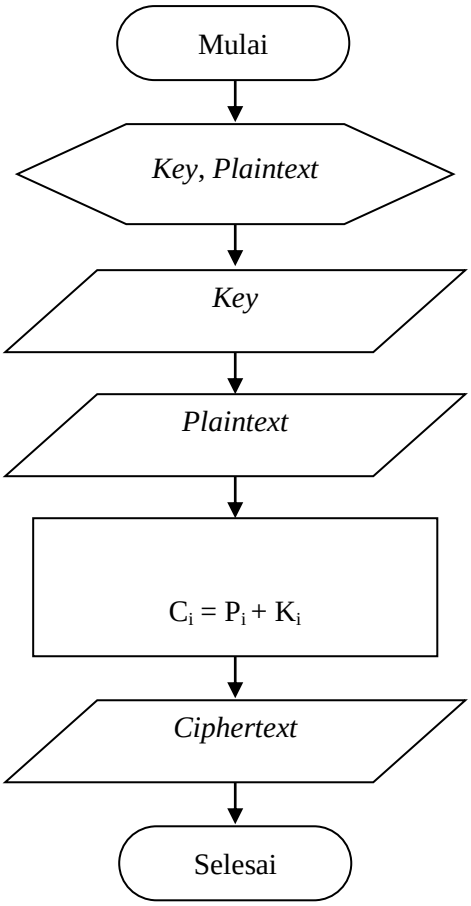
Langkah-langkah algoritma *Vernam Cipher* untuk enkrip dapat dilihat sebagai berikut :

1. Buat pesan untuk di enkrip.
2. Buat kunci untuk proses enkrip.
3. Ubah pesan perkarakter dan kunci ke dalam kode *ascii.*
4. Lakukan proses xor *ascii* kunci terhadap *ascii* pesan pada tiap karakter.

Berikut ini adalah rumus enkrip dari algoritma *Vernam Cipher*:

Ci = (Pi xor Ki) mod 256

Flowchart algoritma enkrip *Vernam Cipher* dapat dilihat sebagai berikut :



Gambar III.1. Flowchart Enkrip Algoritma *Vernam Cipher*

Langkah-langkah algoritma *Vernam Cipher* untuk dekrip dapat dilihat sebagai berikut :

1. Masukkan *ciphertext* untuk di dekrip.
2. Masukkan kunci dekrip *ciphertext.*
3. Ubah pesan perkarakter dan kunci ke dalam kode *ascii.*
4. Lakukan proses xor *ascii* pesan terhadap *ascii* kunci pada tiap karakter.

Contoh Proses Enkrip :

Plaintext : POTENSI

Kunci : 1234567

Solusi :

Ascii Plaintext :

P = 80

O = 79

T = 84

E = 69

N = 78

S = 83

I = 73

Key :

1 = 49

2 = 50

3 = 51

4 = 52

5 = 53

6 = 54

7 = 55

C1 = (P xor k1) mod 256

= (80 xor 49) mod 256

= 97 mod 256

= 97

C2 = (O xor k2) mod 256

= (79 xor 50) mod 256

= 125 mod 256

= 125

C3 = (T xor k3) mod 256

= (84 xor 51) mod 256

= 103 mod 256

= 103

C4 = (E xor k4) mod 256

= (69 xor 52) mod 256

= 113 mod 256

= 113

C5 = (N xor k5) mod 256

= (78 xor 53) mod 256

= 103 mod 256

= 123

C6 = (S xor k6) mod 256

= (83 xor 54) mod 256

= 101 mod 256

= 101

C7 = (I xor k7) mod 256

= (73 xor 55) mod 256

= 126 mod 256

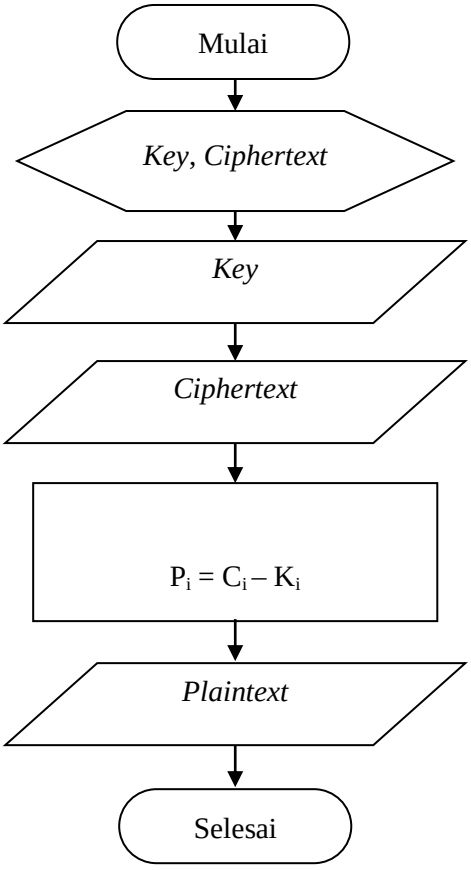
= 126

Chipertext : a}gq{e~

Berikut ini adalah rumus dekrip dari algoritma *Vernam Cipher*:

Pi = (Ci xor Ki) mod 256

Flowchart algoritma dekrip *Vernam Cipher* dapat dilihat sebagai berikut :



Gambar III.2. *Flowchart* Dekrip Algoritma *Vernam Cipher*

Contoh proses dekrip :

Chipertext : a}gq{e~

Kunci : 1234567

Ascii Plaintext :

a = 97

} = 125

g = 103

q = 113

{ = 123

e = 101

~ = 126

Key :

1 = 49

2 = 50

3 = 51

4 = 52

5 = 53

6 = 54

7 = 55

P1 = (C1 xor k1) mod 256

= (97 xor 49) mod 256

= 80 mod 256

= 80 = P

P2 = (C2 xor k2) mod 256

= (125 xor 50) mod 256

= 79 mod 256

= 79 = O

P3 = (C3 xor k3) mod 256

= (103 xor 51) mod 256

= 84 mod 256

= 84 = T

P4 = (C4 xor k4) mod 256

= (113 xor 52) mod 256

= 69 mod 256

= 69 = E

P5 = (C5 xor k5) mod 256

= (123 xor 53) mod 256

= 78 mod 256

= 78 = N

P6 = (C6 xor k6) mod 256

= (101 xor 54) mod 256

= 83 mod 256= 83 = S

P7 = (C7 xor k7) mod 256

= (126 xor 55) mod 256

= 73 mod 256= 73 = I

Plaintext : POTENSI

III.2.2. Algoritma ROT13

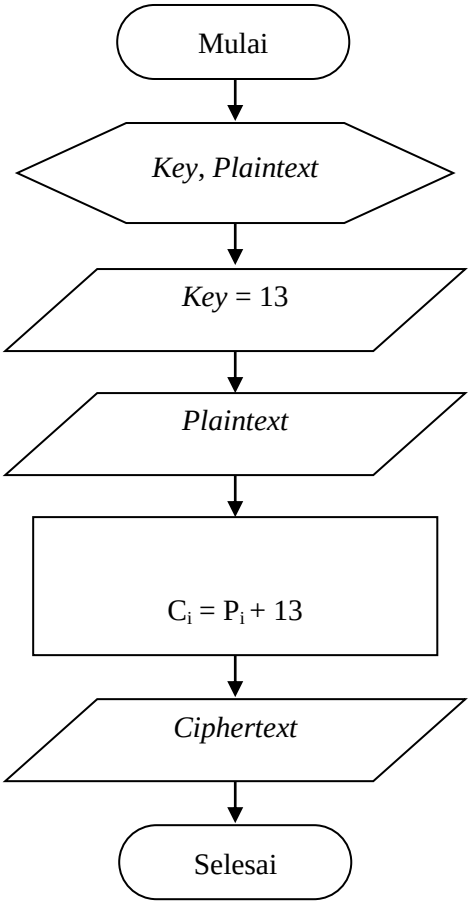
Langkah-langkah algoritma ROT13 untuk enkrip dapat dilihat sebagai berikut :

1. Buat pesan untuk di enkrip.
2. Ubah pesan perkarakter dan kunci ke dalam kode *ascii.*
3. Lakukan proses penjumlahan 13 terhadap *ascii* pesan pada tiap karakter.

Berikut ini adalah rumus enkrip dari algoritma ROT13:

Ci = Pi + 13

Flowchart algoritma enkrip ROT13 dapat dilihat sebagai berikut :



Gambar III.3. *Flowchart* Enkrip Algoritma ROT13

Contoh :

Plaintext (hasil enkripsi algoritma *vernam cipher*) : a}gq{e~

Ascii Plaintext :

a = 97

} = 125

g = 103

q = 113

{ = 123

e = 101

~ = 126

C1 = P1+13 = a+13 = 97+13 = 110 = n

C2 = P2+13 = }+13 = 123+13 = 136 = ˆ

C3 = P3+13 = g+13 = 103+13 = 116 = t

C4 = P4+13 = q+13 = 113+13 = 126= ~

C5 = P5+13 = {+13 = 123+13 = 136= ˆ

C6 = P6+13 = e+13 = 101+13 = 114= r

C7 = P7+13 = ~+13 = 126+13 = 139 = ‹

Ciphertext : n^t~^r‹

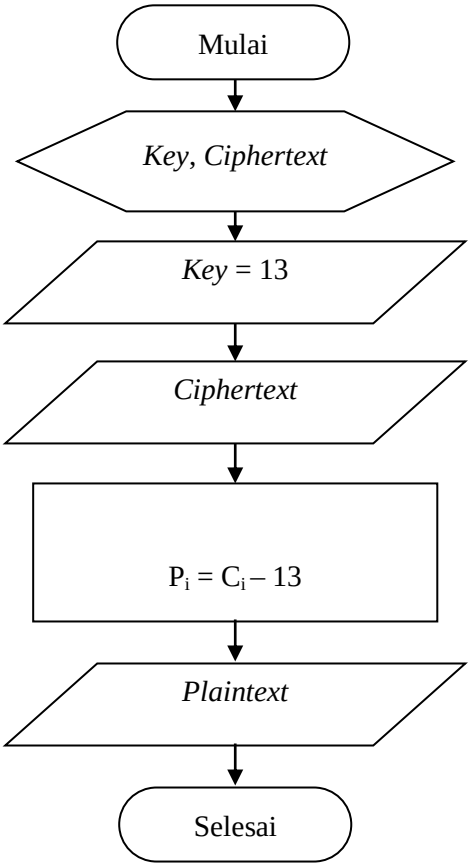
Langkah-langkah algoritma ROT13 untuk dekrip dapat dilihat sebagai berikut :

1. Masukkan *ciphertext* untuk di dekrip.
2. Ubah pesan perkarakter dan kunci ke dalam kode *ascii.*
3. Lakukan proses pengurangan 13 terhadap *ascii* pesan pada tiap karakter.

Berikut ini adalah rumus dekrip dari algoritma ROT13 :

Pi = Ci - 13

Flowchart algoritma dekrip ROT13 dapat dilihat sebagai berikut :



Gambar III.4. *Flowchart* Dekrip Algoritma ROT13

Contoh :

Ciphertext : n^t~^r‹

Ascii Plaintext :

n = 110

^ = 136

t = 116

~ = 126

ˆ = 136

r = 114

‹= 139

P1 = C1-13 = n-13 = 110-13 = 97 = a

P2 = C2-13 = ^-13 = 136-13 = 123 = }

P3 = C3-13 = t-13 = 116-13 = 103 = g

P4 = C4-13 = ~-13 = 126-13 = 113= q

P5 = C5-13 = ^-13 = 136-13 = 123= {

P6 = C6-13 = r-13 = 114-13 = 101= e

P7 = C7-13 = ‹-13 = 139-13 = 126 = ~

Plaintext : a}gq{e~

III.2.3. Algoritma LSB

Langkah-langkah algoritma LSB 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 :

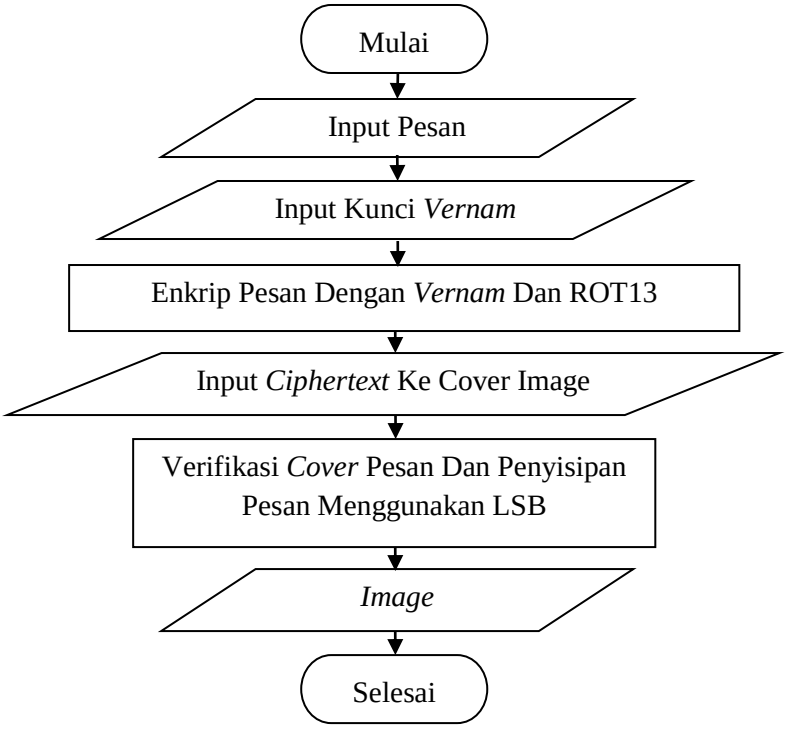
   a. *Input stego image*.

   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. (Farid, dkk, 2016 : 110).

Flowchart penyisipan metode *Least Significant Bit*:



Gambar III.5. Penyisipan *Ciphertext Vernam*, ROT13Dengan LSB

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 :

Plaintext (hasil enkrip algoritma *vernam cipher* dan ROT13) : n^t~^r‹

Ascii Plaintext :

n = 110 = 01101110

ˆ = 136 = 1000 1000

t = 116 = 0111 0100

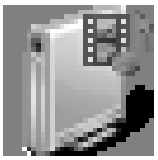
~ = 126 = 0111 1110

ˆ = 136 = 1000 1000

r = 114 = 0111 0010

‹= 139 = 1000 1011

Misalkan matriks tingkat derajat keabuan citra sebagai berikut :



Gambar III.5. File Gambar Asli Format .BMP

Tabel III.1. RGB Matriks Gambar Awal

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

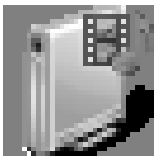
Kemudian pesan n = 01101110 disisipkan sehingga menjadi :

Tabel III.2. RGB Matriks Gambar Akhir

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

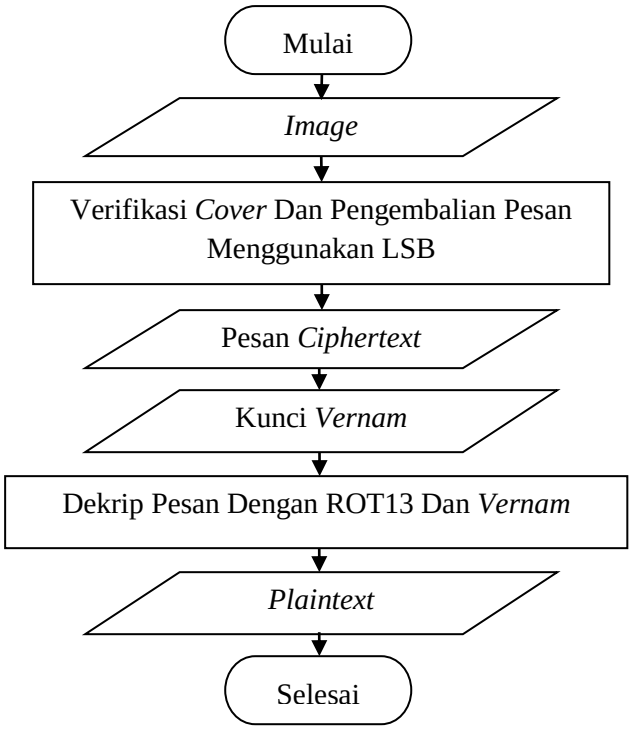
Keterangan :

Lakukan sampai seluruh pesan tersisip. Angka yang digaris bawahi adalah pesan yang disisipkan menggunakan metode LSB. Nilai-nilai tersebut disimpan kembali ke dalam gambar sehingga menghasilkan gambar sebagai berikut :



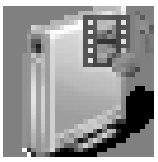
Gambar III.6. File Gambar Format .BMP Tersisip Pesan

Flowchart pengembalian pesan algoritma *Least Significant Bit*:



Gambar III.6. Pengembalian *Plaintext* ROT13, *Vernam* Dengan LSB

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. Gambar tersisip pesan adalah sebagai berikut :



Gambar III.7. File Gambar Format .BMP Tersisip Pesan

Kemudian ekstrak pesan tersebut dari nilai biner gambar RGB :

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 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

Lakukan seterusnya hingga menghasilkan seluruh pesan yang tersisip menjadi sebuah kalimat.

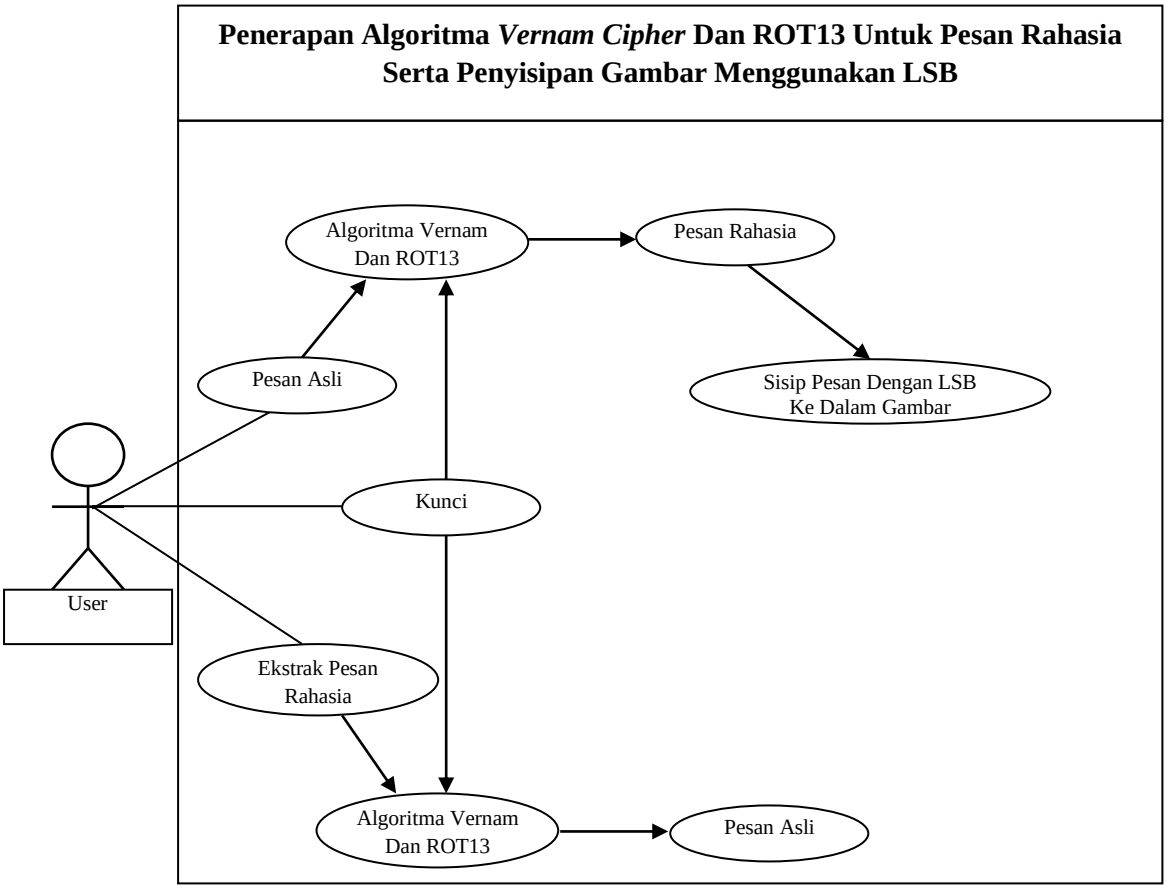
III.3. Desain Sistem

III.3.1. Desain Sistem Secara Global

Bentuk desain 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 CaseDiagram* dapat dilihat pada gambar III.8.



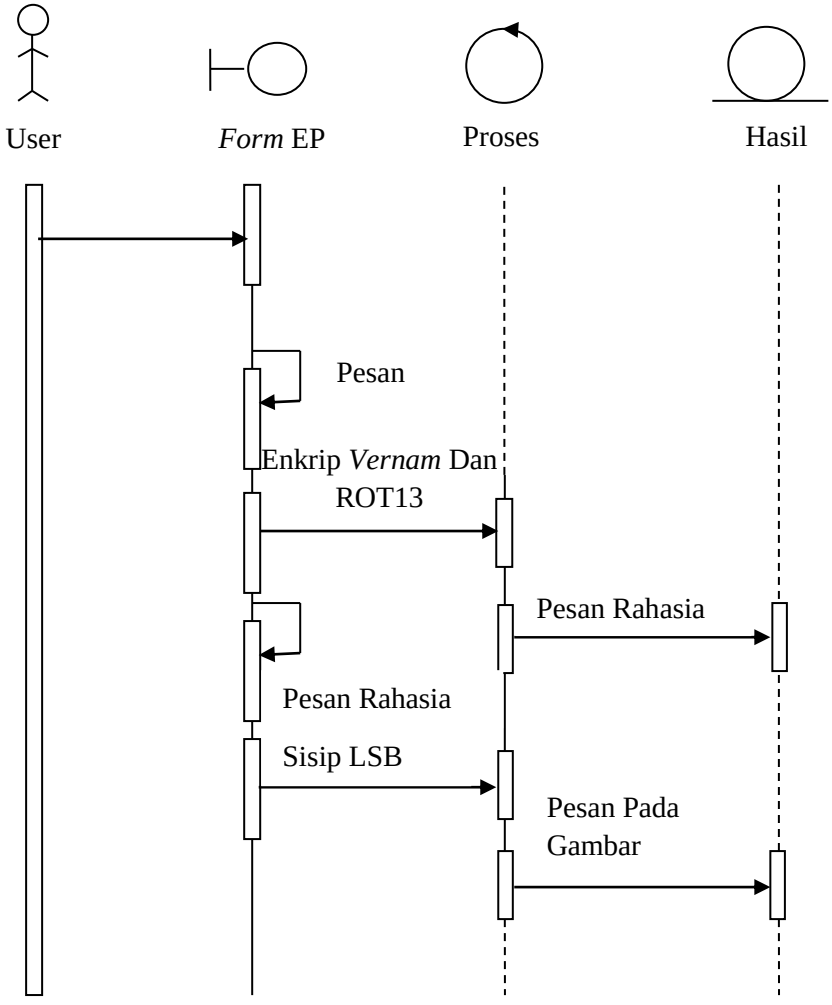
Gambar III.8. *Use Case* Aplikasi Penerapan Algoritma *Vernam Cipher* Dan ROT13 Untuk Pesan Rahasia Serta Penyisipan Gambar Menggunakan LSB

III.3.1.2. *Sequence Diagram*

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

1. *Sequence Diagram* Enkrip Dan Penyisipan

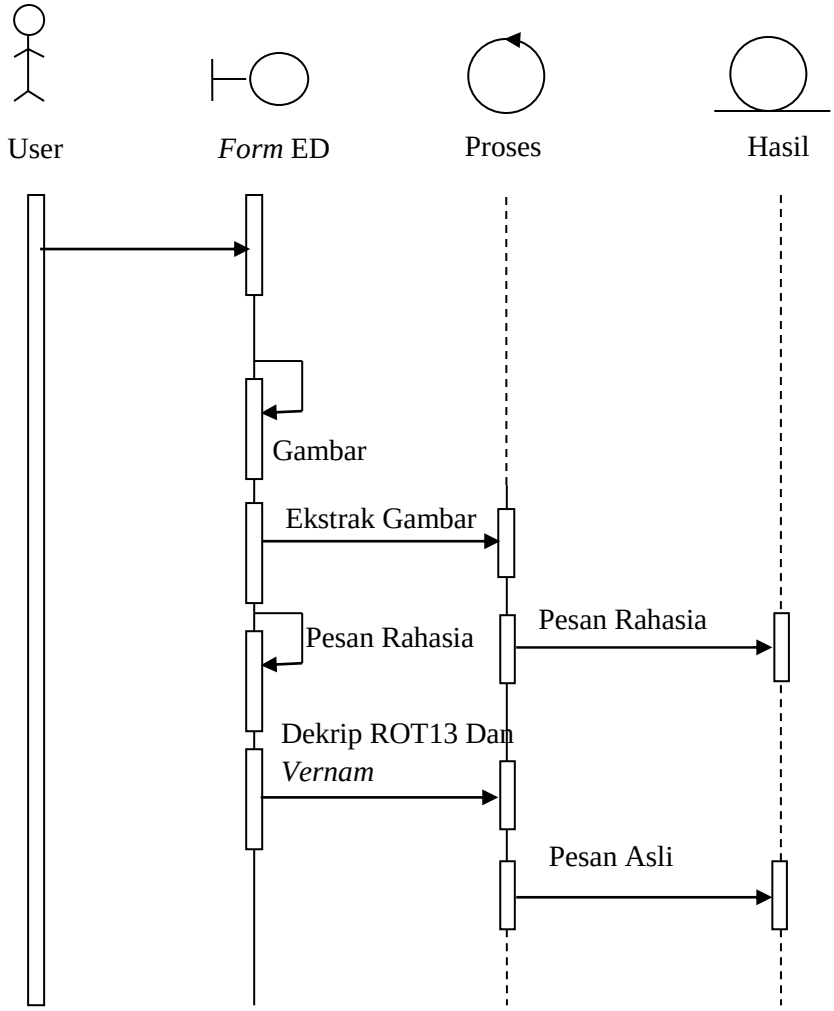
Serangkaian kerja melakukan enkrip dan penyisipan pesan dapat terlihat seperti pada gambar III.9 berikut :



Gambar III.9. *SequenceDiagram* Enkrip Dan Penyisipan

2. *Sequence Diagram* Ekstrak Dan Dekrip

Serangkaian kerja melakukan ekstraksi dan dekrip pesan dapat terlihat seperti pada gambar III.10 berikut :



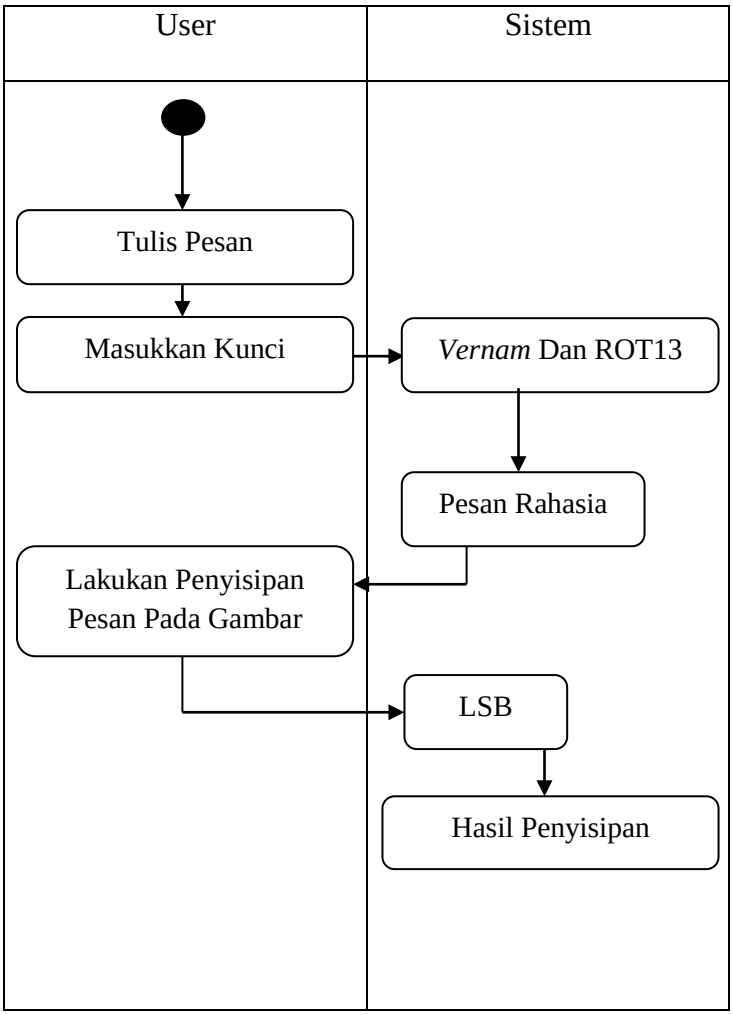
Gambar III.10. *SequenceDiagram* Ekstrak

III.3.1.3. *ActivityDiagram*

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

1. *Activity* Diagram Enkrip Dan Penyisipan

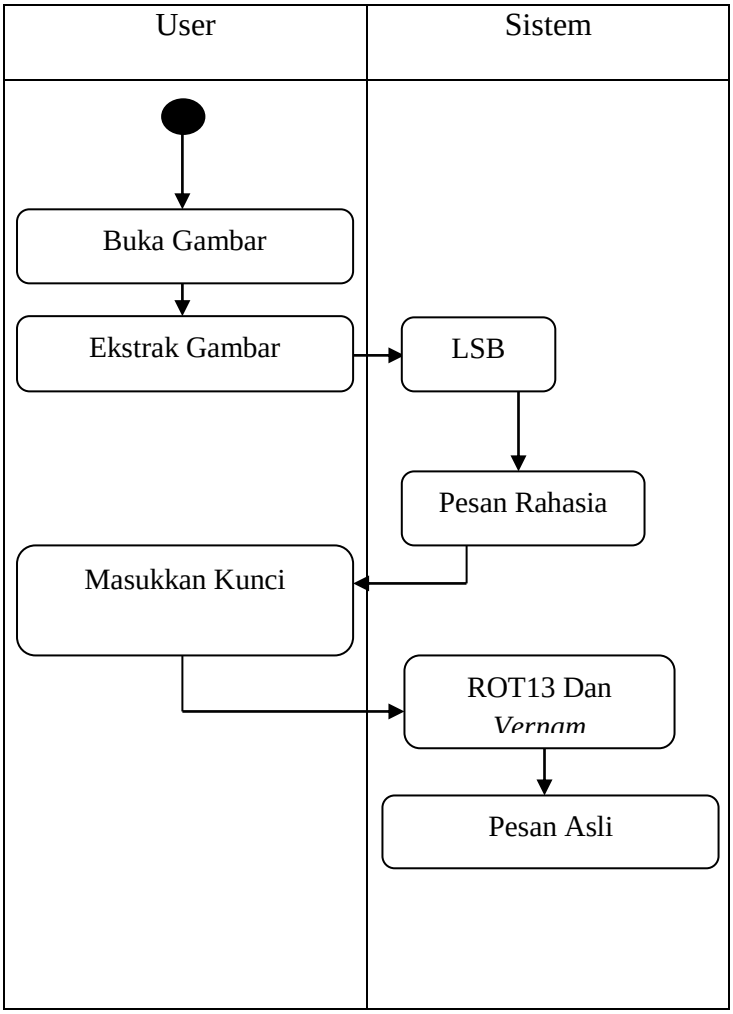
Aktivitas yang dilakukan untuk melakukan enkrip dan penyisipan dapat dilihat seperti pada gambar III.11 berikut :



Gambar III.11. *Activity Diagram* Enkrip Dan Penyisipan

2. *Activity* Diagram Ekstrak Dan Dekrip

Aktivitas yang dilakukan untuk melakukan ekstrak dan dekrip pesan dapat dilihat seperti pada gambar III.12berikut :



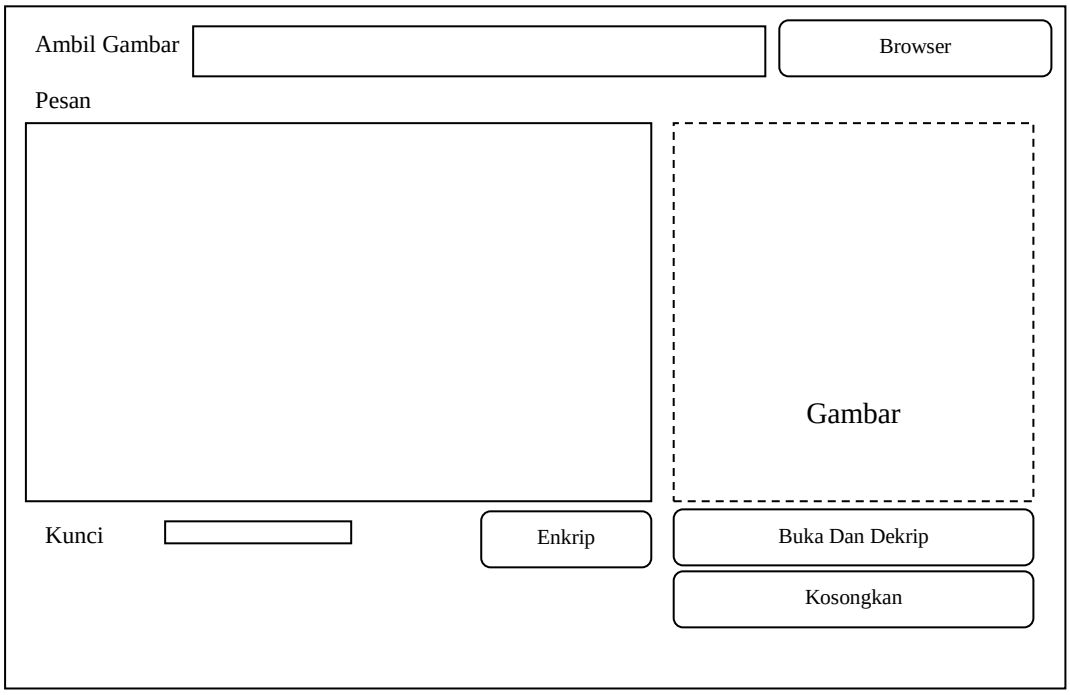
Gambar III.12. *Activity Diagram* Ekstrak Dan Dekrip

III.4. Desain Sistem Tampilan

Desain sistem tampilan Aplikasi Penerapan Algoritma *Vernam Cipher* Dan ROT13 Untuk Pesan Rahasia Serta Penyisipan Gambar Menggunakan LSB adalah sebagai berikut :

1. Perancangan *Form* Enkrip Dan Penyisipan

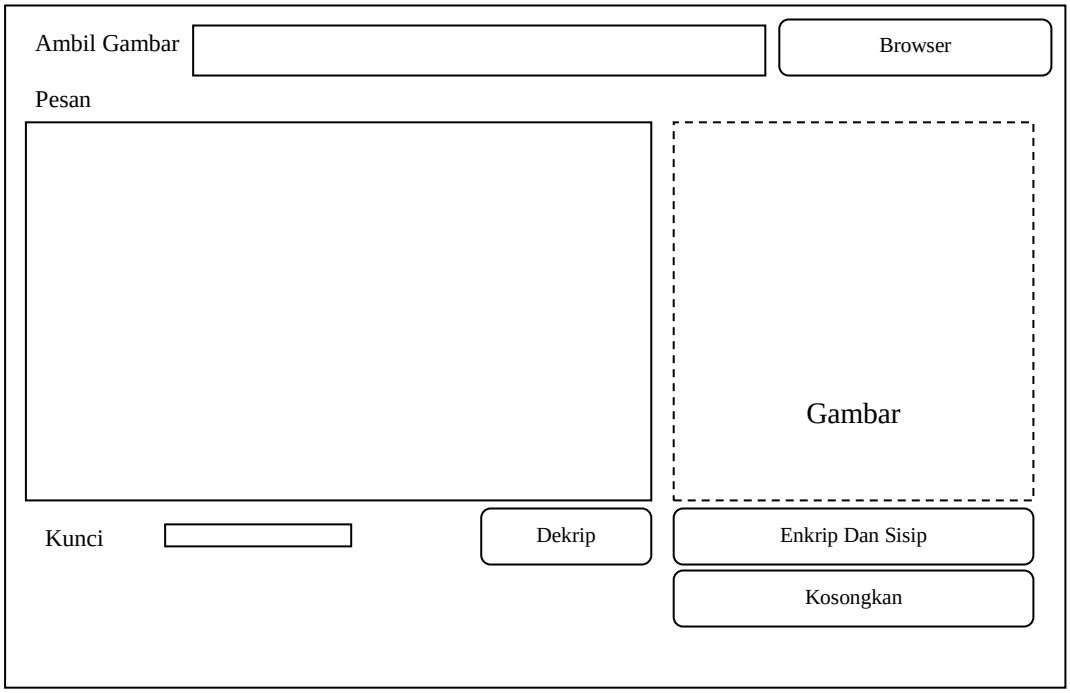
Perancangan *Form* Enkrip Dan Penyisipan berfungsi untuk merahasiakan dan menyisipkan pesan teks ke dalam gambar. Adapun rancangan *form* enkrip dan penyisipan dapat dilihat pada gambar III.13. sebagai berikut :



Gambar III.13. Rancangan *Form* Enkrip Dan Penyisipan

2. Perancangan *Form* Ekstrak Dan Dekrip

Perancangan *Form* Ekstrak Dan Dekrip berfungsi untuk mengekstraksi pesan teks di dalam gambar. Adapun rancangan *form* ekstrak dan dekrip dapat dilihat pada gambar III.14. sebagai berikut :



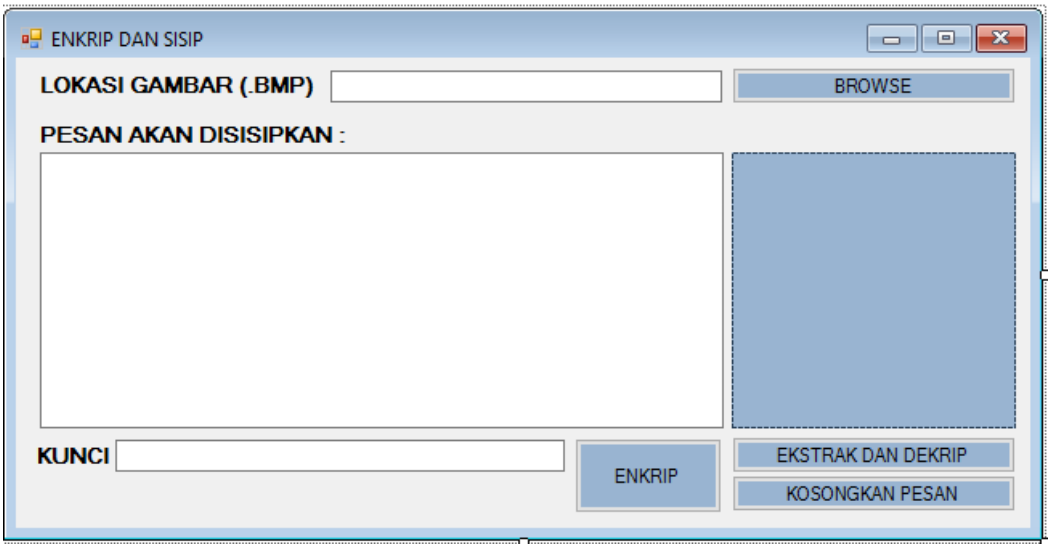
Gambar III.14. Rancangan *Form* Ekstrak Dan Dekrip

BAB IV

HASIL DAN PEMBAHASAN

IV.1. Hasil

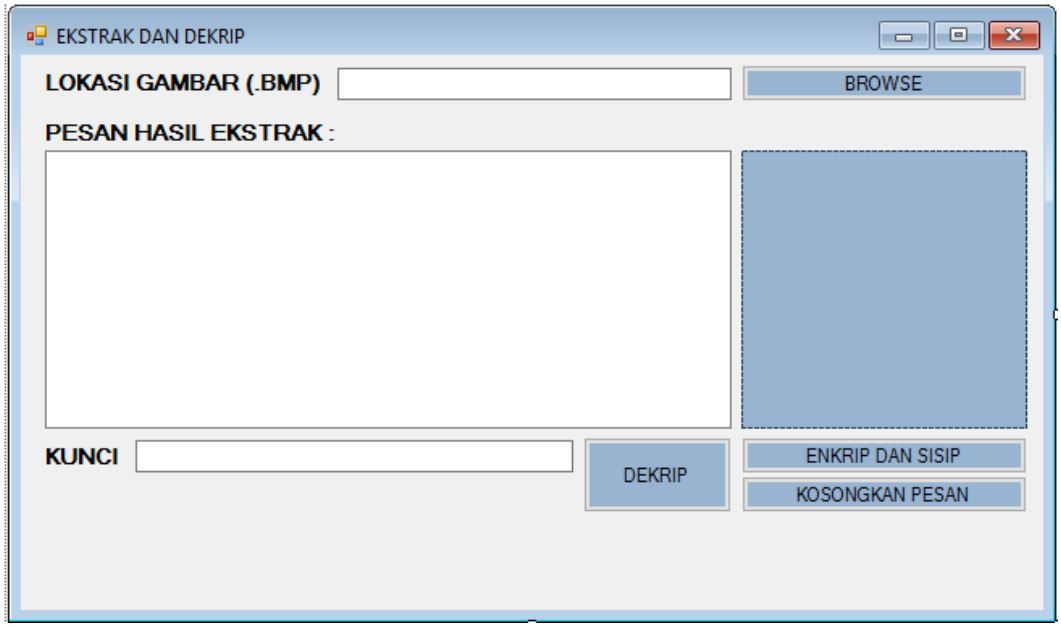
Aplikasi penerapan algoritma *vernam cipher* dan ROT13 untuk pesan rahasia serta penyisipan gambar menggunakan LSB dibuat menggunakan *Visual Basic* 2010, dimulai dengan merancang sistem, menerapkan perhitungan metode dan menerapkan komponen sebenarnya ke dalam *Visual Basic* 2010 sehingga aplikasi dapat diselesaikan sesuai dengan perancangan. Berikut adalah beberapa tampilan yang dihasilkan dari pembuatan aplikasi yang telah dilakukan :



Gambar IV.1. Tampilan *Form* Enkrip

Gambar IV.1 merupakan *form* enkrip dan sisip, untuk merahasiakan pesan yang akan dienkrip kemudian disisipkan ke dalam gambar maka pengguna terlebih dahulu harus mengupload gambar yang akan digunakan dengan menekan tombol browse, setelah gambar berhasil diupload kemudian pengguna harus memasukkan kunci enkrip. Setelah semua kunci dimasukkan kemudian pengguna harus menulis

sebuah pesan di dalam kotak isi pesan dan mengklik tombol enkrip maka aplikasi akan menampilkan jendela untuk pencarian lokasi penyimpanan gambar yang telah disisipkan pesan yang telah dienkripsi. Jika pengguna ingin mengosongkan pesan maka pengguna harus mengklik tombol kosongkan pesan dan jika pengguna ingin mengekstrak kembali pesan yang ada di dalam gambar maka pengguna harus mengklik tombol ekstrak dan dekrip kemudian sistem akan menampilkan *form* ekstrak dan dekrip seperti pada Gambar IV.2.



Gambar IV.2. Tampilan *Form* Ekstrak Dan Dekrip

Gambar IV.2 merupakan *form* ekstrak dan dekrip, penggunaan aplikasi dimulai dari pengambilan gambar pada tombol browse, kemudian pengguna harus menginputkan kunci dan tekan tombol dekrip. Untuk membuka kembali dan menyisipkan pesan ke dalam gambar maka pengguna harus mengklik tombol Enkrip Dan Sisip.

IV.2. Pembahasan

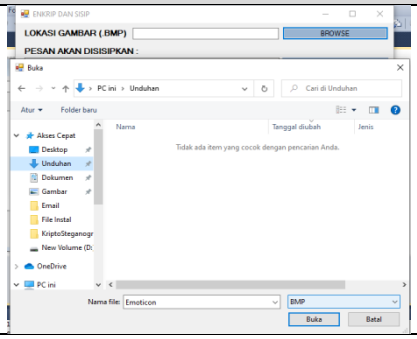
Pembahasan mengenai perangkat keras dan perangkat lunak dalam pembuatan aplikasi penerapan algoritma *vernam cipher* dan ROT13 untuk pesan rahasia serta penyisipan gambar menggunakan LSB dijabarkan sebagai berikut :

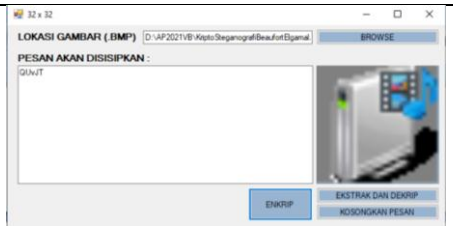
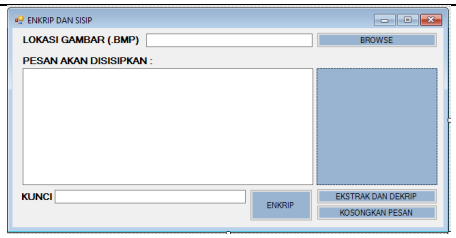
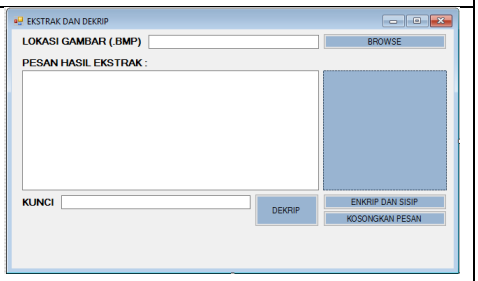
1. Perangkat keraslaptop dengan spesifikasi sebagai berikut :
   a. *Processor* Minimal*Core 2 duo*
   b. RAMminimal 1 Gb
   c. *Hardisk* minimal 80 Gb
2. Perangkat Lunak dengan spesifikasi sebagai berikut :
   a. Sistem Operasi *Windows*
   b. *Microsoft Visual Basic* 2010

IV.2.1Uji Coba

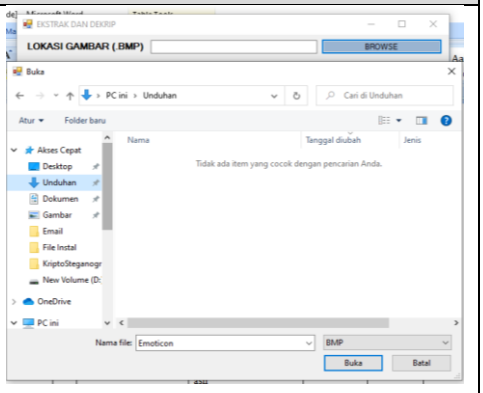
Uji coba pada aplikasi penerapan algoritma *vernam cipher* dan ROT13 untuk pesan rahasia serta penyisipan gambar menggunakan LSByaitu dengan menggunakan *blackbox testing* sebagai berikut :

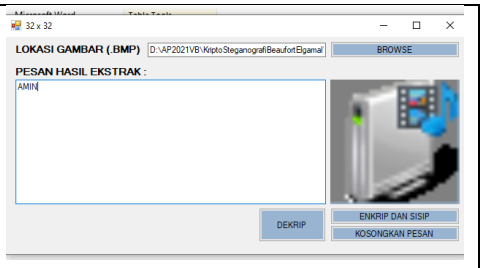
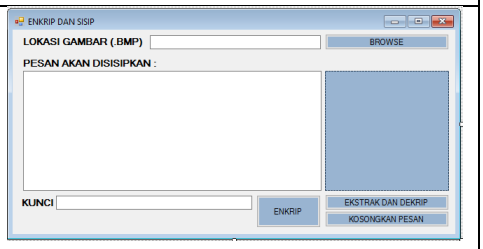
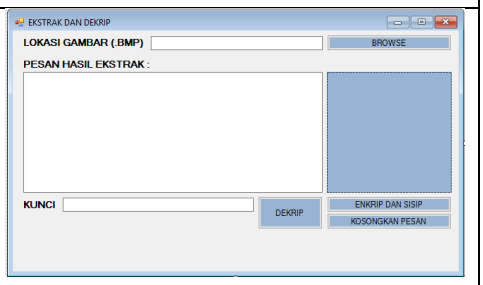
Tabel. IV.1.*Blackbox Testing*Enkrip

No	FormEnkrip Dan Sisip	Keterangan	Hasil	Tampilan Aplikasi
1	Klik tombol browse	Aplikasi membuka jendela sistem	Benar	

2	Klik tombol enkrip	Aplikasi mengubah pesan asli menjadi pesan rahasia	Benar	
3	Klik tombol kosongkan pesan	Aplikasi menghapus pesan di dalam kotak pesan	Benar	
4	Klik tombol buka dan dekrip	Aplikasi membuka *form* ekstrak dan dekrip	Benar	

Tabel. IV.2.*Blackbox Testing*Ekstrak Dan Dekrip

No	FormBuka Dan Dekrip	Keterangan	Hasil	Tampilan Aplikasi
1	Klik tombol browse	Aplikasi membuka jendela sistem	Benar	

2	Klik tombol dekrip	Aplikasi akan mengubah pesan rahasia menjadi pesan asli	Benar	
3	Klik tombol enkrip dan sisip	Aplikasi akan membuka *form* enkrip dan sisip	Benar	
4	Klik tombol kosongkan pesan	Aplikasi membuka *form* ekstrak dan dekrip	Benar	

IV.2.2.Hasil Uji Coba

Berikut ini adalah kesimpulan dari hasil uji coba terhadap sistem yang telah dibuat :

1. Seluruh tools pada aplikasi telah berfungsi dengan baik.
2. Upload gambar ke dalam aplikasi berfungsi dengan baik.
3. Proses enkripsi dan dekripsi pesan berfungsi dengan baik.
4. Proses penyisipan pesan ke dalam gambar telah berjalan dengan baik.
5. Proses ekstraksi pesan dari dalam gambar telah berjalan dengan baik.
6. Perhitungan metode pada aplikasi telah sesuai dengan perhitungan teori.

7. Proses aplikasi tidak memiliki keterlambatan (*bug*).

IV.3. Kelebihan dan Kekurangan Sistem

Setiap sistem memiliki kelebihan dan kekurangan, berikut ini adalah kelebihan dan kekurangan sistem yang telah dibuat.

IV.3.1. Kelebihan Sistem

Adapun kelebihan sistem yang telah dibuat diantaranya yaitu :

1. Sistem yang dibuat menggunakan tiga algoritma sehingga meningkatkan kekuatan kerahasiaan data.
2. Sistem yang dibuat menggunakan dua teknik kerahasiaan data yaitu kriptografi dan steganografi sehingga meningkatkan kerahasiaan data.
3. Sistem yang dibuat tidak merusak gambar walaupun disisipkan pesan rahasia.

IV.3.2. Kekurangan Sistem

Adapun kekurangan sistem yang telah dibuat diantaranya yaitu :

1. Sistem yang dibuat tidak menggunakan kriptografi jenis asimetris.
2. Sistem yang dibuat tidak memiliki petunjuk penggunaan.
3. Sistem yang dibuat tidak dapat menyisipkan pesan pada semua jenis gambar.