Aplikasi Pengamanan Data Menggunakan Algoritma Steganografi *Discrete Cosine Transform* dan Kriptografi AES 128 BIT pada SMK PGRI 15 Jakarta

Rika Rahmawati[#1], Dani Rahardjo[*2]

[#]*Magister Ilmu Komputer, Universitas Budi Luhur*
Jln. Ciledug Raya, Petukangan Utara, Jakarta Selatan 12260 INDONESIA
[1]rika_ubl10@yahoo.co.id
[*]*Magister Ilmu Komputer, Universitas Budi Luhur*
Jln. Ciledug Raya, Petukangan Utara, Jakarta Selatan 12260 INDONESIA
[2]danirahardjo@gmail.com

Abstract **— SMK PGRI 15 Jakarta has a lot of important documents such as financial reports, document collaboration and other important papers. Document security is still not done correctly so that the stored information is still very vulnerable to unknown, retrieved and manipulated by parties who are not entitled to that information. One way to secure the information is by using a steganography technique. So the objective of this research is to build desktop-based applications to secure the data (information) by inserting secret messages that have been encrypted in the form of an image file. Steganography algorithm used is DCT (Discrete Cosine Transform). Data that has been inserted into the previous image is already encrypted with 128 bit AES algorithm, so that data confidentiality is guaranteed. Based on the results of experiments conducted in this study, File Embedded process have an average completion time of 432.3 seconds, and managed to get a good quality Steganography with a relatively small MSE average value of 1.38 dB and an average PSNR of 47.66 dB. While the process Extract Files that have an average completion time of 139,6detik, as well as the success rate of 100%. Steganography DCT (Discrete Cosine Transform) and Cryptographic techniques AES (Advanced Encryption Standard) 128 bit is very helpful in maintaining the confidentiality of documents that are not easily read by people who are not entitled to the document. So the security issues faced by SMK PGRI 15 Jakarta can be resolved.**

Keywords**— Steganography, Discrete Cosine Transform, AES 128, Cryptography.**

I. PENDAHULUAN

Pesatnya perkembangan teknologi informasi saat ini memudahkan manusia dalam melakukan komunikasi dan berbagi informasi. Namun, dengan adanya kemudahan membuat orang lupa bahwa keamanan dan privasi data merupakan bagian yang sangat penting dalam berkomunikasi. SMK PGRI 15 Jakarta memiliki banyak data penting seperti laporan keuangan, dokumen kerja sama, dan surat peting lainnya. Data tersebut masih belum melakukan pengamanan data secara benar. Seiring perkembangan teknologi, hal tersebut memiliki dampak negatif berupa pencurian atau manipulasi data. Sehingga aspek keamanan pada dokumen sangat penting. Dokumen rahasia sekolah akan merugikan sekolah apabila jatuh ke pihak yang tidak berhak untuk disalahgunakan. Karena hal tersebut, diperlukanlah suatu aplikasi pengamanan data yang bisa mengamankan suatu data serta tidak menimbulkan kecurigaan oleh pihak yang tidak berhak.

II. LANDASAN TEORI

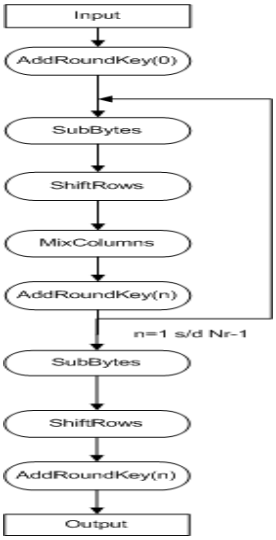
A. Algoritma Kripstografi AES 128 Bit

Algoritma AES termasuk dalam jenis algoritma Kriptografi yang sifatnya simetri dan *cipher block*. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Algoritma AES mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun AES mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan [9].

TABEL I
PERBANDINGAN PANJANG KUNCI AES

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

1) *Proses Enkripsi AES 128 Bit:* AES memiliki ukuran blok dan kunci yang tetap sebesar 128, 192, atau 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Blok-blok data masukan dan kunci dioperasikan dalam bentuk *array*. Setiap anggota array sebelum menghasilkan keluaran *ciphertext* dinamakan dengan state. Setiap state akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, *AddRoundKey*, *SubBytes*, *ShiftRows*, dan *MixColumns*. Kecuali tahap *MixColumns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap *MixColumns* tidak akan dilakukan pada tahap terakhir.



Gambar 1. Ilustrasi Algoritma Enkripsi AES

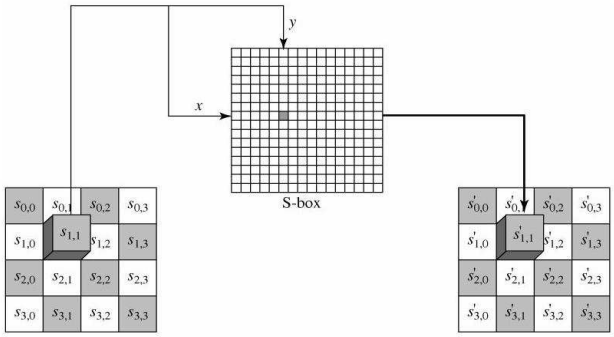
Proses algoritma AES Rijndael yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut: [3]

1. *AddRoundKey*, pada dasarnya adalah melakukan XOR antara *state* awal (*plain text*) dengan *cipherkey*. Tahap ini disebut juga *initial round*.
2. *Round*, merupakan putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
3. *SubBytes*, yaitu substitusi *byte* dengan menggunakan tabel substitusi (S-box).

TABEL II
TABEL S-BOX

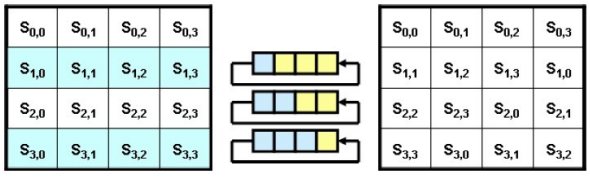
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
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 di dalam tabel substitusi yang merupakan pengaruh pemetaan *byte* pada setiap *byte* dan *state*.



Gambar 2. Ilustrasi Proses *SubBytes*

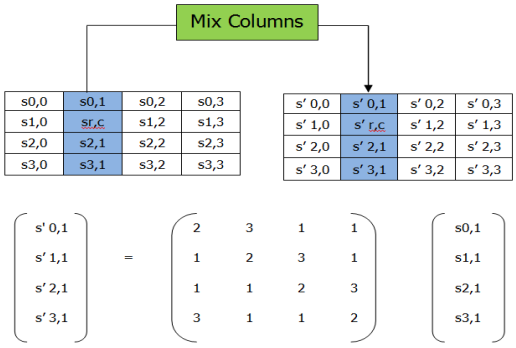
ShiftRow, yaitu sebuah proses yang melakukan *shift* atau pergeseran pada setiap elemen blok/tabel yang dilakukan per barisnya. Baris pertama tidak dilakukan pergeseran, baris kedua dilakukan pergeseran 1 *byte*, baris ketiga dilakukan pergeseran 2 *byte*, dan baris keempat dilakukan pergeseran 3 *byte*. Pergeseran tersebut terlihat dalam sebuah blok adalah sebuah pergeseran tiap elemen ke kiri tergantung berapa *byte* tergesernya, tiap pergeseran 1 *byte* berarti bergeser ke kiri sebanyak satu kali.



Gambar 3. Ilustrasi Proses *ShiftRows*

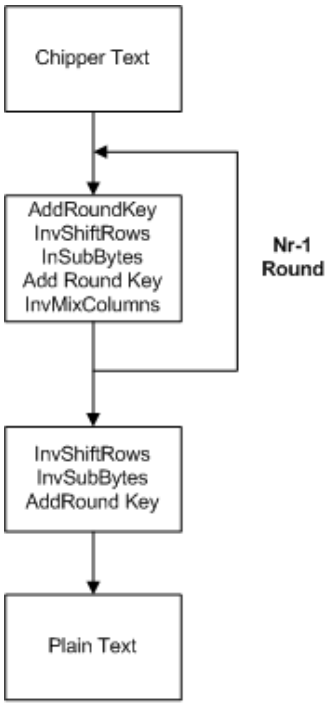
MixColumns, yaitu tahapan untuk mengalikan tiap elemen dari *blokcipher* dengan matriks yang ditunjukkan oleh gambar 2.7. Tabel sudah ditentukan dan siap pakai. Pengalian dilakukan seperti perkalian matriks biasa yaitu menggunakan dot *product* lalu perkalian keduanya dimasukkan ke dalam sebuah *blokcipher* baru. Dengan

begitu seluruh rangkaian proses yang terjadi pada AES telah dijelaskan dan selanjutnya adalah menerangkan mengenai penggunaan tiap-tiap proses tersebut.



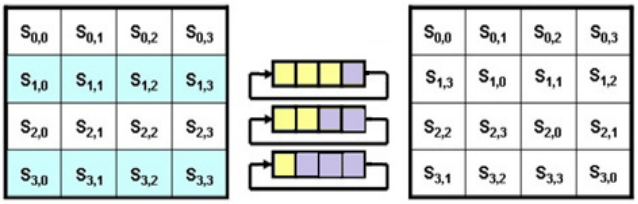
Gambar 4. Ilustrasi Proses *MixColumns*

2) *Proses Dekripsi AES 128 Bit:* 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* pada proses dekripsi AES adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey.



Gambar 5. Skema Proses Dekripsi AES

1. InvShiftRows yaitu transformasi byte yang berbalikan dengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri.



Gambar 6. Ilustrasi Proses InShiftRows

2. InvSub Bytes yaitu transformasi bytes yang berbalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada state dipetakan dengan menggunakan table Inverse S-Box.

TABEL III
TABEL INVERSE S-BOX

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

3. InvMix Columns, yaitu setiap kolom dalam *state* dikalikan dengan matriks perkalian dalam AES.

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

Gambar 7. Perkalian Matriks InvMixColumns

B. Algoritma Discrete Cosine Transform

Discrete cosine transform merupakan sebuah fungsi matematika yang digunakan untuk mengubah nilai-nilai sinyal dari suatu media menjadi komponen frekuensi dasarnya[15]. Dalam citra digital, DCT digunakan untuk mengubah domain spasial gambar menjadi domain frekuensinya.

DCT memiliki fungsi sebagai *energy compaction*, bertujuan mengonsentrasikan energi dari suatu media (dalam hal ini citra) ke dalam sejumlah koefisien dan *decorellation*, yaitu meminimalkan ketergantungan yang terjadi antar koefisien. Proses konsentrasian energi tersebut dilakukan di pojok kiri atas dari koefisien matriks. Di dalam matriks terdapat 3 macam energi frekuensi, yaitu frekuensi

rendah, tengah (bagian yang akan disisipkan), dan frekuensi tinggi.

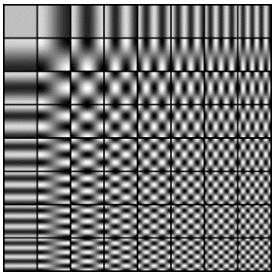
Rumus untuk *forward* DCT [15] adalah sebagai berikut:

$$S(u,v) = \frac{2}{\sqrt{nm}} C(u)C(v) \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} S(x,y) Cos \frac{(2_x + 1)u\pi}{2n} Cos \frac{(2_y + 1)v\pi}{2m}$$

Dengan u=0,...,n-1 ; v=0,...,m-1

Di mana $C(u) = \begin{cases} 2^{-1}/_2, u = 0 \\ 1, u \neq 0 \end{cases}$

Setiap elemen dari hasil transformasi S(u,v) merupakan hasil *dot product* atau *inner product* dari masukan s(x,y) dan basis vektor. DCT juga dapat diperoleh dari produk vektor (masukan) dan n x m matriks ortogonal yang setiap barisnya merupakan basis vektor.



Gambar 8. Delapan basis vektor DCT dengan n=8

Sedangkan invers DCT dapat diperoleh dengan rumus sebagai berikut :

$$S(x,y) = \frac{2}{\sqrt{nm}} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} S(u,v) C(u) C(v) Cos \frac{(2_x + 1)u\pi}{2n} Cos \frac{(2_y + 1)v\pi}{2m}$$

Dengan u=0,...,n-1 ; v=0,...,m-1

Persamaan di atas menyatakan s sebagai kombinasi linier dari basis vektor. *Discrete cosine transform* merepresentasikan sebuah citra dari penjumlahan sinusoidal dari magnitudo dan frekuensi yang berubah-ubah.

III. RANCANGAN SISTEM DAN APLIKASI

A. *Analisis Masalah*

SMK PGRI 15 Jakarta memiliki beberapa *file* dokumen penting berupa *file* dokumen pribadi, sekolah atau organisasi dan lain sebagainya. *File* penting tersebut masih memiliki tingkat keamanan yang rendah karena *file* tersimpan begitu saja di komputer tanpa adanya pengamanan yang berarti. Sering kali masalah keamanan menjadi urutan kedua atau bahkan urutan yang terakhir dalam daftar hal-hal yang dianggap penting. Sebuah *file* dokumen seharusnya dijaga kerahasiaannya agar tidak disalahgunakan oleh orang yang tidak berhak. Walaupun *file* tersebut sudah diamankan masih saja menimbulkan kecurigaan oleh pihak ketiga. Oleh karena itu, *file* dokumen yang tidak diamankan secara maksimal memungkinkan pencurian data sangat mudah dilakukan.

B. *Penyelesaian Masalah*

Dari permasalahan yang telah diuraikan di atas, diperlukan adanya aplikasi yang dapat menjaga kerahasiaan isi *file* dokumen pada SMK PGRI 15 Jakarta tersebut. Aplikasi tersebut nantinya dapat mengubah sebuah *file* dokumen menjadi *file* dokumen yang isinya tidak bisa dibaca dan *file* dokumen tersebut terjaga kerahasiaannya. Untuk menghilangkan faktor kecurigaan oleh pihak ketiga, dokumen tersebut akan disteganografi atau disisipkan ke dalam media lain, yaitu media gambar. Kemudian mengembalikan dokumen tersebut menjadi seperti semula tanpa mengalami perubahan sedikit pun.

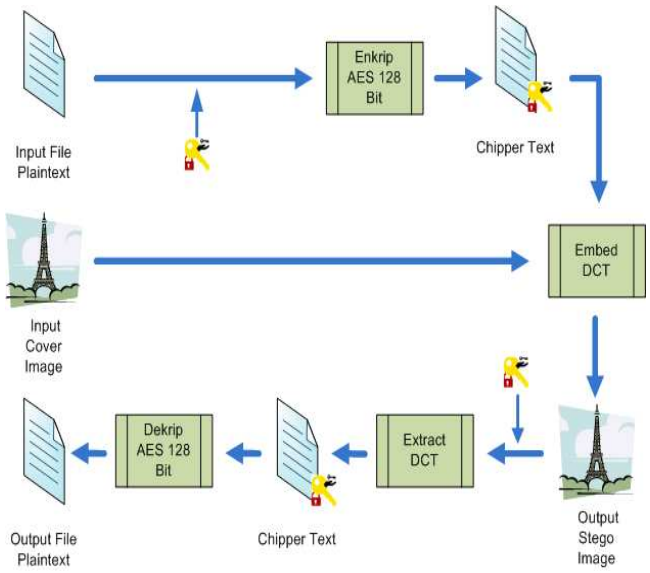
Aplikasi ini dibuat berbasis desktop yang dibangun dengan menggunakan bahasa pemrograman Java. Dalam pembuatan aplikasi ini menggunakan algoritma Kriptografi *Advanced Encryption Standard* (AES) 128 Bit dan metode Steganografi *Discrete Cosine Transform* (DCT). Dengan adanya aplikasi ini diharapkan suatu dokumen atau data penting dapat disimpan dan dikirim ke pihak yang benar-benar berhak, tidak menimbulkan kecurigaan pada saat penyisipan dan tidak disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab.

C. *Perancangan Program*

Program yang dibuat terdiri dari lima buah *form*, yang terdiri dari *Form* Menu Utama, *Form Embedded File*, *Form Extract File*, *Form Guide Note* dan *Form About Program*. Pada Menu Utama terdapat 1 menu, yaitu menu *Help* dan 2 tombol untuk menuju menu *Embedded File* dan *Extract File*. Pada menu *Help* terdapat 2 menu item, yaitu menu item *Guide Note* dan *About* Program.

Untuk melakukan proses *Embed file*, *user* dapat memilih tombol *Embedded File*. Pada *form* ini *user* diharuskan memilih *file* yang ingin dienkripsi namun *file* yang di enkripsi hanya sebatas *file* dokumen saja dan sesuai dengan ukuran yang sudah ditentukan, lalu masukan *password* dan pilih gambar untuk menyisipkan dokumen hasil enkripsi tersebut. Selanjutnya akan tampil *output* berupa informasi hasil *Embedded file* tersebut.

Sedangkan untuk mengembalikan *file* yang sudah di *embed* menjadi *file* semula, *user* dapat memilih tombol *Extract File*. Pada program ini juga disediakan menu *Help* untuk membantu *user* dalam menggunakan program ini.



Gambar 9. Arsitektur Aplikasi Pengaman Data

D. Rancangan Layar

Dalam perancangan layar dibagi menjadi beberapa proses sebagai berikut :

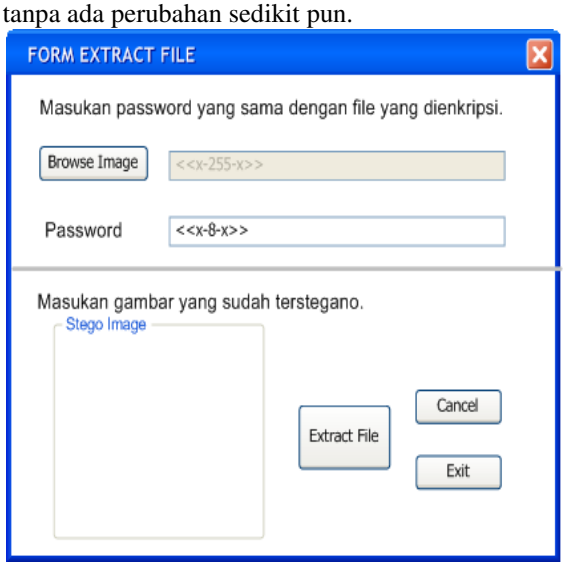
1) Rancangan Layar Form Embedded File

Rancangan layar pada *form Embedded File* digunakan untuk mengenkripsi *file* dokumen dan menyisipkan (*embed*) ke dalam media gambar. Textbox nilai PSNR dan mutu Steganografi berfungsi untuk memperlihatkan hasil penghitungan mutu Steganografi di mana jika nilai PSNR lebih dari 40 maka mutu Steganografi akan menampilkan tulisan baik dan sebaliknya.

Gambar 10. *Form* Menu *Embedded File*

2) Rancangan Layar Form Extract File

Rancangan layar pada *form Extract File* digunakan untuk mengeluarkan *file* dokumen yang sudah disisipkan di media gambar dan setelah itu didekripsi untuk mengembalikan *file* asli tanpa ada perubahan sedikit pun.

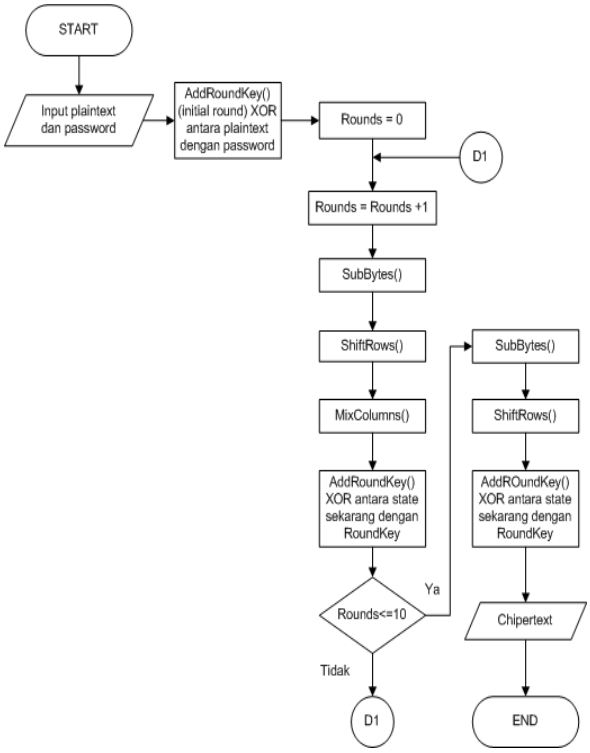


Gambar 11. *Form* Menu *Extract File*

E. Flowchart

1) Flowchart Enkripsi AES 128

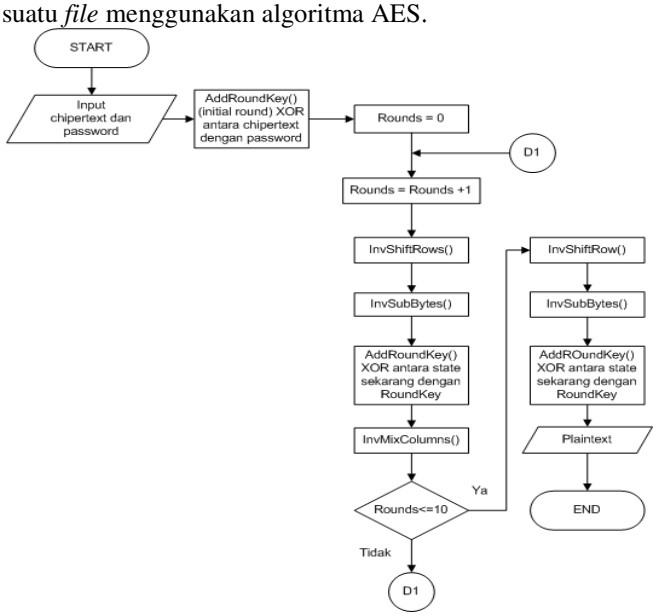
Flowchart ini merupakan alur jalannya proses enkripsi suatu *file* menggunakan algoritma AES.



Gambar 12. *Flowchart* Proses Enkripsi AES

2) Flowchart Dekripsi AES 128

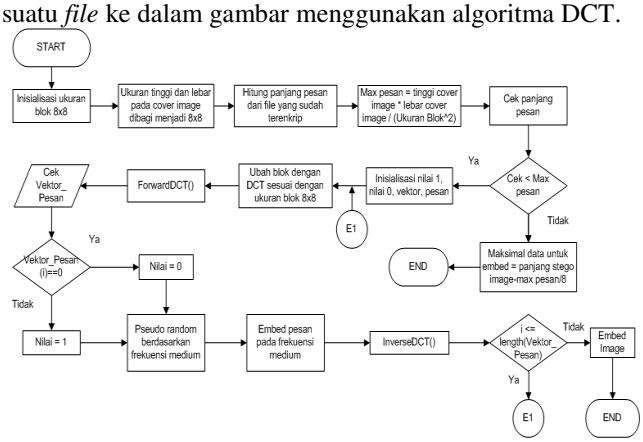
Flowchart ini merupakan alur jalannya proses dekripsi suatu *file* menggunakan algoritma AES.



Gambar 13. *Flowchart* Proses Dekripsi AES

3) Flowchart Embedded DCT

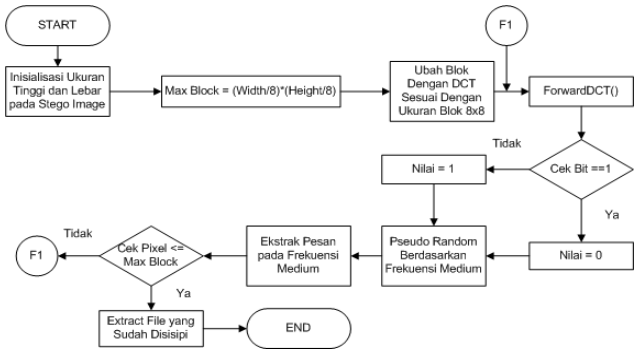
Flowchart ini merupakan alur jalannya proses penyisipan suatu *file* ke dalam gambar menggunakan algoritma DCT.



Gambar 14. *Flowchart* Proses *Embed* DCT

4) Flowchart Extract DCT

Flowchart ini merupakan alur jalannya proses pengeluaran suatu *file* dari gambar yang sudah disisipkan menggunakan algoritma DCT.



Gambar 15. *Flowchart* Proses *Extract* DCT

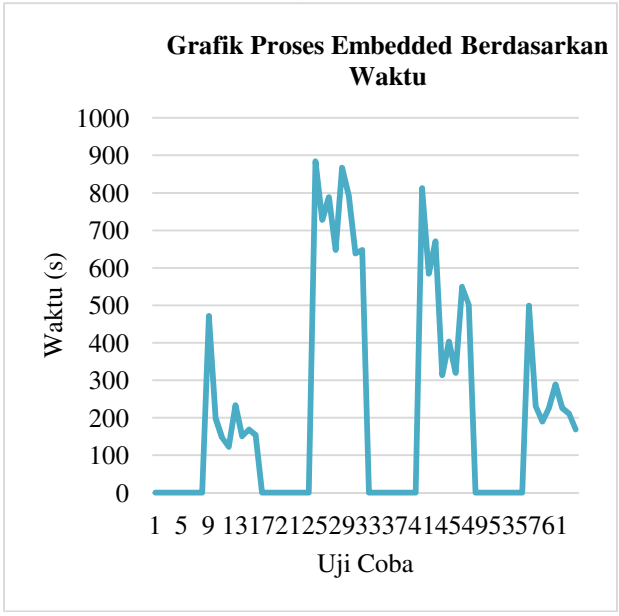
IV. HASIL DAN PEMBAHASAN

A. Hasil Uji Coba

Dalam pengujian ini, akan dibahas perbandingan antara proses *Embedded File* dan *Extract File* yang berisikan file txt, doc, docx, xls, xlsx dan pdf. Pengujian tersebut dilakukan 64 kali dengan menguji 8 *file* dokumen pada setiap 8 file gambar dari sampel yang diberikan untuk mengetahui *performance* embed pesan dengan citra digital, proses ekstrak stego image dan memastikan semua aplikasi berjalan dengan baik.

Berdasarkan grafik di bawah ini (Gambar 16), dari 64 kali proses uji, yang berhasil hanya 32 proses, yaitu proses *embed* dengan *file* di bawah 500 Kb. Sedangkan file dokumen di atas 500 Kb menghasilkan proses yang gagal. Proses *Embedded File* yang gagal dikarenakan besar resolusi gambar tidak mencukupi untuk dilakukan penyisipan pesan.

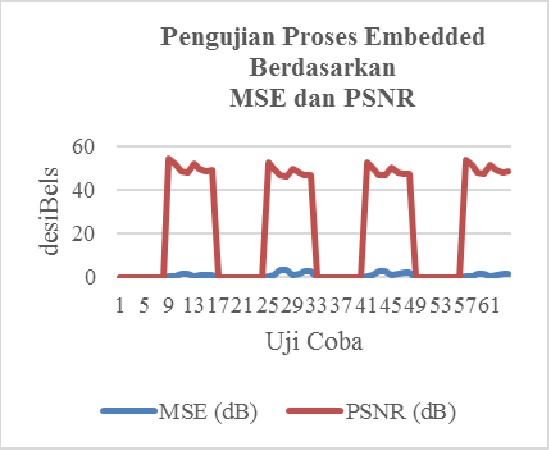
1) Pengujian proses Embedded File berdasarkan waktu



Gambar 16. Grafik Waktu Proses *Embedded*

Selisih waktu dari 64 *file* uji yang dicoba memiliki waktu yang beragam tergantung berapa besar *file* yang disisipkan. Waktu terlama yang dihasilkan adalah 884,2 detik dengan ukuran *file* dokumen sebesar 41 Kb (setelah dienkrip menjadi 108 Kb) dan *file* gambar sebesar 1250 Kb. Sedangkan waktu yang tercepat adalah 122,1 detik dengan ukuran *file* dokumen sebesar 17 Kb (setelah dienkrip menjadi 45 Kb) dan *file* gambar sebesar 326 Kb. Rata-rata waktu yang diperlukan untuk menyelesaikan proses *Embedded File* adalah 432,3 detik.

2) Pengujian proses Embedded File berdasarkan MSE dan PSNR

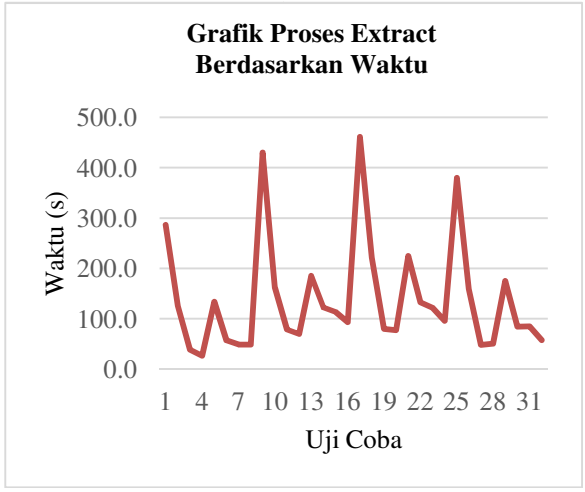


Gambar 17. Grafik MSE dan PSNR

Dari 64 *file* uji yang dicoba memiliki gambar hasil Steganografi yang beragam untuk ukuran dan resolusinya. Penilaian MSE dihasilkan dari berapa besar nilai eror yang dimiliki pada gambar-gambar tersebut. Nilai MSE terbaik adalah nilai yang paling kecil, yaitu sebesar 0,28 dB. Sedangkan nilai MSE yang tertinggi sebesar 3,23 dB. Rata-rata nilai MSE juga relatif kecil, yaitu sebesar 1,38 dB.

Berdasarkan grafik di atas, terlihat bahwa perbandingan nilai MSE dan PSNR sangat jauh. Dari hasil pengujian MSE, maka didapatlah nilai untuk penghitungan PSNR. Penilaian PNSR menunjukkan kualitas mutu dari Steganografi. Nilai PSNR tertinggi 53,65 dB. Sedangkan nilai PSNR yang terendah sebesar 43,04 dB. Rata-rata nilai PSNR yang diuji memiliki kualitas mutu Steganografi yang baik, yaitu sebesar 47,66 dB.

3) Pengujian proses Extract File berdasarkan waktu



Gambar 18. Grafik Waktu Proses *Extract*

V. Evaluasi Program

Berdasarkan hasil uji coba program dan eksekusi aplikasi yang dilakukan, didapat beberapa kelebihan dan kekurangan pada aplikasi pengamanan data menggunakan metode Steganografi DCT dan Kriptografi AES 128 Bit, yaitu sebagai berikut:

Kelebihan Aplikasi, sebagai berikut :

1. Pesan yang berada dalam *stego image* sudah berbentuk *ciphertext* sehingga tidak mudah dikenali.
2. Proses Embedded File akan selalu berhasil jika *cover image* memiliki ukuran yang dapat menampung pesan tersebut.
3. Perubahan yang terjadi antara *cover image* dengan *stego image* tidak terlalu signifikan.
4. *File* yang tadinya terenkripsi otomatis akan kembali normal pada proses *Extract File* apabila dimasukkan password yang sama pada saat proses *Extract File* di aplikasi ini.

Kekurangan Aplikasi, sebagai berikut :

1. Aplikasi ini hanya dapat mengenkripsi *file* berformat .txt, .doc, .docx, .xls,.xlsx, dan .pdf.
2. Media *cover* yang bisa disisipi dengan pesan hanya gambar yang bertipe PNG dan JPEG.

VI. Simpulan

Simpulan yang dapat diambil adalah sebagai berikut:

1. Metode Steganografi DCT (*Discrete Cosine Transform*) dan teknik Kriptografi AES (*Advanced Encryption* Standard) 128 Bit sangat membantu dalam menjaga kerahasiaan pesan agar tidak mudah dibaca oleh orang yang tidak memiliki kepentingan.
2. Proses *Embedded File* yang dilakukan memiliki rata-rata waktu penyelesaian sebesar 432,3 detik, serta berhasil mendapatkan mutu Steganografi yang baik dengan rata-rata nilai MSE yang relatif kecil sebesar 1,38 dB dan rata-rata nilai PSNR sebesar 47,66 dB.

3. Proses *Extract File* yang dilakukan memiliki rata-rata waktu penyelesaian sebesar 139,6 detik, serta tingkat keberhasilan sebesar 100%.

Aplikasi ini dapat dikembangkan menjadi lebih baik lagi melalui berbagai pengembangan sebagai berikut:

1. Waktu proses *Embedded* dan *Extract File* yang rata-rata berukuran besar diharapkan dapat berjalan lebih cepat pada *hardware* yang lebih baik.
2. Proses penyisipan bisa dijalankan pada gambar yang beresolusi kecil untuk pengembangan selanjutnya.
3. Dalam pengembangan lebih lanjut dapat difokuskan bukan hanya media gambar melainkan pada audio atau video untuk media steganografi.

DAFTAR PUSTAKA

[1] Aditya, Y., Pratama, A. dan Nurlifa, A., 2010. Studi pustaka untuk steganografi dengan beberapa metode. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010), 2010 (Snati), pp.32–35.

[2] Ariyus, D., 2008. Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi, Yogyakarta: ANDI OFFSET.

[3] Daemen, J. dan Rijmen, V., 1999. AES Proposal: Rijndael. , 2.

[4] Dwiandiyanta, B.Y., 2011. Perbandingan Watermarking Citra dengan Alihragam Wavelet dan Discrete Cosine Transform. Jurnal Buana Informatika, 2, pp.109–119.

[5] Gunjal, M. dan Jha, J., 2014. Image Steganography Using Discrete Cosine Transform (DCT) and Blowfish Algorithm. International Journal of Computer Trends and Technology (IJCTT), 11(4), pp.144–150.

[6] Ilhamsyah, A., 2014. Steganografi pada Citra JPEG dengan Memanfaatkan Koefisien DCT Terkuantisasi. Universitas Budi Luhur.

[7] Johnson, N.F. dan Mason, G., 1998. Exploring Steganography: Seeing the Unseen.

[8] Karandikar, A. dan Chiddarwar, P.G.G., 2014. Digital Image Protection Using Adaptive Watermarking Techniques. , p.10.

[9] Kromodimeljo, S., 2010. Teori dan Aplikasi Kriptografi, SPK IT Consulting.

[10] Menezes, A.J., Oorschot, P.C. Van dan Vanstone, S. a., 2014. Handbook of Applied Cryptography, New York: CRC Press.

[11] Pratama, A.N., 2014. STEGANOGRAPHY PADA MEDIA GAMBAR DENGAN MENGGUNAKAN DISCRETE COSINE TRANSFORMS (DCT). Universitas Budi Luhur.

[12] Rosyadi, A., 2012. Implementasi Algoritma Kriptografi AES Untuk Enkripsi dan Dekripsi Email. UNDIP Tembalang, pp.2–6.

[13] Sholeh, M. dan Hamokwarong, J. V, 2011. Aplikasi Kriptografi dengan Metode Vernam Cipher dan Metode Permutasi Biner. Momentum, 7(2), pp.8–13.

[14] Solichin, A., 2015. Mengukur Kualitas Citra Hasil Steganografi. , (April), pp.3–6.

[15] Watson, A.B. (Nasa A.R.C., 1994. Image Compression Using the Discrete Cosine Transform. Mathematica Journal, 4(1), pp.81–88.