ANALISA DAN IMPLEMENTASI STEGANOGRAFI UNTUK PELAPORAN INTERNAL PERUSAHAAN MENGGUNAKAN ALGORITMA *DATA ENCRYPTION STANDARD* (DES) DAN METODE *END OF FILE* (EOF) BERBASIS JAVA *PROGRAMMING*

Tri Ika Jaya Kusumawati[1)], Devi Anisah[2)]

[1,2)]Program Studi Magister Ilmu Komputer, Program Pascasarjana, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5869225
[1]tri.ikajaya@budiluhur.ac.id,[2]anisah.devi@gmail.com

ABSTRAK

Keamanan data maupun informasi menjadi bagian yang sangat penting dalam sebuah keamanan perusahaan agar mampu bertahan dalam persaingan dengan perusahan lain. Namun pada kenyataanya timbul masalah tentang bagaimana mengimplementasikan suatu sistem keamanan data yang mampu melakukan proses enkripsi dan deskripsi suatu data file khususnya laporan internal perusahaan. Steganografi menjadi salah satu teknologi untuk menjaga pengiriman informasi maupun data perusahan agar tidak dengan mudah dibajak oleh pihak yang tidak bertanggung jawab. Steganografi ialah penyembunyian pesan dalam sebuah media 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, pesan akan disisipkan pada akhir nilai file. Dalam penerapannya steganografi menggunakan konsep algoritma untuk menyisipkan file, teks maupun gambar lain dalam media utamanya, salah satu yang dapat digunakan adalah algoritma DES (Data Encryption Standard). Penulis telah merancang sebuah aplikasi dengan perpaduan teknik steganografi dengan menggunakan metode EOF dan menerapkan konsep algoritma DES yang dapat digunakan untuk mengamankan laporan internal perusahaan.

Kata Kunci : *Steganografi, DES (Data Encryption Standard), End of File (EOF), Laporan Internal Perusahaan*

I. PENDAHULUAN

Perkembangan teknologi saat ini didukung dengan pentingnya kebutuhan akan mendapatkan sebuah informasi sehingga keamanan data maupun informasi menjadi bagian yang sangat penting. Steganografi menjadi salah satu teknologi untuk menjaga pengiriman informasi maupun data perusahan agar tidak dengan mudah dibajak oleh pihak yang tidak bertanggung jawab.

Steganografi ialah penyembunyian pesan dalam sebuah media dan bersifat tidak mengacak isi file[1]. Sehingga, file yang disisipkan tidak mencurigakan. Pada pengujian ini penerapan steganografi menggunakan metode *End of File (EOF)* yaitu pesan akan disisipkan pada akhir nilai file. Konsep algoritma penerapan untuk menyisipkan file, teks maupun gambar lain dalam media utamanya dengan menggunakan algoritma *DES (Data Encryption Standard*[2]*).*

Keunggulan dari metode *End of File (EOF)* yaitu file yang disisipkan tidak akan mengganggu kualitasnya, keunggulan lainnya adalah metode EOF mempunyai kelebihan dapat menyisipkan pesan dalam jumlah yang tidak terbatas. Sedangkan algoritma *DES (Data Encryption Standard)* yang digunakan sebagai algoritma konsep merupakan salah satu algoritma kriptografi *cipher block* dengan ukuran blok 64 bit dan ukuran kuncinya 56 bit. Keunggulan dari DES salah satunya adalah pada tingkat keamananya, karena panjang kunci eksternal DES adalah 56 bit maka dengan panjang kunci 56 bit akan terdapat 256 atau 72.057.594.037.927.936 kemungkinan kunci. Jika diasumsikan serangan *exhaustive key search* dengan menggunakan prosesor paralel mencoba setengah dari jumlah kemungkinan kunci itu, maka dalam satu detik dapat dikerjakan satu juta serangan. Jadi seluruhnya diperlukan 1142 tahun untuk menemukan kunci yang bena0072[3].

Dengan adanya teknik steganografi dengan penggunaan dari metode EOF dan menekankan konsep algoritma DES dapat membantu dalam pelaporan internal perusahaan sehingga kemanan data dan informasi file penting dapat terjaga keamanannya.

1.1. Tujuan

Tujuan dari penelitian ini yaitu sebagai berikut:

1) Membuat sebuah aplikasi atau software untuk keamanan data laporan internal perusahaan.
2) Memperoleh aplikasi yang menggabungkan algoritma *DES (Data Encryption Standard)* dan teknik steganografi *EOF (End of File).*
3) Mengetahui kelebihan dan kekurangan *DES(Data Encryption Standard)* dan metode *EOF (End of File).*

4) Mengetahui proses enkripsi dan dekripsi file dengan menggunakan algoritma *DES (Data Encryption Standard)*.
5) Mengetahui proses penyisipan file dan pengekstrakan file pada suatu file berformat JPG/bitmap dengan menggunakan metode *End of File*

1.2. Batasan Masalah

Untuk fokusnya penelitian ini, penulis memberi batasan sebagai berikut :

1) Algoritma yang digunakan adalah 32 Bit *DES(Data Encryption Standard).*
2) Metode steganografi yang digunakan adalah *End of File (EOF).*
3) Data yang digunakan adalah file word/excel dan fileJPG/ bitmap.
4) Hanya membahas enkripsi dengan angka.
5) Bahasa pemrograman yang digunakan adalah Java.

II. LANDASAN TEORI

Adapun teori dasar yang penulis jadikan acuan sebagai landasan teori dalam penelitian ini yaitu sebagai berikut :

a. Steganografi

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia[1]. Kata "steganografi" berasal dari bahasa Yunani steganos, yang artinya “tersembunyi atau terselubung”, dan graphein, “menulis”. Kini, istilah steganografi termasuk penyembunyian data digital dalam berkas-berkas (file) komputer. Format yang biasa digunakan adalah:

1) Format image : bitmap (bmp), gif, pcx, jpeg, dll.
2) Format audio : wav, voc, mp3, dll.
3) Format lain : teks file, html, pdf, dll.

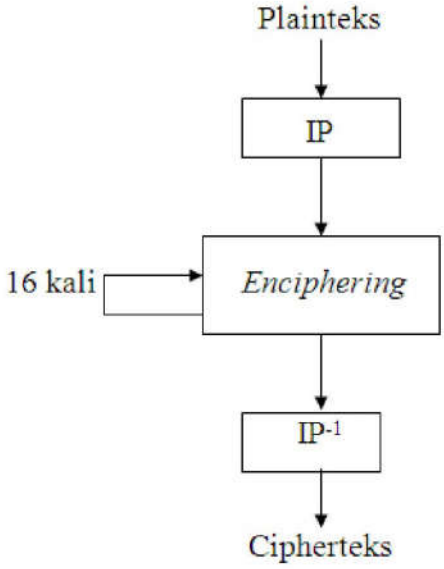
Teknik steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau gambar) di dalam berkas-berkas lain yang mengandung teks, image, bahkan audio tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari berkas semula.

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Kelebihan steganografi jika dibandingkan dengan kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan.

b. Algoritma *Data Encryption Standard (DES)*

DES (Data Encryption Standard) termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *Chiper Blok*. DES beroperasi pada ukuran blok 64 bit . DES mengenkripsi 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal *(internal key)* atau upa-kunci *(subkey)*. Kunci internal dibangkitkan dari kunci eksternal *(external key)* yang panjangnya 64 bit[4].

Berikut ni merupakan skema global dari algoritma DES yang terdpat pada Gambar 1:



Gambar 1 : Skema Global Algoritma DES

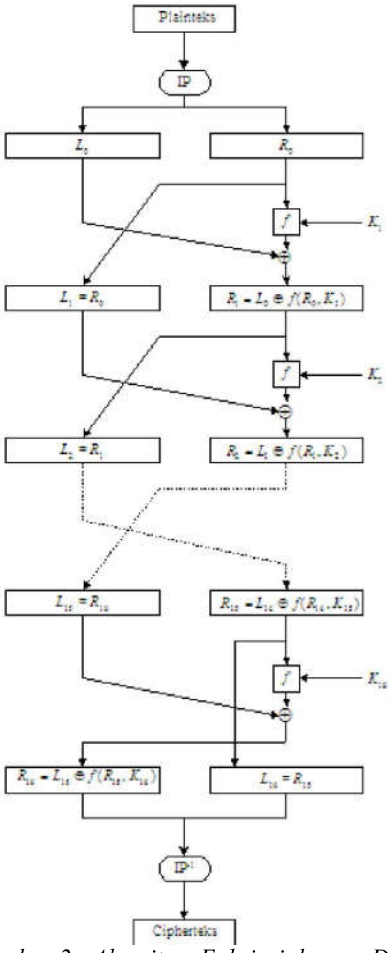
Keterangan gambar :

- Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian, kiri *(L)* dan kanan *(R)*, yang masing-masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES.
- Pada setiap putaran *i*, blok *R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal *Ki*. Keluaran dai fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya. Ini adalah satu putaran DES.

  Secara matematis, satu putaran DES dinyatakan sebagai:

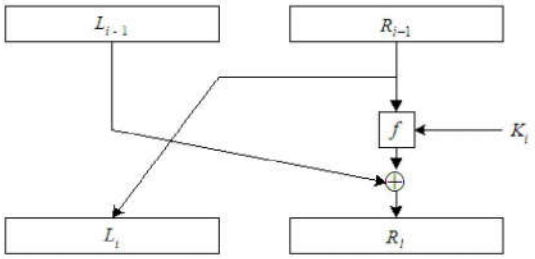
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Berikut merupakan gambar algoritma DES yang lebih rinci:



Gambar 2 : Algoritma Enkripsi dengan DES

Satu putaran DES merupakan model jaringan Feistel (lihat Gambar 3).



Gambar 3 : Jaringan Feistel untuk satuan putaran DES

Dari Gambar 3 bahwa jika (L16, R16) merupakan keluaran dari putaran ke-16, maka (R16, L16) merupakan pra-cipherteks (pre-ciphertext) dari *enciphering* ini. Cipherteks yang sebenarnya diperoleh dengan melakukan permutasi awal balikan, IP-1, terhadap blok pra-cipherteks.

1) Permutasi Awal

Sebelum putaran pertama, terhadap blok plainteks dilakukan permutasi awal (initial permutation atau IP). Tujuan permutasi awal adalah mengacak plainteks sehingga urutan bit-biit di dalamnya berubah. Pengacakan dilakukan dengan menggunakan matriks permutasi awal berikut ini:

Tabel 1 : Tabel/Matriks Permutasi Awal

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Cara membaca tabel 1 diatas: dua entry ujung kiri atas (58 dan 50) berarti:
"pindahkan bit ke-58 ke posisi bit 1"
"pindahkan bit ke-50 ke posisi bit 2", dst.

2) Pembakitan Kunci Internal

Karena ada 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu K1, K2, …, K16. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter. Misalkan kunci eksternal yang tersusun dari 64 bit adalah K. Kunci eksternal ini menjadi masukan untuk permutasi dengan menggunakan matriks permutasi kompresi PC-1 sebagai berikut:

Tabel 2 : Matriks Permutasi Kompresi PC-1

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Dalam permutasi ini, tiap bit kedelapan (parity bit) dari delapan byte kunci diabaikan. Hasil permutasinya adalah sepanjang 56 bit, sehingga dapat dikatakan panjang kunci DES adalah 56 bit.

Selanjutnya, 56 bit ini dibagi menjadi 2 bagian, kiri dan kanan, yang masing-masing panjangnya 28 bit, yang masing-masing disimpan di dalam C0 dan D0:

C0: berisi bit-bit dari K pada posisi
57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18
10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36

D0: berisi bit-bit dari K pada posisi
63, 55, 47, 39, 31, 23,15,7, 62, 54, 46, 38, 30, 22
14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28,20, 12, 4

Selanjutnya, kedua bagian digeser ke kiri (*left shift*) sepanjang satu atau dua bit bergantung pada tiap putaran. Operasi pergeseran bersifat wrapping atau round-shift. Jumlah pergeseran pada setiap putaran ditunjukkan pada Tabel 3:

Tabel 3 : Jumlah Pergeseran Bit Pada Setiap Putaran

Putaran, *i*	Jumlah pergeseran bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Misalkan (Ci, Di) menyatakan penggabungan Ci dan Di. (Ci+1, Di+1) diperoleh dengan menggeser Ci dan Di satu atau dua bit.

Setelah pergeseran bit, (Ci, Di) mengalami permutasi kompresi dengan menggunakan matriks PC-2 berikut:

Tabel 4: Permutasi Kompresi PC-2

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

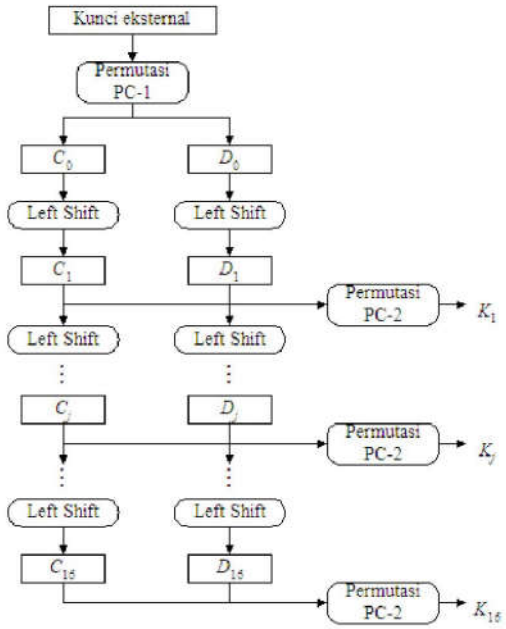
Dengan permutasi ini, kunci internal Ki diturunkan dari (Ci, Di) yang dalam hal ini Ki merupakan penggabungan bit-bit Ci pada posisi:

14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10
23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2

Dengan bit-bit Di pada posisi:

41, 52, 31, 37, 47, 55, 30, 40, 51, 45, 33, 48
44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32

Jadi, setiap kunci internal Ki mempunyai panjang 48 bit. Proses pembangkitan kunci-kunci internal ditunjukkan pada Gambar 4. Bila jumlah pergeseran bit-bit pada Tabel 1 dijumlahkan semuanya, maka jumlah seluruhnya sama dengan 28, yang sama dengan jumlah bit pada Ci dan Di. Karena itu, setelah putaran ke-16 akan didapatkan kembali C16 = C0 dan D16 = D0.



Gambar 4: Proses pembakitan kunci – kunci Internal DES

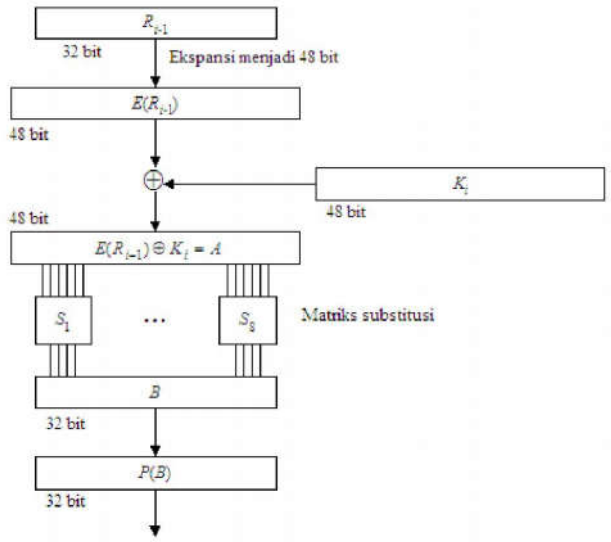
3) *Enciphering*

Proses *enciphering* terhadap blok plainteks dilakukan setelah permutasi awal (lihat Gambar 1). Setiap blok plainteks mengalami 16 kali putaran *enciphering* (lihat Gambar 2). Setiap putaran enciphering merupakan jaringan Feistel yang secara matematis dinyatakan sebagai :

Li = Ri – 1

Ri = Li – 1 $\oplus$ f(Ri – 1, Ki)

Diagram komputasi fungsi f diperlihatkan pada Gambar 5:



Gambar 5 : Rincian Komputasi fungsi F

Selanjutnya, hasil ekpansi, yaitu E(Ri – 1), yang panjangnya 48 bit di-XOR-kan dengan Ki yang panjangnya 48 bit menghasilkan vektor A yang panjangnya 48-bit:

$$E(Ri - 1) \oplus Ki = A$$

Vektor A dikelompokkan menjadi 8 kelompok, masing-masing 6 bit, dan menjadi masukan bagi proses substitusi. Proses substitusi dilakukan dengan menggunakan delapan buah kotak-S (S-box), S1 sampai S8. Setiap kotak-S menerima masukan 6 bit dan menghasilkan keluaran 4 bit. Kelompok 6-bit pertama menggunakan S1, kelompok 6-bit kedua menggunakan S2, dan seterusnya. (cara pensubstitusian dengan kotak-S sudah dijelaskan pada materi "Prinsip-prinsip Perancangan Cipher Blok") Kedelapan kotak-S tersebut adalah:

S1 :

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2:

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3:

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4:

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5:

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	16
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6:

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7:

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8:

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Keluaran proses substitusi adalah vektor B yang panjangnya 48 bit. Vektor B menjadi masukan untuk proses permutasi. Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak-S. Permutasi dilakukan dengan menggunakan matriks permutasi P (P-box) sbb:

Tabel 5: Permutasi P(P-box)

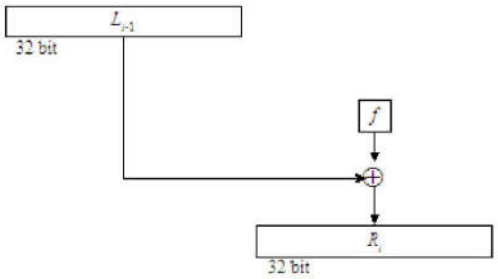
16	7	20	21	29	12	28	17	1	15	23	26	5	8	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Bit-bit P(B) merupakan keluaran dari fungsi f. Akhirnya, bit-bit P(B) di-XOR-kan dengan Li – 1 untuk mendapatkan Ri (lihat Gambar 6):

$$Ri = Li - 1 \oplus P(B)$$

Jadi, keluaran dari putaran ke-i adalah :

$$(Li, Ri) = (Ri - 1 , Li - 1 \oplus P(B))$$



Gambar 6 : Skema Perolehan R1

4) Permutasi Terakhir

Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan. Proses permutasi menggunakan matriks permutasi awal balikan (inverse initial permutation atau IP-1) sebagai berikut :

Tabel 6: Proses Permutasi Menggunakan Matriks Permutasi Awal Balikan

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

5) Deskripsi

Proses dekripsi terhadap cipherteks merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K1, K2, …, K16, maka pada proses dekripsi urutan kunci yang digunakan adalah K16, K15, …, K1.

Untuk tiap putaran 16, 15, …, 1, keluaran pada setiap putaran deciphering adalah:

$$Li = Ri - 1$$

$$Ri = Li - 1 \oplus f(Ri - 1, Ki)$$

Dalam hal ini, (R16, L16) adalah blok masukan awal untuk *deciphering*. Blok (R16, L16) diperoleh dengan mempermutasikan cipherteks dengan matriks permutasi IP-1. Pra-keluaran dari deciphering adalah adalah (L0, R0). Dengan

permutasi awal IP akan didapatkan kembali blok plainteks semula.

Tinjau kembali proses pembangkitan kunci internal pada Gambar 4. Selama deciphering, K16 dihasilkan dari (C16, D16) dengan permutasi PC-2. Tentu saja (C16, D16) tidak dapat diperoleh langsung pada permulaan deciphering. Tetapi karena (C16, D16) = (C0, D0), maka K16 dapat dihasilkan dari (C0, D0) tanpa perlu lagi melakukan pergeseran bit. Catatlah bahwa (C0, D0) yang merupakan bit-bit dari kunci eksternal K yang diberikan pengguna pada waktu dekripsi.

Selanjutnya, K15 dihasilkan dari (C15, D15) yang mana (C15, D15) diperoleh dengan menggeser C16 (yang sama dengan C0) dan D16 (yang sama dengan C0) satu bit ke kanan. Sisanya, K14 sampai K1 dihasilkan dari (C14, D14) sampai (C1, D1). Catatlah bahwa (Ci – 1, Di – 1) diperoleh dengan menggeser Ci dan Di dengan cara yang sama seperti pada Tabel 1, tetapi pergeseran kiri (left shift) diganti menjadi pergeseran kanan (right shift).

6) Mode DES

DES dapat dioperasikan dengan mode ECB, CBC, OFB, dan CFB. Namun karena kesederhanaannya, mode ECB lebih sering digunakan pada paket program komersil meskipun sangat rentan terhadap serangan.

Mode CBC lebih kompleks daripada EBC namun memberikan tingkat keamanan yang lebih bagus daripada mode EBC. Mode CBC hanya kadang-kadang saja digunakan.

7) Implementasi DES

DES sudah diimplementasikan dalam bentuk perangkat keras. Dalam bentuk perangkat keras, DES diimplementasikan di dalam chip. Setiap detik chip ini dapat mengenkripsikan 16,8 juta blok (atau 1 gigabit per detik). Implementasi DES ke dalam perangkat lunak dapat melakukan enkripsi 32.000 blok per detik (pada komputer mainframe IBM 3090).

8) Keamanan DES

Isu-isu yang menjadi perdebatan kontroversial menyangkut keamanan DES[5]:

a) Panjang Kunci

Panjang kunci eksternal DES hanya 64 bit atau 8 karakter, itupun yang dipakai hanya 56 bit. Pada rancangan awal, panjang kunci yang diusulkan IBM adalah 128 bit, tetapi atas permintaan NSA, panjang kunci diperkecil menjadi 56 bit. Alasan pengurangan tidak diumumkan.

Tetapi, dengan panjang kunci 56 bit akan terdapat 256 atau 72.057.594.037.927.936 kemungkinan kunci. Jika diasumsikan serangan exhaustive key search dengan menggunakan prosesor paralel mencoba setengah dari jumlah kemungkinan kunci itu, maka dalam satu detik dapat dikerjakan satu juta serangan. Jadi seluruhnya diperlukan 1142 tahun untuk menemukan kunci yang benar.

Tahun 1998, Electronic Frontier Foundation (EFE) merancang dan membuat perangkat keras khusus untuk menemukan kunci DES secara exhaustive search key dengan biaya $250.000 dan diharapkan dapat menemukan kunci selama 5 hari. Tahun 1999, kombinasi perangkat keras EFE dengan kolaborasi internet yang melibatkan lebih dari 100.000 komputer dapat menemukan kunci DES kurang dari 1 hari.

b) Jumlah Putaran

Sebenarnya, delapan putaran sudah cukup untuk membuat cipherteks sebagai fungsi acak dari setiap bit plainteks dan setiap bit cipherteks. Jadi, mengapa harus 16 kali putaran?

Dari penelitian, DES dengan jumlah putaran yang kurang dari 16 ternyata dapat dipecahkan dengan *known-plaintext attack* lebih mangkus daripada dengan *brute force attack.*

c) Kotak-S

Pengisian kotak-S DES masih menjadi misteri tanpa ada alasan mengapa memilih konstanta-konstanta di dalam kotak itu.

DES mempunyai beberapa kunci lemah (*weak key*). Kunci lemah menyebabkan kunci-kunci internal pada setiap putaran sama (K1 = K2 = … = K16). Akibatnya, enkripsi dua kali berturut-turut terhadap plainteks menghasilkan kembali plainteks semula. Kunci lemah terjadi bila bit-bit di dalam Ci dan Di semuanya 0 atau 1, atau setengah dari kunci seluruh bitnya 1 dan setengah lagi seluruhnya 0. Kunci eksternal (dalam notasi HEX) yang menyebabkan terjadinya kunci lemah adalah (ingat bahwa setiap bit kedelapan adalah bit paritas).

Selain kunci lemah, DES juga mempunyai sejumlah pasangan kunci setengah-lemah (*semiweak key*). Pasangan kunci setengah- lemah mengenkripsikan plainteks menjadi cipherteks yang sama. Sehingga, satu kunci dalam pasangan itu dapat mendekripsi pesan yang dienkripsi oleh kunci yang lain di dalam pasangan itu.

Kunci setengah-lemah terjadi bila:

- Register C dan D berisi bit-bit dengan pola 0101…0101 atau 1010…1010
- Register yang lain (C atau D) berisi bit-bit dengan pola 0000…0000, 1111…1111, 0101…0101, atau 1010…1010

Ada 6 pasang kunci setengah lemah (dalam notasi HEX):

- 01FE 01FE 01FE 01FE dan FE01 FE01 FE01 FE01
- 1FE0 1FE0 0EF1 0EF1 dan E01F E01F F10E F10E
- 01E0 01E0 01F1 01F1 dan E001 E001 F101 F101
- 1FFE 1FFE 0EFE 0EFE dan FE1F FE1F FE0E FE0E
- 011F 011F 010E 010E dan 1F01 1F01 0E01 0E01
- E0FE E0FE F1FE F1FE dan FEE0 FEE0 FEF1 FEF1

c. *Ending Of File (EOF)*

Metode *End of File (EOF)* merupakan salah satu metode yang digunakan dalam steganografi. Teknik ini menggunakan 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. Contoh hasil penyisipan pesan rahasia dengan menggunakan metode *End of File* dapat dilihat pada Gambar 7:

Gambar 7: Sebelum dan Sesudah Penyisipan Pesan dengan Menggunakan Metode End of File

d. Laporan Internal Perusahaan

Laporan Internal Perusahaan/Laporan internal auditor merupakan saran pertanggungjawaban internal auditor atas penugasan pemeriksaan oleh pimpinan. Melalui laporan ini internal auditor akan mengungkapkan dan menguraikan kelemahan yang terjadi dan keberhasilan yang telah dicapai.

III. METODOLOGI PENELITIAN

a. Fase Analisis

Pada tahap ini akan dilakukan pengumpulan bahan referensi yang terkait dengan *DES(Data Encryption Standard)* 32Bit dan metode *End of File* yang dapat berupa buku-buku, artikel-artikel atau e-book serta jurnal nasional dan internasional yang didapatkan melalui internet.

b. Fase Pembuatan Program

Perancangan dan implementasi sistem yang dilakukan secara ekperimental, yaitu bereksperimen membuat program berdasarkan materi dan algoritma yang telah dipelajari. Desain antarmuka sistem dan struktur proses kerja sistem dalam bentuk perangkat lunak dengan menggunakan bahasa pemrograman Java.

c. Pengujian Sistem

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.

3.1 Pola Pikir

A. Analisa Permasalahan dan Pemecahannya

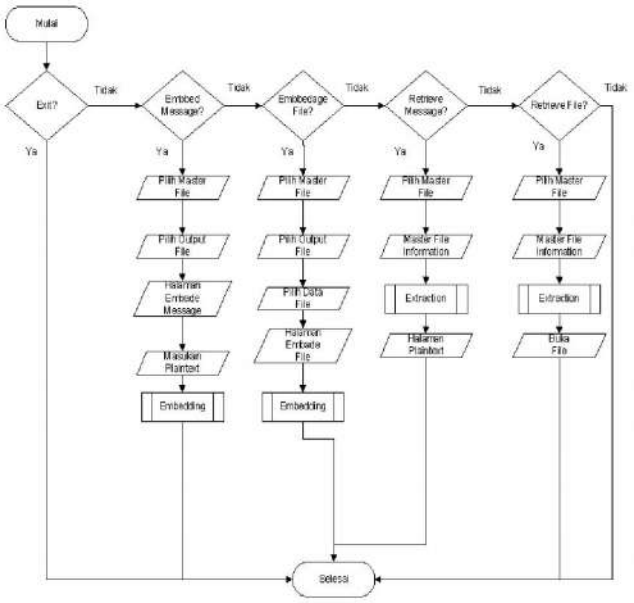
Pada perusahaan, perkembangan teknologi saat ini didukung dengan pentingnya kebutuhan akan mendapatkan sebuah informasi. Dalam sebuah instansi atau perusahaan membutuhkan pengolahan data dan informasi guna keberlangsungan kegiatan usaha, oleh sebab itu keamanan data maupun informasi menjadi bagian yang sangat penting dalam sebuah keamanan perusahaan agar mampu bertahan dalam persaingan dengan perusahan lain.

Salah satu bentuk aset terpenting dalam sebuah perusahaan adalah pelaporan internal perusahaan. Laporan Internal Perusahaan / Laporan internal auditor merupakan saran pertanggungjawaban internal auditor atas penugasan pemeriksaan oleh pimpinan. Melalui laporan ini internal auditor akan mengungkapkan dan menguraikan kelemahan yang terjadi dan keberhasilan yang telah dicapai. Oleh sebab itu apabila laporan internal perusahaan jatuh ke pihak yang tidak bertanggung jawab akan berpengaruh terhadap keberlangsungan kegiatan usaha.

Untuk menghindari hal-hal tersebut maka perlu dibuatnya suatu sistem atau aplikasi yang berfungsi untuk melindungi keamanan data atau informasi perusahaan yaitu dengan menerapkan steganografi. Dalam penelitian ini membahas bagaimana mengimplementasikan suatu sistem keamanan data yang mampu melakukan proses enkripsi dan dekripsi suatu data file khususnya laporan internal perusahan dengan menggunakan algoritma *DES (Data Encryption Standard)* 32 Bit, kemudian file yang telah dienkripsi tersebut disisipkan ke dalam suatu file gambar berformat JPG/bitmap dengan menggunakan metode *End of File.*

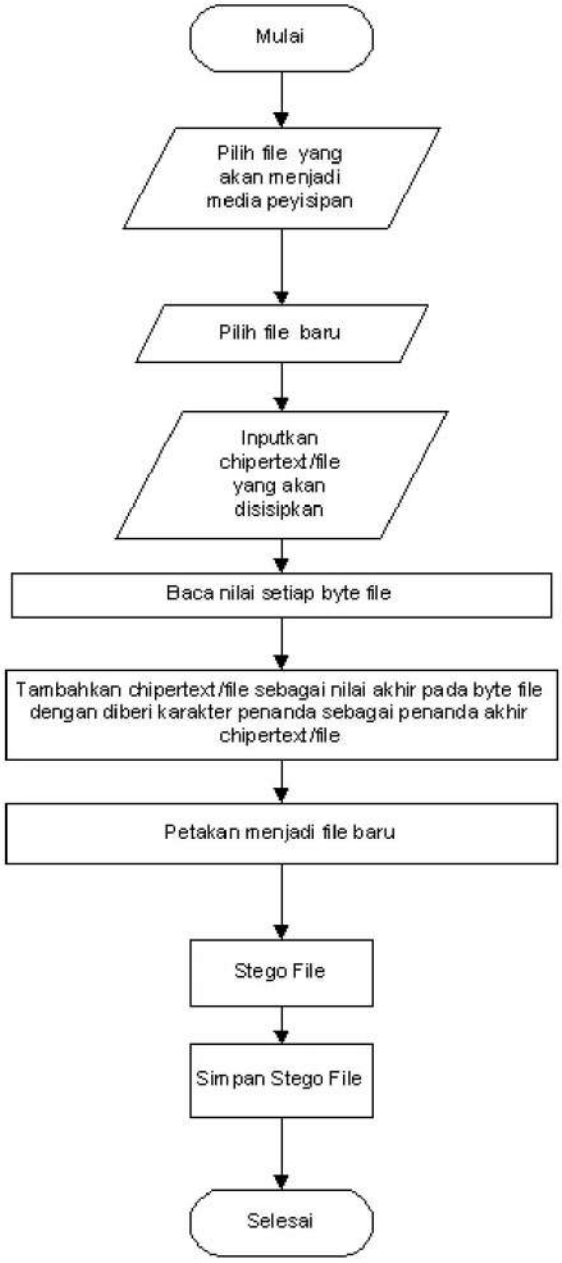
B. Algoritma

Berikut merupakan algoritma dari aplikasi yang sesuai dengan rancangan dari alur konsep pada tahapan pola pikir diatas. Gambar 8 menunjukan *flowchart* dari 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*).



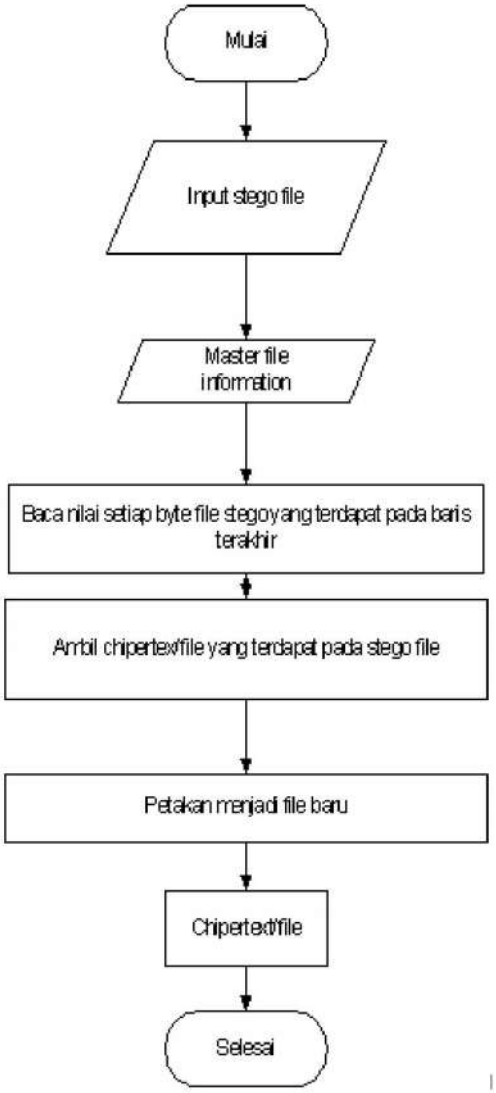
Gambar 8: Flowchart Gambaran Umum Sistem

Flowchart proses penyisipan ciphert ext ke dalam media file citra dengan menggunakan metode End of File dapat dilihat pada Gambar 9 :



Gambar 9 : Flowchart proses Embedding pada metode End Of File (EOF)

Flowchart proses pengambilan *ciphertext* dari media file citra dengan menggunakan metode *End of File* dapat dilihat pada Gambar 10:



Gambar 10 : : Flowchart Proses Exctract EOF

IV. IMPLEMENTASI

Tahap implementasi sistem merupakan lanjutan dari tahap perancangan sistem. Pada tahap ini dilakukan implementasi sistem ke dalam bahasa pemrograman berdasarkan hasil analisis dan perancangan sistem. Pada tahap implementasi ini digunakan perangkat lunak dan perangkat keras, sehingga sistem yang dibangun dapat diselesaikan dengan baik.

1) Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam implementasi sistem ini adalah sebagai berikut :

- Processor Intel Core 2 Duo 2.00 GHz

- Memory (RAM) 3.00 GB
- Harddisk 78 GB
- Monitor dengan resolusi layar 1200 x 800 pixel
- Mouse dan Keyboard

2) Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem ini adalah sebagai berikut :

- Sistem Operasi Windows 7 Professional
- Java JDK 1.7

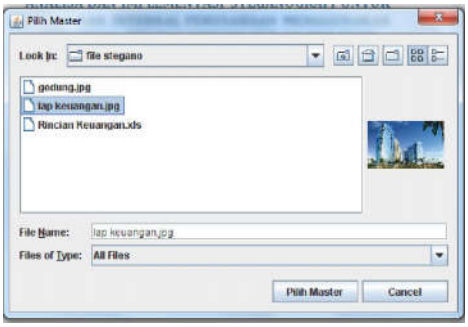
3) Tampilan Antar Muka

Tampilan antar muka dari sistem ini diimplementasikan berdasarkan dari tahap analisis dan perancangan sistem. Tampilan antar muka sistem ini terdiri dari 5 (lima) halaman utama, yaitu Halaman Login, Halaman Menu Utama, Halaman *Embed/Encode* dan Halaman *Extract/Decode*.

Berikut merupakan tampilan program dari menu *Embed File* yang terdapat pada Gambar 11:

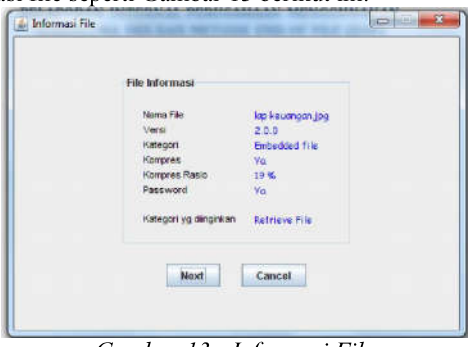
Gambar 11 : Menu Embed File

Menu *Embed File* digunakan untuk menyisipkan file laporan internal perusahaan kedalam sebuah media gambar. Selain menu *Embed File* juga terdapat *Menu Retrieve File* yang digunakan untuk mengembalikan data file yang telah disisipkan, namun terlebih dahulu akan tampil seperti Gambar 12 untuk memilih file gambar yang akan *diretrieve*. Berikut tampilannya:



Gambar 12 : Proses Retrieve File Pilih Master

Setelah gambar master dipilih maka akan tampil informasi file seperti Gambar 13 berikut ini:

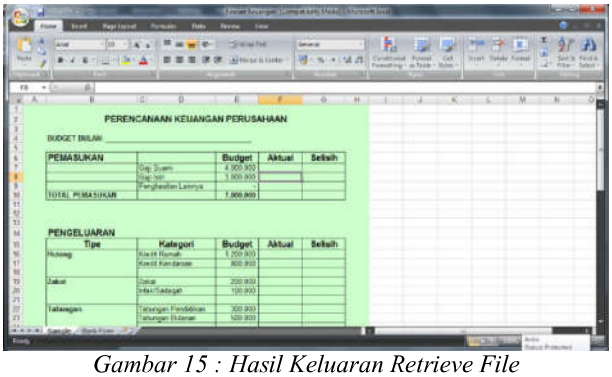


Gambar 13 : Informasi File

Kemudian Pilih 'Next' untuk melanjutkan proses *retrieve* pada aplikasi steganografi ini, dan apabila file yang akan *diretrieve* dilengkapi dengan password maka akan tampil pop up menu sebagai berikut:

Gambar 14: Konfirmasi Password

Selanjutnya file yang disisipkan akan bisa langsung di-load ke memori dan dapat langsung dijalankan di sistem operasi windows, berikut tampilannya yang terdapat pada Gambar 15:

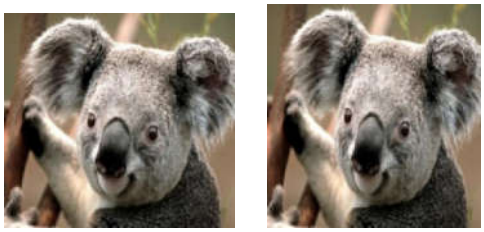


Gambar 15 : Hasil Keluaran Retrieve File

4) Analisa Hasil Pengujian

Tahap pengujian sistem merupakan lanjutan dari tahap implementasi sistem. Fungsi dari tahap pengujian sistem adalah untuk membuktikan bahwa sistem yang telah diimplementasikan dari hasil analisis dan perancangan sistem telah berjalan dengan baik.

a. Pengujian Pada File *.jpg

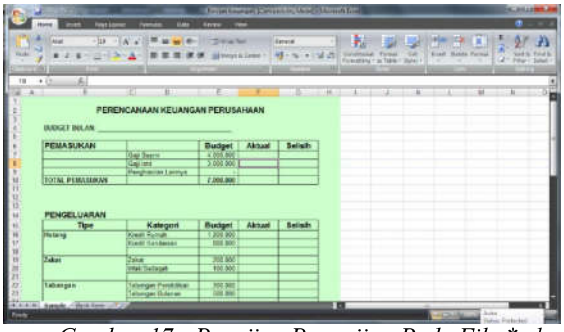


Gambar 16: Perbandingan Media File .Jpg Tanpa Data (Kiri) Dan Dengan Data(Kanan)

1. Ukuran media file tanpa data : 763KB
2. Ukuran data yang disispkan : 2 KB
3. Ukuran media file : 764 KB
4. Selisih file : 1 KB

b. Pengujian Pada File *.xls

Pada file yang berextension .xls file yang sudah disisipi data masih dapat digunakan dengan baik seperti sebelumnya.



*Gambar 17 : Penujian Pengujian Pada File *.xls*

c. Pengujian Pada File *.exe

Setelah melakukan testing maka file *executable* yang merupakan file yang bisa langsung di-load ke memory dan langsung bisa dijalankan di sistem operasi windows. Berdasarkan dari hasil pengujian file *executable* yang disisipkan data tidak rusak.

V. PENUTUP

Berdasarkan hasil analisa aplikasi steganografi ini maka dapat ditarik kesimpulan dan saran yaitu sebagai berikut:

5. 1. Kesimpulan

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

a. Perusahaan dapat menerapkan aplikasi ini untuk mengirimkan laporan internal perusahan kepada pemilik perusahaan dengan aman tanpa kekhawatiran akan adanya sabotasi laporan, ketika kan dikirimkan melalui media internet.
b. Data yang disipkan dengan media file dapat terlihat dengan jelas seperti aslinya, jika dengan metode enkripsi maka file tidak terbaca seperti aslinya
c. Selisih ukuran file sesuai dengan data yang disisipkan pada media file.
d. Pada pengujian file biner secara kasat mata tidak terjadi kerusakan sehingga aplikasi yang digunakan masih bisa berjalan dengan baik.

5. 2. Saran

Adapun saran-saran yang dapat penulis berikan untuk pengembangan dan perbaikan sistem ini adalah sebagai berikut :

a. Algoritma kriptografi dan teknik steganografi yang digunakan pada sistem ini dapat diganti dengan algoritma kriptografi dan teknik steganografi yang lebih baik untuk meningkatkan keamanan data.
b. Untuk penelitian selanjutnya, sebaiknya menggunakan teknik steganografi yang tidak menimbulkan penambahan garis-garis pada bagian bawah gambar, seperti mengkombinasikan metode *End of File (EOF)* dan *Least Significant Bit (LSB).*
c. Perlunya perbaikan pada proses dekripsi agar tidak didapat hasil dekripsi yang tidak sesuai den gan plain text yang sebenarnya.
d. Sistem ini dapat dikembangkan lebih lanjut dengan menambahkan pilihan data rahasia yang akan dienkripsi dan didekripsi, seperti data teks yang bukan angka, data gambar, data suara dan data video.
e. Sistem ini dapat dikembangkan lebih lanjut dengan menambahkan pilihan file gambar berformat lain, seperti PNG dan bitmat.

DAFTAR PUSTAKA

[1] Y. Kurniawan, *Keamanan Internet dan Jaringan Telekomunikasi*. Bandung: Bandung : Informatika, 2004.
[2] Sutoyo.T, *Teori Pengolahan Citra Digital.* Yogyakarta: ANDI., 2009.
[3] R. Munir, *Kriptografi*. Bandung: Informatika Bandung, 20006.
[4] "DES." [Online]. Available: Kur2003.if.itb.ac.id/file/DES.doc. [Accessed: 01-Nov-2014].
[5] A. Widyarnako, "Teknik Kriptografi Rabin, Serangan yang Dapat Dilakukan dan Perbandingannya dengan RSA.," *Inst. Teknol. Bandung*, vol. 2.