Implementasi Sistem Keamanan Data dengan Menggunakan Teknik Steganografi *End of File* (EOF) dan Rabin *Public Key Cryptosystem*

Henny Wandani[1], Muhammad Andri Budiman, S.T, M.Comp.Sc, MEM[2], Amer Sharif. S.Si, M.Kom[3]

Program Studi S1 Ilmu Komputer, Universitas Sumatera Utara

Jalan Universitas No. 9 Kampus USU Medan 20155

[1] wandanihenny@gmail.com

[2]mandrib@gmail.com

[3]amersharifdjamin@yahoo.com

***Abstrak*— Semakin meningkatnya perkembangan komunikasi data membuat semakin pentingnya aspek keamanan dan kerahasiaan data. Kriptografi merupakan suatu seni atau ilmu menjaga keamanan data atau pesan yang bersifat mengacak suatu data atau pesan. Sedangkan steganografi adalah ilmu menyembunyikan pesan atau data ke dalam suatu media. Kedua teknik tersebut dapat digabungkan dan akan menghasilkan suatu sistem keamanan data yang tinggi. Pada penelitian ini, terlebih dahulu dilakukan proses enkripsi data atau pesan rahasia yang berupa data teks angka dengan jumlah maksimum yang dimasukkan adalah 24 digit angka, kemudian hasil enkripsi (*ciphertext*) akan disembunyikan ke dalam suatu *file* gambar yang berformat bitmap dengan ukuran minimum 25x25. Selanjutnya, dilakukan proses ekstraksi dan dekripsi *ciphertext*, sehingga diperoleh kembali *plaintext* yang berupa data teks angka. Algoritma kriptografi yang digunakan adalah algoritma Rabin *Public Key* dan teknik steganografi yang digunakan adalah metode *End of File*.**

Kata kunci— Keamanan Data, *End of File*, Enkripsi, Dekripsi, Rabin *Public Key*.

I. PENDAHULUAN

Seiring perkembangan teknologi, teknik dan metode penyampaian pesan rahasia pun semakin beragam. Terdapat berbagai bentuk pesan rahasia seperti pesan teks, pesan citra, pesan *audio* dan pesan *video* yang umum digunakan. Pengamanan pesan teks dapat dilakukan dengan berbagai macam teknik kriptografi. Salah satunya adalah pengamanan pesan teks menggunakan kriptografi kunci asimetris. Kriptografi kunci asimetris terdiri dari dua kunci, yaitu kunci publik dan kunci privat. Dalam kriptografi kunci asimetris, kunci publik berfungsi untuk mengenkripsi suatu pesan dan kunci privat berfungsi untuk mendekripsi suatu pesan. Sehingga tingkat keamanan suatu pesan lebih baik dibandingkan menggunakan kriptografi kunci simetris yang hanya memiliki satu kunci privat saja.

Terdapat berbagai macam metode kriptografi kunci asimetris yang telah digunakan. Salah satunya adalah algoritma Rabin *Public Key*. Algoritma Rabin *Public Key* diperkenalkan oleh Michael O. Rabin pada tahun 1979. Algoritma Rabin menggunakan pemfaktoran bilangan untuk melakukan pengamanan. Metode pemfaktoran bilangan secara cepat sampai saat ini belum terpecahkan. Selain itu, Rabin *Public Key* ini akan menghasilkan empat kemungkinan hasil pendekripsian yang mengharuskan si penerima pesan menentukan hasil dekripsi yang benar.

Namun, teknik kriptografi yang sifatnya mengacak suatu pesan rahasia menimbulkan kecurigaan. Sehingga muncullah teknik steganografi yang merupakan pengembangan dari kriptografi. Steganografi ialah penyembunyian pesan dalam sebuah media penyimpanan dan bersifat tidak mengacak isi *file*. Sehingga, *file* yang disisipkan tidak mencurigakan. Saat ini telah ada beberapa metode steganografi yang umum digunakan. Salah satunya adalah metode *End of File* (EOF). Pada metode *End of File* ini, pesan disisipkan pada akhir nilai *file*.

Berdasarkan latar belakang yang telah penulis uraikan, maka dilakukan penelitian dengan judul "Implementasi Sistem Keamanan Data dengan Menggunakan Teknik Steganografi *End of File* (EOF) dan Rabin *Public Key Cryptosystem*."

A. Rumusan Masalah

Yang menjadi rumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan suatu sistem keamanan data yang mampu melakukan proses enkripsi dan dekripsi suatu data teks dengan menggunakan algoritma Rabin *Public Key*, kemudian pesan yang telah dienkripsi tersebut disisipkan ke dalam suatu *file* gambar berformat bitmap dengan menggunakan metode *End of File*.

B. Batasan Masalah

Untuk fokusnya penelitian ini, penulis memberi batasan sebagai berikut :

1. Algoritma kriptografi yang digunakan adalah Rabin *Public Key*.
2. Metode steganografi yang digunakan adalah *End of File* (EOF).
3. Data yang digunakan adalah data teks dan *file* bitmap.
4. Hanya membahas enkripsi dengan angka.
5. Bahasa pemrograman yang digunakan adalah Matlab 7.5.0 (R2007b).

C. Tujuan Penelitian

Tujuan dari penelitian adalah sebagai berikut:

1. Memperoleh aplikasi yang menggabungkan algoritma kriptografi Rabin *Public Key* dan teknik steganografi *End of File*.
2. Mengetahui kelebihan dan kekurangan algoritma Rabin *Public Key* dan metode *End of File*.
3. Mengetahui proses enkripsi dan dekripsi pesan teks dengan menggunakan algoritma Rabin *Public Key*.
4. Mengetahui proses penyisipan pesan dan pengekstrakan pesan pada suatu *file* berformat bitmap dengan menggunakan metode *End of File*.

D. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Manfaat bagi penulis :
   a. Menambah pengetahuan penulis dalam melakukan proses enkripsi dan dekripsi suatu pesan teks dengan menggunakan algoritma Rabin *Public Key*.
   b. Menambah pengetahuan penulis dalam melakukan proses penyisipan dan pengekstrakan suatu pesan rahasia pada *file* citra dengan menggunakan metode *End of File* (EOF).

2. Manfaat bagi bidang ilmu :
   a. Menambah pengetahuan tentang kelebihan dan kekurangan algoritma Rabin *Public Key* dan metode *End of File* (EOF).
   b. Sebagai bahan referensi bagi peneliti lain yang ingin merancang aplikasi kriptografi dan steganografi sejenis.

3. Manfaat bagi masyarakat adalah membantu masyarakat dalam mengamankan pesan teks angka.

E. Metodologi Penelitian

Dalam penelitian ini, ada beberapa tahapan yang akan dilakukan adalah sebagai berikut:

1) Studi Literatur: Pada tahap ini akan dilakukan pengumpulan bahan referensi yang terkait dengan Rabin *Public Key Cryptosystem* dan metode *End of File* yang dapat berupa buku-buku, artikel-artikel atau *e-book* serta jurnal nasional dan internasional yang didapatkan melalui internet.

2) Analisis Data: Pada tahap ini dilakukan pengolahan data yang didapat dan kemudian dilakukan analisis terhadap hasil studi literatur yang diperoleh sehingga menjadi suatu informasi.

3) Perancangan Desain Sistem: Pada tahap ini akan dilakukan perancangan desain antarmuka sistem dan struktur proses kerja sistem.

4) Implementasi Sistem: Pada tahap ini akan dilakukan implementasi dari hasil perancangan desain sistem dalam bentuk perangkat lunak.

5) Pengujian Sistem: Pada tahap ini akan dilakukan pengujian sistem yang bertujuan untuk mengetahui kesalahan-kesalahan yang terjadi pada sistem, sehingga dapat dilakukan perbaikan. Kemudian dilakukan analisis pada sistem untuk mengetahui apakah sistem sesuai dengan permasalahan dari penelitian.

6) Penyusunan Laporan: Pada tahap ini dilakukan penyusunan laporan dari hasil analisis dan perancangan sistem dalam format penulisan penelitian.

II. TINJAUAN TEORETIS

Kriptografi berasal dari bahasa Yunani, yaitu "*cryptos*" yang berarti rahasia dan "*graphein*" yang berarti tulisan. Jadi, kriptografi adalah tulisan rahasia. Namun, menurut Kurniawan dalam bukunya yang berjudul "*Kriptografi Keamanan Internet dan Jaringan Komunikasi"*, menjelaskan bahwa kriptografi merupakan seni dan ilmu untuk menjaga keamanan pesan.

A. Algoritma Asimetris

Algoritma kriptografi asimetris merupakan algoritma kriptografi yang kunci enkripsi dan kunci dekripsinya berbeda. Algoritma asimetris disebut juga dengan algoritma kunci publik karena kunci enkripsi yang digunakan bersifat publik atau boleh diketahui semua orang. Pada algoritma ini, kunci yang digunakan untuk mengenkripsi pesan disebut dengan kunci publik. Sedangkan kunci yang digunakan untuk mendekripsi pesan disebut dengan kunci privat. Kunci privat bersifat rahasia atau tidak boleh diketahui orang lain.

B. Algoritma Rabin Public Key

Algoritma Rabin *Public Key* pertama kali diperkenalkan pada tahun 1979 oleh Michael O. Rabin. Algoritma Rabin *Public Key* adalah salah satu sistem kriptografi asimetris yang menggunakan kunci publik dan kunci privat. Algoritma Rabin *Public Key* merupakan varian algoritma Rivest Shamir Adleman (RSA). Fungsi dasar algoritmanya mirip dengan fungsi dasar dari algoritma RSA. Hanya saja komputasinya lebih sederhana dibandingkan algoritma RSA.

C. Proses Enkripsi dan Dekripsi Algoritma Rabin Public Key

a. Proses Pembangkitan Kunci

Pada algoritma Rabin *Public Key*, proses pembangkitan kuncinya dilakukan sebagai berikut :

1. Pilih 2 (dua) buah bilangan prima besar sembarang yang saling berbeda (*p* dan *q*), dimana $p \equiv q \equiv 3$ (*mod* 4). Atau dengan kata lain jika *p* dan *q* di modulo 4 akan menghasilkan 3.
2. Hitung nilai *n* yang merupakan kunci publik dengan rumus sebagai berikut:
   $$n = p * q$$
   dengan *p* dan *q* adalah kunci privat.

Untuk mengenkripsi pesan hanya dibutuhkan kunci publik *n*, sedangkan untuk dekripsi, dibutuhkan bilangan *p* dan *q* sebagai kunci privat.

b. Proses Enkripsi

Proses enkripsi pada algoritma Rabin *Public Key* menggunakan kunci publik *n*. Pada proses dekripsi menggunakan Algoritma Rabin *Public Key* akan menghasilkan 4 (empat) buah kemungkinan *plaintext*. Oleh karena itu, diperlukan modifikasi dalam proses enkripsi dan dekripsi untuk menentukan *plaintext* yang sebenarnya. Berikut langkah–langkah proses enkripsi pesan rahasia menggunakan algoritma Rabin *Public Key* yang telah dimodifikasi adalah :

1. Ubah nilai *plaintext m* menjadi nilai biner, kemudian tambahkan dengan nilai biner *m* itu sendiri (*redundant information*) atau dengan kata lain *plainteks* digandakan.
2. Ubah hasil penggandaan nilai biner *plaintext* menjadi nilai desimalnya.
3. Hitung nilai *k* yang merupakan kongruen nilai desimal dari hasil penggandaan *plaintext m* terhadap kunci publik *n* dengan menggunakan rumus :
   $$k = \frac{m - (m \bmod n)}{n}$$
4. Hitung nilai *ciphertext c* dengan menggunakan rumus :
   $$c = m^2 \bmod n$$
   dengan *c* adalah *ciphertext*, *n* adalah kunci publik, dan *m* adalah nilai desimal dari hasil penggandaan nilai biner *plaintext*.

c. Metode dekripsi

Proses enkripsi pada algoritma Rabin *Public Key* menggunakan kunci privat *p* dan *q*. Berikut langkah–langkah proses dekripsi dengan menggunakan algoritma Rabin *Public Key* yang telah dimodifikasi:

1. Tentukan nilai *Yp* dan *Yq* yang merupakan pembagi GCD (*Greatest Common Divisor*) dari *p* dan *q* dengan menggunakan Algoritma *Extended Euclidean*. Karena GCD bilangan prima adalah 1, maka dapat ditulis sebagai berikut :
   $$Yp*p + Yq * q = 1$$
2. Hitunglah nilai akar kuadrat dari *ciphertext* terhadap *p* dan *q* dengan rumus:
   $$m_p = c^{\left(\frac{p+1}{4}\right)} \bmod p$$
   $$m_q = c^{\left(\frac{q+1}{4}\right)} \bmod q$$
   dengan m_p adalah akar kuadrat dari *ciphertext* terhadap *p* dan m_q adalah akar kuadrat dari *ciphertext* terhadap *q*.
3. Hitung nilai *r, s, t* dan *u* dengan menggunakan *Chinese Remainder Theorem*, dengan persamaan berikut :
   $$r = (Yp*p* m_q + Yq * q* m_p) \bmod n$$
   $$s = (Yp*p* m_q - Yq * q* m_p) \bmod n$$
   $$t = (-Yp*p* m_q + Yq * q* m_p) \bmod n$$
   $$u = (-Yp*p* m_q - Yq * q* m_p) \bmod n$$
4. Tambahkan *r,s,t,u* dengan kongruen nilai desimal hasil penggandaan *plainteks k* yang dikalikan dengan kunci publik *n*.
   $$R = (k*n)+r$$
   $$S = (k*n)+s$$
   $$T = (k*n)+t$$
   $$U = (k*n)+u$$
5. Ubahlah nilai desimal *R,S,T,U* ke dalam bentuk biner. Kemudian nilai biner *R,S,T,U* dibagi menjadi 2 (dua) bagian. Bandingkan kedua bagian tersebut. Jika kedua bagian tersebut menghasilkan bentuk biner yang sama, maka didapatlah hasil dekripsi *ciphertext c* dengan mengubah bentuk biner salah satu bagian yang telah dibagi menjadi 2(dua) bagian yang sama.

D. Steganografi

Kata steganografi berasal dari bahasa Yunani, yaitu dari kata steganos (tersembunyi atau terselubung) dan graphien (tulisan) yang berarti tulisan tersembunyi. Secara umum steganografi merupakan seni atau ilmu yang digunakan untuk menyembunyikan pesan rahasia dengan segala cara sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari pesan rahasia tersebut.

Metode *End of File* (EOF) merupakan salah satu metode yang digunakan dalam steganografi. Teknik ini menggunakana cara dengan menyisipkan data pada akhir *file*. Sehingga, tidak akan mengganggu kualitas data awal yang akan disisipkan pesan. Namun, ukuran *file* setelah disisipkan pesan rahasia akan bertambah. Sebab, ukuran *file* yang telah disisipkan pesan rahasia sama dengan ukuran *file* sebelum disisipkan pesan rahasia ditambah dengan ukuran pesan rahasia yang disisipkan. Untuk mengenal data yang disisipkan pada akhir *file*, diperlukan suatu tanda pengenal atau simbol pada awal dan akhir data yang akan disisipkan.

III. ANALISIS DAN PERANCANGAN

Pada penelitian ini digunakan algoritma Rabin *Public Key* untuk melakukan proses enkripsi dan dekripsi pesan. Hasil enkripsi pesan yang berupa *ciphertext*, akan disisipkan ke dalam suatu *file* citra dengan menggunakan metode *End of File* (EOF). Selanjutnya, *file* citra yang telah disisipkan *ciphertext* tersebut akan diekstraksi dan hasilnya akan didekripsi.

A. Analisis Algoritma Rabin Public Key

a. Pembangkitan Kunci

Tahapan – tahapan yang dilakukan dalam proses pembangkitan kunci pada algoritma Rabin *Public Key* adalah :

1. Pilih 2 (dua) buah bilangan prima besar sembarang yang saling berbeda *p* dan *q*, dimana $p \equiv q \equiv 3$ (*mod* 4). Atau dengan kata lain jika *p* dan *q* di modulo 4 akan menghasilkan 3.
   Contoh :
   $p = 11$ dan $q = 23$
2. Hitung nilai *n* yang merupakan kunci publik dengan rumus sebagai berikut:
   $$n = p * q$$
   dengan *p* dan *q* adalah kunci privat.
   Contoh :
   $n = p*q$
   $= 11*23$
   $n = 253$

b. Proses Enkripsi

Proses enkripsi pada algoritma Rabin *Public Key* menggunakan kunci publik *n*. Berikut langkah-langkah proses enkripsi menggunakan algoritma Rabin *Public Key* yang telah dimodifikasi :

1. Ubah nilai *plaintext m* menjadi nilai biner, kemudian nilai biner *plaintext m* digabungkan dengan nilai biner *plaintext m* itu sendiri (*redundant information*) atau dengan kata lain *plainteks* digandakan. Penggabungan ini bertujuan untuk dapat menentukan *plaintext* yang sebenarnya dari keempat hasil dekripsi yang diperoleh.
   Contoh :
   $m = 8$, nilai binernya adalah 1000.
   Maka nilai $m_{[m\ m]} = 10001000$.
2. Ubah hasil penggandaan nilai biner *plaintext* menjadi nilai desimalnya.
   Contoh :
   $m_{[m\ m]} = 10001000$, nilai desimalnya adalah 136.
   Maka nilai $m = 136$.

3. Hitung nilai *k* yang merupakan kongruen nilai desimal dari hasil penggandaan *plaintext m* terhadap kunci publik *n* dengan menggunakan rumus :
   $$k = \frac{m - (m \bmod n)}{n}$$
   Contoh :
   $$k = \frac{m - (m \bmod n)}{n}$$
   $$k = \frac{136 - (136 \bmod 253)}{253}$$
   $k = 0$

4. Hitung nilai *ciphertext c* dengan menggunakan rumus:
   $$c = m^2 \bmod n$$
   dengan *c* adalah *ciphertext*, *n* adalah kunci publik, dan *m* adalah nilai desimal dari hasil penggandaan nilai biner *plaintext*.
   Contoh :
   $m = 136$ dan $n = 253$, maka *c* adalah :
   $c = m^2 \bmod n$
   $= 136^2 \bmod 253$
   $c = 27$
   Maka, *plaintext* = 8 dienkripsi dengan nilai $c = 27$.

Untuk kombinasi *plaintext* dan kunci yang merupakan angka kelipatan "11" akan diperoleh *ciphertext* yang juga merupakan angka kelipatan "11". Hal ini menyebabkan kriptanalis dapat mengetahui bentuk *plaintext* yang sebenarnya atau dengan kata lain algoritma Rabin *Public Key* tidak aman untuk serangan *chosen-ciphertext attack*. Tabel 1 akan menunjukkan hasil enkripsi angka kelipatan "11" dengan rentang angka "11-99" dan kunci publik yang digunakan adalah "77".

TABEL 1
HASIL ENKRIPSI ANGKA KELIPATAN "11" DENGAN KUNCI PUBLIK "77"

Plaintext	Ciphertext
11	11
22	11
33	44
44	44
55	11
66	11
77	0
88	11
99	44

c. Proses Dekripsi

Berikut langkah–langkah proses dekripsi dengan menggunakan algoritma ini yang telah dimodifikasi :

1. Tentukan nilai *Yp* dan *Yq* yang merupakan pembagi GCD (*Greatest Common Divisor*) dari *p* dan *q* dengan menggunakan Algoritma *Extended Euclidean*. Karena GCD bilangan prima adalah 1, maka dapat ditulis sebagai berikut :
   $$Yp*p + Yq * q = 1$$
   Contoh *:*
   $Yp*p + Yq * q = 1$
   $11\ Yp + 23\ Yq = 1$

TABEL 2

PROSES MENENTUKAN NILAI *x* DAN *y* DENGAN MENGGUNAKAN ALGORITMA *EXTENDED EUCLIDEAN*

Hasil Bagi	Sisa Bagi	Subsitusi	Penggabungan
-	11	-	11*1+23*0=11
-	23	-	11*0+23*1=23
0	11	(11*1+23*0) –(11*0 + 23*1)*0 = 11	11*1+23*0=11
2	1	(11*0 + 23*1) – (11*1 + 23*0)*2 = 1	11*(-2)+23*1=1
11	0	*Karena sisa bagi mencapai 0, maka proses berakhir*	

Hasil akhir yang diperoleh adalah 11*(-2) + 23*1=1, sehingga didapat nilai x = -2 dan y = 1.

2. Hitunglah nilai akar kuadrat dari *ciphertext* terhadap *p* dan *q* dengan rumus:

$$m_p = c^{\left(\frac{p+1}{4}\right)} \bmod p$$

$$m_q = c^{\left(\frac{q+1}{4}\right)} \bmod q$$

dengan m_p adalah akar kuadrat dari *ciphertext* terhadap *p* dan m_q adalah akar kuadrat dari *ciphertext* terhadap *q*.

Contoh :

$m_p = c^{\left(\frac{p+1}{4}\right)} \bmod p$

$= 27^{\left(\frac{11+1}{4}\right)} \bmod 11$

$= 27^3 \bmod 11$

$m_p = 4$

$m_q = c^{\left(\frac{q+1}{4}\right)} \bmod q$

$= 27^{\left(\frac{23+1}{4}\right)} \bmod 23$

$= 27^6 \bmod 23$

$m_q = 2$

Maka, didapatlah m_p =4 dan m_q = 2

3. Hitung nilai *r, s, t* dan *u* dengan menggunakan *Chinese Remainder Theorem.*

$r = (Yp*p* m_q + Yq * q* m_p)$ mod n

$s = (Yp*p* m_q - Yq * q* m_p)$ mod n

$t = (-Yp*p* m_q + Yq * q* m_p)$ mod n

$u = (-Yp*p* m_q - Yq * q* m_p)$ mod n

Contoh :

$r = (Yp*p* m_q + Yq * q* m_p)$ mod n

= ((-2)*11*2 + 1*23*4) mod 253

= 48

$s = (Yp*p* m_q - Yq * q* m_p)$ mod n

= ((-2)*11*2 - 1*23*4) mod 253

= 117

$t = (-Yp*p* m_q + Yq * q* m_p)$ mod n

= (2*11*2 + 1*23*4) mod 253

= 136

$u = (-Yp*p* m_q - Yq * q* m_p)$ mod n

= (2*11*2 + 1*23*4) mod 253

= 205

4. Tambahkan *r,s,t,u* dengan kongruen nilai desimal hasil penggandaan *plainteks k* yang dikalikan dengan kunci publik *n*.

$R = (k*n)+r$

$S = (k*n)+s$

$T = (k*n)+t$

$U = (k*n)+u$

Contoh :

$R = (k*n) + r$

= (0*253) + 48

R = 48

$S = (k*n) + s$

= (0*253) + 117

S = 117

$T = (k*n) + t$

= (0*253) + 136

T = 136

$U = (k*n) + u$

= (0*253) + 205

U = 205

5. Ubahlah nilai desimal *R,S,T,U* ke dalam bentuk biner. Kemudian nilai biner *R,S,T,U* dibagi menjadi 2 (dua) bagian. Bandingkan kedua bagian tersebut. Jika kedua bagian tersebut menghasilkan bentuk biner yang sama, maka didapatlah hasil dekripsi *ciphertext c* dengan mengubah bentuk biner salah satu bagian yang telah dibagi menjadi 2(dua) bagian yang sama.

Contoh :

R = 48, nilai binernya adalah 110000. Jika nilai binernya dibagi menjadi 2 (dua) bagian, maka akan menghasilkan nilai biner 110 dan 000. Karena tidak menghasilkan bentuk biner yang sama, maka *R* bukan pesan rahasia yang sebenarnya.

S = 117, nilai binernya adalah 1110101. Panjang nilai biner yang dihasilkan adalah ganjil, sehingga sudah dapat dipastikan *S* bukan pesan rahasia yang sebenarnya.

T = 136, nilai binernya adalah 10001000. Jika nilai binernya dibagi menjadi 2 (dua) bagian, maka akan menghasilkan nilai biner 1000 dan 1000. Karena menghasilkan bentuk biner yang sama, maka *T* adalah pesan rahasia yang sebenarnya.

U = 205, nilai binernya adalah 11001101. Jika nilai binernya dibagi menjadi 2 (dua) bagian, maka akan menghasilkan nilai biner 1100 dan 1101. Karena tidak menghasilkan bentuk biner yang sama, maka *U* bukan pesan rahasia yang sebenarnya.

Maka, didapatlah dekripsi dari *ciphertext c=27* adalah {48, 117, 136, 205} dengan 136

menghasilkan bentuk biner yang sama jika dibagi menjadi 2 (dua) bagian, yaitu 1000 dan 1000. Dan jika diubah menjadi nilai desimal akan menghasilkan nilai *plaintext* yang sebenarnya yaitu 8.

B. Analisis Metode End of File (EOF)

a. Analisis Proses Embedding

Proses *embedding* atau penyisipan pesan menggunakan metode *End of File* adalah sebagai berikut :

1. Inputkan *ciphertext* yang akan disisipkan.
2. Inputkan citra yang akan menjadi media penyisipan *ciphertext* (*cover image*).
3. Baca nilai setiap *pixel* citra.
4. Tambahkan *ciphertext* sebagai nilai akhir *pixel* citra dengan diberi karakter penanda sebagai penanda akhir *ciphertext*.
5. Petakan menjadi citra baru.

Berikut contoh penyisipan *ciphertext* menggunakan metode *End of File* :

Terdapat suatu citra RGB 8x8 yang memiliki nilai setiap *pixel* seperti pada Gambar 1.

104	38	55	104	96	96	77	92
80	93	60	60	60	51	56	94
91	79	16	62	90	69	73	87
97	98	70	52	60	62	52	99
85	83	37	18	82	88	51	56
87	84	56	65	68	39	106	101
69	37	44	74	80	68	99	99
66	62	60	32	105	88	71	77

Gambar 1 Matriks *pixel* citra RGB

Citra RGB tersebut akan disisipkan *ciphertext* "101 120 97 109 112 108 101". *Ciphertext* akan ditambahkan sebagai nilai akhir pada *pixel* citra RGB. Pada akhir *ciphertext* diberi karakter penanda "ÿ" yang memiliki nilai desimal "255". Maka didapatlah matriks *pixel* seperti pada Gambar 2.

104	38	55	104	96	96	77	92
80	93	60	60	60	51	56	94
91	79	16	62	90	69	73	87
97	98	70	52	60	62	52	99
85	83	37	18	82	88	51	56
87	84	56	65	68	39	106	101
69	37	44	74	80	68	99	99
66	62	60	32	105	88	71	77
101	**120**	**97**	**109**	**112**	**108**	**101**	**255**

Gambar 2 Matriks *pixel* citra RGB yang telah disisipkan *ciphertext*

Dan matriks tersebut akan dipetakan kembali dalam bentuk citra RGB dan citra ini disebut *stego image.*

b. Analisis Proses Extraction

Proses *extraction* atau pengambilan *ciphertext* dari media menggunakan metode *End of File* adalah sebagai berikut :

1. Inputkan citra yang telah disisipkan *ciphertext* (*stego image*).
2. Baca nilai *pixel stego image* yang terdapat pada baris terakhir matriks *pixel* citra.
3. Ambil *ciphertext* yang terdapat pada *stego image*, yaitu nilai *pixel* awal yang terdapat pada baris terakhir matriks *pixel* citra sampai nilai desimal karakter penanda.

Berikut contoh pengambilan *ciphertext* menggunakan metode *End of File*:

Terdapat suatu citra RGB 8x8 yang telah disisipkan *ciphertext* (*stego image*) dengan karakter penanda *ciphertext* adalah "ÿ" yang memiliki nilai desimal "225". Nilai setiap *pixel file* citra RGB tersebut dapat dilihat pada Gambar 3.

104	38	55	104	96	96	77	92
80	93	60	60	60	51	56	94
91	79	16	62	90	69	73	87
97	98	70	52	60	62	52	99
85	83	37	18	82	88	51	56
87	84	56	65	68	39	106	101
69	37	44	74	80	68	99	99
66	62	60	32	105	88	71	77
101	120	97	109	112	108	101	255

Gambar 3 Matriks *pixel* citra RGB yang telah disisipkan *ciphertext*

Kemudian dibaca nilai *pixel stego image* yang terdapat pada baris terakhir matriks *pixel* citra seperti pada Gambar 4.

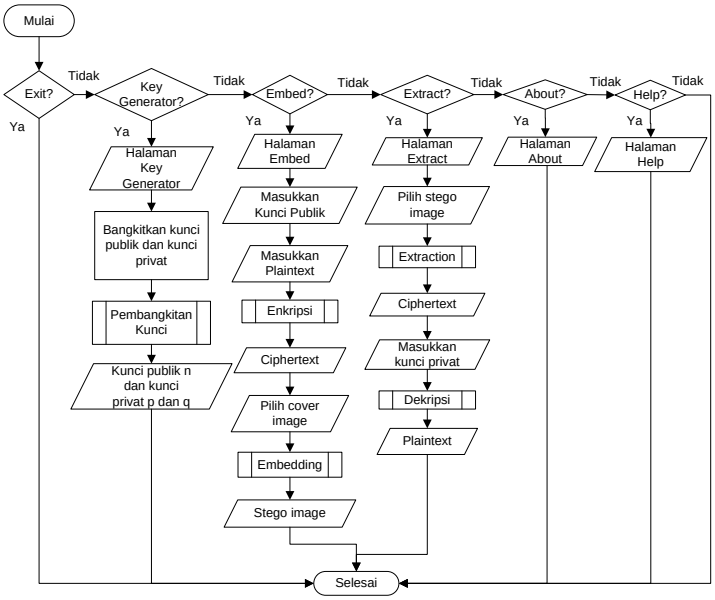
101	120	97	109	112	108	101	255

Gambar 4 Matriks *pixel stego image* yang terdapat pada baris terakhir matriks *pixel* citra

Dengan mengambil nilai *pixel* awal pada baris terakhir matriks *pixel* citra sampai dengan nilai desimal karakter penanda "ÿ" yaitu "255", didapatlah nilai *ciphertext* yaitu "101 120 97 109 112 108 101".

C. Flowchart Gambaran Umum Sistem

Terdapat 5 (lima) proses utama yang terjadi pada sistem ini, yaitu proses pembangkitan kunci, proses enkripsi, proses penyisipan pesan (*embedding*), proses dekripsi dan proses ekstraksi pesan (*extraction*). Keseluruhan proses tersebut dapat dilihat pada *flowchart* gambaran umum sistem pada Gambar 5.



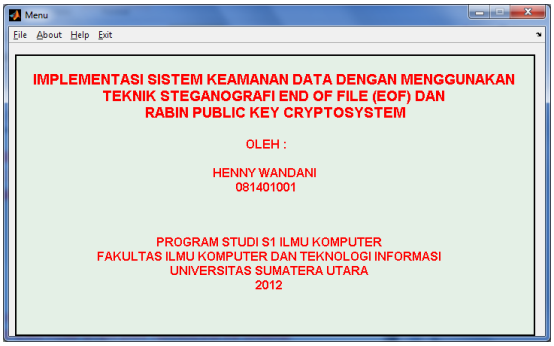
Gambar 5 *Flowchart* gambaran umum sistem

IV. IMPLEMENTASI

Pada sistem ini terdiri dari 5 (lima) buah halaman, yaitu :

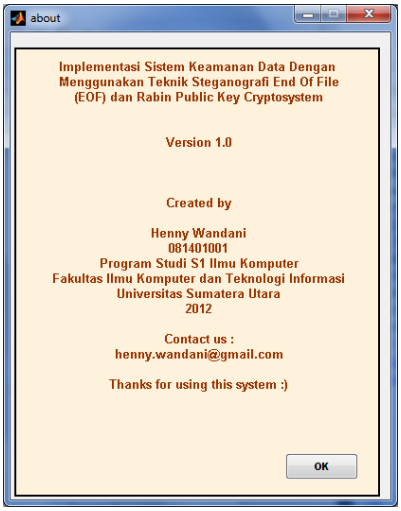
1. Halaman Menu Utama
2. Halaman *Embed/Encode*
3. Halaman *Extract/Decode*
4. Halaman *About*
5. Halaman *Help*

Pada saat sistem ini dijalankan, maka akan muncul halaman Menu Utama yang dapat dilihat pada Gambar 6.

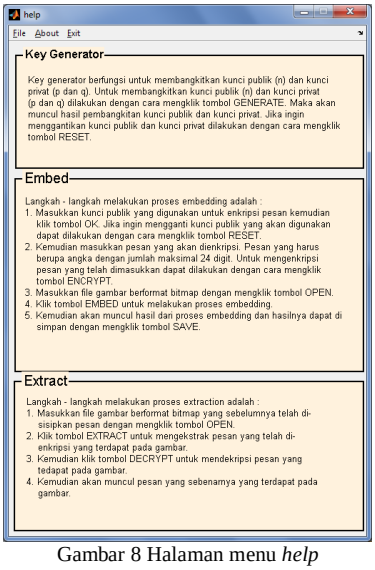


Gambar 6 Halaman menu utama

Halaman menu utama terdiri dari 4 (empat) *menubar*, yaitu menu *File*, *About*, *Help* dan *Exit*. Menu *About* menghubungkan pengguna dengan halaman menu *About* yang dapat dilihat pada Gambar 7 dan menu *Help* menghubungkan pengguna dengan halaman menu *Help* yang dapat dilihat pada Gambar 8. Sedangkan menu *Exit* akan mengeluarkan pengguna dari sistem.



Gambar 7 Halaman menu *about*

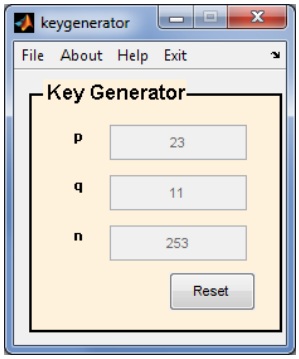


Gambar 8 Halaman menu *help*

Pada menu *File* terdapat 3 (tiga) *submenubar*, yaitu *submenubar Key Generator* yang akan menghubungkan pengguna dengan menu *Key Generator*, *submenubar Embed/Encode* yang akan menghubungkan pengguna dengan menu *Embed/Encode* dan submenu *Extract/Decode* yang akan menghubungkan pengguna dengan menu *Extract/Decode*.

A. Proses Pembangkitan Kunci

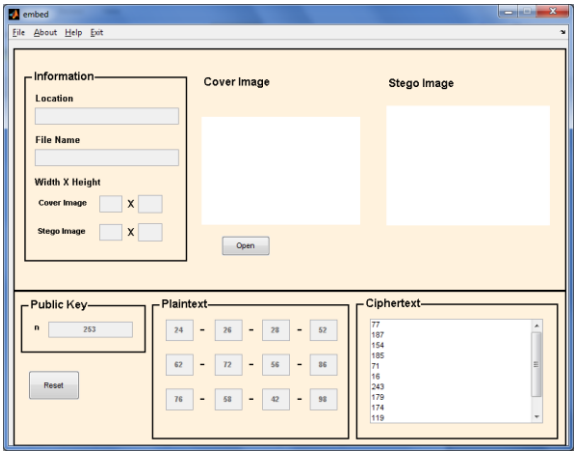
Untuk melakukan proses enkripsi dan dekripsi diperlukan kunci publik dan kunci privat. Sehingga, diperlukan proses pembangkitan kunci dengan cara memilih *menubar File*, kemudian memilih *submenubar Key Generator*. Selanjutnya pengguna dapat mengeksekusi tombol *Generate* untuk membangkitkan kunci. Hasil proses pembangkitan kunci pada sistem dapat dilihat pada Gambar 9.



Gambar 9 Hasil proses pembangkitan kunci

B. Proses Enkripsi

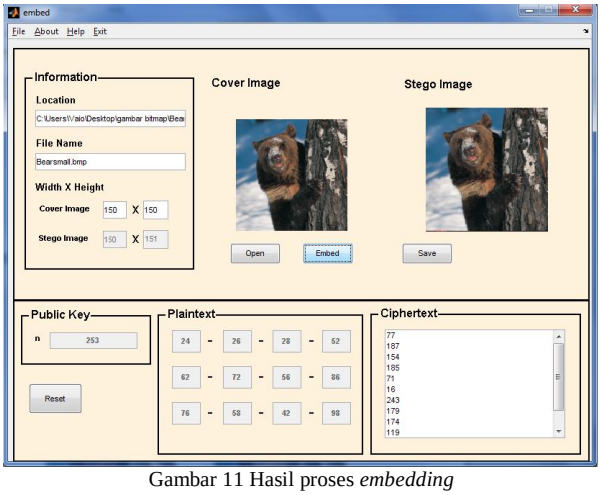
Setelah dilakukan pembangkitan kunci, pengguna dapat melakukan proses enkripsi yang terdapat pada halaman *Embed/Encoding*. Proses enkripsi dapat dilakukan dengan cara memasukkan kunci publik yang telah dibangkitkan dan plainteks yang akan dienkripsi. Selanjutnya, pengguna mengeksekusi tombol *Encrypt* dan sistem akan menampilkan hasil cipherteks hasil enkripsi. Hasil proses enkripsi pada sistem dapat dilihat pada Gambar 10.



Gambar 10 Hasil proses enkripsi

C. Proses Penyisipan Pesan (Embedding)

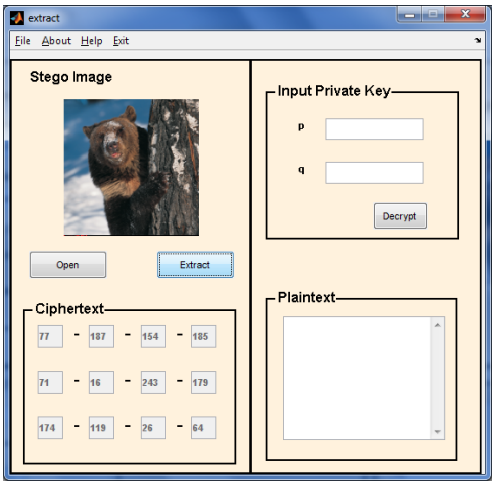
Setelah dilakukan proses enkripsi, cipherteks hasil enkripsi akan disisipkan ke dalam suatu *file* citra berformat bitmap (*embedding*). Untuk melakukan proses *embedding*, pengguna harus memilih *file* citra yang akan digunakan sebagai media penyimpanan pesan (*cover image*) terlebih dahulu dengan cara mengeksekusi tombol *Open*. Kemudian, dilakukan proses *embedding* dengan cara mengeksekusi tombol *Embed*. Sistem akan menampilkan hasil *embedding* yang berupa *stego image*. *Stego image* hasil proses *embedding* ini dapat disimpan dengan cara mengeksekusi tombol *Save*. Hasil proses *embedding* pada sistem dapat dilihat pada Gambar 11.



Gambar 11 Hasil proses *embedding*

D. Proses Ekstraksi

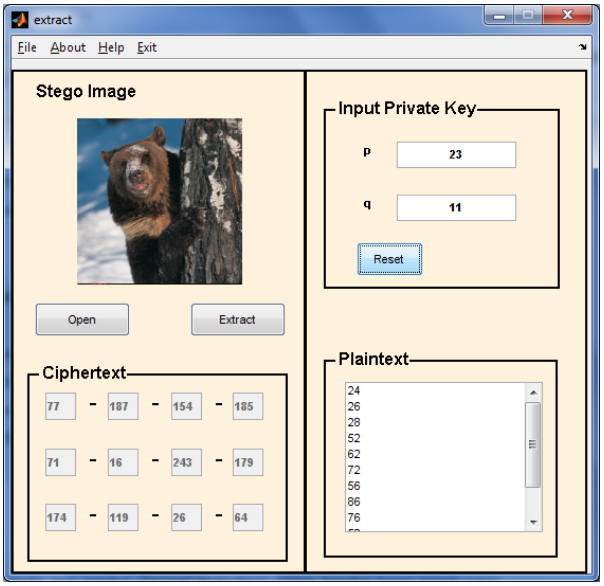
Proses ekstraksi dapat dilakukan pada halaman *Extract/Decoding*. Pertama sekali, pengguna harus memilih *stego image* yang akan diekstraksi dengan cara mengeksekusi tombol *Open*. Setelah *stego image* telah dipilih, pengguna dapat melakukan proses ekstraksi dengan cara mengeksekusi tombol *Extract*. Selanjutnya, sistem akan menampilkan hasil dari proses ekstraksi yang berupa cipherteks. Hasil proses ekstraksi pada sistem dapat dilihat pada Gambar 12.



Gambar 12 Hasil proses ekstraksi

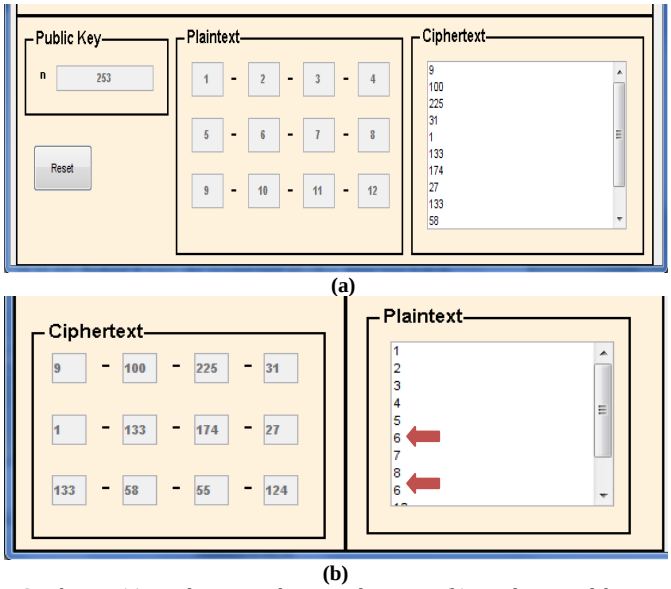
E. Proses Dekripsi

Setelah dilakukan proses ekstraksi, pengguna dapat melakukan proses dekripsi dengan cara menginputkan kunci privat yang akan digunakan untuk proses dekripsi. Selanjutnya, dilakukan proses dekripsi dengan cara mengeksekusi tombol *Decrypt*. Sistem akan menampilkan hasil dekripsi yang berupa plainteks. Proses dekripsi pada sistem dapat dilihat pada Gambar 13.



Gambar 13 Hasil proses dekripsi

Namun, terdapat masalah dari sistem ini yaitu untuk kombinasi *plaintext* dan kunci tertentu dapat diperoleh hasil dekripsi yang berbeda dari *plaintext* yang sebenarnya. Hal ini terjadi karena pada saat sistem memeriksa 4 (empat) kemungkinan nilai *plaintext*, terdapat 2 (dua) atau lebih nilai kemungkinan *plaintext* yang memenuhi syarat sebagai *plaintext* yang sebenarnya. Sehingga sistem ini akan mengambil nilai kemungkinan *plaintext* yang pertama sekali memenuhi syarat sebagai *plaintext* yang sebenarnya. Hal ini dapat dilihat pada Gambar 14 (a) dan Gambar 14 (b).



Gambar 14 (a) Hasil proses enkripsi pada sistem, (b) Hasil proses dekripsi pada sistem

Pada Gambar 14 (a) dilakukan proses enkripsi *plaintext* "1 2 3 4 5 6 7 8 9 10 11 12" dengan kunci publik "253" dan kunci privat "23" dan "11" menghasilkan *ciphertext* "9 100 225 31 1 133 174 27 133 58 55 124". Kemudian pada Gambar 14 (b) dilakukan proses dekripsi dan sistem menghasilkan nilai *plaintext* "1 2 3 4 5 6 7 8 6 10 11 12". Terdapat 1 (satu) nilai *plaintext* yang dihasilkan sistem yang berbeda dengan nilai yang *plaintext* yang sebenarnya, yaitu "9". Hal ini disebabkan karena terdapat 2 (dua) nilai kemungkinan *plaintext* yang memenuhi syarat, yaitu "6" dan "9". Karena nilai yang pertama sekali memenuhi syarat sebagai *plaintext* adalah "6", maka sistem menampilkan "6" sebagai *plaintext* yang sebenarnya.

V. KESIMPULAN

Berdasarkan hasil studi literatur, analisis, perancangan, implementasi dan pengujian sistem ini, maka didapat kesimpulan sebagai berikut :

1. Sistem ini menggabungkan algoritma kriptografi Rabin *Public Key* dan teknik steganografi *End of File* untuk menjaga keamanan dan kerahasiaan suatu data rahasia.
2. Pada sistem ini, data rahasia akan dienkripsi, kemudian hasil enkripsi yang berupa *ciphertext* akan disembunyikan ke dalam suatu *file* gambar berformat bitmap sehingga tidak akan muncul kecurigaan pihak lain dan keamanan dan kerahasiaan pesan terjaga.
3. Pada sistem ini, untuk beberapa kombinasi *plaintext* dan kunci tertentu terdapat hasil dekripsi yang berbeda dari *plaintext* yang sebenarnya dikarenakan pada saat sistem memeriksa 4 (empat) kemungkinan nilai *plaintext*, terdapat 2 (dua) atau lebih nilai kemungkinan *plaintext* yang memenuhi syarat sebagai *plaintext* yang sebenarnya. Sehingga sistem ini akan mengambil nilai kemungkinan *plaintext* yang pertama sekali memenuhi syarat sebagai *plaintext* yang sebenarnya.
4. Algoritma Rabin *Public Key* tidak aman untuk serangan *chosen-ciphertext attack* karena untuk kombinasi *plaintext* dan kunci yang merupakan angka kelipatan "11" akan menghasilkan *ciphertext* yang merupakan angka kelipatan "11" juga. Sehingga, seorang kriptanalis dapat mengetahui bentuk *plaintext* yang sebenarnya.
5. Pada metode *End of File*, data yang telah dienkripsi akan disisipkan pada nilai akhir *file* gambar, sehingga akan menambah ukuran *file* dan terdapat penambahan garis-garis pada bagian bawah *file* gambar tersebut.

VI. REFERENSI

[1] Aditya, Yogie, Andhika Pratama, Alfian Nurlifa. 2010. *Studi Pustaka untuk Steganografi dengan Beberapa Metode.* Universitas Islam Indonesia : Vol. 1

[2] Alexander G, Renald. 2012. *Analisis Perbandingan Algoritma RSA dan Diffie - Hellman untuk Pertukaran Kunci.* Jurnal. Institut Teknologi Bandung : Vol.1.

[3] Budiman, M. Andri. 2012. *Teori Bilangan dan Kriptografi.* Medan, Indonesia.

[4] Binanto, Iwan. 2010. *Multimedia Digital Dasar Teori + Pengembangan.* Yogyakarta: Penerbit ANDI.

[5] Bishop, Matt. 2004. *Introduction of Computer Security.* Boston : Pearson Education, Inc. Hal. 97,113.
[6] Menezes, J. Alfred, Paul C. Van Oorschot, Scott A.Vanstone. 1996. *Handbook of Applied Cryptography*. CRC Press.
[7] Krisnawati. 2008. *Metode Least Significant Bit (LSB) dan End of File (EOF) untuk Menyisipkan Teks pada Citra Grayscale.* Jurnal. STMIK "AMIKOM" : Vol.1
[8] Kurniawan, Yusuf. 2004. *Keamanan Internet dan Jaringan Telekomunikasi.* Bandung : Informatika.
[9] Munir, Rinaldi. 2003. *Matematika Diskrit*. Bandung, Indonesia : Penerbit Informatika Bandung.
[10] Munir, Rinaldi. 2006. *Kriptografi*. Bandung, Indonesia : Penerbit Informatika Bandung.
[11] Rhee, Man Young. 1993. *Cryptography and Secure Communication.* New York. McGraw-Hill.
[12] Smart, Nigel. 2003. *Cryptography - An Introduction*. 3rd Ed. California. McGraw-Hill.
[13] Schneier, Bruce. 1996. *Applied Criptography*. 2nd Ed. New York. John Wiley & Sons.
[14] Sutoyo. T dan Kawan-kawan. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.
[15] Widyarnako, Arya. 2008. *Teknik Kriptografi Rabin, Serangan yang Dapat Dilakukan dan Perbandingannya dengan RSA.* Jurnal. Institut Teknologi Bandung : Vol.2.