Teknik Penyembunyian Data Menggunakan Kombinasi Kriptografi Rijndael dan Steganografi Least Significant Bit (LSB)

Dwi Ely Kurniawan[#1], Narupi[*2]

[#]*Jurusan Teknik Informatika Politeknik Negeri Batam*
Jl. Ahmad Yani, Batam Center, Batam
[1]dwialikhs@gmail.com
[2]narupi.narupi@yahoo.com

Abstract — ***So many users do save, send and share data in a network. Investigated that happened cyber crime such as theft and misuse of data so that the security level should be further enhanced with the data hiding techniques. In this study, the data hiding technique is done by combining Rijndael algorithm and Least Significant Bit (LSB). The system will encrypt and insert messages into picture, and then the system decipher hidden message so that can be read by a receiver. Testing is done by sending the results of Rijndael and Least Significant Bits (LSB) to various media sender. Results from study indicate that the data is encrypted before and after have same relative size, but changing the image quality. Change of image quality depending on size of bit values of media and message.***

Keywords— ***Encryption, Least Significant Bit, Rijndael***

I. PENDAHULUAN

Komunikasi dalam komputer saat ini menjadi tren seperti menyimpan data, mengirim data dan melakukan *sharing* data dalam suatu jaringan, terkadang pengguna tidak menyadari pentingnya mengamankan data dari pencurian atau serangan seseorang (pihak ketiga) yang tidak berhak terhadap data dari pengguna tersebut. Suatu hal yang dirasa perlu dan penting bagi pengguna adalah teknik dalam penyembunyian data. Berdasarkan informasi Kepala Subdirektorat IV Cyber Crime Ditreskrimsus Polda Metro Jaya jumlah laporan penipuan mencapai 40 persen dari seluruh kasus *cyber crime*, kasus pencemaran nama baik sekitar 30 persen dan sisanya adalah kejahatan pencurian data (*hacking*) dan kejahatan *cyber* lainnya [7]. Hal ini menunjukkan bahwa tingkat keamanan data haruslah ditingkatkan.

Aspek yang sangat penting dalam komunikasi data adalah masalah keamanan dan kerahasiaan data [10]. Keamanan suatu informasi sangat penting, baik pada saat pengiriman ataupun pada saat informasi tersebut diterima. Apabila informasi jatuh ke pihak lain, hal tersebut dapat menimbulkan kerugian bagi pemilik informasi tersebut. Oleh karena itu diperlukan adanya teknik atau seni untuk mengamankan data atau informasi. Data tersebut dapat berupa teks, gambar, audio, video, file kompresi ataupun data lainnya. Salah satu teknik yang dapat digunakan untuk mengamankan data adalah dengan menggunakan steganografi dan kriptografi.

Steganografi merupakan teknik menyembunyikan informasi ke dalam sebuah media, bisa berupa media gambar, suara ataupun video [9]. Steganografi secara teknis berarti pesan yang ditutupi, disisipkan atau pesan yang disembunyikan. Artinya steganografi dapat menyisipkan pesan rahasia ke dalam media lain dan mengirimkannya tanpa ada yang menyadari keberadaan pesan tersebut. Salah satu metode steganografi adalah *Least Significant Bit* (LSB), metode yang sederhana tidak terlalu kompleks namun pesan yang disembunyikan cukup aman [2] [13].

Sedangkan kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut dengan melakukan pembangkitan kunci, enkripsi dan dekripsi [3]. Kriptografi mempunyai dua bagian penting, yaitu enkripsi dan deskripsi. Enkripsi adalah proses penyandian dari pesan asli (*plaintext*) menjadi pesan yang tidak dapat diartikan (*ciphertext*). Sedangkan dekripsi sendiri berarti merubah pesan yang sudah disandikan (*ciphertext*) menjadi pesan aslinya (*plaintext*). *Advanced Encryption Standards* (AES) merupakan algoritma kriptografi yang didesain oleh Vincent Rijmen dan John Daemen dan pada tahun 2000. AES sering disebut algoritma Rijndael terpilih sebagai standar algoritma kriptografi karena implementasi aman dan juga efisien [10].

Steganografi dan kriptografi merupakan seni dan ilmu untuk menjaga pesan atau data. Steganografi menyisipkan data rahasia ke dalam sebuah media, sedangkan kriptografi merubah data dari *plaintext* menjadi *ciphertext*. Berdasarkan permasalahan keamanan data dengan tren teknologi yang ada maka penelitian ini mencoba mengembangkan dan mengimplementasikan teknik penyembunyian data dengan kriptografi rijndael dan steganografi LSB. Meskipun kedua metode ini memiliki perbedaan namun bila dikombinasikan maka dapat meningkatkan keamanan data tersebut.

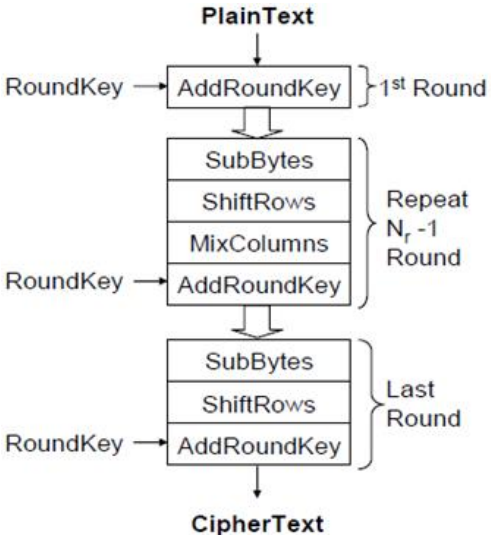
II. LANDASAN TEORI

A. Kriptografi

Kriptografi menggunakan suatu algoritma (*chiper*) dan kunci (*key*). Chiper merupakan fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi data. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi. Tidak sekedar mengandalkan keamanannya pada kerahasian algoritma tetapi kerahasian kunci. *Plaintext* yang sama bila disandikan dengan kunci yang berbeda akan menghasilkan *chipertext* yang berbeda pula. Artinya algoritma kriptografi yang digunakan boleh saja diketahui umum namun tanpa pengetahuan kunci, data tetap tidak terpecahkan.

Salah satu metode kriptografi adalah *Advanced Encryption Standard* (AES) atau biasa disebut dengan algoritma rijndael. Rijndael mendukung panjang kunci 128 bit sampai 256 bit, maka dikenal dengan AES-128, AES-192, dan AES-256. Proses enkripsi rijndael terdiri atas 4 jenis transformasi byte yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey.

Gambar 1 menjelaskan pada awal proses enkripsi, masukan yang telah berbentuk *array state* dilakukan transformasi AddRoundKey(). Tahap ini disebut *initial round* yakni melakukan XOR antara *state* awal *plaintext* dengan *chipper key*. Selanjutnya, *array state* ditransformasikan sebanyak Nr secara berulang-ulang pada proses transformasi SubBytes(), ShiftRows(), MixColumns() dan AddRoundKey() setiap putarannya. Proses ini dalam algoritma rijndael disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya di mana pada *round* terakhir, *array state* tidak mengalami transformasi MixColumns().[4]



Gambar 1. Proses Enkripsi Rijndael

B. Steganografi

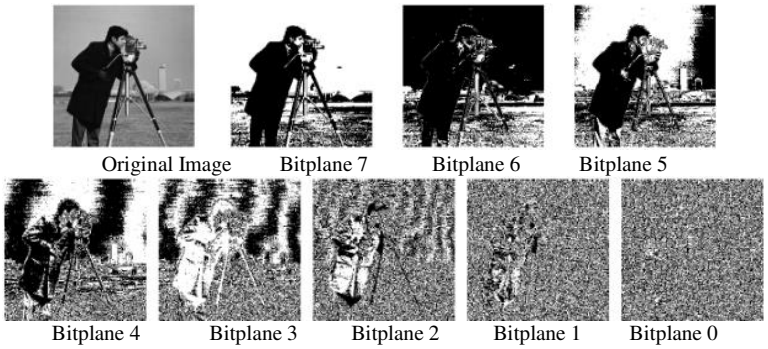
Teknik steganografi diharapkan dapat membantu dalam upaya peningkatan pengamanan pengiriman informasi. Secara teknis steganografi berarti menyembunyikan pesan atau menyisipkan pesan ke media penampung. Pesan rahasia disisipkan ke dalam media penampung sehingga seseorang tidak akan menyadari keberadaan pesan tersebut [9]. Penilaian sebuah algoritma steganografi yang baik dinilai dari beberapa faktor diantaranya; keberadaan pesan rahasia dalam media penampung tidak dapat dipersepsi oleh indera manusia, kualitas atau mutu media penampung tidak berubah banyak akibat penyisipan, jumlah atau kapasitas informasi yang dapat disisipkan dan tahan terhadap berbagai operasi manipulasi media penampung serta pesan yang disembunyikan harus dapat diungkapkan kembali [8],[11]. *Least Significant Bit* (LSB) merupakan salah satu jenis steganografi dengan menyembunyikan pesan pada media digital sebagai penampungnya. Berkas citra bitmap 24 bit, setiap piksel tersusun atas tiga warna merah, hijau dan biru (RGB) yang masing-masing terdiri dari susunan bilangan 8 bit (byte) dari 0 sampai 255 atau format biner 00000000 sampai 11111111 [12]. Misalkan terdapat citra 1024 x 768 berarti memiliki 786.432 piksel setiap piksel panjangnya n-bit. Citra biner 1 bit/piksel, citra grayscale 8 bit/piksel dan citra true color 24 bit/piksel.

True Color (24-bit) Grayscale image (8-bit) Binary image (1-bit)

Gambar 2. Citra Lena 24-bit, 8-bit dan 1-bit

Gambar 2 merupakan citra 24-bit (real image) 1 piksel = 24-bit, terdiri dari komponen RGB (Red, Green, Blue) sehingga setiap piksel berukuran 3 byte (24-bit). Setiap byte bit-bit tersusun dari kiri ke kanan dalam urutan yang kurang berarti (*least significant bit*) hingga bit-bit yang berarti *Most Significant Bit* (MSB). Jika setiap bit ke-i dari MSB ke LSB pada setiap piksel diekstrak dan diplot ke dalam setiap bitplane image maka diperoleh delapan buah citra biner.

Gambar 3 terlihat perbandingan dari delapan buah citra biner. Bitplane LSB yaitu bitplane 0 terlihat seperti citra acak (*random image*). Bitplane LSB merupakan bagian yang redundan pada citra. Artinya perubahan nilai bit pada bagian tersebut tidak mengubah persepsi citra secara keseluruhan. Inilah yang mendasari metode steganografi yang paling sederhana yakni LSB [12].



Gambar 3. Ekstrak Image dari MSB ke LSB dalam 8 Bitplane

C. Media Penampung

Data dapat berupa file text ataupun file dokumen. Semua informasi atau data pada komputer disimpan serta dimanipulasi dalam format biner yaitu 0 dan 1 dan sering disebut dengan bit (binary digit) [4]. BMP adalah representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan di memori komputer yang tidak terkompresi. BMP merupakan citra yang paling mudah untuk dijadikan media dalam teknik steganografi karena citra BMP itu tidak terkompresi.

Sebenarnya banyak jenis citra lain yang bisa dijadikan media untuk steganografi seperti PNG, JPEG dan lain-lain. Namun demikian BMP adalah format yang paling baik dan paling sesuai karena semua file bitmap selain BMP menggunakan fungsi atau algoritma kompresi. Itu berarti tidak semua pixel yang ada bisa digunakan untuk menyimpan data karena akan berakibat file menjadi rusak.

D. PSNR

Peak Signal to Noise Ratio (PSNR) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besaran derau yang berpengaruh pada sinyal tersebut. Rentang nilai PSNR yang baik antara 20dB – 40dB. Nilai PSNR yang lebih tinggi artinya kemiripan lebih erat antara hasil *stego* dengan gambar asli. Rumus yang dapat digunakan:

$$PSNR = 10 \log \left(\frac{MAXi^2}{\sqrt{MSE}} \right) \quad (1)$$

dimana;

$$MSE = \frac{\sum_{y=1}^{m} \sum_{x=1}^{n} [I(x,y) - I'(x,y))]^2}{mn} \quad (2)$$

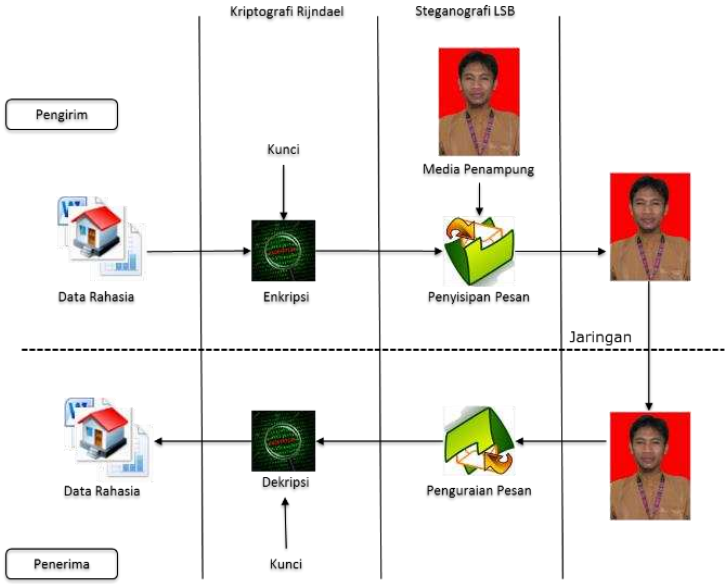
Penjelasan dari rumus:
PSNR : nilai PSNR citra (dalam satuan *decible* citra)
MAXi : nilai maksimum piksel i
MSE : nilai *means square error*
m : pajang citra *stego* dalam piksel
n : lebar citra *stego* dalam piksel
I(x,y) : nilai piksel dari citra cover (asli)
I'(x,y) : nilai piksel dari citra *stego*

III. ANALISIS PERANCANGAN

Secara umum sistem yang dibangun menyembunyikan pesan rahasia atau informasi ke dalam media penampung berupa citra digital, lalu proses penguraian atau pengembalian informasi sehingga pesan yang tersembunyi bisa dibaca kembali seperti aslinya. Teknik penyembunyian secara keseluruhan sistem digambarkan pada gambar 4.

Data rahasia berupa file atau data yang ingin disimpan atau disisipkan baik itu berupa text atau file seperti gambar, music, video, dokumen word, excel dan file lainnya. Sebelum informasi disisipkan ke dalam media penampung maka data dienkripsi terlebih dahulu dengan menggunakan password. Enkripsi merupakan proses penyandian dari pesan asli (*plaintext*) menjadi pesan yang tidak dapat diartikan (*ciphertext*). Enkripsi menggunakan rijndael nantinya sebagai input penyisipan pesan. Proses selanjutnya penyisipan informasi yang berupa *chippertext* ke dalam media penampung yang mana proses ini menggunakan steganografi LSB dengan menghasilkan *stegofile*. Kemudian stegofile tersebut dapat dikirim ke penerima.

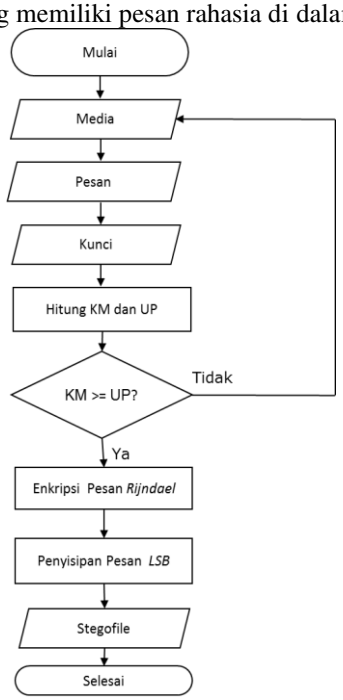
Selanjutnya penerima melakukan penguraian pesan dari media penampung. Data yang didapat dari proses penguraian ini masih berupa data *chippertext*, kemudian didekripsi yakni proses penyusunan kembali, data dirubah dari bentuk *chippertext* menjadi *plaintext* dengan algoritma rijndael. Sehingga orang yang mempunyai kunci tadi bisa membaca pesan yang ada di media penampung tersebut.



Gambar 4. Ekstrak Image dari MSB ke LSB dalam 8 Bitplane

A. *Flowchart Penyisipan Pesan*

Tahap penyisipan adalah tahapan di mana proses penyisipan pesan dilakukan. Proses ini diawali dengan pengisian atau input data dan diakhiri dengan hasil sebuah media citra yang memiliki pesan rahasia di dalamnya.



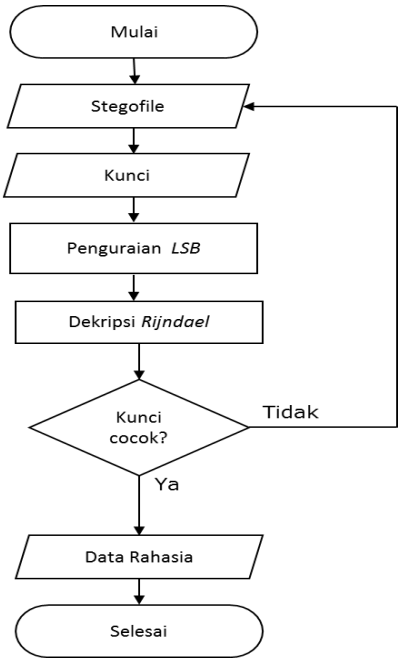
Gambar 5. Flow Penyisipan Pesan

Gambar 5 menjelaskan proses pertama pengirim memilih file citra yang nantinya digunakan sebagai media penampung, mengisi *password* dan menulis atau memilih file pesan/teks yang akan disisipkan. Kemudian sistem melakukan pengecekan kapasitas media (KM) dan ukuran pesan (UD). Jika kapasitas media tidak cukup untuk menyimpan pesan maka sistem kembali ke proses awal yaitu pemilihan media dan jika kapasitas media cukup maka sistem melakukan proses enkripsi dengan algoritma rijndael dan selanjutnya penyisipan menggunakan LSB. Hasil keluaran berupa *stegofile*.

B. *Flowchart Penguraian Pesan*

Tahap penguraian adalah tahapan di mana proses pembacaan atau ekstrak pesan dilakukan. Proses ini diawali dengan pengisian atau input citra yang memiliki pesan dan diakhiri dengan hasil sebuah pesan yang berupa teks atau file.

Gambar 6 menjelaskan pengguna (penerima) memilih dan memasukkan stegofile dan kunci dari pesan yang akan dibaca. Sistem akan melakukan penguraian dengan algoritma LSB dan pendekripsian dengan algoritma rijndael. Sistem mencocokan kunci *stegofile* dan kunci *password* yang dimasukkan oleh pengguna. Apabila kunci sesuai maka sistem akan menampilkan data yang tersembunyi dan sebaliknya bila kunci tidak sesuai maka sistem kembali ke proses awal.



Gambar 6. Flow Penguraian Pesan

C. Proses Perancangan

Proses perancangan merupakan tahapan yang dilakukan dari awal sampai akhir dalam hal pembuatan aplikasi. Tahapan ini untuk memudahkan pembuat aplikasi dalam menentukan langkah-langkah yang harus dilakukan serta memudahkan pengecekan kesalahan algoritma jika terjadi *error/bug*. Adapun tahapan-tahapannya adalah sebagai berikut.

1) Inisialisasi: Proses inisialisasi adalah proses pembacaan data yang diinputkkan oleh pengguna. Inisialisasi dilakukan dalam rangka untuk memastikan tipe data yang diinput sesuai dengan tipe data yang digunakan dalam proses selanjutnya.

2) Pengecekan Data Input: Proses pengecekan data input adalah proses pengecekan data yang ada di variabel. Proses ini mengecek format dari media yang digunakan. Jika media berupa JPEG atau PNG, maka media tersebut akan dilakukan pengubahan formatnya ke dalam media BMP. Selanjutnya pengecekan kapasitas daya tampung media dan juga besarnya pesan, jika daya tampung media cukup untuk menyimpan pesan maka akan dilakukan proses selanjutnya.

3) Proses Enkripsi: Proses enkripsi adalah proses untuk merubah file yang bisa dibaca menjadi file yang tidak bisa dibaca. Enkripsi menggunakan algoritma rijndael. Algoritma rijndael ini membagi-bagi file yang dienkripsi menjadi bagian-bagian atau blok, untuk setiap bloknya merupakan kelipatan 16 byte.

Ori File dalam byte			Enkripsi File dalam byte	
Dari	Ke	Enkripsi	Dari	Ke
0	15	-->	0	15
16	31	-->	16	31
32	47	-->	32	47
48	63	-->	48	63
64	79	-->	64	79
80	95	-->	80	95
96	111	-->	96	111
112	127	-->	112	127

Gambar 7. Enkripsi Data Blok 16 Bytes

Gambar 7 mejelaskan sistem akan melakukan enkripsi data secara berulang dari 16 bytes pertama hingga bytes terakhir. Untuk mengetahui jumlah perulangan dalam melakukan enkripsi perlu menggunakan rumus berikut.

$$JPn = \frac{\text{UP}}{16} \quad (3)$$

Penjelasan dari rumus:
JPn : jumlah pengulangan
UP : ukuran pesan dalam ukuran byte
16 : ukuran 16 bytes blok dalam enkripsi.

4) Proses Penyisipan Pesan: Pesan atau data yang telah dienkripsi diplotkan per bit, kemudian disisipkan pada media secara berurutan di bagian bit terakhirnya. 1 byte sama dengan 8 bit, ini berarti jika ada data pesan dengan ukuran 10 bytes itu memerlukan media 80 bytes. Ada beberapa yang perlu diperhitungkan yaitu jumlah byte yang diperlukan untuk menyimpan file ekstensi, kunci, ukuran pesan (UP) dan penutup pesan.

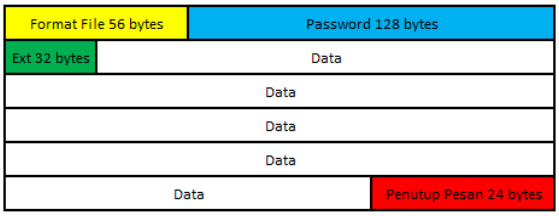
TABEL I
TABEL ANALISA KAPASITAS MEDIA

Byte			Keterangan
Dari	Ke	Jumlah	
0	56	56	Format File
57	184	128	Password 16 *bytes*
185	216	32	Ext File 4 *bytes*
217	UP	UP	Data
UP + 216	UP + 240	24	Penutup Pesan "$$$"

Tabel I memberikan pengetahuan bahwa kapasitas media adalah ukuran media dikurangi 240, dengan cara penyisipan pesan dilakukan secara terurut. Sebagai contoh jika sebuah file media memiliki ukuran sebesar 1250 bytes, maka media tersebut mampu menyimpan data rahasia paling banyak 1010 bytes. Adapun kapasitas media yang bisa digunakan untuk menyimpan pesan dapat digunakan rumus berikut.

$$KM = UM - 240 \quad (4)$$

Penjelasan dari rumus tersebut:
KM : kapasitas media dalam ukuran bit
UM : ukuran media dalam ukuran bytes
240 : total bytes yang diperlukan untuk menyimpan format file, ext pesan, kunci dan penutup pesan.



Gambar 8. Ilustrasi Penyisipan Pesan

Gambar 8 menjelaskan ilustrasi penyisipan pesan. Warna kuning merupakan tempat yang digunakan untuk menyimpan file format media sebesar 56 bytes, warna biru digunakan untuk menyimpan *password* data sebesar 128 bytes, warna hijau digunakan untuk menyimpan ekstensi file sebesar 32 bytes sedangkan warna putih merupakan tempat yang digunakan untuk menyimpan data atau pesan. Sedangkan di akhir data akan ditutup dengan tanda penutup pesan sebagaimana ditunjukkan pada warna merah sebesar 24 bytes.

5) Proses Pembacaan Pesan: Pembacaan nilai bit terakhir dari media secara berurutan. Bit-bit tersebut akan disusun ulang kembali menjadi file atau pesan yang masih terenkripsi.

6) Proses Dekripsi: Proses dekripsi adalah kebalikan dari pada proses enkripsi. Pesan yang didapat dari proses pembacaan pesan akan dibagi-bagi per 16 byte yang kemudian akan dilakukan dekripsi. Hasil dari dekripsi ini disusun ulang menjadi file yang semula sebelum dilakukan enkripsi, sehingga pesan yang disisipkan bias dibaca oleh penerima.

7) Penampilan Informasi: Tahap ini adalah tahap untuk memberikan informasi dari pada proses yang telah dilakukan pada tahapan diatas. Informasi yang ditampilakan berupa laporan penyisipan dan laporan penguraian. Laporan penyisipan pesan diantaranya status enkripsi, path file, kapasitas stegofile, perubahan kualitas media, status steganografi. Laporan penguraian pesan diantaranya status enkripsi, path dan ukuran pesan.

IV. IMPLEMENTASI DAN PENGUJIAN

A. *Implementasi Sistem*

1) Penyisipan Pesan: bagian ini merupakan menu untuk membuat pesan penyembunyian digital ke dalam sebuah file media gambar. Pengguna akan diminta untuk memasukkan tipe data atau pesan, file media penampung, folder hasil dan *password*, kemudian sistem akan melakukan verifikasi data input yang telah dimasukkan.

Gambar 9. Penyisipan Pesan

Gambar 9 merupakan halaman penyisipan pesan, setelah tombol OK ditekan, maka sistem melakukan pengecekan kapasitas media. Bila kapasitas media sesuai maka sistem memproses penyembunyian bit-bit dari file yang sudah dienkripsi ke dalam file cover atau media. Selanjutnya sistem memberikan informasi laporan hasil penyembunyian data seperti yang ditunjukkan pada gambar 10.

Gambar 10. Laporan Hasil Penyembunyian Pesan

2) *Penguraian Pesan:* Setelah file diterima, pengguna mengurai pesan untuk mengetahui pesan tersembunyi. Prosesnya kebalikan dari pembuatan pesan. Sistem mengambil bit terakhir dari setiap piksel yang ada pada media dengan melakukan pencocokan password. Jika password sesuai maka proses dekripsi dilakukan dengan menyusun ulang kembali potongan bit tersebut menjadi file seperti yang semula, lalu sistem memberikan informasi pesan gambar 11.

Gambar 11. Laporan Hasil Penguraian Pesan

B. *Pengujian Sistem*

 e-ISSN : 2443-2229

Pengujian ini dilakukan untuk menguji kelayakan dari aplikasi. Apakah aplikasi berjalan dengan benar sesuai dengan permintaan. Beberapa pengujian yang dilakukan diantaranya, pengujian enkripsi, penyisipan dengan perbandingan histogram kualitas citra dan pengujian PSNR.

1) *Pengujian Enkripsi:* Pengujian enkripsi untuk mengetahui keberadaan pesan rahasia tidak dapat dimengerti dan dipahami. Artinya dengan algoritma rijndael dapat meningkatkan keamanan pesan pada steganografi.

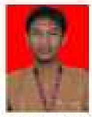
TABEL II
PENGUJIAN ENKRIPSI PESAN

Jenis Data	Sebelum Enkripsi	Sesudah Enkripsi
Home.ico	 Ukuran file : 19.2 KB	Ukuran file : 19.2 KB
Pesan.txt	File Edit Format View Help Ini adalah Pesan Rahasia jangan dibaca Ukuran file : 1 KB	File Edit Format View Help 0ZZAs;7*UO•à|$ 'CIè|O>œA÷¤ í"yo Dl«Q1‹(ó¤qBéi!¾ Ukuran file: 1 KB

Tabel II hasil enkripsi pesan menunjukkan bahwa sebuah data sebelum dan sesudah dienkripsi memiliki ukuran yang sama, karena pada saat enkripsi hanya melakukan pengubahan nilai masing-masing bytes dengan menggunakan algoritma rijndael sehingga data yang dihasilkan tidak bisa dibaca atau dipahami lagi. Uji coba ini dapat diketahui bahwa data yang sudah di enkripsi tidak bisa lagi dibaca oleh orang lain kecuali memiliki *password* untuk mendekripsi ulang. Hal ini memberikan pengertian bahwa dengan menggunakan enkripsi data akan lebih aman.

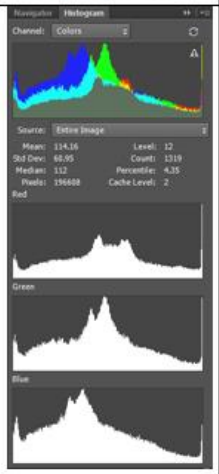
2) *Pengujian Penyisipan:* Pengujian penyisipan untuk mengetahui kualitas atau mutu media penampung tidak berubah banyak akibat penyisipan. Pengujian penyisipan menggunakan histogram analisis dan *Peak Signal to Noise Ratio* (PSNR).

TABEL III
PENGUJIAN PENYISIPAN PESAN

Media Penampung	File Pesan	StegoText
Fotoku.bmp  Ukuran 489 KB	Pesan.txt File Edit Format View Help Ini adalah Pesan Rahasia jangan dibaca Ukuran file : 1 KB	Fotoku + Pesan.bmp  Ukuran file : 489 KB 320 B yang berubah
Bunga.bmp  Ukuran file: 2.250 KB	Home.png  Ukuran file : 19.2 KB	Bunga + Home.bmp  Ukuran file: 2.250 KB 82 KB yang berubah

Tabel III hasil penyisipan pesan menunjukkan bahwa sebuah media citra sebelum dan sesudah disisipkan memiliki ukuran yang sama. Hal ini karena pada saat penyisipan hanya merubah nilai masing-masing bytes dengan tidak menambah atau mengurangi jumlah piksel. Adapun citra yang dihasilkan setelah proses penyisipan mengalami perubahan kualitas yang mana besarnya perubahan kualitas citra tergantung dari nilai bit dari pada bit media dan bit pesan. Semakin banyak perbedaan nilai bit maka semakin besar pula perubahannya. Perubahan yang terjadi pada citra tidak akan terlihat oleh mata karena hanya kurang lebih 1 byte saja. Namun demikian untuk kualitas citra sebelum dan sesudah proses steganografi dapat dianalisa dengan menggunakan histogram analisis pada photoshop. Pengujian ini dilakukan untuk membuktikan bahwa proses staganografi memberikan dampak atau perubahan pada kualitas citra yang sangat kecil. Hasil analisa histogram dengan bantuan photoshop pada salah satu media penampung sebagai berikut.

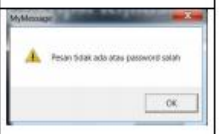
TABEL IV
PERBANDINGAN HISTOGRAM KUALITAS CITRA

	Sebelum disisipi	Sesudah disisipi	Keterangan
Media			Secara inderawi media sebelum dan sesudah tidak ada perbedaan
Histogram Analysis			Dengan menggunakan histogram analysis yang ada pada photo shop, media sebelum dan sesudah disisipi tidak menunjukkan adanya perbedaan, baik itu nilai dari *mean, standard deviation, median* dan juga jumlah *pixel*nya.

Tabel IV terlihat pada media penampung kualitas citra tidak memberikan dampak yang besar bagi media yang disisipkan sehingga tidak akan menghilangkan rasa curiga bagi orang lain bahwa di dalam media tersebut terdapat pesan yang disembunyikan. Selain itu dilakukan penghitungan PSNR terhadap lima citra uji dengan variasi tabel berikut.

Selain itu dalam penguraian pesan dilakukan pengujian normal dan abnormal. Pengujian normal terjadi bila pengguna melakukan input data secara benar seperti password, stegofile belum dimanipulasi baik *rotate image*, edit, atau kompresi. Hasilnya bahwa setiap nilai dari pada stegofile sangat sensitive dari perubahan data yang mengakibatkan data tidak dapat dibaca. Maksimal media yang bisa disisipkan pesan adalah 107 MB.

TABEL V
PENGUJIAN PSNR PADA CITRA

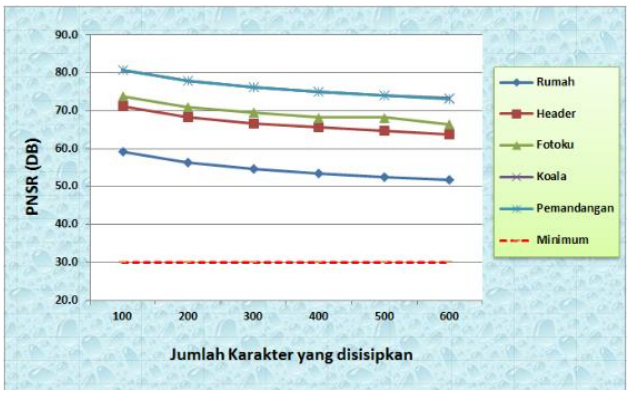
Stegofile	Problem	Pesan Error
Koala.bmp 	Password salah	
Penguin.bmp 	Stegofile tidak memiliki pesan	
Gunung.bmp 	Stegofile telah di*rotate* atau di edit dari horizontal menjadi vertikal	
Koala3.bmp 	Ukuran media yang akan disisipi lebih dari lebih dari 107 MB	

Tabel V menunjukkan pesan kesalahan bila password, stegofile dan ukuran tidak sesuai. Selanjutnya pengujian citra menggunakan ukuran pesan yang disisipkan secara bervariasi serta jumlah karakter yang berbeda untuk tiap pengujiannya yakni 100, 200, 300, 400, 500 dan 600 karakter.

TABEL VI
PENGUJIAN PSNR PADA CITRA

File citra (.BMP)	Password	Ukuran Citra (KB)	Jumlah Karakter yang Disisipkan					
			100	200	300	400	500	600
Rumah	salma	65	59.2	56.4	54.7	53.5	52.6	51.8
Header	harits	272	71.1	68.2	66.7	65.6	64.6	63.8
Fotoku	welcome	490	73.8	71.1	69.4	68.2	68.2	66.4
Koala	newyear	2305	80.8	77.9	76.2	75.0	74.0	73.2
Pemandangan	passwordku	2305	80.7	77.9	76.1	74.9	74.0	73.2

Tabel VI menunjukkan hasil yang diperoleh nilai decibel (db) masing-masing berbeda untuk setiap jumlah karakter yang disisipkan, hal ini menunjukkan kemampuan steganografi terhadap berbagai macam ukuran. Adapun untuk memudahkan dalam menganalisa nilai PSNR citra yang diuji ditunjukkan pada grafik berikut.



Gambar 12. Grafik Pengujian PSNR Aplikasi

Gambar 12 pada grafik tersebut menunjukkan bahwa semakin banyak jumlah karakter yang disisipkan maka kualitas citra akan semakin berkurang, semakin kecil media penampung yang disisipkan maka kualitas citra akan semakin berkurang.

V. KESIMPULAN

Kesimpulan yang dapat diambil dari pembahasan ini adalah sebagai berikut.

1) Kombinasi algoritma rijndael dan least significant bit mampu menyembunyikan data atau pesan dengan tidak merubah ukuran dari media yang disisipkan karena metode ini tidak menambahkan jumlah piksel tetapi hanya merubah nilai dari pada piksel itu sendiri.

2) Metode *least significant bit* pada media memberikan sedikit perubahan kualitas citra. Namun demikian perubahan warna piksel yang dihasilkan akibat penyisipan bit-bit pesan tersebut tidak dapat dideteksi oleh mata manusia karena perubahannya sangat kecil.

3) Penelitian ini menggunakan media paling besar 107 MB dengan maksimum pesan 13,4 MB. Semakin banyak jumlah karakter yang disisipkan maka kualitas citra akan semakin berkurang, semakin kecil media penampung yang disisipkan maka kualitas citra akan semakin berkurang. Dari pengujian PSNR, semua media yang disisipkan pesan menggunakan aplikasi ini memiliki kualitas citra yang bagus yaitu diatas 30 db.

DAFTAR PUSTAKA

[1] A. Cheddad, J. Condell, K. Curran, P.Mc Kevitt, Digital Image Steganography: Survey and Analysis of Current Methods. Elsevier. Northern Ireland, UK, Signal Processing 90, 2010, pp.727-752.
[2] C.T.E. Yuliana. 2014 Implementasi Algoritma Kriptografi Blowfish dan Metode Steganografi End of File (EOF) untuk Keamanan Data. Jurnal ePrint Udinus.
[3] D. Ariyus, Kriptografi: Keamanan Data dan Komunikasi, Yogyakarta, Graha Ilmu, 2006.
[4] R. Munir. 2004. Bahan Kuliah Ke-13 Kriptografi Advance Encryption Standar (AES). Departemen Teknik Informatika, ITB.
[5] Kristoforus. Implementasi Algoritma Rijndael untuk Enkripsi dan Dekripsi pada Citra Digital, Jurnal UII, 2012.
[6] M. Nosrati, R. Karimi, M. Hariri, An Introduction To Steganography Methods, World Applied Programming, vol 1, No. 3, August 2011.
[7] Norma G. Kasus Penipuan Dominasi Kejahatan Cyber, Harian Kompas 15 April 2013.
[8] P. Alatas, Implementasi Teknik Steganografi dengan Metode LSB pada Citra Digital, Library Univ. Gunadarma, 2009.
[9] R. Firmansyah. Implementasi Kriptografi dan Steganografi pada Media Gambar dengan Menggunakan Metode DES dan Region Embed Data Density, Digilib ITS, 2011.
[10] R.O. Pradana, Analisis Perbandingan Algoritma Rijndael dan Algoritma Twofish pada Proses Pengiriman Data Teks Menggunakan Jaringan LAN (Local Are Network), Skripsi UNIKOM Bandung, 2011.
[11] A. Prabowo, A. Hidayatno, Y. Christiyono. 2011. Penyembunyian Data Rahasia pada Citra Digital Berbasis Chaos dan Discrete Cosine Transform. Jurnal Transmisi. Vol 13 No.2. Hal.46-52. 2011
[12] R. Munir.2015. Bahan Kuliah Kriptografi; Steganografi. Program Studi Informatika. STEI-ITB.
[13] A. Prihanto, Peningkatan Kapasitas Informasi Tersembunyi pada Image Steganografi Menggunakan Teknik Hybrid, Digilib ITS, 2010.z