Implementasi Sistem Keamanan Data Untuk Semua Jenis File Dengan Menggunakan Teknik Steganografi End Of File (EOF) dan Algoritma Kriptografi Rivest Code 4 (RC4)

Syapriadi
Jurusan Teknik Informatika
STMIK-AMIK Riau
syapriadi@yahoo.com

Rahmiati
Jurusan Manajemen Informatika
STMIK-AMIK Riau
rahmiati_06@yahoo.com

Susi Erlinda
Jurusan Teknik Informatika
STMIK-AMIK Riau
erlinda_susi@yahoo.com

Abstrak

Keamanan dan kerahasiaan data merupakan salah satu aspek terpenting dalam bidang komunikasi, khususnya komunikasi yang menggunakan media komputer. Proses pengamanan informasi dapat dilakukan dengan menyembunyikan informasi tersebut pada media lain atau dengan metode tertentu, sehingga orang lain tidak menyadari ada suatu informasi didalam media tersebut. Dikenal dengan teknik Steganografi dan Kriptografi. Steganografi adalah teknik menyembunyikan atau menyamarkan keberadaan pesan rahasia dalam media penampungnya. Sedangkan kriptografi menyamarkan arti dari suatu pesan,tetapi tidak menyembunyikan bahwa ada suatu pesan. Pada penelitian ini akan mengoptimalkan keamanan data yang tersimpan pada komputer atau data yang akan dikirim ke pihak lain. Dengan mengadopsi konsep steganografi end of file 4dan algoritm kriptografi RC4, dan implementasinya dengan menggunkan bahasa pemograman Delphi 2007. Penulis telah merancang sebuah aplikasi dengan perpaduan teknik steganografi dan kriptografi sebagai sistem untuk menjaga keamanan data.

Kata Kunci : Keamanan data, *steganografi(EOF)*, Kriptografi, Algoritma RC4.

1. Pendahuluan

Keamanan informasi pada zaman global ini makin menjadi sebuah kebutuhan yang vital dalam berbagai aspek kehidupan. Suatu informasi akan memiliki nilai lebih tinggi apabila menyangkut tentang aspek-aspek keputusan bisnis, keamanan, ataupun kepentingan umum. Dimana informasi-informasi tersebut tentunya akan banyak diminati oleh berbagai pihak yang juga memiliki kepentingan di dalamnya.

Seiring dengan perkembangan teknologi, teknik dan metode penyampaian pesan rahasia pun semakin beragam. Terdapat berbagai bentuk pesan rahasia seperti pesan teks, pesan citra, pesan audiodan pesan video yang umum digunakan. Masalah keamanan dan kerahasiaan data merupakan hal yang sangat penting dan menjadi isu yang sangat penting dan terus berkembang, baik dalam suatu organisasi yang berupa komersial (perusahaan), perguruan tinggi, lembaga pemerintahan, maupun dalam hal individual (pribadi).

Untuk itu, salah satu cara pengamanan yang dapat di lakukan adalah dengan mengunakan teknik kriptografi. Kriptografi merupakan bidang pengetahuan yang menggunakan persamaan matematis untuk melakukan proses enskripsi maupun deskripsi. Teknik ini digunakan untuk merubah data biasa dengan kunci tertentu menjadi data yang tidak diketahui oleh orang lain kecuali bagi orang yang berhak. Berbagai macam algortima kriptografi salah satu diantaranya adalah *rivest code* (RC4), model ini merupakan salah satu algoritma kunci simetris yang berbentuk stream cipher. Algoritma kunci simetris termasuk algoritma yang masih sering di gunakan dalam kriptografi.

Namun, teknik kriptografi yang sifatnya mengacak suatu pesan rahasia dapat menimbulkan kecurigaan. Sehingga munculah teknik steganografi yang merupakan pengembangan dari kriptografi. Teknik steganografi merupakan suatu teknik menyembunyikan informasi di balik *cover* media teks, gambar, *audio* dan *video* sehingga informasi atau data yang sesungguhnya tidak terlihat dan tidak menimbulkan kecurigaan bagi orang lain. Saat ini telah ada beberapa metode steganografi yang digunakan. Salah satunya adalah metode *End of File* (EOF).

Keamanan informasi akan lebih menjadi lebih tangguh dengan memadukan teknik kriptografi dan steganografi. Dimana apabila informasi rahasia yang telah disembunyikan dapat terdeteksi oleh pihak yang tidak berhak, Maka informasi rahasia tersebut masih terlindungi oleh suatu metode kriptografi.

Berdasarkan pada semua hal yang telah dijabarkan di atas, maka pada penelitian ini algoritma kriptografi *rivest code* 4 (RC4) dan steganografi *end of file* (EOF) digunakan untuk merancang dan membangun sebuah aplikasi untuk mengamankan data.

Pada penelitian ini penulis menggunakan beberapa jurnal dan skripsi yang terkait dengan keamanan data khususnya kriptografi dan steganografi sebagai referensi bagi penulis. Berikut list literatur yang digunakan sebagai bahan referensi:

Tabel 1. Referensi Literatur

No	Nama penulis	Judul	Keterangan
1	Jamaludin, FTI-UIN Jakarta [1]. 2010	Aplikasi keaman informasi menggunakan teknik steganografi dengan metode least significant bit (lsb) insertion dan rc4.	Aplikasi steganografi hanya dapat menyembunyikan pesan rahasia berupa file teks dan file pembawa pesan rahasia hanya file citra atau gambar.
2	Endang Fiansyah, FTI UI Depok [2]. 2008	Implementasi algoritma dasar rc4 stream cipher dan pengacakan plainteks dengan teknik dynamic blocking pada APLIKASI SISTEM INFORMASI KEGIATAN SKRIPSI.DI DEPARTEMEN TEKNIK ELEKTRO	Sistem yang dibangun untuk pengamanan pada jaringan computer dan pengamanan database.

Berdasarkan pada referensi literatur diatas, penulis mencoba melakukan pengembangan dengan menggabungkan 2 buah metode keamanan data steganografi end of file dan kriptografi RC4.

2. Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan membuat aplikasi untuk keamanan data dengan mengunakan algoritma kriptografi *Rivest Code* (RC4) dan steganografi *End Of File* (EOF).

3. Rasional Penelitian

Hasil Penelitian ini diharapkan berguna sebagai sarana pengembangan keilmuan dalam bidang teknik informatika dan kemampuan dalam membangun perangkat lunak pada ilmu kriptografi dan steganografi yaitu sebuah aplikasi dengan mengimplementasikan algoritma *Rivest Code 4* (RC4) dan Steganografi *End Of File* (EOF). Sehingga penulis bisa merasakan manfaatnya. Dan bagi masyarakat pada umum bisa dipergunakan sebagai aplikasi alternatif untuk keamanan data.

4. Landasan Teori

4.1.Definisi Kriptografi

Menurut Ariyus [3], kriptografi, secara terminologinya adalah ilmu dan seni untuk menjaga kerahasiaan pesan ketika pesan kirim dari suatu tempat ketempat yang lain. Namun pada pengertian moderen kriptografi adalah ilmu yang berdasarkan pada teknik matematika untuk urusan dengan keamanan informasi seperti kerahasian, keutuhan data dan otentikasi identitas.

Kriptografi adalah dibidang ilmu yang sangat penting keberadaannya untuk menjaga kerahasian dan keamanan suatu informasi. Data yang ingin diacak dan data tersebut dapat dan dimengerti disebut teks asli *(Plain Text)* [4]. *Plain Text* diacak dengan mengunakan kunci enskripsi *(Encryption Key). Plain Text* yang tersandi disebut *ciphertext*. Proses pengacakan tersebut enskripsi *(Encrytion).* Kemudian proses untuk mengembalikan *ciphertext* ke *Plain Text* dinamakan deskripsi *(Decryption).*kunci yang digunakan pada tahap deskripsi di sebut kunci deskripsi *(Decryption Key)*

Kriptografi mempunyai dua komponen utama yaitu enskripsi dan deskripsi. Selain itu dibutuhkan kunci untuk mengubah plainteks menjadi chiperteks dan juga sebaliknya. Tanpa kunci plainteks tidak bisa mengenskrip masukan menjadi chiperteks, demikian juga sebaliknya. Kerahasiaan kunci sangatlah penting, apabila kerahasiaannya terbongkar maka isi pesan akan terbongkar Berikut adalah skema yang mengilustrasikan enkripsi dan deskripsi.kunci enkripsi kunci deskripsi .

Menurut Anjik dan Rianto [4], kriptografi mempunyai beberapa tujuan sebagai berikut :

1. Kerahasian (*Confidentiality)*
   Memberikan kerahasian pesan dan menyimpan data dengan menyembunyikan informasi lewat teknik-teknik enkripsi.
2. Integritas *(Integrity)*
   Memberikan jaminan untuk tiap bagian bahwa pesan tidak akan mengalami perubahan dari saat pesan dibuat sampai pesan dibuka.
3. Penghindaran Penolakan *(Non-Repudantion)*
   Memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang apabila ia mencoba menyangkal memiliki dokumen tersebut.
4. Autentikasi *(Authentication)*
   Memberikan dua layanan. *Pertama*, mengidentifikasi keaslian suatu pesan dan memberikan jaminan keautentikannya. *Kedua*, menguji identitas seseorang apabila ia memasuki

suatu sistem. Proses ini untuk menjamin keaslian suatu data.

4.2. Jenis-Jenis Algoritma Kriptografi

Algoritm kriptografi adalah algoritma yang berfungsi untuk melakukan tujuan dari ilmu kriptografi itu sendiri. Algoritma kriptografi terdiri dari 2 bagian fungsi, yaitu :

1. ENKRIPSI *(encryption),* Proses tranformasi dari *plaintext* menjadi *ciphertext* disebut proses *Encipherment* atau enkripsi *(encryption).*
2. DEKRIPSI (*decryption*), Proses mentransformasikan kembali *ciphertext* menjadi *plaintext* disebut proses dekripsi (*decryption*).

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya [5]. Algoritma yang memakai kunci simetri di antaranya adalah :

a. Data Encryption Standard (DES),
b. RC2, RC4, RC5, RC 6,
c. International Data Encryption Algorithm (IDEA),
d. Advanced Encryption Standard (AES),
e. On Time Pad (OTP),
f. A5, dan lain sebagainya.

4.3.Algoritma Kode Rivest's

Ariyus [3] menyatakan bahwa kunci simetri *(secret key)* adalah salah satu algoritma kriptografi yang digunakan untuk melakukan enkripsi dan deskripsi, dengan menggunakan kunci rahasia untuk setiap bit dan bit per blok. Salah satu ilmuwan yang mendalami metode ini adalah Ronald Linn Rivest dari Laboratorium RSA. RSA Security Inc, sebuah organisasi komersial yang merupakan hasil marger antara security dynamic dan RSA, Ron Rivest, Adi Shamir dan Leonard Adleman. Berkat mereka empat buah algoritma kunci simetri RC2, RC4, RC5 dan RC6 mendapat penghargaan Turing Award dari ACM (Association for Computing Machinery) pada tahun 2002 silam.

4.4.Kode Rivest 4 (RC4)

Kriptografi terdiri atas kunci simetris dan asimetris. Algoritma kunci simetris termasuk algoritma yang masih sering digunakan dalam pembuatan algoritma kriptografi. Bila dibandingkan dengan block cipher, maka stream cipher memiliki algoritma yang lebih sederhana. Salah satunya adalah algoritma RC4, yaitu algoritma yang masih banyak digunakan karena kesederhanaannya secara matematis dan kecepatan prosesnya yang tidak kalah cepat dibandingkan dengan algoritma yang lebih rumit.

Algoritma RC4 merupakan jenis *stream cipher*, yaitu operasi enkripsi dilakukan per karakter 1 byte untuk sekali operasi. Dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variable. Algoritma RC4 memiliki dua fase, yaitu setup kunci dan pengenkripsian. Dengan kunci yang sama maka pada proses deskripsi data kembali ke bentuk semula. Sampai saat ini algoritma RC4 masih banyak digunakan orang dalam mengenkripsi informasi karena algoritmanya yang sederhana namun memiliki kecepatan yang hampir sama dibandingkan algoritma yang lebih rumit. Dalam aplikasinya data-data penting tidak hanya berupa dokumen atau teks yang dapat di enkripsi tetapi data gambar, video dan suara yang sebagian orang merasa perlu mengamankannya sebagai data pribadi dapat juga untuk di enkripsi.

4.5. Definisi Steganografi

Setelah mengamati runtutan sejarah dan perkembangan pengertian serta penggunaan, maka dapat disimpulkan bahwa steganografi merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan sesuatu informasi. Steganografi dapat dipandang atau dikelompokkan sebagai salah satu bagian atau cabang dari ilmu komunikasi. Pada era informasi digital, steganografi merupakan teknik dan seni menyembunyikan informasi dan data digital dibalik informasi digital lain, sehingga informasi digital yang sesungguhnya tidak kelihatan atau dapat tersamarkan. Terdapat beberapa istilah yang berkaitan dengan steganografi [5].

1. *Hiddentext* atau *embedded massage* pesan yang disembunyikan.
2. *Covertext* atau *cover-object* pesan yang digunakan untuk menyembunyikan *embedded massage.*
3. *Stegotext* atau *stego-object* pesan yang berisi *embedded massage.*

Steganografi membutuhkan dua properti yaitu media penampung dan data rahasia atau informasi yang akan disembunyikan. Untuk penerapan steganografi pada komputer banyak menggunakan file digital sebagai wadah penyembunyiannya. File tersebut dapat berupa file citra, file audio (suara), video, maupun file teks. Data rahasia dapat berupa file dengan tipe apa saja, misalnya file exe, file teks, file suara dan lain sebagainya.

4.6.Teknik Steganografi EOF

Metode EOF (*End Of File*) merupakan salah satu teknik yang menyisipkan data pada akhir file dan pengembangan daripada metode LSB. Teknik ini dapat digunakan untuk menyisipkan data yang ukurannya sama dengan ukuran file sebelum

disisipkan data ditambah dengan ukuran data yang disisipkan kedalam file tersebut.

Teknik EOF tidak akan mengubah isi awal dari file yang disisipi. Sebagai contoh, jika akan menyisipkan sebuah pesan kedalam sebuah file gambar, file tersebut tidak akan berubah. Ini yang menjadi salah satu keunggulan metode EOF dibandingkan metode steganografi yang lain. Karena disisipkan pada akhir file, pesan yang disisipkan tidak akan bersinggungan dengan isi file, hal ini menyebabkan integritas data dari file yang disisipi tetap dapat terjaga. Namun, metode EOF akan mengubah besar ukuran file sesuai dengan ukuran pesan yang disisipkan kedalam file awal namun tidak mengubah citra daripada media yang dipakai sebagai tempat penyisipan pesan tersebut [6].

5. Analisis Masalah

Kriptografi merupakan salah satu cara yang paling umum dalam mengamankan suatu informasi. Selain mudah dalam mengimplementasikannya teknik kriptografi juga tidak membutuhkan medium perantara. Saat ini banyak algoritma yang digunakan dalam kriftografi salah satunya adalah RC4. Tetapi seberapan rumit algoritma yang digunakan untuk enkripsi, masalah keamanan data tetap ada karena ada pihak-pihak yang juga mengembangkan teknik deskripsi untuk mencuri data tersebut. Hal ini dikarenakan enkripsi tidak dilakukan secara tersembunyi.

Salah satu contoh yang dapat digunakan dalam menganalogikan kriptografi dan steganografi adalah persoalan tahanan dipenjara. Umpama, ada dua orang tahan anggaplah si A dan si B yang mendekam dalam sel yang terpisah, salah satu cara mereka berkomunikasi adalah lewat surat dengan perantara seorang sipir si C. Si A menulis surat pada selembar kertas, lalu surat diberikan kepada C untuk diantar kepada B , C tentu dapat membaca isi surat A sebelum surat tersebut sampai ketangan B. Si A menulis pesan rahasia kepada si B mengenai rencana waktu pelarian mereka dari penjara. Pesan yang dikirim berbunyi “Lari jam satu”. Jika menggunakan kriptografi, maka bunyi pesan tersebut akan diubah menjadi *chiphertext*. “f4yc*fdf^gy”. Si C yang menyampaikan pesan tersebut pasti curiga dan menduga si A mengirim pesan rahasia dan mulai berusaha untuk memecahkan *chiphertext* tersebut.

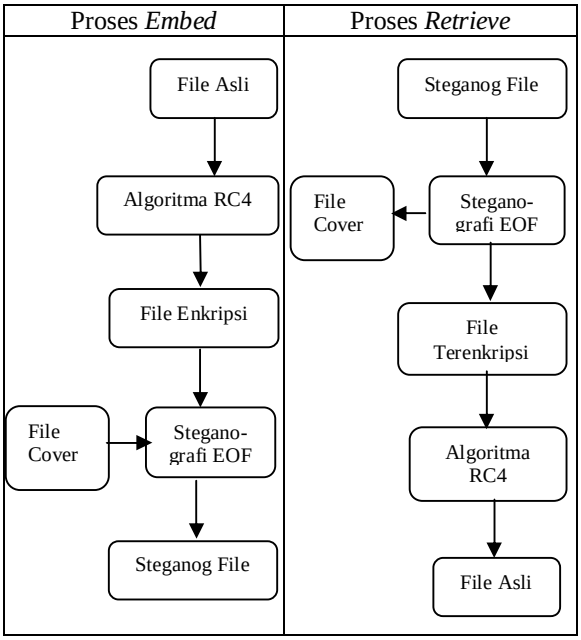
Steganografi memiliki sudut pandang yang berbeda kepada pemecahan masalah pengiriman pesan tersebut. Dengan mengunakan steganografi, maka setiap huruf pesan rahasia akan disisipkan kedalam setiap awal pada pesan yang akan dikirim, menjadi : “**L**upakan **A**sal **R**umor **I**tu, **J**aga **A**gar **M**atamu **S**ehat **A**tau **T**urunkan **U**banmu”. Si C yang menyampaikan pesan tidak akan curiga dan menganggap si A sedang bergurau dengan si B.

6. Perancangan Algoritma Steganografi EOF dan Kriptografi RC4

Setelah melihat masalah keamanan data tersebut, maka dibutuhkan suatu program aplikasi yang dapat menggabungkan kriptografi dan steganografi. Masalah yang diselesaikan dalam skripsi ini adalah menerapkan algoritma steganografi EOF dan algoritma kriptografi RC4 stream chipher yang digunakan untuk enkripsi dan dekripsi, penyisipan dan ektrasi file.

6.1. Perancangan Algoritma

Algoritma yang diimplementasikan adalah menggabungkan metode steganografi end of file dan metode kriptografi RC4 stream cipher dalam meng*embed* dan men*retrieve* file. Pada proses *embed* dan *retrieve* file dapat dilihat melalui gambar 1 dibawah ini:

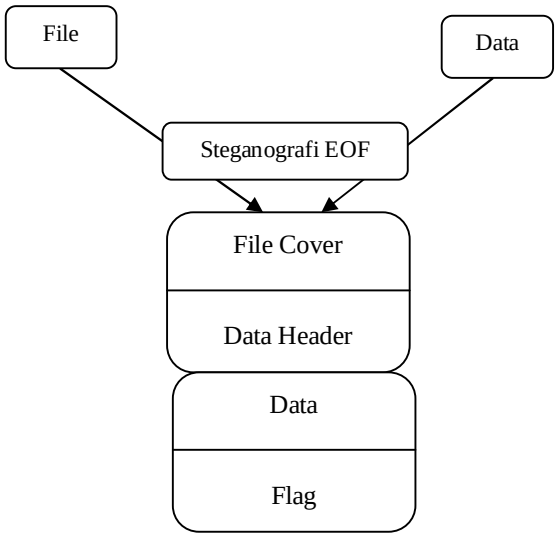


Gambar 1. Proses *Embed* dan *Retrieve*

6.2 Algortima Steganografi EOF

Metode EOF merupakan sebuah metode yang di adaptasi dari metode penanda akhir file. Metode ini bekerja dengan cara meletakkan data secara langsung di akhir file. EOF digunakan sebagai salah satu metode dalam melakukan penyembunyian data digital kedalam digital lainnya. Metode EOF memanfaatkan kelemahan indra manusia yang tidak sensitif, sehingga seakan-akan tidak terjadi perubahan yang terlihat antara sebelum dan sesudah pesan disisipkan kedalam file digital yang digunakan sebagai penampung dari pesan

yang disisipkan. Secara umum media steganografi yang akan disisipkan memiliki struktur sebagai berikut:



sumber: http://www.cenadep.org

Gambar 2. Struktur Steganografi End Of File

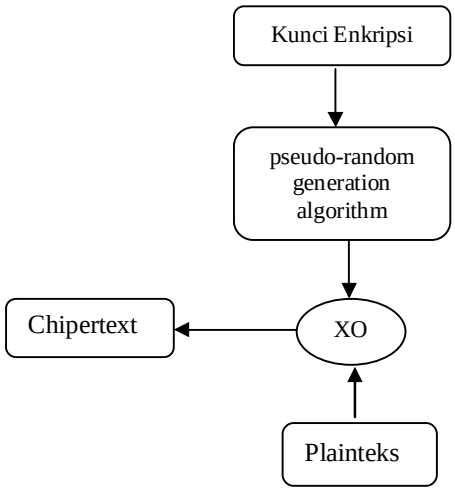
Blok pertama pada diagram diatas merupakan file asli atau file cover yang akan menjadi media steganografi end of file, blok kedua yaitu data header terdiri dari password, posisi awal data dan panjang blok data, blok data berisi data yang disisipkan, blok terakhir adalah flag yang mencatat posisi header data dan sebuah penanda untuk menentukan apakah file cover sudah berisi data atau belum. Penanda (FLAG) digunakan sebagai penentu posisi awal data header pada file media.

6.3. Algortima RC 4 Stream Cipher

Algoritma kriptografi RC4 merupakan salah satu algoritma berjenis *stream cipher*. Algoritma ini akan memproses data dalam ukuran byte demi byte. Algortima ini dapat melakukan enkripsi dan deskripsi pada panjang data yang variabel atau dinamis.

Algoritma RC4 menggunakan dua buah indeks yaitu *i* dan *j* di dalam algoritmanya. Indeks *i* digunakan untuk memastikan bahwa suatu elemen berubah, sedangkan indeks *j* akan memastikan bahwa suatu elemen berubah secara random. Secara garis besar algoritma dari metode RC4 Stream Cipher ini terbagi menjadi dua bagian, yaitu key setup dan stream RC4 Stream Cipher generation. Pada key setup terdapat tiga tahapan proses di dalamnya, yaitu Inisialisasi S-Box, menyimpan key dalam Key Byte Array, permutasi pada S-Box. Pada Stream Generation akan menghasilkan nilai pseudo random yang akan dikenakan operasi XOR untuk menghasilkan *ciphertext* ataupun sebaliknya yaitu untuk menghasilkan plaintext.

Enkripsi dihasikan dari setup kunci dimana kunci akan di XOR-kan dengan plaintext untuk menghasilkan teks yang terenkripsi. XOR merupakan operasi logika yang membandingkan dua bit biner. Jika bernilai beda maka akan dihasilkan nilai 1. Jika kedua bit sama maka hasilnya adalah 0. Kemudian penerima pesan akan mendeskripkannya dengan meng XOR-kan kembali dengan kunci yang sama agar dihasilkan pesan dari plaintext tersebut.



sumber: informatika.stei.itb.ac.id

Gambar 3. Deskripsi Algoritma RC4

Algoritma RC4 menginisalisasi state-array dan penghasilan kunci enkripsi serta pengenkripsiannya. Dalam penginisialisasian state-array, terdapat 2 state array yang harus diinisialisasi, S dan K. Array S sebesar 256 byte diinisialisasi dengan angka dari 0 sampai dengan 255. Sedangkan array K sebesar 256 byte diisi dengan key dengan panjang 1-256 byte secara berulang sampai seluruh array K terisi penuh. Setelah itu, dilakukan Key Scheduling Algoritma untuk menghasilkan permutasi dari array S berdasarkan key yang tersedia. Langkah-langkah algoritma kriptografi RC4 sebagai berikut:

1. Inisialilasi S-Box.

a. Isi S-Box secara berurutan, yaitu S0, S1, S2, ..., S255

b. Lakukan padding kunci K sehingga panjang kunci K= 256.

c. Lakukan pertukaran dan pengisian pada S-Box dengan kunci K sebagai berikut:
   j=0
   for i=0 to 255
   j=(j=+Si+Ki)mod 256
   swap Si dan Sj

Fungsi swap merupakan penukarkan nilai S ke-i dengan nilai S ke-j. Jika panjang kunci K<256 lakukan pedding sehingga panjang kunci menjadi 256. Contoh

jika K="123" terdiri dari 3 byte, maka lakukan padding dengan menambahkan byte K="123123123123" sampai panjang K mencapai 256 byte.

2. Proses enkripsi dn deskripsi RC4
   i=0
   j=0
   for idx=0 to PanjangPlaintext -1 do
   i= (i+1)mod 256
   j= (j+Si)mod 256
   swap Si dengan Sj
   t= (Si+Sj)mod 256
   k=St
   buffidx=k XOR buffidx

buff merupakan pesan yang akan dienkripsi atau dideskripsi. Hasil akhir dari
proses diatas adalah buff yang berisi pesan yang telah dienkripsi atau deskripsikan.
Untuk menujukkan cara kerja algoritma RC4 dengan mengunakan kata "STMIK" sebagai plaintext dengan kunci "12345abc". Tahap awal inisialisasi S-Box untuk i=0 sampai 255.

Tabel 2. S-Box inisialisasi awal

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Selanjutnya menyimpan key dalam key byte array, kunci (key) yang akan digunakan untuk mengenkripsi atau dekripsi akan diinputkan kedalam array berukuran 256 byte secara berulang sampai array terisi penuh. Urutan kunci yang digunakan dalam kode ASCII bisa dilihat dalam tabel 3 berikut.

Isi array key (K-Box) dengan kunci yang digunakan. Jika panjang kunci K<256 lakukan pedding sehingga panjang kunci menjadi 256.

Tabel 3. Kode ASCII untuk setiap kunci yang digunakan

Kunci	Kode ASCII (Desimal)
1	49
2	50
3	51
4	52
5	53
A	97
B	98
C	99

Tabel 4. Urutan Array Key (K-Box)

49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99

49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99
49	50	51	52	53	97	98	99	49	50	51	52	53	97	98	99

Algoritma untuk melakukan pertukaran dan pengisian pada S-Box dengan kunci K (K-Box) sebagai berikut:

j=0
for i=0 to 255
j=(j+Si+Ki)mod 256
swap Si dan Sj

dari algoritma tersebut akan diperoleh nilai S-Box yang telah mengalami proses transposisi, sehingga urutan S-Boxnya menjadi acak.

Tabel 5. S-Box setelah ditransposisi

49	239	20	208	9	186	215	130	83	176	58	81	109	219	75	189
169	7	121	205	144	40	153	124	132	179	47	240	253	123	222	125
206	211	118	64	54	27	36	162	140	98	212	11	10	157	170	17
157	207	91	141	231	168	66	190	188	146	255	88	80	41	241	178
24	102	13	152	203	113	86	220	210	95	126	34	247	232	85	65
171	59	107	143	175	19	248	63	110	198	101	226	14	133	201	42
223	142	181	242	52	114	60	120	28	93	199	166	195	163	44	155
92	194	159	139	117	158	228	2	249	134	151	116	193	32	246	160
150	230	106	119	55	77	209	187	149	135	243	245	128	99	148	90
22	0	97	252	43	72	221	165	214	132	30	68	156	105	37	164
235	173	229	16	225	238	161	115	26	62	51	136	56	4	177	131
67	39	217	244	29	184	3	234	191	197	111	69	122	202	71	137
218	236	46	145	8	89	78	182	192	174	21	185	183	84	79	224
31	112	45	1	35	100	82	94	104	227	76	38	108	53	251	103
154	6	50	70	157	138	23	74	196	57	129	87	73	127	216	250
172	25	200	12	227	204	33	18	254	213	96	61	15	233	180	237

Tahapan selanjutnya mencari penghasilan kunci yang digunakan untuk mengenkripsi plaintext "STMIK" dengan mengunakan S-Box yang telah ditransposisi. karena plaintext berjumlah lima karakter dibutuhkan satu kunci dan satu kali pengoprasian XOR untuk setiap karakter pada plaintext. Inisialisasi i dan j sama dengan 0 kemudian set i=(i+1) mod 256 dan j=(j+Si)mod 256 dan swap Si dengan Sj (Si+Sj)mod 256 byte yang acak untuk enkripsi.

Iterasi Pertama
i=0 j=0
i=(0+1)mod 256 = 1
j=(0+S[1])mod 256 = (0+239)mod 256 = 239
swap S[1]dengan S[239]

Tabel 6. Transposisi S[1] dengan S[239]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
49	250	20	208	9	186	215	130	83	176	58	81	109	219	75	189
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
169	7	121	205	144	40	153	124	132	179	47	240	253	123	222	125
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
206	211	118	64	54	27	36	162	140	98	212	11	10	157	170	17
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
157	207	91	141	231	168	66	190	188	146	255	88	80	41	241	178
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
24	102	13	152	203	113	86	220	210	95	126	34	247	232	85	65
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
171	59	107	143	175	19	248	63	110	198	101	226	14	133	201	42
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111

223	142	181	242	52	114	60	120	28	93	199	166	195	163	44	155
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
92	194	159	139	117	158	228	2	249	134	151	116	193	32	246	160
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
150	230	106	119	55	77	209	187	149	135	243	245	128	99	148	90
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
22	0	97	252	43	72	221	165	214	132	30	68	156	105	37	164
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
235	173	229	16	225	238	161	115	26	62	51	136	56	4	177	131
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
67	39	217	244	29	184	3	234	191	197	111	69	122	202	71	137
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
218	236	46	145	8	89	78	182	192	174	21	185	183	84	79	224
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
31	112	45	1	35	100	82	94	104	227	76	38	108	53	251	103
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
154	6	50	70	157	138	23	74	196	57	129	87	73	127	216	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
172	25	200	12	227	204	33	18	254	213	96	61	15	233	180	237

K1= S(S[1]+S[239])mod 256= (250+239)mod 256= 233
K1=11101001
i=1 dan j=239
Iterasi Kedua
i=1 j=239
i=(1+1)mod 256= 2
j=(0+S[2])mod 256 = (239+20)mod 256 =3
swap S[2]dengan S[3]

Tabel 7. Transposisi S[2]dengan S[3]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
49	250	208	20	9	186	215	130	83	176	58	81	109	219	75	189
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
169	7	121	205	144	40	153	124	132	179	47	240	253	123	222	125
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
206	211	118	64	54	27	36	162	140	98	212	11	10	157	170	17
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
157	207	91	141	231	168	66	190	188	146	255	88	80	41	241	178
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
24	102	13	152	203	113	86	220	210	95	126	34	247	232	85	65
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
171	59	107	143	175	19	248	63	110	198	101	226	14	133	201	42
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
223	142	181	242	52	114	60	120	28	93	199	166	195	163	44	155
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
92	194	159	139	117	158	228	2	249	134	151	116	193	32	246	160
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
150	230	106	119	55	77	209	187	149	135	243	245	128	99	148	90
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
22	0	97	252	43	72	221	165	214	132	30	68	156	105	37	164
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
235	173	229	16	225	238	161	115	26	62	51	136	56	4	177	131
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
67	39	217	244	29	184	3	234	191	197	111	69	122	202	71	137
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207

218	236	46	145	8	89	78	182	192	174	21	185	183	84	79	224
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
31	112	45	1	35	100	82	94	104	227	76	38	108	53	251	103
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
154	6	50	70	157	138	23	74	196	57	129	87	73	127	216	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
172	25	200	12	227	204	33	18	254	213	96	61	15	233	180	237

K1= S(S[2]+S[3])mod 256 = (208+20)mod 256=228
K2=11100100
i=2 dan j=3
Iterasi Ketiga
i=(2+1)mod 256 = 3
j=(0+S3)mod 256 = (3+20)mod 256 = 23
swap S[3]dengan S[23]

Tabel 8. Transposisi S[3] dengan S[23]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
49	250	208	124	9	186	215	130	83	176	58	81	109	219	75	189
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
169	7	121	205	144	40	153	20	132	179	47	240	253	123	222	125
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
206	211	118	64	54	27	36	162	140	98	212	11	10	157	170	17
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
157	207	91	141	231	168	66	190	188	146	255	88	80	41	241	178
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
24	102	13	152	203	113	86	220	210	95	126	34	247	232	85	65
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
171	59	107	143	175	19	248	63	110	198	101	226	14	133	201	42
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
223	142	181	242	52	114	60	120	28	93	199	166	195	163	44	155
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
92	194	159	139	117	158	228	2	249	134	151	116	193	32	246	160
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
150	230	106	119	55	77	209	187	149	135	243	245	128	99	148	90
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
22	0	97	252	43	72	221	165	214	132	30	68	156	105	37	164
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
235	173	229	16	225	238	161	115	26	62	51	136	56	4	177	131
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
67	39	217	244	29	184	3	234	191	197	111	69	122	202	71	137
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
218	236	46	145	8	89	78	182	192	174	21	185	183	84	79	224
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
31	112	45	1	35	100	82	94	104	227	76	38	108	53	251	103
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
154	6	50	70	157	138	23	74	196	57	129	87	73	127	216	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
172	25	200	12	227	204	33	18	254	213	96	61	15	233	180	237

K3= S(S3+S23)mod 256 = (124+20)mod 256= 144
K3=10010000
i=3 dan j=23
Iterasi Keempat

i=(3+1)mod 256 = 4
j=(0+S[4])mod 256 = (23+9)mod 256= 32
swap S[4] dengan S[32]

Tabel 9. Transposisi S[4] dengan S[32]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
49	250	208	124	206	186	215	130	83	176	58	81	109	219	75	189
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
169	7	121	205	144	40	153	20	132	179	47	240	253	123	222	125
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
9	211	118	64	54	27	36	162	140	98	212	11	10	157	170	17
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
157	207	91	141	231	168	66	190	188	146	255	88	80	41	241	178
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
24	102	13	152	203	113	86	220	210	95	126	34	247	232	85	65
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
171	59	107	143	175	19	248	63	110	198	101	226	14	133	201	42
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
223	142	181	242	52	114	60	120	28	93	199	166	195	163	44	155
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
92	194	159	139	117	158	228	2	249	134	151	116	193	32	246	160
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
150	230	106	119	55	77	209	187	149	135	243	245	128	99	148	90
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
22	0	97	252	43	72	221	165	214	132	30	68	156	105	37	164
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
235	173	229	16	225	238	161	115	26	62	51	136	56	4	177	131
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
67	39	217	244	29	184	3	234	191	197	111	69	122	202	71	137
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
218	236	46	145	8	89	78	182	192	174	21	185	183	84	79	224
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
31	112	45	1	35	100	82	94	104	227	76	38	108	53	251	103
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
154	6	50	70	157	138	23	74	196	57	129	87	73	127	216	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
172	25	200	12	227	204	33	18	254	213	96	61	15	233	180	237

K4= S(S[4]+S[32])mod 4 = (206+9)mod 256= 215
K4=11010111
i=4 dan j=32
Iterasi Kelima
i=(4+1)mod 256 = 5
j=(0+S[5])mod 256 = (32+186)mod 256 = 218
swap S[5] dengan S[218]

Tabel 10. Transposisi S[5] dengan S[218]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
49	250	208	124	206	76	215	130	83	176	58	81	109	219	75	189
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
169	7	121	205	144	40	153	20	132	179	47	240	253	123	222	125
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
9	211	118	64	54	27	36	162	140	98	212	11	10	157	170	17

48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
157	207	91	141	231	168	66	190	188	146	255	88	80	41	241	178
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
24	102	13	152	203	113	86	220	210	95	126	34	247	232	85	65
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
171	59	107	143	175	19	248	63	110	198	101	226	14	133	201	42
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
223	142	181	242	52	114	60	120	28	93	199	166	195	163	44	155
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
92	194	159	139	117	158	228	2	249	134	151	116	193	32	246	160
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
150	230	106	119	55	77	209	187	149	135	243	245	128	99	148	90
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
22	0	97	252	43	72	221	165	214	132	30	68	156	105	37	164
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
235	173	229	16	225	238	161	115	26	62	51	136	56	4	177	131
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
67	39	217	244	29	184	3	234	191	197	111	69	122	202	71	137
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
218	236	46	145	8	89	78	182	192	174	21	185	183	84	79	224
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
31	112	45	1	35	100	82	94	104	227	165	38	108	53	251	103
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
154	6	50	70	157	138	23	74	196	57	129	87	73	127	216	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
172	25	200	12	227	204	33	18	254	213	96	61	15	233	180	237

K1= S(S[5]+S[168])mod 256 = (76+186)mod 256= 6
K5=0011 0110

Setelah menemukan kunci untuk tiap karakter, maka di lakukan operasi XOR antara karakter plaintext dengan kunci yang dihasilkan.

Tabel 11. ASCII Tiap Karakter yang di gunakan

HURUF	Kode ASCII (Binary 8 bit)
S	01010011
T	01010100
M	01001101
I	01001001
K	01001011

Sumber*: http://www.ascii-code.com/*

Tabel 12. Proses XOR kunci Enkripsi dengan Plaintext

	S	T	M	I	K
Plain-text	0101 0011	0101 0100	0100 1101	0100 1001	0100 1011
Key	1110 1001	1110 0100	1001 0000	1101 0111	0011 0110
Cipher-text	1011 1010	1011 0000	1101 1101	1001 1110	0111 1101
	º	°	Ý	Ž	}

Sumber http://www.ascii-code.com/

Setelah terkirim, pesan yang telah dienkripsi akan didekripsikan. Proses pendekripsian dilakukan dengan proses XOR antara kunci dekripsi yang sama dengan kunci dekripsi dengan cipherteks.

Tabel 13. Proses XOR kunci Deskripsi dengan ciphertext

	º	°	Ý	Ž	}
Ciphertext	1011 1010	1011 0000	1101 1101	1001 1110	0111 1101
Key	1110 1001	1110 0100	1001 0000	1101 0111	0011 0110
Plaintext	0101 0011	0101 0100	0100 1101	0100 1001	0100 1011
	S	T	M	I	K

Sumber http://www.ascii-code.com/

7. Hasil dan Pembahasan

Membahas tentang pengujian dan analisa hasil program yang telah dibuat. Tujuan dari pengujian ini adalah untuk mengetahui apakah aplikasi yang telah dibuat sesuai dengan perancangannya.

8. Implementasi Sistem

Untuk menjalankan aplikasi pengamanan data menggunakan kriptografi dan steganografi ini memerlukan beberapa komponen pendukung. Komponen-komponen pendukungnya adalah perangkat keras dan perangkat lunak yang digunakan dalam membuat program tersebut hingga selesai.

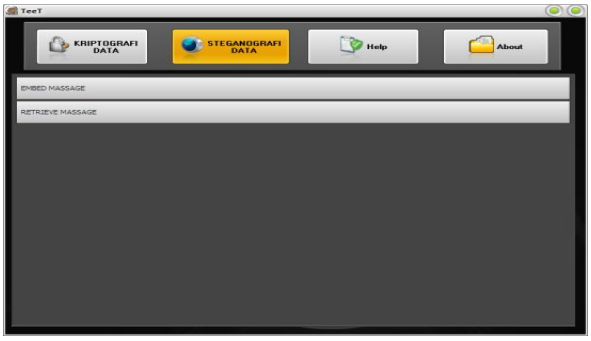
User Interface

Pada saat aplikasi dijalankan, akan muncul form layar utama dengan menampilkan menu kriptografi data, steganografi data, help, dan about.

Gambar 4. Tampilan Layar Utama

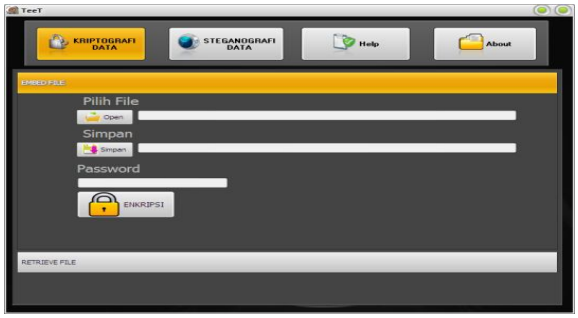
Tampilan Menu Kriptografi Data

Dari keempat menu yang tampil pada layar utama kriptografi data, steganografi data, help, dan about. Jika pengirim ingin melakukan pengacakan file dengan mengunakan kriptografi, pengirim dapat memilih menu kriptografi data.



Gambar 5.Tampilan Menu Kriptografi Data

Didalam menu kriptografi data, ada dua submenu yaitu embed file dan retreive file. Jika pengirim ingin melakukan pengacakan file, maka pilih submenu embed file maka akan tampil layar seperti gambar di bawah ini.

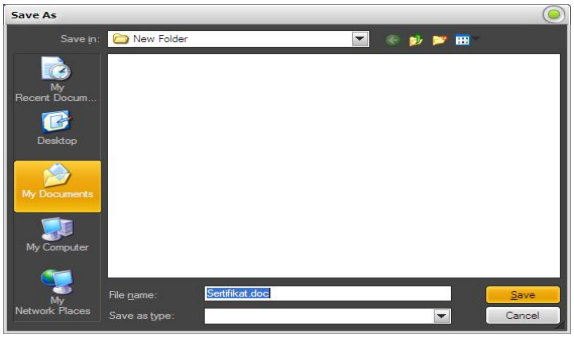


Gambar 6. Tampilan Sub Menu Embed File

Setelah sub menu embed file ditampilkan, maka yang harus dilakukan adalah mencari file yang akan dienkripsi dengan menekan tombol open, setelah itu akan tampil kotak menu open dialog sebagai berikut:

Gambar 7. Tampilan Sub Menu pilih File

Setelah itu menempatkan destinasi file hasil enkripsi. dengan menekan tombol Save setelah itu akan tampil kotak menu Save dialog sebagai berikut:

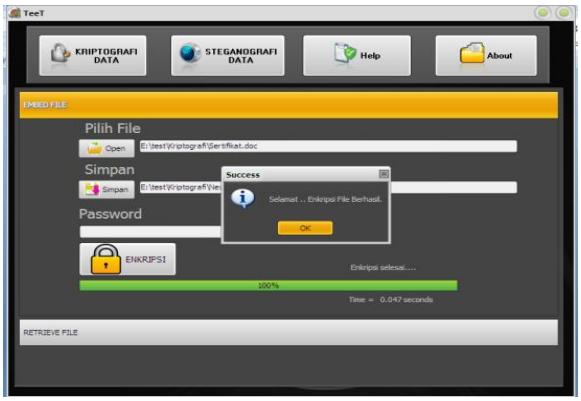


Gambar 8.Tampilan Sub Menu Simpan

Lalu memasukkan kunci enkripsi, dan menekan tombol "Enkripsi" maka pengirim diminta untuk memasukkan ulang kunci enkripsi pada form konfirmasi password

Gambar 9. Konfirmasi password

Apabila password yang diinput sama dengan sebelum maka aplikasi akan menampilkan pesan success, setelah itu file akan terenkripsi



Gambar 10. Informasi File Sukses Dienkripsi

9. Pengujian Sistem

Pengujian sistem dilakukan pada lingkungan perangkat lunak sebagai berikut:

1. Pengujian terhadap fungsi-fungsi aplikasi
   Hal ini dilakukan apakah semua fungsi berjalan dengan baik sesuai dengan yang diharapkan. Pengujian terhadap fungsi-fungsi aplikasi dengan metode blackbox. Pengujian blackbox berfokus pada persyaratan fungsional perangkat lunak yang dibangun/dikembangkan.
2. Penggujian terhadap *attack*/serangan
   Tujuan aplikasi ini adalah untuk pengamanan data, jadi keamanan dari *attack*/serangan adalah syarat mutlak dari sebuah aplikasi. Pada aplikasi ini pengujian dilakukan pada file cipherteks hasil proses enkripsi menggunakan algoritma RC4. Sedangkan pengujian terhadap *attack*/serangan dilakukan dengan metode *trial and error*. Dan dapat dilihat pada tabel 2.

9.1 Rencana Pengujian

Pengujian perangkat lunak dilaksanakan dengan tujuan agar perangkat lunak / sistem yang dibangun memiliki kualitas dan kinerja yang optimal baik dalam performa perangkat lunak dalam melakukan proses enkripsi, deskripsi, penyisipan dan ekstraksi. Adapun batasan pengujian yang akan dilakukan sebagai berikut:

1. Pengujian kesesuaian kapasitas file yaitu pengujian kesesuaian antara data yang berhasil dideskripsi dengan data yang dienkripsi
2. Pengujian kualitas file yaitu pengujian sama tidaknya file original dengan file yang sudah tersisipkan data.
3. pengujian ketahanan file yang terenkripsi tehadap *attack*/serangan dengan metode trial and error

Tabel 14. Rencana Pengujian Aplikasi

Kelas Uji	Butir Uji	Jenis Pengujian
Enkripsi	Melakukan enkripsi plaintext berformat *.doc, *.jpg, *.mp3 dan *.flv	blackbox
Deskripsi	Mendeskripsikan cipherteks ke plainteks	blackbox
Penyisipan	Menyisipkan ciphertext kedalam file cover yang berformat *.doc, *.jpg, *.mp3 dan *.flv	blackbox
Ekstraksi	Mengekstraksi ciphertext dari file cover yang berformat *.doc, *.jpg, *.mp3 dan *.flv	blackbox
Attack/Serangan	Mencoba memasukkan kemungkinan password untuk mendeskripsikan ciphertext.	Trial and Error

10. Kesimpulan

Dari pengujian-pengujian yang dilakukan, dapat ditarik kesimpulan secara keseluruhan sebagai tersebut:

1. Ukuran besaran file setelah disisipkan adalah penjumlahan antara media yang disisipkan dengan media aslinya. Pengujian aplikasi ini berhasil sesuai dengan metode algoritma EOF.
2. Sesuai dengan prinsip steganografi antara media asli dan media yang disisipkan tidak terlalu terlihat perbedaannya secara kasat mata, yang dimaksudkan untuk menyembunyikan kecurigaan.
3. Media file pembawa juga tidak akan mengalami kerusakan data terhadap pesan yang disisipkan apabila nama dari media pembawa tersebut diganti, selama yang diganti bukan dari ektensi filenya saja.

Referensi

[1] Jamaludin, (2010), *"Aplikasi Keamanan Informasi Menggunakan Teknik Steganografi dengan Metode Least Significant Bit (LSB) Insertion dan RC4"*, Jurnal FTI-UIN, Jakarta

[2] Fiansyah, Endang, (2008), *"Implementasi Algoritma Dasar RC4 Stream Chiper dan Pengacakan Plain Teks dengan Teknik Dynamic Blocking pada Aplikasi Sistem Informasi Kegiatan Skripsi di Departemen Teknik Elektro"*, Jurnal FTI-UI, Depok.

[3] Ariyus, Dony. (2006). *Pengantar Ilmu Kriptografi Teori, Analisa dan Implementasi*. Yogyakarta : Andi

[4] Munir, Rinaldi. (2006). *Kriptografi*. Bandung : Informatika

[5] Sadikin, Rifki. 2012. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta : Andi

[6] http://jurnal.usu.ac.id/index.php/alkhawarizmi/article/download/500/262