Applied Information Systems and Management (AISM)
Volume 3, (2) 2020, hal 69-78
P-ISSN: 2621-2536; E-ISSN: 2621-2544

Implementasi Kriptografi Dengan Algoritma Advanced Encryption Standard (AES) 128 Bit Dan Steganografi Menggunakan Metode End Of File (EOF) Berbasis Java Desktop Pada Dinas Pendidikan Kabupaten Tangerang

Anggraeni Eka Putri, Aghistina Kartikadewi, Lina Audina Abdul Rosyid

Abstract*— Faktor keamanan data dalam proses pertukaran maupun penyimpanan data menjadi hal yang sangat penting untuk diperhatikan seiring dengan kerahasiaan atau pentingnya informasi tersebut. Pertukaran data sangat beresiko ketika pihak yang tidak berkepentingan dapat mengakses data tersebut. Mengingat pentingnya keamanan dan kerahasiaan data, maka dibutuhkan system keamanan pengiriman maupun penyimpanan data menggunakan teori penyamaran dalam bentuk sandi atau kode khusus. Teknik yang umum digunakan adalah dengan mengacak informasi dan menggantinya dengan sandi khusus yang telah ditetapkan yaitu kriptografi. Tetapi informasi yang diacak sering menimbulkan kecurigaan, maka dibutuhkan teknik lain yaitu dengan menyamarkan data ke dalam data lain yaitu steganografi. Teknik steganografi akan menyamarkan pesan dengan cara disisipkan dalam sebuah gambar digital. Kombinasi dari teknik kriptografi dan steganografi akan menghasilkan tingkat keamanan data yang sangat tinggi guna menjaga keamanan data tanpa mengubah gamabar secara visual. Algoritma kriptografi yang akan digunakan adalah metode kriptografi Advanced Encryption Standard (AES) 128 bit dan teknik steganografi End Of File (EOF)

Index Terms*— Kriptografi AES, Encode, Decode, Steganografi EOF, Embed, Retrieve

I. PENDAHULUAN

Perkembangan dunia digital saat ini membuat lalu lintas pengiriman data elektronik semakin ramai dan sensitif. Sebagai contoh perkembangan jaringan internet yang memungkinkan orang untuk saling bertukar data melalui jaringan internet. Seiring dengan perkembangan tersebut, kejahatan teknologi komunikasi dan informasi juga turut berkembang. Dengan adanya pencurian data maka aspek keamanan dalam pertukaran informasi serta penyimpanan data dianggap penting, karena suatu komunikasi data jarak jauh, belum tentu memiliki jalur transmisi yang aman dari penyadapan, serta penyimpanan data belum tentu aman dari pencurian sehingga keamanan informasi menjadi bagian penting dalam dunia informasi.

Keamanan data di Dinas Pendidikan Kabupaten Tangerang selama ini, hanya melakukan pengamanan data secara sederhana dan tidak terlalu memperhatikan tingkat keamanan. Padahal Dinas Pendidikan Kabupaten Tangerang adalah intansi pemerintah yang memiliki banyak data yang tidak boleh di ketahui olah masyarakat umum. Beberapa data yang seharusnya diamankan adalah Instrumen Seleksi Calon Kepala Sekolah dan Pengawas, Hasil seleksi yang belum waktunya diumumkan, SPJ (Surat Pertanggung Jawaban) keuangan yang belum diperiksa oleh pemeriksa internal dan eksternal, dan masih banyak lagi data yang bersifat rahasia lainnya. Dengan demikian aplikasi keamanan terhadap data atau file sangat diperlukan untuk menghidari tindakan-tindakan tertentu yang dapat merugikan.

Kriptografi dapat digunakan untuk menjaga keamanan pesan yang dikirim dari suatu tempat ke tempat yang lain. Dengan teknik kriptografi pesan asli yang ingin dikirimkan (plaintext) diubah atau dienkripsi dengan suatu kunci menjadi suatu informasi acak yang tidak bermakna (ciphertext). Kunci yang hanya diketahui oleh pengirim dan penerima, dari kunci tersebut bisa digunakan untuk mengembalikan ciphertext ke plaintext kembali oleh penerima. Dengan begitu, orang lain yang tidak memiliki hak akses terhadap pesan tersebut tidak dapat mengetahui isi pesan sebenarnya, hanya mengetahui pesan acaknya saja.

Namun karena sifatnya yang acak itu, timbul suatu kecurigaan terhadap pesan yang dikirim. Karena.

terlihat pesan tersebut seperti tidak mempunyai arti, maka bisa saja pihak luar akan merusak pesan tersebut dengan tujuan agar penerima tidak mendapatkan pesan tersebut secara utuh. Untuk

Received: 02 Febuari 2020; Revised: 9 juni 2020; Accepted: 7 November 2020

A.E. Putri, Postgraduate Program, Master of Computer Science, Budi Luhur University. (putri.anggraeni.eka@gmail.com)

A. Kartikadewi, Postgraduate Program, Master of Computer Science, Budi Luhur University. (aghistinakd@gmail.com)

L.A.A Rosyid, Postgraduate Program, Master of Computer Science, Budi Luhur University (linaaudina16@gmail.com)

mengatasi masalah ini, dapat digunakan teknik lain yaitu teknik steganografi. Steganografi lebih mengurangi kecurigaan karena pesan yang disamarkan disembunyikan dalam gambar.
Algoritma kriptografi yang akan digunakan adalah dengan menggunakan metode *Advanced Encryption Standard* (AES) 128 bit dan teknik steganografi yang digunakan dengan metode *End Of File* (EOF). Teknik kriptografi dan steganografi sama-sama memiliki kekurangan, oleh karena itu menggabungkan kedua teknik enkripsi ini dimaksudkan akan menambah tingkat keamanan pada saat pertukaran data, dan menyisipkan pesan rahasia.

A. Batasan Masalah

Agar pembahasan masalah tidak menyimpang, maka penulis memberikan beberapa batasan masalah sebagai berikut yaitu, algoritma yang digunakan untuk mengenkripsi *file* dokumen atau data rahasia yakni algoritma AES, dan metode steganografi digunakan untuk penyisipan *file* dokumen setelah proses pengenkripsian *file* dokumen atau data rahasia yakni metode EOF. Media steganografi yang digunakan sebagai *cover object* adalah *file* berupa citra *digital image* (jpg, jpeg, png, bmp, gif). Tipe pengenkripsian dan penyisipan seperti *file word* (.doc, .docx), *file excel* (.xls, .xlsx), dan *image* (.jpg, .jpeg, .png, .bmp, .gif). Mekanisme penentuan tipe *file* yang ingin di enkripsi atau di dekripsi hanya berdasarkan ekstensi *file* (format *file*) dan bahasa pemrograman yang digunakan adalah Java.

B. Tujuan Penulisan

Tujuannya adalah mengimplementasikan algoritma kriptografi *Advanced Encryption Standard* (AES) dan metode steganografi *End Of File* (EOF) untuk mengamankan file atau informasi dengan membuat aplikasi pengamanan data berbasis desktop. Mengamankan sebuah *file* berupa .docx, .doc, .xlsx, .xlx, dan .jpg agar tidak bisa diakses oleh pengguna yang memang tidak mempunyai hak akses. Menghasilkan proses enkripsi dan dekripsi data, serta penyisipan data secara optimal tanpa adanya kerusakan data setelah proses tersebut. Hasil proses encode berupa gambar, sehingga tidak menimbulkan kecurigaan

II. METODE PENELITIAN

A. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya kedalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Pesan yang akan dienkripsi disebut sebagai plaintext (teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah plaintext melibatkan penggunaan suatu bentuk kunci. Pesan plaintext yang telah dienkripsi (atau dikodekan) dikenal sebagai ciphertext (teks sandi)[1].

B. Konsep Dasar Kriptografi

Konsep kriptografi sendiri telah lama digunakan oleh manusia misalnya pada peradaban Mesir dan Romawi walau masih sangat sederhana. Prinsip-prinsip yang mendasari kriptografi yakni:
Confidelity (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap 1)rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
Data integrity (keutuhan data) yaitu layanan yang mampu 2)mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
Authentication (keotentikan) yaitu layanan yang berhubungan dengan 3)identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
Non-repudiation (anti-penyangkalan) yaitu layanan yang dapat mencegah 4)suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya)[2].

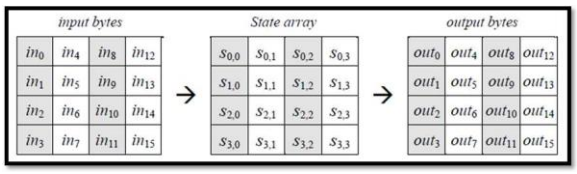
C. Algoritma AES (Advanced Encryption Standard)

Input dan output dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau plaintext yang nantinya akan dienkripsi menjadi ciphertext. Cipher key dari AES terdiri dari key dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah round yang akan diimplementasikan pada algoritma AES ini. Berikut ini adalah Tabel 1 yang memperlihatkan jumlah round/putaran (Nr) yang harus diimplementasikan pada masing-masing panjang kunci[3].

Tabel 1.
Jumlah round/putaran (Nr)

Tipe	Jumlah Key (Nk)	Besar Blok (Nb)	Jumlah Round (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Pada dasarnya, operasi AES dilakukan terhadap array of byte dua dimensi yang disebut dengan state. State mempunyai ukuran NROWS X NCOLS. Pada awal enkripsi, data masukan yang berupa in0, in2, in3, in4, in5, in6, in7, in8, in9, in10, in11, in12, in13, in14, in15 disalin ke dalam array state. State inilah yang nantinya dilakukan operasi enkripsi/dekripsi. Kemudian keluarannya akan ditampung kedalam array out. Gambar 1mengilustrasikan proses penyalinan dari input bytes, state array, dan output bytes :



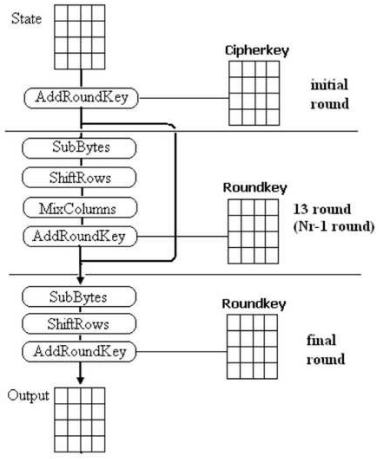
Gambar 1: Proses Input Bytes, State Array, dan Output Bytes

Pada saat permulaan, input bit pertama kali akan disusun menjadi suatu array byte dimana panjang dari array byte yang

digunakan pada AES adalah sepanjang 8 bit data. Array byte inilah yang nantinya akan dimasukkan atau dicopy ke dalam state dengan urutan dimana r (row/baris) dan c (column/kolom):
s[r,c] = in[r+4c] untuk $0 \le r < 4$ dan $0 \le c < Nb$
sedangkan dari state akan dicopy ke output dengan urutan :
out[r+4c] = s[r,c] untuk $0 \le r < 4$ dan $0 \le c < Nb$

a. Proses Enkripsi AES

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey. Pada awal proses enkripsi, input yang telah dicopykan ke dalam state akan mengalami transformasi byte AddRoundKey. Setelah itu, state akan mengalami transformasi SubBytes, ShiftRows, MixColumns, dan AddRoundKey secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai round function. Round yang terakhir agak berbeda dengan round-round sebelumnya dimana pada round terakhir, state tidak mengalami transformasi MixColumns. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada gambar 2 di bawah ini :



Gambar 2. Ilustrasi Enkripsi AES 256 Bit (Yuniati, dkk, 2009)

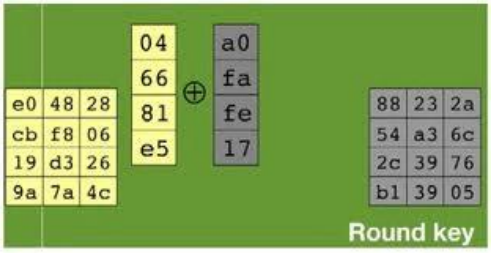
1) AddRoundkey

Pada proses enkripsi dan dekripsi AES proses AddRoundKey sama, sebuah round key ditambahkan pada state dengan operasi XOR. Setiap round key terdiri dari Nb word dimana tiap word tersebut akan dijumlahkan dengan word atau kolom yang bersesuaian dari state sehingga :

[s'o,c , s'1,c , s'2,c , s'3,c] [so,c , s1,c , s2,c , s3,c] □ [wround*Nb+c] untuk $0 \le c \le Nb$

[wi] adalah word dari key yang bersesuaian dimana i = round*Nb+c. Transformasi AddRoundKey pada proses enkripsi pertama kali pada round = 0 untuk round selanjutnya round = round + 1, pada proses dekripsi pertama kali pada round = 14 untuk round selanjutnya round = round - 1.

Agar lebih mudah dipahami lihatlah gambar 3, pada gambar tersebut di sebelah kiri adalah ciphertext dan sebelah kanan adalah roundkeynya. XOR dilakukan perkolom yaitu kolom-1 ciphertext di XOR dengan kolom-1 roundkey dan seterusnya.



Gambar 3. Skema addroundkey

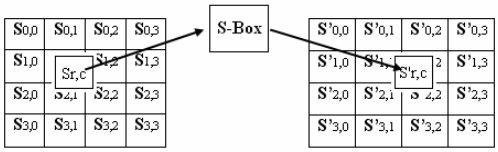
2) SubBytes

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel subtitusi (S-Box). Tabel subtitusi S-Box akan dipaparkan dalam tabel 2.

Tabel 2.
S-box subbytes (Yuniati dkk, 2009)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

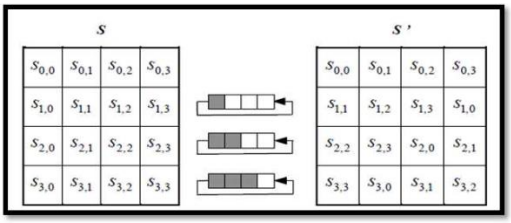
Untuk setiap byte pada array state, misalkan S[r,c]=xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r,c], maka nilai subtitusinya, dinyatakan dengan S[r,c], adalah elemen didalam tabel subtitusi yang merupakan pengaruh pemetaan byte pada setiap byte dan state.



Gambar 4. Pengaruh Pemetaan pada setiap byte dalam state (Yuniati dkk, 2009)

3) *ShiftRows*

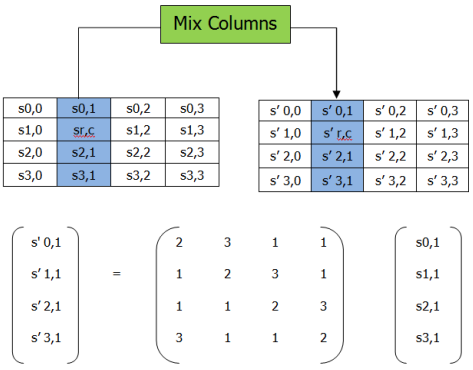
Transformasi Shiftrows pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Proses pergeseran Shiftrow ditunjukkan dalam Gambar 5. berikut:



Gambar 5. Transformasi shiftrows (Yuniati dkk, 2009)

4) *Mix Columns*

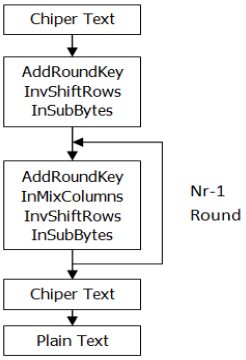
Yang terjadi saat MixColumn adalah mengalikan tiap elemen dari blokcipher dengan matriks yang ditunjukkan oleh gambar 2.9. Tabel sudah ditentukan dan siap pakai. Pengalian dilakukan seperti perkalian matriks biasa yaitu menggunakan dot product lalu perkalian keduanya dimasukkan ke dalam sebuah blokcipher baru. Ilustrasi dalam gambar 6. akan menjelaskan mengenai bagaimana perkalian ini seharusnya dilakukan. Dengan begitu seluruh rangkaian proses yang terjadi pada AES telah dijelaskan dan selanjutnya adalah menerangkan mengenai penggunaan tiap-tiap proses tersebut.



Gambar 6. Perkalian matriks (Yuniati dkk, 2009)

b. Proses Dekripsi AES

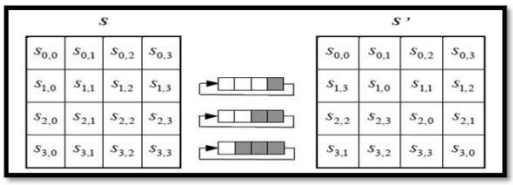
Transformasi cipher dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada invers cipher adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey. Algoritma dekripsi dapat dilihat pada skema berikut ini :



Gambar 7. Ilustrasi proses dekripsi aes

1) InvShiftRows

InvShiftRows adalah transformasi byte yang berkebalikan dengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri. Ilustrasi transformasi InvShiftRows terdapat pada gambar 8.



Gambar 8. Transformasi invshiftrows

2) InvSubBytes

InvSubBytes juga merupakan transformasi bytes yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada state dipetakan dengan menggunakan tabel Inverse S-Box. Tabel Inverse S-Box akan ditunjukkan dalam tabel berikut:

Tabel 3
Invers s-box (Yuniati dkk, 2009)

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	f
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	c
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	2
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	9
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	8
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	0
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	7
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	e
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	6
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7

3) InvMixColumns

Setiap kolom dalam state dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 9. Perkalian matriks invmixcolumns (Yuniati dkk, 2009)

Hasil dari perkalian dalam matrik adalah :

s'o,c = ({0E} • so,c) □ □({0B} • s1,c) □ ({0D} • s2, c) □ ({09} • s3, c)
s'1,c = ({09} • so,c) □ ({0E} • s1,c) □ ({0B} • s2,c) □ ({0D} • s3, c)
s'2,c = ({0D} • so,c) □ ({09} • s1,c) □ ({0E} • s2,c) □ ({0B} • s3, c)
s'3,c = ({0B} • so,c)□□ ({0D} • s1,c) □ ({09} • s2,c) □ ({0E} • s3, c

D. Steganografi

Dalam Steganografi (steganography) adalah ilmu dan seni menyembunyikan pesan rahasia(hiding message) sedemikian sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indera manusia. Kata "steganografi" berasal dari bahasa Yunani “steganos”, yang artinya “tersembunyi atau terselubung”, dan “graphein”, “menulis”.

Sebuah pesan steganografi (plaintext), dienkripsikan dengan beberapa arti tradisional, yang menghasilkan ciphertext. Kemudian, covertext dimodifikasi dalam beberapa cara sehingga berisi ciphertext, yang menghasilkan stegotext. Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik covertext lainnya dapat dimanipulasi untuk membawa pesan tersembunyi. Hanya penerima (yang harus mengetahui teknik yang digunakan) dapat membuka pesan dan mendekripsikannya.

Selain pengertian tersebut terdapat pula pengertian bahwa Steganografi membutuhkan dua properti yaitu wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video.

E. Metode *End Of File* (EOF)

Metode ini merupakan metode pengembangan LSB (Least Significant Bit). Dalam metode ini pesan disisipkan diakhir berkas. Pesan yang disisipkan dengan metode ini jumlahnya tidak terbatas. Akan tetapi efek sampingnya adalah ukuran berkas menjadi lebih besar dari ukuran semula. Ukuran berkas yang terlalu besar dari yang seharusnya, tentu akan menimbulkan kecurigaan bagi yang mengetahuinya.

Misalnya pada sebuah citra skala keabuan 6x6 piksel disisipkan pesan yang berbunyi “#aku”. Kode ASCII dari pesan tersebut adalah :

35 97 107 117

Misalkan matriks tingkat derajat keabuan citra sebagai berikut :

196 10 97 182 101 40
67 200 100 50 90 50
25 150 45 200 75 28
176 56 77 100 25 200
101 34 250 40 100 60
44 66 99 125 190 200

Pada akhir data gambar akan diberikan suatu penanda data gambar dan pesan, dalam contoh tanda sebagai berikut :

196 10 97 182 101 40
67 200 100 50 90 50
25 150 45 200 75 28
176 56 77 100 25 200
101 34 250 40 100 60
44 66 99 125 190 200
255 255 255 255

Kode biner pesan disisipkan di akhir citra yang telah diberikan penanda, sehingga gambar menjadi :

196 10 97 182 101 40
67 200 100 50 90 50
25 150 45 200 75 28
176 56 77 100 25 200
101 34 250 40 100 60
44 66 99 125 190 200
255 255 255 255
35 97 107 117

F. Perbedaan Kriptografi dan Steganografi

Steganografi dan kriptografi mempunyai prinsip kerja yang berbeda, meskipun keduanya mempunyai hubungan yang dekat dalam dunia keamanan data Hasil dari kriptografi biasanya berupa data yang berbeda dari bentuk aslinya dan biasanya data seolah-olah berantakan sehingga tidak dapat diketahui informasi apa yang terkandung didalamnya (namun sesungguhnya dapat dikembalikan ke bentuk semula lewat proses dekripsi), sedangkan hasil keluaran dari steganografi memiliki bentuk persepsi yang sama dengan bentuk aslinya. Kesamaan persepsi tersebut adalah oleh indera manusia (khususnya visual), namun bila digunakan komputer atau perangkat pengolah digital lainnya dapat dengan jelas dibedakan antara sebelum proses dan setelah proses.

.

III. HASIL DAN PEMBAHASAN

A. Analisa Masalah

Dinas Pendidikan Kabupaten Tangerang memiliki masalah dalam mengamankan dokumen penting. Dokumen tersimpan begitu saja dikomputer tanpa adanya pengamanan. Oleh karena itu kemungkinan pencurian data sangat mudah dilakukan. Sehingga dibutuhkan aplikasi pengamanan data, namun suatu pengamanan data saja tidak cukup karena menimbulkan

kecurigaan pada pihak ketiga. Untuk itu perlu penggabungan suatu metode agar data yang diamankan tidak dicurigai sebagai data yang penting oleh pihak ketiga, sehingga data tetap terjaga aman

B. Perancangan Program

Program yang dibuat terdiri dari sembilan buah form, yang terdiri dari form menu utama, form signin, form signup, form encode, form decode, form embed, form retrieve, form help, dan form about.

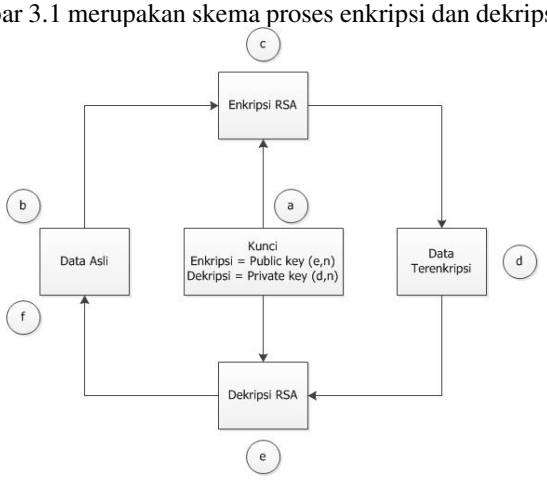
Untuk melakukan proses encode, user dapat memilih menu encode. Pada menu ini user diharuskan mengisi kata sandi sebagai kunci kemudian memilih master image sebagai media citra digital (gambar), memilh dokumen rahasia, dan menentukan lokasi penyimpanan hasil proses encode. Namun file yang diencode hanya sebatas file dokumen saja dan sesuai dengan ukuran yang sudah ditentukan.

Sedangkan untuk mengembalikan file yang sudah diencode menjadi file asli, user dapat memilih menu decode. Pada aplikasi ini juga disediakan menu help untuk membantu user dalam menggunakan aplikasi ini.

Skema proses enkripsi dan dekripsi *file* dapat diuraikan sebagai berikut :

1) Langkah awal untuk menggunakan aplikasi ini *user* diharuskan membuat kunci pada *form key* yang akan digunakan untuk melakukan proses enkripsi dan dekripsi.
2) Setelah membuat kunci *user* diharuskan untuk menginput *file* yang akan dienkripsi.
3) Kemudian dilakukan proses enkripsi RSA.
4) Setelah itu maka *file* akan terenkripsi.
5) Kemudian dilakukan proses dekripsi RSA pada *file* yang terenkripsi.
6) Lalu *file* tersebut akan kembali seperti semula.

Gambar 3.1 merupakan skema proses enkripsi dan dekripsi *file*.

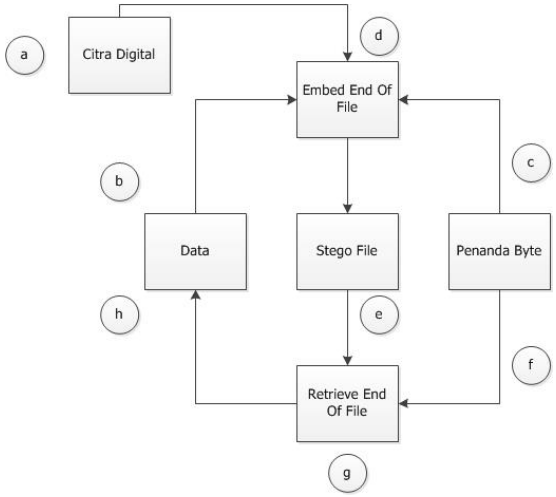


Gambar 10. Skema Proses Enkripsi Dan Dekripsi *File*

Skema proses *embed* dan *retrieve file* dapat diuraikan sebagai berikut :

1) Langkah awal *user* menginput citra *digital* sebagai media penyisipan.
2) Kemudian *user* menginput *file* yang akan disisipkan.
3) Lalu akan disisipkan penanda byte di akhir byte gambar.
4) Kemudian proses *embed* berjalan.
5) Setelah itu maka akan mendapat *stego file*.
6) Untuk melakukan *retrieve file*, dibutuhkan penanda byte.
7) Kemudian proses *retrieve file* berjalan.
8) Setelah itu *file* akan kembali.

Gambar 3.2 merupakan skema proses *embed* dan *retrieve file*.



Gambar 11. Skema Proses *Embed* dan *Retrieve File*.

IV. IMPLEMENTASI DAN UJICOBA

A. Data Uji Coba

Dalam pengujian program kali ini akan membahas antara lain :

1) Perbandingan antara proses enkripsi dan proses dekripsi antara *file word* (.doc, .docx), *file* excel (.xls, .xlsx), *file power point* (.ppt, .pptx), *file* PDF (.pdf), *file* text (.txt) dan *image* (.jpg, .png), yaitu meliputi ukuran awal *file* yang ingin dienkripsi, waktu proses enkripsi, waktu proses dekripsi dan hasil yang dicapai dalamproses enkripsi maupun dekripsi.
2) Perbandingan antara proses *embed* dan proses *retrieve* antara *file word* (.doc, .docx), *file excel* (.xls, .xlsx), *file power point* (.ppt, .pptx), *file* PDF (.pdf) dan *file text* (.txt), yaitu meliputi ukuran awal *file* yang ingin di-*embed*, waktu proses *embed*.

Tabel 4.
Data Uji Coba

No.	Nama *File* Dokumen	Jenis *File*	Ukuran *File*
1.	IPv6 Addressesing May2015.docx	Word 97-2003 (.doc)	1.748 KB
2.	Apakah Linux bebas virus.doc	Word Document (.docx)	283 KB
3.	Report.xls	Excel 97-2003 (.xls)	10 KB
4.	Tabel mencari netmask.xlsx	Excel Document (.xlsx)	335 KB
5.	pertemuan04_ver05.pdf	File PDF (.pdf)	1.897 KB
6.	Keamanan Data.txt	Text Document (.txt)	1.732 KB
7.	Hakikat Budi Luhur.ppt	Powerpoint 97-2003 (.ppt)	869 KB
8.	Chap8 Single Area OSPF.pptx	Powerpoint (.pptx)	1.869 KB
9.	Panitia Ultah BL.jpg	Image (.jpg)	675 KB
10.	admin.PNG	Image (.png)	237 KB

B. Hasil Uji Coba Proses Enkripsi

Pada uji coba proses enkripsi, *file* enkripsi yang dihasilkan diberi nama sesuai dengan *file* rahasia dan *file* rahasia yang digunakan adalah semua jenis *file* yang terdefinisi pada Tabel 4.1 Data Uji Coba. Kunci publik yang digunakan yaitu bonta.*PublicKey* dengan nilai n=1961 dan d=7. Hasil uji coba proses enkripsi dapat dilihat pada Tabel 5.

Tabel 5.
Hasil Uji Coba Proses Enkripsi

No.	Nama *File* Dokumen	Ukuran *File* Asli	Waktu Enkripsi (*second*)	Ukuran *File* Enkripsi	Status
1.	IPv6 Addressesing May2015.docx	1.748 KB	9.531 s	3.200 KB	Enkripsi Berhasil
2.	Apakah Linux bebas virus.doc	283 KB	0.733 s	488 KB	Enkripsi Berhasil
3.	Report.xls	10 KB	0.016 s	13 KB	Enkripsi Berhasil
4.	Tabel mencari netmask.xlsx	335 KB	0.795 s	620 KB	Enkripsi Berhasil
5.	pertemuan04_ver05.pdf	1.897 KB	15.787 s	3.502 KB	Enkripsi Berhasil
6.	Keamanan Data.txt	1.732 KB	10.748 s	2.990 KB	Enkripsi Berhasil
7.	Hakikat Budi Luhur.ppt	869 KB	2.387 s	1.512 KB	Enkripsi Berhasil
8.	Chap8 Single Area OSPF.pptx	1.869 KB	13.806 s	3.429 KB	Enkripsi Berhasil
9.	Panitia Ultah BL.jpg	675 KB	1.622 s	1.251 KB	Enkripsi Berhasil
10.	admin.PNG	237 KB	0.546 s	438 KB	Enkripsi Berhasil

C. Hasil Uji Coba Proses Dekripsi

Pada uji coba proses dekripsi dilakukan dengan dua cara, yaitu menggunakan kunci privat yang benar dan kunci privat yang salah pada masing-masing *file* yang telah dienkripsi. Proses dekripsi dengan kunci privat yang benar menggunakan bonta.*PrivateKey* dengan nilai n=1961 dan d=535 yang dapat dilihat pada Tabel 6.

Tabel 6.
Hasil Uji Coba Dekripsi Dengan Kunci Privat Yang Benar

No.	Nama *File* Dokumen	Ukuran *File* Enkripsi	Waktu Dekripsi (*second*)	Ukuran *File* Dekripsi	Status
1.	en.IPv6 Addressesing May2015.docx	3.200 KB	150.556 s	1.748 KB	Dekripsi Berhasil
2.	en.Apakah Linux bebas virus.doc	488 KB	18.97 s	283 KB	Dekripsi Berhasil
3.	en.Report.xls	13 KB	0.499 s	10 KB	Dekripsi Berhasil
4.	en.Tabel mencari netmask.xlsx	620 KB	25.1 s	335 KB	Dekripsi Berhasil
5.	en.pertemuan04_ver05.pdf	3.502 KB	193.518 s	1.897 KB	Dekripsi Berhasil
6.	en.Keamanan Data.txt	2.990 KB	143.832 s	1.732 KB	Dekripsi Berhasil
7.	en.Hakikat Budi Luhur.ppt	1.512 KB	58.859 s	869 KB	Dekripsi Berhasil
8.	en.Chap8 Single Area OSPF.pptx	3.429 KB	182.786 s	1.869 KB	Dekripsi Berhasil
9.	en.Panitia Ultah BL.jpg	1.251 KB	50.528 s	675 KB	Dekripsi Berhasil
10.	en.admin.PNG	438 KB	17.753 s	237 KB	Dekripsi Berhasil

Pada Tabel 7. dijelaskan hasil uji coba proses dekripsi dengan kunci privat yang salah yaitu mengunakan dewi.*PrivateKey*.

Tabel 7.
Hasil Uji Coba Dekripsi Dengan Kunci Privat Yang Salah

No.	Nama *File* Dokumen	Ukuran *File* Enkripsi	Waktu Dekripsi (*second*)	Ukuran *File* Dekripsi	Status
1.	en.IPv6 Addressesing May2015.docx	3.200 KB	491.214 s	1.748 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
2.	en.Apakah Linux bebas virus.doc	488 KB	70.387 s	283 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
3.	en.Report.xls	13 KB	1.389 s	10 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
4.	en.Tabel mencari netmask.xlsx	620 KB	90.262 s	335 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
5.	en.pertemuan04_ver05.pdf	3.502 KB	566.203 s	1.897 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
6.	en.Keamanan Data.txt	2.990 KB	465.989 s	1.732 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
7.	en.Hakikat Budi Luhur.ppt	1.512 KB	210.709 s	869 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
8.	en.Chap8 Single Area OSPF.pptx	3.429 KB	561.039 s	1.869 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
9.	en.Panitia Ultah BL.jpg	1.251 KB	50.528 s	675 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka
10.	en.admin.PNG	438 KB	17.753 s	237 KB	Dekripsi Berhasil, tetapi *file* tidak dapat dibuka

D. *Hasil Uji Coba Proses* Embed

Pada uji coba proses *embed* dilakukan dengan media gambar yang digunakan sebagai *image cover* yaitu b.jpg dengan ukuran *file* 18 KB dan Dimensi 960 x 540. Hasil uji coba proses *embed file* dengan file belum terenkripsi dapat dilihat pada Tabel 8.

Tabel 8.
Hasil Uji Coba Proses *Embed File* dengan *File* yang Belum Dienkripsi

No.	Nama *File* Dokumen	Ukuran *File* Asli	Ukuran *Embedded File*	Status
1.	IPv6 Addressesing May2015.docx	1.748 KB	1.766 KB	*Embed File* Berhasil
2.	Apakah Linux bebas virus.doc	283 KB	301 KB	*Embed File* Berhasil
3.	Report.xls	10 KB	29 KB	*Embed File* Berhasil
4.	Tabel mencari netmask.xlsx	335 KB	353 KB	*Embed File* Berhasil
5.	pertemuan04_ver05.pdf	1.897 KB	1.915 KB	*Embed File* Berhasil
6.	Keamanan Data.txt	1.732 KB	1.750 KB	*Embed File* Berhasil
7.	Hakikat Budi Luhur.ppt	869 KB	888 KB	*Embed File* Berhasil
8.	Chap8 Single Area OSPF.pptx	1.869 KB	1.887 KB	*Embed File* Berhasil

Pada Tabel 9. dijelaskan hasil uji coba proses *embed file* dengan *file* yang telah dienkripsi.

Tabel 9.
Hasil Uji Coba Proses *Embed File* dengan *File* yang Telah Dienkripsi

No.	Nama *File* Dokumen	Ukuran *File* Enkripsi	Ukuran *Embedded File*	Status
1.	en.IPv6 Addressesing May2015.docx	3.200 KB	3.218 KB	*Embed File* Berhasil
2.	en.Apakah Linux bebas virus.doc	488 KB	506 KB	*Embed File* Berhasil
3.	en.Report.xls	13 KB	31 KB	*Embed File* Berhasil
4.	en.Tabel mencari netmask.xlsx	620 KB	638 KB	*Embed File* Berhasil
5.	en.pertemuan04_ver05.pdf	3.502 KB	3.520 KB	*Embed File* Berhasil
6.	en.Keamanan Data.txt	2.990 KB	3.008 KB	*Embed File* Berhasil
7.	en.Hakikat Budi Luhur.ppt	1.512 KB	1.531 KB	*Embed File* Berhasil
8.	en.Chap8 Single Area OSPF.pptx	3.429 KB	3.447 KB	*Embed File* Berhasil

E. *Hasil Uji Coba Proses* Retrieve

Pada uji coba proses *retrieve* dilakukan dengan dua cara, yaitu melakukan proses *retrieve file* pada *embedded file* yang tidak berisi *file* enkripsi seperti yang terdefinisi pada Tabel 4.5 Hasil Uji Coba Proses *Embed File* dengan *File* yang Belum Dienkripsi dan melakukan proses *retrieve file* pada *embedded file* yang berisi *file* enkripsi yang terdefinisi pada Tabel 4.6 Hasil Uji Coba Proses *Embed File* dengan *File* yang Dienkripsi. Hasil uji coba proses *retrieve file* pada *embedded file* yang tidak berisi *file* enkripsi dapat dilihat pada Tabel 10.

Tabel 10.
Hasil Uji Coba Proses *Retrieve File* Pada *Embedded File* Yang Tidak Berisi *File* Enkripsi

No.	Nama *File* Dokumen	Ukuran *Embedded File*	Ukuran *Retrieve File*	Status
1.	IPv6 Addressesing May2015.jpg	1.766 KB	1.748 KB	*Retrieve File* Berhasil
2.	Apakah Linux bebas virus.jpg	301 KB	283 KB	*Retrieve File* Berhasil
3.	Report.jpg	29 KB	10 KB	*Retrieve File* Berhasil
4.	Tabel mencari netmask.jpg	353 KB	335 KB	*Retrieve File* Berhasil
5.	pertemuan04_ver05.jpg	1.915 KB	1.897 KB	*Retrieve File* Berhasil
6.	Keamanan Data.jpg	1.750 KB	1.732 KB	*Retrieve File* Berhasil
7.	Hakikat Budi Luhur.jpg	888 KB	869 KB	*Retrieve File* Berhasil
8.	Chap8 Single Area OSPF.jpg	1.887 KB	1.869 KB	*Retrieve File* Berhasil

Pada Tabel 11. dijelaskan hasil uji coba proses *retrieve file* pada *embedded file* yang berisi *file* enkripsi.

Tabel 11.
Hasil Uji Coba Proses *Retrieve File* Pada *Embedded File* Yang Berisi *File* Enkripsi

No.	Nama *File* Dokumen	Ukuran *Embedded File*	Ukuran *Retrieve File*	Status
1.	IPv6 Addressesing May2015.jpg	3.218 KB	3.200 KB	*Retrieve File* Berhasil
2.	Apakah Linux bebas virus.jpg	506 KB	488 KB	*Retrieve File* Berhasil
3.	Report.jpg	31 KB	13 KB	*Retrieve File* Berhasil
4.	Tabel mencari netmask.jpg	638 KB	620 KB	*Retrieve File* Berhasil
5.	pertemuan04_ver05.jpg	3.520 KB	3.502 KB	*Retrieve File* Berhasil
6.	Keamanan Data.jpg	3.008 KB	2.990 KB	*Retrieve File* Berhasil
7.	Hakikat Budi Luhur.jpg	1.531 KB	1.512 KB	*Retrieve File* Berhasil
8.	Chap8 Single Area OSPF.jpg	3.447 KB	3.429 KB	*Retrieve File* Berhasil

F. Evaluasi Program

Setelah dilakukan pengujian program terhadap program aplikasi ini, didapatkan beberapa kelebihan dan kekurangan dari aplikasi ini, sebagai berikut :

Kelebihan Program

1) Memberikan sistem pengamanan kunci yang terbaik.
2) Memberikan pengamanan ganda kepada *file* agar lebih aman.
3) Program memliki tampilan yang sederhana dan diharapkan dapat dengan mudah dijalankan oleh *user*.

4) *File* yang telah terenkripsi tidak dapat dibuka, sehingga meminimalkan kebocoran isi *file*.
5) *File* yang telah terenkripsi dapat disisipkan ke media gambar sehingga menghindari upaya pembobolan.
6) Proses penyisipan *file* dan pengembalian *file* yang cepat.

Kekurangan Program

1) Waktu yang dibutuhkan cukup lama untuk melakukan proses enkripsi dan dekripsi.
2) Ukuran *file* yang dihasilkan setelah proses enkripsi lebih besar dari ukuran *file* aslinya.

V. KESIMPULAN

Berdasarkan perancangan, pembuatan, analisa program dan serangkaian uji coba dari aplikasi ini, maka dapat diambil suatu kesimpulan antara lain :

1) Proses penyimpanan dan pertukaran informasi menjadi lebih aman, seperti *file word* (.doc, .docx), *file excel* (.xls, .xlsx), *file power point* (.ppt, .pptx), *file* pdf (.pdf), *file text* (.txt) dan *image* (.jpg, .png), baik file dokumen yang akan dikirim melalui email maupun *file* dokumen yang disimpan secara internal oleh PT. L7 Systems.
2) Tindakan pembobolan *file* yang terenkripsi akan berkurang karena adanya keamanan ganda menggunakan steganografi.
3) Tindakan pencurian, penyalahgunaan dan manipulasi data tidak dapat terjadi karena isi *file* dokumen sudah teracak.
4) Dengan menggunakan kunci yang berbeda saat enkripsi dan dekripsi maka keamanan data rahasia semakin terjaga dan aman.
5) Proses dekripsi dengan kunci yang asli akan mengembalikan *file* menjadi *file* semula tanpa mengalami perubahan sedikitpun.
6) Waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi berbanding lurus dengan ukuran *file* yang diproses (semakin kecil ukuran *file* yang diproses, semakin cepat proses enkripsi dan dekripsi dilakukan, semakin besar ukuran *file* yang diproses, semakin lama proses enkripsi dan dekripsi dilakukan).
7) Proses penyisipan *file* dan media *cover* sangat cepat sehingga tidak memakan waktu yang lama.

A. *Saran*

Adapun saran yang mungkin diperlukan untuk membuat aplikasi ini dapat berjalan lebih baik lagi antara lain :

1) Proses enkripsi dan penyisipan *file* pada aplikasi ini diharapkan dapat ditingkatkan kinerjanya sehingga tidak hanya dapat mengenkripsi dan menyisipkan *file* seperti *file word* (.doc, .docx), *file excel* (.xls, .xlsx), *file power point* (.ppt, .pptx), *file pdf* (.pdf), *file text* (.txt) dan *image* (.jpg, .png), namun *file* dokumen lainnya.
2) Proses enkripsi dan dekripsi dengan ukuran *file* yang besar diharapkan dapat bejalan lebih cepat dengan *hardware* yang lebih baik.

Ukuran *file* hasil enkripsi dan dekripsi diharapkan menjadi lebih kecil lagi dengan menggunakan algoritma kompresi yang lebih baik.

REFERENCES

[1] Maharani, Septya, Fahrul, Agus, Februari 2009, "*Implementasi Perangkat Lunak Penyandian Pesan Menggunakan Algoritma RSA*". Jurnal Informatika Mulawarman. Vol. 4, No. 1, https://informatikamulawarman.files.wordpress.com/2010/02/06-jurnal-vol4no1-2009-v-1-2hal13-20.pdf. 20 Oktober 2015.

[2] Martono, Irawan, September 2013, "*Penggunaan Steganografi dengan Metode End of File (Eof) pada Digital Watermarking*". Jurnal TICOM. Hal. 229-235, Vol. 2,No. 1, http://aptikom3.or.id/files/Jurnal%20TICOM%20Vol.%202%20No.%201%20Tahun%202013.pdf. 21 Oktober 2015.

[3] Meidina, 2013, "*Visualisasi Algoritma RSA Dengan Menggunakan Bahasa Pemrograman Java"*. Depok, Universitas Gunadarma, http://repository.gunadarma.ac.id/1331/1/VISUALISASI%20ALGORITMA%20RSA%20DENGAN%20MENGGUNAKAN%20_UG.pdf. 1 Oktober 2015.

[4] Prasestyo, Bagus, Galang, Santoso, Edy, Marji, 2013, "Kompresi File Audio Wave Menggunakan Algoritma *Huffman Shift Coding*". Malang, Universitas Brawijaya. http://filkom.ub.ac.id/doro/download/article/file/dr00016201306. 21 Oktober 2015.

[5] Rahajoeningroem, Tri, Muhammad, Aria, Mei 2011, "*STUDI DAN IMPLEMENTASI ALGORITMA RSA UNTUK PENGAMANAN DATA TRANSKRIP AKADEMIK MAHASISWA*". Majalah Ilmiah UNIKOM. Vol. 8, No. 1, http://jurnal.unikom.ac.id/jurnal/studi-dan-implementasi.pdf. 13 Oktober 2015.

[6] Sembiring, Sandro, Agustus 2013, "*Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan teks Pada Gambar Dengan Metode End of File"*. Pelita Informatika Budi Dharma. Vol. 4, No. 2, http://pelita-informatika.com/berkas/jurnal/429.pdf. 2 Oktober 2015.

[7] Tuturoong, J Nancy, Agustus 2010, "*Perbandingan Rasio dan Kecepatan Kopresi Menggunakan Algoritma Huffman, LZW dan DMC".* TEKNO. Vol. 8, No. 53. http://ejournal.unsrat.ac.id/index.php/tekno/article/viewFile/4320/3849. 2 Oktober 2015.

[8] Wahyuni, ana, Mei 2011, "*Keamanan Pertukaran Kunci Kriptografi dengan Algoritma Hybrid : Diffie-Hellman dan RSA"*. Majalah Ilmiah INFORMATIKA. Vol. 2, No. 2, http://www.unaki.ac.id/ejournal/index.php/jurnal-informatika/article/download/58/57. 20 Oktober 2015.