Pembangunan Aplikasi Penyembunyian Pesan Menggunakan Metode *End Of File* (EOF) ke dalam Citra *Digital* Terhadap Pesan yang Terenkripsi Dengan Algoritma RSA

Nina Anindyawati
Informatika
Fakultas MIPA UNS
Jl.Ir Sutami No.36 A Kentingan Surakarta
laelnino@gmail.com

Esti Suryani
Informatika
Fakultas MIPA UNS
Jl.Ir Sutami No.36 A Kentingan Surakarta
suryapalapa@yahoo.com

ABSTRAK

Kriptografi adalah ilmu tulisan rahasia dengan tujuan menyembunyikan makna pesan. Salah satu metodenya adalah dengan algoritma RSA (Rivest-Shamir-Adleman). Sedangkan steganografi yaitu ilmu dan seni menyembunyikan pesan rahasia (hiding message) sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia. End Of File (EOF) merupakan salah satu teknik yang digunakan dalam steganografi. Paper ini membahas bagaimana membangun aplikasi steganografi dengan menggabungkan kriptografi RSA untuk menyandikan pesan dan steganografi EOF untuk menyembunyikan pesan ke dalam media gambar, dan akan dilakukan pengujian untuk ukuran data yang berbeda dari gambar serta pesan teks untuk mengetahui seberapa besar pengaruh ukuran file pesan terhadap waktu eksekusi. Hasil penelitian menunjukkan bahwa telah dihasilkan sistem steganografi untuk menyembunyikan pesan ke dalam media gambar dengan gabungan kriptografi RSA dan steganografi EOF. Dari hasil pengujian dapat disimpulkan Untuk setiap plainteks yang diuji dan apabila ukuran plainteksnya dinaikkan, maka ukuran chipertext yang dihasilkan mengalami kenaikan secara linier. Pengaruh ukuran gambar dan format gambar yang dipakai sebagai media pembawa pesan relatif sama terhadap waktu penyisipan maupun ekstraksi. Rata-rata waktu eksekusi untuk enkripsi, dekripsi, penyisipan dan ekstraksi mengalami kenaikan secara linier

Keywords
Kriptografi, Steganografi, RSA, *End Of File* (EOF)

1. PENDAHULUAN

Teknologi komunikasi dan informasi berkembang dengan pesat dan memberikan pengaruh besar bagi kehidupan manusia, terutama dalam aktivitas berkirim informasi. Memungkinkan orang untuk saling bertukar data melalui jaringan internet, terutama data-data yang bersifat rahasia dan sangat penting, dan tidak boleh diketahui oleh pihak lain. Seiring dengan perkembangan tersebut, kejahatan teknologi komunikasi dan informasi juga turut berkembang, berupa perusakan maupun pencurian data oleh pihak yang tidak berkepentingan. Pada saat ini telah dilakukan berbagai upaya untuk menjaga keamanan data. Salah satu cara untuk mengatasi hal tersebut adalah dengan kriptografi dan steganografi.

Kriptografi adalah ilmu tulisan rahasia dengan tujuan menyembunyikan makna pesan. Kriptografi mempunyai banyak metode penyembunyian pesan. Salah satunya adalah penyembunyian pesan dengan algoritma RSA (Rivest-Shamir-Adleman). RSA termasuk dalam kriptografi kunci publik, dirancang oleh 3 orang peneliti dari MIT pada tahun 1976, singkatan RSA sendiri berasal dari inisial nama depan mereka, Ron Rivest, Adi Shamir, dan Leonard Adleman [7]. Algoritma RSA dipilih karena kelebihannya yaitu sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor primanya. Semakin panjang suatu kunci publik, maka usaha yang harus dikeluarkan untuk memecahkan kunci tersebut akan lebih lama [1]. Sedangkan steganografi, yaitu ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indera manusia [8]. *End Of File* (EOF) merupakan salah satu teknik yang digunakan dalam steganografi. Teknik ini menggunakan cara dengan menyisipkan data pada akhir file. Dalam teknik ini, data yang disisipkan pada akhir file diberi tanda khusus sebagai pengenal start dari data tersebut dan pengenal akhir dari data tersebut. [12]. EOF dipilih karena sifatnya yang *redundant bits* yaitu dimana setiap penambahan karakter *ctrl-z* pada sebuah *file* tidak akan mengubah nilai atau ukuran file tersebut. Karakteristik inilah yang menyebabkan *ctrl-z* dipilih sebagai penanda dari sebuah akhir file karena sifat null (kosong) yang dimilikinya, sehingga tidak mengubah isi awal dari *file* yang disisipi.

Penelitian ini akan membuat aplikasi penyembunyian pesan dengan menggabungkan dua buah metode, yaitu kriptografi dan steganografi. Salah satu penelitian yang memperkenalkan konsep penggabungan kriptografi dan steganografi adalah [2] dalam jurnalnya yang berjudul "*Text Steganography: A Novel Approach*". Penelitiannya menggunakan konsep Kriptografi dan Steganografi (sebagai dua lapisan keamanan). Diperkenalkan dua metode baru yaitu pemetaan code_matrix dan pemetaan matrix_pix, serta menggunakan steganografi (*Least Significant Bit*) sebagai lapisan ketiga keamanan, proses steganografi dalam penelitian ini yaitu memetakan matrix_pix ke dalam piksel gambar.

Penulis dalam penelitian ini berfokus pada enkripsi pesan teks dengan algoritma RSA dan menghasilkan sebuah *chipertext*. Selanjutnya *chipertext* tersebut akan disisipkan ke dalam media gambar dengan metode EOF, sehingga akan dihasilkan sebuah *stego object*. Selanjutnya akan dilakukan pengujian untuk ukuran data yang berbeda dari gambar serta pesan teks untuk mengetahui seberapa besar pengaruh ukuran file pesan terhadap waktu eksekusi. Penggunaan dua buah metode dalam penyembunyian pesan tersebut diharapkan agar pesan yang akan dikirim aman.

2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: "*cryptós*"artinya "*secret*" (rahasia), sedangkan "*gráphein*" artinya "*writing*"(tulisan). Jadi, kriptografi berarti "*secret writing*" (tulisan rahasia). Kriptografi adalah ilmu tulisan rahasia dengan tujuan menyembunyikan makna pesan [3]. Selain itu kriptografi juga didefinisikan sebagai ilmu untuk menjaga rahasia dari rahasia [4].

2.2 Metode RSA

RSA termasuk dalam kriptografi kunci public yang paling dikenal oleh orang banyak. RSA dirancang oleh 3 orang peneliti dari MIT pada tahun 1976. Oleh Ron Rivest, Adi Shamir, dan Leonard Adleman. Singkatan RSA berasal dari inisial nama depan mereka [8]. Pada algoritma RSA terdapat 3 langkah utama yaitu *key generation* (pembangkitan kunci), enkripsi, dan dekripsi. Kunci pada RSA mencakup dua buah kunci, yaitu *public key* dan *private key*. *Public key* digunakan untuk melakukan enkripsi, dan dapat diketahui oleh orang lain. Sedangkan *private key* tetap dirahasiakan dan digunakan untuk melakukan dekripsi. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima.

2.3 Steganografi

Steganografi berasal dari bahasa Yunani yaitu *steganos* yang artinya adalah penyamaran atau penyembuyian dan graphein yang artinya adalah tulisan. Jadi steganografi, yaitu ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan(eksistensi) pesan tidak terdeteksi oleh indera manusia [8]. Penyembunyian atau penyamaran pesan ini dibuat sedemikian rupa sehingga pihak lain tidak mengetahui bahwa ada 'pesan lain' di dalam pesan yang dikirimkan. Hanya pihak penerima yang sah saja yang dapat mengetahui 'pesan lain' tersebut.

Unsur-unsur Steganografi:

a. Embedded message (hiddentext)

Pesan yang disembunyikan. Bisa berupa teks, gambar, audio,video, dll

b. Cover-object (covertext)

Pesan yang digunakan untuk menyembunyikan *embedded message*. Bisa berupa teks, gambar, audio, video, dll

c. Stego-object (stegotext)

Pesan yang sudah berisi pesan *embedded message*.

d. Stego-key

Kunci yang digunakan untuk menyisipan pesan dan mengekstraksi pesan dari *stegotext*.

2.4 Metode *End Of File*

End Of File (EOF) merupakan salah satu metode steganografi yang teknik penyisipan datanya diletakkan pada akhir *file*. Data atau *file* yang akan disembunyikan nantinya, dapat berukuran lebih besar dari ukuran *file image*. EOF menggunakan karakter Ctrl-Z (ASCII 26) sebagai penanda akhir file, seperti pada sistem DOS. Pembacaan file berbasiskan DOS membaca file hingga karakter Ctrl-Z yang dapat disebut elemen dari EOF. Prinsip penggunaan Ctrl-Z ini adalah pembatas antara *covertext* (media pembawa) dengan pesan rahasia yang disisipkan. Jadi apabila kita membuka file teks dengan binary pada DOS kemudian kita gabungkan dengan sebuah teks setelah Ctrl-Z, maka pesan tersebut akan tersembunyi dibelakang EOF [11]

2.5 *Image* (Citra)

Definisi citra atau gambar adalah suatu benda yang tidak bergerak

atau statis. Setiap elemen pada citra dibentuk dari *pixel-pixel*. Sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi, f (x, y), dimana x dan y adalah koordinat spasial, dan amplitudo f pada setiap pasang koordinat (x, y) disebut tingkat intensitas atau tingkat keabuan [5]

2.6 Penelitian Terkait

Pengamanan dokumen yang dapat digunakan sebagai salah satu instrumen sistem pengamanan dokumen khususnya untuk dokumen teks. Adapun prinsip pengamanan dokumen ini adalah bagaimana sistem dapat mengamankan proses penyimpanan dan pengiriman dokumen. Metode yang digunakan adalah metode RSA, menggunakan dua macam kunci (private key dan public key) sehingga amat sulit untuk ditembus. Hasil pengujian ini menunjukkan bahwa sistem dapat menyimpan dan mengirimkan dokumen baik pengiriman melalui internet maupun intranet dalam bentuk susunan huruf yang terenkripsi dan mengembalikan ke bentuk dokumen semula dengan cara dekripsi. Semakin besar memori dokumen semakin lama waktu enkripsi dan waktu dekripsi, untuk waktu dekripsi juga lebih lama dari waktu enkripsi [13].

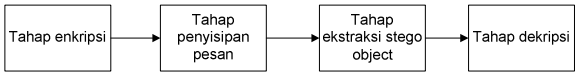
Penilitan yang dilakukan oleh [10] bertujuan untuk melakukan riset singkat tentang steganografi. Mengembangkan dan menerapkan aplikasi steganografi untuk menyembunyikan data dalam file gambar, serta mengambil data tersembunyi dari gambar yang berisi data tersembunyi dan meningkatkan kapasitas bersembunyi dengan mengenkripsi dan mengompresi data. Format gambar yang akan digunakan adalah GIF dan JPEG. Menggunakan dua macam metode yaitu, End Of File dan Matrix Embedding. Untuk metode EOF, teknik penyisipan akan digunakan penanda EOF gambar. Metode ini akan digunakan untuk menyembunyikan data di kedua gambar GIF dan JPEG. Untuk metode ini, tidak akan ada pembatasan untuk jumlah byte yang dapat disembunyikan.

Digunakan enkripsi Rijndael dan cipher pergeseran untuk mengenkripsi data. Sementara untuk enkripsi kunci menggunakan Rijndael akan hash md5 hash oleh fungsi. Menggunakan teknik steganografi adalah *End Of File* (EOF) teknik. Hasil penelitian ini membuktikan bahwa teknik steganografi EOF tidak merusak file media yang digunakan untuk menyembunyikan data [11]

3. METODE PENELITIAN

3.1 Skema Tahapan Implementasi

Ada dua buah proses utama yang akan dilakukan yaitu kriptografi untuk enkripsi teks dan steganografi pada penyisipan *ciphertext* ke dalam media gambar. Secara lebih rinci ada beberapa tahap yang akan dilakukan, yaitu tahap enkripsi, tahap penyisipan pesan, tahap ekstraksi stego object, dan tahap dekripsi, dapar ditunjukkan pada gambar 1.



Gambar 1. Skema Tahapan Implementasi Kriptografi dan Steganografi

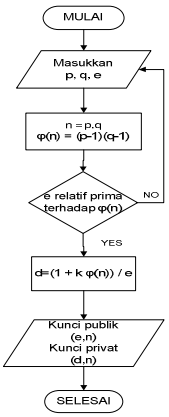
3.1.1 Tahap Enkripsi

Pada tahap ini sebuah pesan teks akan dilakukan enkripsi untuk menghasilkan *chipertext*. Enkripsi pesan teks menggunakan algoritma RSA. Algoritma RSA sendiri akan dibagi menjadi

tiga tahap, yaitu tahap pembentukan kunci, tahap enkripsi, dan tahap dekripsi. Berikut adalah penjelasan lebih rinci dari algoritma RSA.

Gambar 2. Skema Enkripsi Pesan Teks dengan algoritma RSA

Tahap Pembentukan Kunci RSA



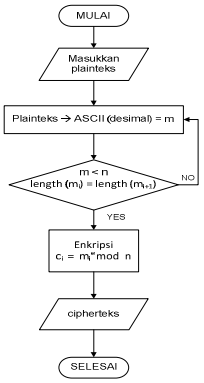
Gambar 3. Diagram Alir Pembentukan Kunci RSA

Diasumsikan bahwa Alice ingin mengirimkan pesan kepada Bob. Langkah-langkah pembangkitan kunci [9] :

a. Bob memilih dua buah bilangan prima besar secara acak, p dan q. dan $p \neq q$.
b. Lalu Bob menghitung nilai dari n = p.q
   Hitung nilai dari φ(n) = (p-1)(q-1)
   Bilangan n pada RSA disebut dengan modulus.
c. Selanjutnya Bob memilih e secara acak. Dimana n sedemikian sehingga 1 <e <φ (n) dan gcd (e, φ (n)) = 1
   Bilangan e pada RSA disebut dengan eksponen enkripsi.
d. Bob menghitung d dimana 1 <d<φ(n). menggunakan rumus
   ed ≡ 1 (mod φ(n)) yang ekuivalen dengan $d = \frac{1+k\phi(n)}{e}$
   untuk membangkitkan kunci privatnya.
e. Selanjutnya Bob menerbitkan (n, e) dalam beberapa database publik dan menjaga nilai d, p, q, dan φ (n) agar tetap rahasia. Dengan demikian, kunci public RSA adalah (n, e) dan kunci privat RSA adalah d. Bilangan d disebut eksponen dekripsi.
   Hasil yang didapatkan setelah dilakukan pembangkitan kunci :
   a. Kunci publik merupakan pasangan (e, n)
   b. Kunci privat adalah pasangan (d, n)

Tahap pembangkitan pasangan kunci RSA, adalah langkah awal yang harus dilakukan untuk mendapatkan kunci publik dan kunci privat, yang akan digunakan pada saat enkripsi dan dekripsi pesan

Tahap Enkripsi Pesan



Gambar 4. Diagram Alir Enkripsi RSA

Proses enkripsi RSA adalah sebagai berikut :

a. Mengambil nilai e dan n dari proses pembangkitan kunci.
b. Memasukkan teks yang akan dienkripsi.
c. Mengubah berkas teks yang akan dienkripsi ke dalam bentuk decimal, yang sesuai dengan tabel ASCII.
d. Membagi berkas tersebut menjadi beberapa blok (m_i), dengan syarat $m_i < n$ dan length (m_i) = length (m_{i+1})
e. Setelah itu setiap blok dienkripsi menggunakan pasangan kunci publik menggunakan persamaan $\mathbf{c_i = m_i^e \bmod n}$ yang dalam hal ini p_i adalah blok plainteks, c_i adalah cipherteks yang diperoleh , dan e adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai p_i harus terletak dalam himpunan nilai **0, 1, 2, ..., n-1** untuk menjamin hasil perhitungan tidak berada di luar himpunan.

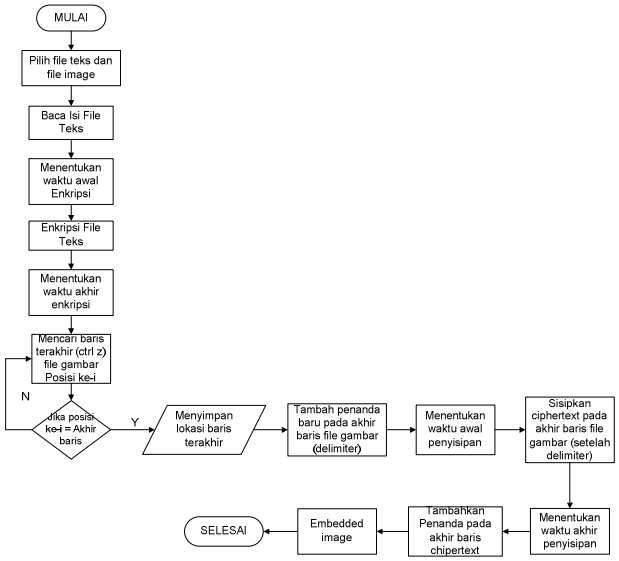
   Atau secara sederhana seperti berikut ini [9] :
   Diasumsikan bahwa pesan plaintext m ∈ M adalah dalam bentuk numerik dengan m <n. Kita juga mengasumsikan bahwa gcd(m, n) = 1.
   (1) Alice Bob memperoleh kunci publik (n, e) dari database.
   (2) Alice mengenkripsi m dengan menghitung $c \equiv m^e \pmod n$
   (3) Alice mengirimkan c ∈ C kepada Bob.
   Pada tahap ini, sebuah plainteks akan dienkripsi dengan menggunakan kunci yang telah didapatkan dari proses sebelumnya. Enkripsi RSA menggunakan kunci publik.

3.1.2 Tahap Penyisipan Pesan

Pada tahap ini sebuah pesan yang terenkripsi akan disisipkan ke dalam media gambar. Proses penyisipannya menggunakan metode *End Of File*, yaitu penyisipan datanya diletakkan pada akhir file.

Gambar 5. Skema Penyisipan Chipertext ke Dalam Media Gambar

Gambar 6 berikut adalah langkah-langkah dalam proses penyisipan ciphertext ke dalam media gambar :



Gambar 6. Diagram Alir Penyisipan Chipertext ke Media Gambar

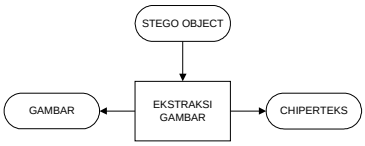
Algoritma dari metode EOF ini adalah sebagai berikut :

a. Membaca informasi file, tentukan dimana posisi akhir baris (*ctrl-z)* berada.
b. Menandai posisi akhir baris (*ctrl-z)* dan menambahkan penanda baru (delimiter) sebagai awal baris penyisipan pesan.
c. Menyisipkan pesan dimulai dari posisi akhir baris (*ctrl-z)* setelah delimiter hingga akhir pesan.
d. Menyisipkan penanda pada akhir baris (*ctrl-z)* kedua pada akhir pesan.

Prosedur penyisipan ini dilakukan dengan cara menambahkan baris baru pada akhir file teks setelah karakter ctrl-z. Setelah citra digital selesai disisipkan, ditambahkan karakter ctrl- z kedua sebagai penanda akhir penyisipan file. Pesan cipherteks yang telah didapatkan dari proses enkripsi akan disisipkan ke dalam file image. Sehingga akan dihasilkan *stego object*.

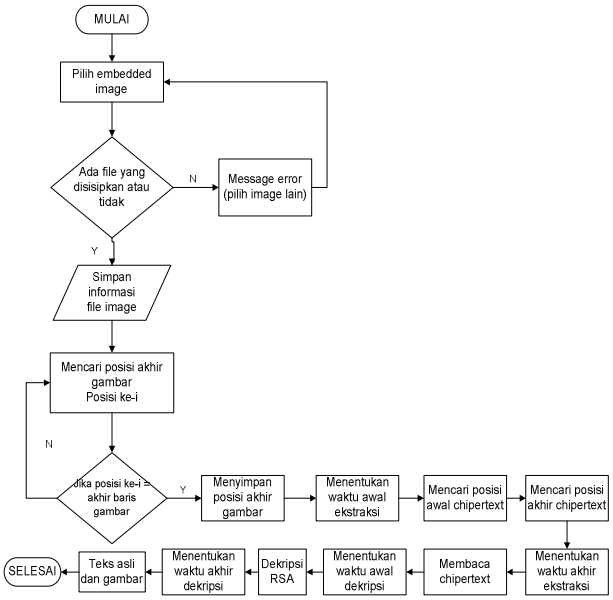
3.1.3 Tahap Ekstraksi Stego Object

Pada tahap ini gambar yang telah disisipi oleh pesan akan dipisahkan kembali menjadi gambar dan pesan teks terenkripsi.



Gambar 7. Skema Ekstraksi Stego Object

Gambar 8 berikut adalah langkah-langkah dalam proses ekstraksi ciphertext :

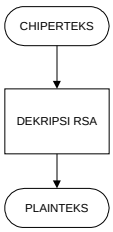


Gambar 8. Diagram Alir Pemisahan Chipertext dari Stego Object

Langkah pertama, merancang sistem untuk membaca seluruh data *digital* dengan cara melacak posisi ctrl-z pada file tersebut. Jika karakter ctrl-z ditemukan, akan disimpan lokasinya. Selanjutnya sistem akan terus membaca file hingga menemukan karakter ctrl-z kedua yang menjadi akhir dari citra *digital* yang disisipkan. Kemudian, barulah sistem menampilkan hasil pembacaan citra *digital* ini pada tampilan perangkat lunak. Pesan yang dalam bentuk *stego object*, akan di ekstrak dan dipisahkan kembali. Hasil ekstraksi berupa *ciphertext* dan *image*, selanjutnya *ciphertext* tersebut dilakukan dekripsi untuk didapatkan teks asli.

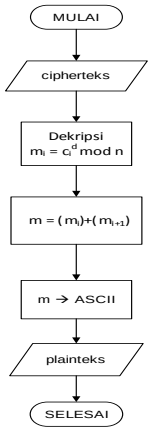
3.1.3 Tahap Dekripsi

Pada tahap ini pesan yang terenkripsi akan dilakukan dekripsi untuk diubah kembali ke pesan asli.



Gambar 9. Skema Dekripsi RSA

Berikut langkah-langkah dalam proses dekripsi ciphertext :



Gambar 10. Diagram Alir Dekripsi RSA

Proses dekripsi :

Setelah Bob menerima *chipertext* c, dia menggunakan nilai d untuk menghitung $m \equiv c^d \pmod n$ [9].

Cipherteks yang telah didapat dari langkah sebelumnya akan didekripsi untuk mendapatkan plainteks (pesan awal) dengan menggunakan kunci privat. Kunci privat didapatkan pada tahap pembangkitan pasangan kunci.

Proses dekripsi dilakukan dengan menggunakan persamaan $\mathbf{m_i = c_i^d \bmod n,}$ yang dalam hal ini d adalah kunci dekripsi.

3.2 Tahap Pengujian

Aplikasi yang telah selesai dibangun perlu diuji untuk mengetahui apakah aplikasi tersebut bekerja sesuai dengan tujuan yang telah ditentukan sebelumnya. Pengujian akan dilakukan dengan menghitung waktu eksekusi. Waktu eksekusi yang akan dihitung antara lain : waktu enkripsi, waktu dekripsi, waktu penyisipan *chipertext* ke dalam media gambar, dan waktu ekstraksi *chipertext* dari *stego object*. Plainteks yang diujikan menggunakan format .txt dan ukurannya akan dinaikkan tiap kelipatan 5 KB, dimulai dari ukuran 1 KB. Ukuran gambar yang dipakai menggunakan format .jpg dan .gif dengan dimensi 600 x 600, 500 x 500, dan 300 x 300. Pengujian dilakukan dengan menggunakan blok 58 dan kunci yang sama.

4. PEMBAHASAN

4.1 Hasil Implementasi

Aplikasi yang dibangun dibuat terdiri dari beberapa fungsi yang terbagi menjadi 3 buah bagian, yaitu bagian *file* teks, bagian *file image*, dan *file info*.

Bagian *File* Teks

a. *Input File*
Digunakan untuk membuka file teks, dalam aplikasi ini yang dipakai adalah file teks dengan format .txt.

b. *Output File*
Digunakan untuk menyimpan *chipertext* hasil enkripsi dan teks asli hasil dekripsi. Pada *output file* akan ditentukan *destination folder* untuk menyimpan hasil enkripsi.

c. *Text Preview*
Digunakan untuk menampilkan plainteks yang akan dilakukan enkripsi dan *chipertext* hasil ekstraksi akan ditampilkan disini.

d. Tombol pilihan enkripsi dan dekripsi
Tombol ini digunakan untuk memilih apakah kita akan melakukan proses enkripsi RSA atau dekripsi RSA.

e. Tombol Enkripsi/Dekripsi
Tombol ini digunakan untuk melakukan proses enkripsi dan dekripsi RSA.

Bagian *File Image*

f. *Input Image*
Digunakan untuk membuka file image, dalam aplikasi ini *image* yang digunakan menggunakan format .jpg dan .gif dengan dimensi 600x600, 500x500, dan 300x300

g. *Pic Preview*
Digunakan untuk menampilkan *image* yang akan digunakan sebagai media penyisipan pesan dan *stego object* yaitu *image* yang telah disisipi oleh *chipertext*.

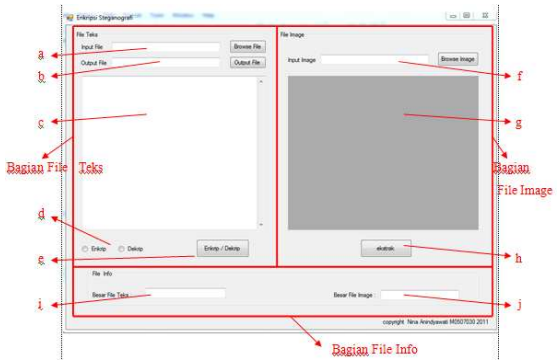
h. Ekstraksi
Digunakan untuk melakukan proses ekstraksi atau pemisahan *chipertext* dari *stego object*, hasil ekstraksinya akan ditampilkan ke dalam *Text Preview*.

Bagian *File Info*

i. Besar *File* Teks
Bagian ini digunakan untuk menampilkan besar ukuran dari file teks yang ditampilkan di *Text Preview*.

j. Besar *File Image*
Bagian ini digunakan untuk menampilkan besar ukuran dari file *image* yang ditampilkan di *Pic Preview*.

Berikut adalah tampilan aplikasi yang telah dibangun :



Gambar 11. Tampilan Utama Aplikasi Enkripsi Steganografi

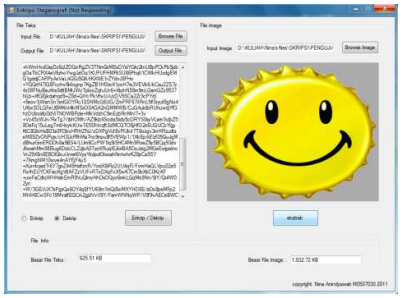
Langkah-langkah Proses Enkripsi dan Penyisipan Chipertext

1. Menentukan plainteks yang akan dienkripsi dengan RSA. File plainteksnya berformat .txt
2. Menentukan *destination folder* untuk menyimpan hasil enkripsi plainteks.
3. Menentukan gambar yang akan dijadikan sebagai media pembawa pesan. Gambar bisa berformat .jpg atau .gif
4. Melakukan enkripsi dengan RSA sehingga akan terbentuk sebuah file *chipertext*.
5. *Chipertext* yang dihasilkan selanjutnya akan disisipkan ke dalam media gambar.
6. Terbentuklah *image* baru atau *stego object*.
7. Sebagai gambaran dapat dilihat pada Gambar 11

Gambar 11. Tampilan Proses Sebelum dan Setelah Enkripsi dari Aplikasi Enkripsi Steganografi

Langkah-langkah Proses Ekstraksi dan Dekripsi Chipertext

1. Menentukan gambar stego object.
2. Melakukan ekstraksi, yaitu mengambil chipertext dari stego object.
3. Menentukan destination folder untuk menyimpan hasil dekripsi chipertext.
4. Melakukan dekripsi chipertext dengan RSA sehingga akan terbentuk file dekripsi atau file asli.
5. Sebagai gambaran dapat dilihat pada Gambar 12



Gambar 12. Tampilan Proses Sebelum dan Setelah Ekstraksi dari Aplikasi Enkripsi Steganografi

4.2 Hasil Pengujian dan Analisa

Pengujian dilakukan dengan cara menghitung waktu eksekusi enkripsi, dekripsi, penyisipan, dan ekstraksi. Plainteks yang diujikan menggunakan format .txt dan ukurannya dinaikkan tiap kelipatan 5 KB. Ukuran gambar yang dipakai menggunakan format .jpg dan .gif dengan dimensi 600x600, 500x500, dan 300x300. Pengujian dilakukan dengan menggunakan blok 58 dan kunci yang sama. Hasil pengujiannya adalah sebagai berikut :

Tabel 1. Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi untuk Ukuran Plainteks yang Berbeda pada Media Gambar Dimensi 600 x 600 dengan Format .jpg

Ukuran (KB)				Waktu (s)			
Plainteks	Chiperteks	Image sebelum	Image setelah	enkripsi	dekripsi	penyisipan	ekstraksi
1	3	50.9	53.9	0,0200000	0,0600001	0,0100001	6,5500092
5	14.8	50.9	65.7	0,0312000	0,2184004	0,0156000	19,9000794
10	29,5	50.9	80,4	0,0468001	0,4836009	0,0156001	34,6008607
15	44,3	50.9	95,2	0,0624001	0,6396011	0,0312001	51,2460901
20	59	50.9	109	0,0780001	0,8470485	0,0468001	71,7141018
25	73.6	50.9	124	0,0820047	0,9400013	0,0500029	88,7888473
30	88,5	50.9	139	0,1000057	1,1530660	0,0570032	110,462318
35	103	50.9	154	0,1200069	1,3010744	0,0640037	127,03051
40	118	50.9	168	0,1220070	1,5110865	0,0700040	143,535209
45	132	50.9	183	0,1290074	1,6740958	0,0800046	156,986979
50	147	50.9	198	0,1550088	1,8241043	0,0880050	174,081956

Tabel 2. Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi untuk Ukuran Plainteks yang Berbeda pada Media Gambar Dimensi 600 x 600 dengan Format .gif

Ukuran (KB)				Waktu (s)			
Plainteks	Chiperteks	Image sebelum	Image setelah	enkripsi	dekripsi	penyisipan	ekstraksi
1	3	134	137	0,0200000	0,0600001	0,0100000	5,7800081
5	14.8	134	149	0,0300000	0,2100003	0,0200001	19,5400273
10	29,5	134	163	0,0450026	0,5760329	0,0210012	35,5450330
15	44,3	134	178	0,0550031	0,5770330	0,0280016	52,2909909
20	59	134	193	0,0660038	0,7820447	0,0360021	68,7859343
25	73.6	134	208	0,0800001	0,9300013	0,0490028	87,9201231
30	88,5	134	222	0,1000057	1,1190640	0,0540031	105,6040402
35	103	134	237	0,1070061	1,2830734	0,0600035	119,2968234
40	118	134	252	0,1190068	1,4660838	0,0740042	136,0937841
45	132	134	267	0,1390080	1,6430940	0,0830047	153,9938080
50	147	134	281	0,1440082	1,8281046	0,0920053	176,0400689

Tabel 3. Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi untuk Ukuran Plainteks yang Berbeda pada Media Gambar Dimensi 500 x 500 dengan Format .jpg

Ukuran (KB)				Waktu (s)			
Plainteks	Chiperteks	Image sebelum	Image setelah	enkripsi	Dekripsi	penyisipan	ekstraksi
1	3	38.55	41.5	0,0100001	0,0900001	0,0100000	6,9500097
5	14.8	38.55	53.3	0,0380022	0,2200003	0,0120006	19,9800280
10	29,5	38.55	68	0,0468000	0,4950284	0,0156001	36,4652431
15	44,3	38.55	82,8	0,0650037	0,5980342	0,0300017	54,0100892
20	59	38.55	97,5	0,0670038	0,7640437	0,0370021	70,6430405
25	73.6	38.55	112	0,0850048	0,9420539	0,0450026	85,9459158
30	88,5	38.55	127	0,0960055	1,1570662	0,0560032	105,0200068
35	103	38.55	141	0,1030059	1,3540774	0,0630036	122,6230136
40	118	38.55	156	0,1110063	1,4730843	0,0740042	139,0769548
45	132	38.55	171	0,1380079	1,6530954	0,0820047	158,0650408
50	147	38.55	185	0,1460084	1,8461056	0,0910052	174,1339599

Tabel 4. Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi untuk Ukuran Plainteks yang Berbeda pada Media Gambar Dimensi 500 x 500 dengan Format .gif

Ukuran (KB)				Waktu (s)			
Plainteks	Chiperteks	Image sebelum	Image setelah	enkripsi	Dekripsi	penyisipan	ekstaksi
1	3	92	95	0,0156001	0,0936002	0,0150009	5,8344103
5	14.8	92	106	0,0320018	0,3350192	0,0156000	19,8388462
10	29,5	92	121	0,0440025	0,4010229	0,0210012	35,2570166
15	44,3	92	136	0,0620036	0,5730328	0,0290017	52,1239813
20	59	92	151	0,0660038	0,7740443	0,0410024	68,8969406
25	73.6	92	165.72	0,0780045	0,9360144	0,0440026	86,7569622
30	88,5	92	180	0,0950054	1,1240643	0,0550031	104,3489685
35	103	92	195	0,1020058	1,2920739	0,0600034	122,3679990
40	118	92	210	0,1050060	1,4750844	0,0720041	139,7749947
45	132	92	224	0,1250071	1,6540946	0,0760044	156,5839561
50	147	92	239	0,1500086	1,8221042	0,0840048	173,9349486

Tabel 5. Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi untuk Ukuran Plainteks yang Berbeda pada Media Gambar Dimensi 300 x 300 dengan Format .jpg

Ukuran (KB)				Waktu (s)			
Plainteks	Chiperteks	Image sebelum	Image setelah	enkripsi	dekripsi	penyisipan	ekstraksi
1	3	19.1	22.1	0,0162502	0,0825012	0,0112502	6,8888588
5	14.8	19.1	33.9	0,0290017	0,2200003	0,0150000	18,6200260
10	29,5	19.1	48,6	0,0440025	0,3990288	0,0200012	36,0140599
15	44,3	19.1	63,4	0,0490028	0,5750329	0,0270015	53,1200383
20	59	19.1	78,1	0,0600034	0,7580434	0,0350020	60,4350287
25	73.6	19.1	92.8	0,0780045	0,9300013	0,0430025	87,5801226
30	88,5	19.1	107	0,0910053	1,0970628	0,0510029	105,2490199
35	103	19.1	122	0,0970055	1,2940741	0,0610035	123,5620673
40	118	19.1	137	0,1190068	1,4740844	0,0700040	139,9220031
45	132	19.1	151	0,1290073	1,6690955	0,0790045	158,4940653
50	147	19.1	166	0,1440083	1,8331048	0,0860050	175,3880317

Tabel 6. Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi untuk Ukuran Plainteks yang Berbeda pada Media Gambar Dimensi 300 x 300 dengan Format .gif

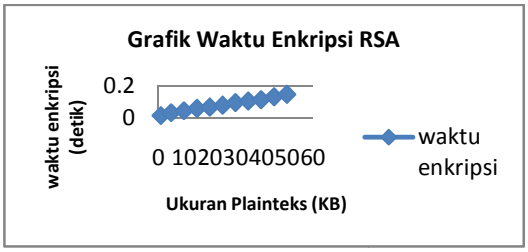
Ukuran (KB)				Waktu (s)			
Plainteks	Chiperteks	Image Asli	Image setelah	enkripsi	dekripsi	penyisipan	ekstraksi
1	3	34.5	37.5	0,0100000	0,1300002	0,0100000	6,7000094
5	14.8	34.5	49.3	0,0400000	0,2200003	0,0200001	20,2100283
10	29,5	34.5	64	0,0480028	0,3990228	0,0210012	35,6720403
15	44,3	34.5	78,8	0,0660037	0,5800332	0,0260015	52,7390165
20	59	34.5	93,5	0,0700040	0,7550432	0,0340019	69,0749508
25	73.6	34.5	108.1	0,0740042	0,9200526	0,0430024	87,3129940
30	88,5	34.5	123	0,0980056	1,1170639	0,0520030	105,731047
35	103	34.5	137	0,1040059	1,3000744	0,0610035	123,843083
40	118	34.5	152	0,1120064	1,4820848	0,0680039	139,911002
45	132	34.5	167	0,1340077	1,6850964	0,0750043	157,301997
50	147	34.5	181	0,1490085	1,8341049	0,0840048	171,893831

Rata-rata Waktu Eksekusi Enkripsi, Dekripsi, Penyisipan, dan Ekstraksi berdasarkan hasil pengujian dapat disimpulkan bahwa rata-rata waktu eksekusinya adalah sebagai berikut :

Tabel 7. Rata-rata waktu eksekusi untuk proses enkripsi, dekripsi, penyisipan, dan ekstraksi pada plainteks dengan kelipatan 5 KB

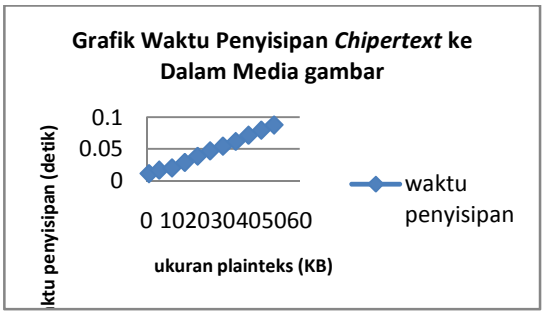
Ukuran Plainteks (KB)	Enkripsi (detik)	Dekripsi (detik)	Penyisipan (detik)	Ekstraksi (detik)
1	0,0153084	0,0860170	0,01104187	6,45055092
5	0,0333676	0,2372368	0,0163668	19,6815059
10	0,0453022	0,4432885	0,01993435	35,3910045
15	0,0599028	0,5904612	0,02853468	52,5883677
20	0,0678365	0,7800446	0,03830177	68,2583328
25	0,0798364	0,9330208	0,04666935	87,4408276
30	0,0966722	1,1278979	0,05416975	106,069233
35	0,1055060	1,3040746	0,06150353	123,120583
40	0,1146732	1,4802514	0,0713374	139,718991
45	0,1323409	1,6630953	0,0791712	156,904308
50	0,148008	1,8312714	0,08750502	174,245466

Untuk Tabel 7 lebih jelasnya dapat dilihat pada grafik berikut ini :

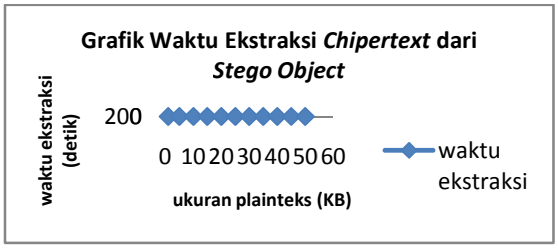


Gambar 13. Grafik Rata-rata Waktu Eksekusi untuk Enkripsi RSA

Gambar 14. Grafik Rata-rata Waktu Eksekusi untuk Dekripsi RSA



Gambar 15. Grafik Rata-rata Waktu Eksekusi untuk Penyisipan Chipertext Ke Dalam Media Gambar



Gambar 16. Grafik Rata-rata Waktu Eksekusi untuk Ekstraksi Chipertext Dari Stego Object

Berdasarkan hasil pada Tabel 7, untuk melihat rata-rata kenaikan waktu eksekusi enkripsi, dekripsi, penyisipan, dan ekstraksi, dapat dilihat pada tabel 8 di bawah ini :

Tabel 8. Kenaikan waktu eksekusi untuk proses enkripsi, dekripsi, penyisipan, dan ekstraksi pada plainteks dengan kelipatan 5 KB

Ukuran Plainteks (KB)	Enkripsi	Dekripsi	Penyisipan	Ekstraksi
1-5	2,18	2,76	1,48	3,05
5-10	1,36	1,87	1,22	1,79
10-15	1,32	1,33	1,43	1,49
15-20	1,32	1,32	1,34	1,29
20-25	1,18	1,19	1,21	1,28
25-30	1,21	1,21	1,16	1,21
30-35	1,09	1,16	1,14	1,16
35-40	1,09	1,14	1,16	1,35
40-45	1,15	1,12	1,11	1,12
45-50	1,12	1,01	1,11	1,11
Rata-rata	1,302	1,411	1,236	1,485

Dari hasil tabel-tabel di atas, untuk menghitung rata-rata tiap waktu eksekusi dapat menggunakan rumus berikut :

Dari hasil pengujian pada tabel-tabel di atas dapat disimpulkan sebagai berikut :

1. Ukuran plainteks yang digunakan dinaikkan tiap kelipatan 5 KB, yaitu 1 KB, 5 KB, 10 KB, 15 KB, 20 KB, 25 KB, 30 KB, 35 KB, 40 KB, 45 KB, 50 KB.
2. Untuk setiap plainteks yang diuji, ukuran chipertext yang dihasilkan naik kurang lebih 3 kali lipat dari ukuran plainteksnya. (Tabel 1 – Tabel 6)
3. Untuk setiap plainteks, apabila ukurannya dinaikkan tiap kelipatan 5 KB maka ukuran chipertext yang dihasilkan naik kurang lebih 5 kali dari ukuran chipertext sebelumnya. (Tabel 1 – Tabel 6)
4. Untuk setiap *image* yang telah disisipkan chiperteks ukurannya merupakan hasil dari ukuran *image* sebelum disisipkan ditambah dengan ukuran chiperteksnya. (Tabel 1 – Tabel 6)
5. Waktu untuk dekripsi *chipertext* membutuhkan waktu yang lebih lama dari waktu enkripsinya.
6. Ukuran gambar dan format gambar yang dipakai sebagai media pembawa pesan relatif tidak mempengaruhi terhadap waktu penyisipan maupun ekstraksi.
7. Rata-rata waktu eksekusi untuk enkripsi, dekripsi, penyisipan dan ekstraksi mengalami kenaikan secara linier. Dengan menaikkan ukuran plainteks tiap kelipatan 5KB, maka pengaruhnya terhadap waktu eksekusi adalah sebagai berikut:
   Waktu enkripsi kurang lebih naik 1 kali dari waktu enkripsi sebelumnya.
   Waktu dekripsi kurang lebih naik 1 kali dari waktu dekripsi sebelumnya.
   Waktu penyisipan *chipertext* ke dalam media gambar kurang lebih 1 kali dari waktu penyisipan sebelumnya.
   Waktu ekstraksi *chipertext* dari *stego object* kurang lebih 1 kali dari waktu ekstraksi sebelumnya, (dapat dilihat pada Gambar 13 – Gambar 15)

5. PENUTUP

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan sebelumnya, dapat ditarik kesimpulan sebagai berikut:

a. Telah dihasilkan sistem steganografi untuk menyembunyikan pesan ke dalam media *image* dengan menggabungkan 2 buah metode, yaitu kriptografi RSA dan steganografi *End Of File* (EOF).
b. Hasil pengujian menunjukkan bahwa :
   - Untuk setiap plainteks yang diuji dan apabila ukuran plainteksnya dinaikkan, maka ukuran chipertext yang dihasilkan mengalami kenaikan secara linier.
   - Pengaruh ukuran gambar dan format gambar yang dipakai sebagai media pembawa pesan relatif sama terhadap waktu penyisipan maupun ekstraksi.
   - Rata-rata waktu eksekusi untuk enkripsi, dekripsi, penyisipan dan ekstraksi mengalami kenaikan secara linier.

6. DAFTAR PUSTAKA

[1] Arifin, Zainal. 2009. *Studi Kasus Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi yang Aman*. Samarinda : Program Studi Ilmu Komputer, FMIPA Universitas Mulawarwan. Jurnal Informatika Mulawarman Vol 4 No. 3

[2] Bhattacharyya, Debnath., et all. 2009. *Text Steganography: A Novel Approach*. India. International Journal of Advanced Science and Technology Vol. 3

[3] C. Paar, J. Pelzl. 2010. *Understanding Cryptography*. Springer-Verlag Berlin Heidelberg

[4] Delfs, Hans and Knebl, Helmut. 2007. *Introduction to Cryptography Principles and Applications Second Edition*. Springer-Verlag Berlin Heidelberg

[5] Gonzalez, Rafael. C. 2001. *Digital Image Processing Second Edition*. New Jersey USA: Prentice Hall

[6] Katz, Jonathan, and Lindell, Yehuda. 2007. *Introduction To Modern Cryptography*. Chapman & Hall/CRC

[7] Menezes, A, van Oorschot, P and Vanstone, S. 1997. *Handbook of Applied Cryptography*. CRC Press, Inc

[8] Munir, Rinaldi. 2004. *Pengantar Kriptografi*. Bandung : Institut Teknologi Bandung

[9] Mollin, Rhichard A. 2003. *RSA and Public-Key Cryptography*. CRC Press LLC

[10] Shantala C.P. and Dr. K.V. Vishwanatha. 2009. *Advanced Steganography for Lossy Compression Images.* India. International Journal of Cryptography and Security Volume 2, Number 1

[11] Sejati, Adiputra. 2007. Studi dan Perbandingan Steganografi Metode EOF (End of File) dengan DCS (Dynamic Cell Spreading). Bandung : Teknik Informatika Institut Teknologi Bandung

[12] Sukrisno, dan Utami, Ema. 2007. *ImplementasiSteganografi Teknik Eof Dengan Gabungan Enkripsi Rijndael, Shift Cipher Dan Fungsi Hash Md5*. Seminar Nasional Teknologi 2007 (SNT 2007). Yogyakarta : STMIK AMIKOM ISSN : 1978 – 9777

[13] Supriyono. 2008. *Pengujian Sistem Enkripsi-Dekripsi Dengan Metode RSA untuk Pengamanan Dokumen*. Yogyakarta : Sekolah Tinggi Teknik Nuklir. JFN, Vol 2 No. 2. ISSN 1978-8738