IMPLEMENTASI DAN ANALISIS KEAMANAN PESAN MENGGUNAKAN TEKNIK STEGANOGRAFI LSB DAN ALGORITMA KRIPTOGRAFI *RELATIVE DISPLACEMENT CIPHER*

IMPLEMENTATION AND ANALYSIS OF MESSAGE SECURITY USING LSB STEGANOGRAPHY AND RELATIVE DISPLACEMENT CIPHER CRYPTOGRAPHY ALGORITHM

Muhammad Tezar[1], Rita Magdalena[2], Nur Andini[3]

[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[1]tezar.muhammad@gmail.com, [2]ritamagdalena@telkomuniversity.ac.id, [3]nurandini@telkomuniversity.ac.id

Abstrak

Saat ini teknik steganografi telah banyak digunakan untuk menyimpan informasi rahasia pada media digital. Salah satu metode yang digunakan adalah Least Significant Bit (LSB). Namun steganografi yang menggunakan metode LSB masih sangat sederhana dan sudah terlalu umum digunakan untuk menyisipkan berkas teks pada media digital sehingga masih memungkinkan pihak ketiga untuk membuka informasi yang dirahasiakan. Untuk itu dibutuhkan teknik kriptografi untuk mengacak plaintext sebelum disisipkan ke dalam gambar RGB dengan teknik steganografi. Algoritma kriptografi yang digunakan adalah *Relative Displacement Cipher* (RDC) dan teknik steganografi yang digunakan adalah *Least Significant Bit* (LSB). Hasil yang diperoleh dari tugas akhir ini adalah sebuah citra yang memiliki pesan terenkripsi pada bit terakhir *pixel* penyusunnya. Dengan PSNR bernilai 45.2209dB setelah disisipkan panjang pesan maksimum yang mampu ditampung *cover image*. Namun ketika diberi *noise*, bit pesan yang tersimpan di dalam *cover image* akan berubah sehingga informasi yang tersimpan akan terganggu. Dengan demikian, sistem ini dapat berjalan dengan baik jika *stego image* tidak diberi serangan.

Kata kunci : Steganografi, Kriptografi, LSB, *Relative Displacement*

Abstract

Currently steganography has been widely used to store confidential information on digital media. One methods used is Least Significant Bit (LSB). However LSB steganography method is very simple and commonly used to embed text in digital media files that allows third parties to disclose the information. Therefore cryptography is used to encrypt the plaintext before it is embedded into the RGB image. Cryptographic algorithms that used in this project is Relative Displacement Cipher (RDC) and steganographic technique used is Least Significant Bit (LSB). The results of this final project is an image containing the encrypted text on the last bit pixels constituent. With 45.2209dB PSNR after it inserted the maximum message length that can be accommodated by the cover image. But when given the noise, the information bits that stored in the cover image will be changed so that the stored message will be disrupted. It can be concluded that this system will work well if the stego image is not given the attack.

Keywords: Steganography, Criptography, LSB, Relative Displacement Cipher

1. Pendahuluan

Sebagai makhluk sosial, komunikasi merupakan bagian penting pada kehidupan manusia. Komunikasi telah menjadi sarana untuk berinteraksi dan bertukar informasi. Namun ada kalanya seseorang tidak ingin informasi yang disampaikannya diketahui oleh pihak yang tidak diinginkan. Maka dari itu dibutuhkan suatu metode komunikasi yang dirancang sedemikian rupa agar informasi yang disampaikan menjadi lebih aman, yaitu dengan menggunakan teknik steganografi. Terdapat beberapa metode pada steganografi, salah satunya adalah metode LSB. Teknik ini memanfaatkan bit terendah pada pixel penyusun suatu media untuk menyisipkan informasi. Saat ini teknik steganografi LSB telah umum digunakan dan teknik pengungkapan informasinya telah

diketahui khalayak banyak sehingga dianggap kurang efektif sebagai metode komunikasi yang aman. Namun kita dapat meningkatkan keamanan teknik ini dengan menambahkan algoritma kriptografi. Kriptografi sendiri merupakan suatu teknik yang dapat mengubah atau mengacak data asli menjadi bentuk yang sulit untuk diketahui lagi maknanya. Banyak algoritma kriptografi yang dapat digunakan dalam penyandian informasi, salah satunya adalah kriptografi RDC. Algoritma kriptografi ini merupakan inovasi baru pada teknik kriptografi karena teknik enkripsi yang digunakan berbeda dari teknik kriptografi lainnya. Dengan penambahan algoritma kriptografi RDC pada teknik steganografi LSB, penyerang butuh mendekripsikan informasi tersebut untuk mendapat informasi asli yang terkandung pada media penampung. Berdasarkan latar belakang tersebut, maka dilakukan penelitian dengan memanfaatkan teknik steganografi LSB dan algoritma kriptografi RDC dalam penyisipan pesan rahasia pada media gambar. Pesan berupa berkas teks yang kemudian disisipkan pada sebuah media gambar RGB.

2. Dasar Teori

A. Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani *cryptós* (*secret*) dan *gráphein* (*writing*). Jadi, kriptografi berarti *secret writing* (tulisan rahasia). Kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya.

Ada empat tujuan dari kriptografi ini yang juga merupakan aspek keamanan informasi yaitu:

1. *Privacy/Confidentiality*
2. *Integrity*
3. *Authentication*
4. *Non-repudiation*

B. Algoritma Relative Displacement Cipher (RDC)

RDC merupakan algoritma baru untuk metode enkripsi dan dekripsi dengan kunci simetris. Pada kriptografi simetris, proses enkripsi dan dekripsi dilakukan dengan bantuan kunci yang sama, dengan demikian harus diketahui atau dikirim ke kedua sisi baik pengirim maupun penerima sebelum proses enkripsi dan dekripsi dilakukan. Teknik inovatif ini berdasar pada pembagian masukan *string* menjadi matriks orde genap, penggunaan *magic square matrix*, dan pola rotasi. Hal-hal inilah yang membuat hasil enkripsinya lebih aman karena selain mengatasi masalah pengulangan karakter, algoritma ini juga mengurangi kemungkinan serangan *brute-force*. Beberapa istilah yang digunakan pada algoritma RDC adalah sebagai berikut:

1. *Magic Square Matrix*, matriks persegi yang memiliki jumlah tiap elemen di setiap kolom dan baris adalah sama.
2. *Pattern Rotation*, serangkaian langkah-langkah untuk menggeser posisi tiap elemen dengan aturan yang telah ditentukan sebelumnya.
3. Matriks Persegi Orde Genap, matriks dengan jumlah baris dan kolomnya sama serta berupa angka genap yang bisa dibagi dua. Contoh matriks dengan orde 2x2, 4x4, 6x6, dst.

C. Steganografi

Steganografi (*Steganography*) berasal dari bahasa Yunani *steganos* (*hidden*) dan *gráphein* (*writing*). Jadi, steganografi berarti *hidden writing* (tulisan tersembunyi). Steganografi adalah seni dan ilmu menyembunyikan pesan ke dalam sebuah media dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa sebenarnya ada suatu pesan rahasia.

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia.

Aspek terpenting pada steganografi adalah tingkat keamanan penyembunyian informasinya yang mengacu pada seberapa besar ketidakmampuan pihak ketiga dalam mendeteksi keberadaan informasi yang tersembunyi.

D. *Least Significant Bit* (LSB)

LSB adalah bagian dari barisan data biner (basis dua) yang mempunyai nilai paling tidak berarti/paling kecil. Letaknya adalah paling kanan dari barisan bit. Sedangkan most significant bit adalah sebaliknya, yaitu angka yang paling berarti/paling besar dan letaknya disebelah paling kiri. Contohnya adalah bilangan biner dari 255 adalah 11111111

Dari barisan angka 1 tersebut, angka 1 paling kanan bernilai 1, dan itu adalah yang paling kecil. Bagian tersebut disebut dengan *least significant bit* (bit yang paling tidak berarti), sedangkan bagian paling kiri bernilai 128 dan disebut dengan *most significant bit* (bit yang paling berarti). *Least significant bit* sering kali digunakan untuk kepentingan penyisipan data ke dalam suatu media digital.

Metode penyisipan LSB (*least significant bit*) ini adalah menyisipi pesan dengan cara mengganti bit ke-8, 16 dan 24 pada representasi biner *file* gambar dengan representasi biner pesan rahasia yang akan disembunyikan. Dengan demikian pada setiap *pixel* file gambar BMP 24 bit dapat disisipkan 3 bit pesan, misalnya terdapat data *raster original* file gambar adalah sebagai berikut:

00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101001

Sedangkan representasi biner huruf A adalah 01000001, dengan menyisipkannya ke dalam *pixel* di atas maka akan dihasilkan:

00100110 11101001 11001000
00100110 11001000 11101000
11001000 00100111 11101001

E. Parameter Pengujian

a. *Fidelity*

Parameter pengujian *fidelity* bermaksud untuk mengetahui seberapa besar perubahan *stego image* dibandingkan dengan *cover image*. Akan digunakan dua cara untuk melakukan pengujian dari aspek *fidelity*. Pertama, pengujian secara subjektif dengan menggunakan *Mean Opinion Score* (MOS). Kedua, pengujian secara objektif dengan melihat *Peak Signal to Noise Ratio* (PSNR).

i. Pengujian MOS

Mean Opinion Score (MOS) merupakan suatu metode pengujian dengan mengukur kualitas *stego image* berdasarkan deskripsi kualitatif dari apa yang kita lihat. MOS memberikan indikasi numerik kualitas gambar yang didapatkan setelah disisipkan pesan. Terminologi MOS merupakan hasil rekomendasi dari International Telecommunication Union (ITU-T P.800).

Nilai MOS yang diperoleh tidak harus bilangan bulat. Teknisnya beberapa orang diminta untuk membandingkan *stego image* dengan *cover image* dengan rentang 1 sampai 5. Nilai 1 menyatakan nilai terburuk dan 5 menyatakan nilai yang terbaik. Kemudian dihitung rata-rata dari penilaian seluruh responden sehingga didapatkan nilai *Mean Opinion Score* (MOS) dari media gambar tersebut. Tabel penilaian parameter MOS ditunjukan pada Gambar 3.6.

Nilai	Kualitas Gambar	Keterangan
5	*Excellent*	Tidak terlihat derau sama sekali
4	*Good*	Derau terlihat, namun tidak merusak kualitas gambar
3	*Fair*	Derau sedikit merusak kualitas gambar
2	*Poor*	Derau merusak kualitas gambar
1	*Bad*	Derau sangat merusak kualitas gambar

Tabel 1 Tabel parameter penilaian MOS

$$= \sum_{=1} \frac{0 \quad \ell \; -}{} \qquad (1)$$

ii. Pengujian PSNR

Peak Signal to Noise Ratio (PSNR) digunakan untuk mengetahui perbandingan kualitas gambar sebelum dan sesudah disisipkan pesan. PSNR biasanya diukur dalam satuan desibel (dB). Untuk menentukan PSNR, terlebih dahulu ditentukan nilai *Mean Square Error* (MSE). Perhitungan MSE dapat dirumuskan sebagai berikut:

$$= \frac{1}{} \sum \sum \; (\quad - K \;)^2 \qquad (2)$$

dimana:

MSE = Nilai *Mean Square Error* dari gambar
m = panjang gambar (dalam *pixel*)
n = lebar gambar (dalam *pixel*)
i,j = koordinat *pixel*
I = nilai bit gambar pada koordinat *i,j*
K = nilai derajat keabuan citra pada koordinat *i,j*

Setelah menentukan MSE, kita dapat menentukan nilai PSNR dengan memasukkan nilai MSE pada rumus berikut:

$$= 20. \left(\frac{255}{\sqrt{\quad}} \right) \qquad (3)$$

dimana:

PSNR = nilai PSNR gambar (dalam dB)
MSE = nilai *MSE*

Nilai PSNR yang wajar memiiliki nilai minimal 30dB. Jika hasil yang didapat berada dibawah 30dB maka *noise* gambar akan terlihat oleh pandangan manusia normal. Berdasarkan hal tersebut, hasil uji dianggap baik jika memiliki nilai PSNR ≥ 30dB.

b. *Recovery*

Parameter pengujian *recovery* bermaksud untuk mengetahui apakah pesan yang disisipkan pada gambar dapat diungkapkan kembali. Pada tahap ini akan dilakukan beberapa kali pengujian untuk memastikan bahwa pesan rahasia yang disisipkan pada gambar dapat diambil kembali dalam keadaan utuh. Indikator keberhasilan pengujian ini adalah jika panjang teks pesan asli sama dengan panjang pesan teks hasil *extracting*. Setelah dilakukan pengujian tersebut, dilakukan penghitungan persentase keberhasilan pada proses pengujian *recovery* dengan menggunakan rumus berikut:

$$\% \; K = \frac{} {} x100\% \qquad (4)$$

c. *Robustness*

Parameter pengujian *robustness* bermaksud untuk mengetahui apakah pesan rahasia yang tersimpan pada gambar tahan terhadap berbagai manipulasi sinyal yang dapat dilakukan terhadap *stego image*. Pengujian dilakukan dengan dengan *Gaussian noise* dan *Salt & Pepper noise* serta serangan geometris pada citra seperti *rescaling* dan *cropping*. Kemudian melakukan ekstraksi *stego image* untuk melihat apakah pesan masih bisa diungkap atau tidak. Indikator keberhasilan pengujian adalah jika pesan berhasil diungkap secara utuh dari *stego image*.

d. *Security*

Parameter pengujian *security* lebih ditujukan untuk menguji ketahanan algoritma RDC terhadap kemungkinan serangan oleh pihak ketiga. Salah satu cara untuk menguji ketahanan kriptografi adalah dengan mengukur nilai *avalanche effect* kriptografi tersebut. *Avalanche effect* merupakan perubahan kecil pada *plaintext*

maupun *key* yang menyebabkan perubahan terhadap *ciphertext* yang dihasilkan. Umumnya bit pada *ciphertext* mengalami perubahan dari jumlah bit *plaintext* sebesar 50%. Suatu *avalanche effect* dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45-60% (sekitar separuhnya). Semakin banyak perubahan yang terjadi mengakibatkan kriptanalis akan semakin sulit untuk menemukan *plaintext* atau *key*. Nilai *avalanche effect* dirumuskan dengan:

$$\text{Avalanche Effect} = \frac{\sum \text{Bit Berubah}}{\sum \text{Bit Keseluruhan}} \times 100\% \qquad (5)$$

3. Perancangan Sistem

Algoritma *Relative Displacement Cipher* (RDC) digunakan untuk melakukan proses enkripsi dan dekripsi pesan. Pada sisi pengirim pesan teks dienkripsi untuk mencari *ciphertext* kemudian disisipkan ke dalam berkas gambar RGB berformat BMP dengan metode *Least Significant Bit* (LSB). Pada sisi penerima *stego image* tersebut diekstrak dan hasilnya didekripsikan untuk mendapatkan *plaintext*.

Gambar 1 Blok Diagram Sistem

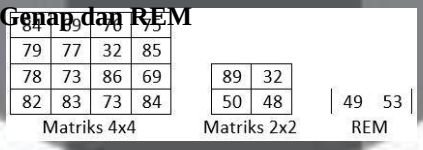
A. *Encryption*

Gambar 2 Diagram Alir Proses Enkripsi Algoritma RDC

a. Konversi ke Bentuk Nilai ASCII

Misalkan *plaintext*: TELKOM UNIVERSITY 2015
ASCII: [84 69 76 75 79 77 32 85 78 73 86 69 82 83 73 84 89 32 50 48 49 53]

b. Atur *String* pada Matriks Orde Genap dan REM



Gambar 3 Mengatur String menjadi Matriks dan REM sebagai Sisa

c. Menentukan Kunci Simetris KEY

Untuk mencari KEY kita harus menentukan KEY1 dan KEY2 terlebih dahulu.
KEY1 = jumlah kolom dari seluruh matriks yang dihasilkan dan jumlah elemen pada REM
KEY2 = penjumlahan *Magic Square Matrix* untuk matriks yang terakhir dihasilkan
Catatan: Jika KEY > 9 lakukan pnjumlahan seluruh digit hingga mendapatkan satu buah digit KEY
KEY1 = 4 + 2 + 2 = 8
Ukuran matriks terakhir = 2x2, hasil penjumlahan *magic square matrix* = 5

Diperoleh KEY2 = 5
KEY = 8 + 5 = 13 = 1 + 3 = 4

d. Rotasi Matriks dengan Pola Rotasi Enkripsi

Urutan langkah-langkah pada pola rotasi ini dibuat sebagai berikut, asumsikan matriks yang dibentuk adalah sebagai berikut:

84	69	76	75
79	77	32	85
78	73	86	69
82	83	73	84

89	32
50	48

Gambar 1 Matriks Awal

(i) Pola rotasi pertama yaitu melakukan pertukaran kolom genap.
(ii) Pola rotasi kedua yaitu melakukan pertukaran baris genap.
(iii) Pola rotasi ketiga yaitu menggeser keatas kiri dengan arah diagonal.
(iv) Pola rotasi keempat yaitu menggeser keatas kanan dengan arah diagonal.
(v) Pola rotasi kelima yaitu menggeser keatas sebanyak satu kali untuk setiap kolom genap.
(vi) Pola rotasi keenam dan yang terakhir yaitu merotasi lingkaran terluar searah jarum jam kemudian merotasi lingkaran didalamnya berlawanan arah dengan jarum jam. Begitupun selanjutnya.

Hasil yang didapatkan melalui pola rotasi ini adalah seperti yang ditunjukkan pada matriks dibawah ini:

82	84	86	76
78	69	77	83
69	79	85	73
75	32	73	84

32	48
50	89

Gambar 5 Hasil Matriks setelah Proses Pola Rotasi

e. Menjumlahkan Matriks dengan *Magic Square*

Hasil matriks dijumlahkan dengan *magic square matrix* berdimensi sama dengan matriks masukan.

$$\begin{vmatrix} 82 & 84 & 86 & 76 \\ 78 & 69 & 77 & 83 \\ 69 & 79 & 85 & 73 \\ 75 & 32 & 73 & 84 \end{vmatrix} + \begin{vmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{vmatrix} = \begin{vmatrix} 98 & 86 & 89 & 89 \\ 83 & 80 & 87 & 91 \\ 78 & 86 & 91 & 85 \\ 79 & 46 & 88 & 85 \end{vmatrix}$$

$$\begin{vmatrix} 32 & 48 \\ 50 & 89 \end{vmatrix} + \begin{vmatrix} 1 & 3 \\ 4 & 2 \end{vmatrix} = \begin{vmatrix} 33 & 51 \\ 54 & 91 \end{vmatrix}$$

Gambar 6 Penjumlahan Matriks Masukan dengan *Magic Square Matrix*

f. Merubah Nilai REM berdasarkan KEY

Hitung variable *F* sebagai penjumlahan *magic square matrix* untuk orde KEY. Jumlahkan $F*(n + 1)$ untuk setiap n^{th} pada REM, untuk menghasilkan REM yang baru.
REM = [49 53]
KEY = 4, penjumlahan *magic square matrix order* 4 = 34
Dengan demikian *F* = 34
REM baru = [49 + (34 * (1 + 1)) 53 + (34 * (2 + 1))] = [117 155]

g. Menyusun Matriks dan REM Baru

Atur seluruh elemen pada matriks persegi dan sisanya (REM) dengan urutan saat mereka disusun.

98	86	89	89
83	80	87	91
78	86	91	85
79	46	88	85

33	51
54	91

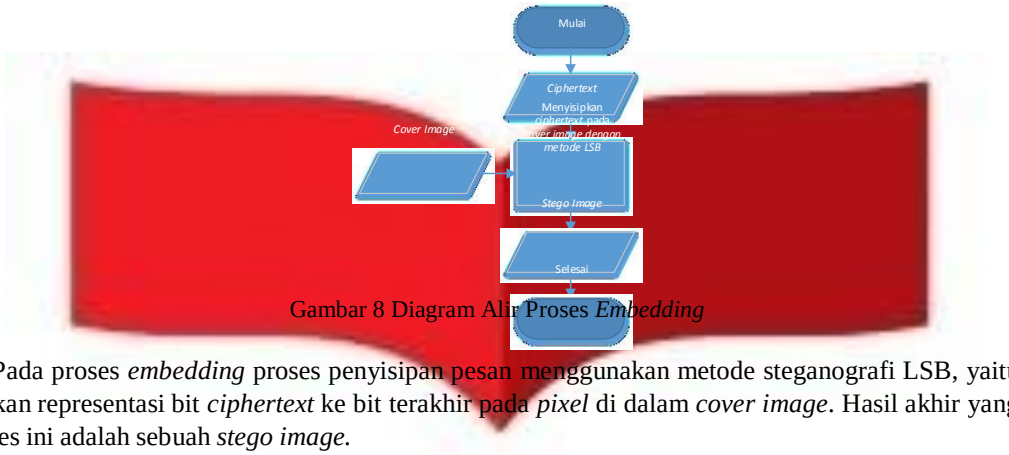
[117 155]

Gambar 7 Hasil Enkripsi Matriks

h. Konversi ke Bentuk Karakter ASCII

Hasil enkripsi yang berupa ASCII disusun dan diubah ke dalam bentuk karakter.
ASCII: [98 86 89 89 83 80 87 91 78 86 91 85 79 46 88 85 33 51 54 91 117 155]
Ciphertext: **bVYYSPW[NV[UO.XU!36[u›**

B. Embedding



Gambar 8 Diagram Alir Proses *Embedding*

Pada proses *embedding* proses penyisipan pesan menggunakan metode steganografi LSB, yaitu dengan menyisipkan representasi bit *ciphertext* ke bit terakhir pada *pixel* di dalam *cover image*. Hasil akhir yang didapat pada proses ini adalah sebuah *stego image.*

C. *Extracting*

Gambar 9 Diagram Alir Proses *Extracting*

Proses *extracting* merupakan kebalikan dari proses *embedding*, yaitu mengungkap pesan pada *stego image* dengan mengambil bit terakhir pada *pixel* citra kemudian menyusunnya menjadi suatu representasi bit yang membentuk sebuah karakter.

D. *Decryption*

Gambar 10 Diagram Alir Proses *Decryption*

a. Konversi ke Bentuk Nilai ASCII

Pada proses awal dekripsi, *ciphertext* yang diungkap dirubah ke dalam bentuk ASCII untuk mempermudah kalkulasi bit pesan yang akan didekripsi.
Diketahui *ciphertext*: **bVYYSPW[NV[UO.XU!36[u›**
ASCII: [98 86 89 89 83 80 87 91 78 86 91 85 79 46 88 85 33 51 54 91 117 155]

b. Atur *String* pada Matriks Orde Genap dan REM

Nilai ASCII yang diperoleh pada proses sebelumnya diatur menjadi matriks persegi orde genap dengan ukuran maksimum yang mungkin, dan menempatkan elemen yang sisa kedalam variabel REM.

98	86	89	89
83	80	87	91
78	86	91	85
79	46	88	85

Matriks 4x4

33	51
54	91

Matriks 2x2

[117 155]
REM

Gambar 11 Hasil Perubahan String menjadi Matriks Orde Genap dan REM

c. Menentukan Kunci Simetris KEY

Untuk mencari KEY dekripsi kita harus menentukan KEY1 dan KEY2 terlebih dahulu. Cara menemukan KEY1 dan KEY2 sama seperti pada proses enkripsi.
KEY1 = 4 + 2 + 2 = 8
Ukuran matrix terakhir = 2, hasil penjumlahan magic square matrix = 5
Diperoleh KEY2 = 5
KEY = 8 + 5 = 13 = 1 + 3 = 4

d. Mengurangkan Matriks dengan *Magic Square*
Hasil matriks orde genap yang didapat dikurangkan dengan *magic square matrix* yang memiliki ukuran atau dimensi yang sama:

$$\begin{vmatrix} 98 & 86 & 89 & 89 \\ 83 & 80 & 87 & 91 \\ 78 & 86 & 91 & 85 \\ 79 & 46 & 88 & 85 \end{vmatrix} - \begin{vmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{vmatrix} = \begin{vmatrix} 82 & 84 & 86 & 76 \\ 78 & 69 & 77 & 83 \\ 69 & 79 & 85 & 73 \\ 75 & 32 & 73 & 84 \end{vmatrix}$$

$$\begin{vmatrix} 33 & 51 \\ 54 & 91 \end{vmatrix} - \begin{vmatrix} 1 & 3 \\ 4 & 2 \end{vmatrix} = \begin{vmatrix} 32 & 48 \\ 50 & 89 \end{vmatrix}$$

Gambar 12 Pengurangan Matriks Orde Genap dengan *Magic Square Matrix*

e. Rotasi Matriks dengan Pola Rotasi Dekripsi

Urutan langkah-langkah pada pola rotasi ini dibuat sebagai berikut:

82	84	86	76
78	69	77	83
69	79	85	73
75	32	73	84

32	48
50	89

Gambar 132 Matriks Awal

(i) Pola rotasi pertama yaitu merotasi lingkaran terluar berlawanan arah jarum jam kemudian merotasi lingkaran didalamnya searah dengan jarum jam. Begitupun selanjutnya.
(ii) Pola rotasi kedua yaitu menggeser kebawah sebanyak satu kali untuk setiap kolom genap.
(iii) Pola rotasi ketiga yaitu menggeser kebawah kiri dengan arah diagonal.
(iv) Pola rotasi keempat yaitu menggeser kebawah kanan dengan arah diagonal.
(v) Pola rotasi kelima yaitu menukar posisi baris genap seperti ditunjukkan pada gambar:
(vi) Pola rotasi keenam dan yang terakhir yaitu menukar posisi kolom genap seperti ditunjukkan pada gambar :

Hasil yang didapatkan melalui pola rotasi ini adalah seperti yang ditunjukkan pada matriks.

84	69	76	75
79	77	32	85
78	73	86	69
82	83	73	84

89	32
50	48

Gambar 14 Hasil Akhir Pola Rotasi Matriks

f. Merubah Nilai REM berdasarkan KEY

Hitung variable *F* sebagai penjumlahan *magic square matrix* untuk orde KEY. Kurangi *F**(*n* + 1) untuk setiap n^{th} pada REM, untuk mengembalikan REM.
REM = [117 155]
KEY = 4, penjumlahan *magic square matrix order* 4 = 34
Dengan demikian *F* = 34
REM yang baru = [117 - (34 * (1 + 1)) 155 - (34 * (2 + 1))] = [49 53]

g. Menyusun Matriks dan REM Baru

Atur seluruh elemen pada matriks persegi dan sisanya (REM) dengan urutan saat mereka disusun. Kemudian konversikan nilai tiap elemen menjadi karakter ASCII untuk memperoleh *plaintext*.

84	69	76	75
79	77	32	85
78	73	86	69
82	83	73	84

Matriks 4x4

89	32
50	48

Matriks 2x2

49	53

REM

Gambar 15 Hasil Matriks Setelah Proses Dekripsi

h. Konversi ke Bentuk Karakter ASCII

ASCII: [84 69 76 75 79 77 32 85 78 73 86 69 82 83 73 84 89 32 50 48 49 53]
Hasil *plaintext* nya menjadi: **TELKOM UNIVERSITY 2015**

4. Hasil dan Analisis

A. *Recovery*

Tabel 1 Pengujian *Recovery* Pesan pada Stego Image dengan Jumlah Karakter Berbeda

No	Jumlah Karakter Sisip	Jumlah Pengujian	Persentase Keberhasilan *recovery* pada tiap Citra Uji			
			512x512	256x256	128x128	64x64
1	30	10 kali	100%	100%	100%	100%
2	60	10 kali	100%	100%	100%	100%
3	120	10 kali	100%	100%	100%	100%
4	240	10 kali	100%	100%	100%	100%
5	480	10 kali	100%	100%	100%	100%

Pada hasil uji *Recovery* yang ditunjukan pada Tabel 2 dapat dilihat bahwa sistem mampu menyisipkan dan mengekstrak kembali informasi yang sama pada citra yang digunakan dengan menggunakan panjang pesan yang bervariasi. Pengujian dilakukan secara berulang sebanyak 10 kali pada tiap citra uji. Dari pengujian ini dapat disimpulkan bahwa sistem dapat berjalan baik pada pengujian Recovery karena mampu mengembalikan informasi secara utuh.

B. *Robustness*

Tabel 2 Pengujian *Gaussian Noise* pada *Stego Image*

Variance Noise	Ukuran CItra	Jumlah Karakter Sisipan	BER
0.01	512x512	32768	99.62%
	256x256	8192	99.69%
	128x128	2048	99.61%
	64x64	512	99.80%

0.05	512x512	32768	99.69%
	256x256	8192	99.67%
	128x128	2048	99.80%
	64x64	512	99.76%

Tabel 3 Pengujian *Salt & Pepper Noise* pada *Stego Image*

Intensity	Ukuran CItra	Jumlah Karakter Sisipan	BER
0.05	512x512	32768	99.71%
	256x256	8192	99.83%
	128x128	2048	99.80%
	64x64	512	99.61%
0.1	512x512	32768	99.70%
	256x256	8192	99.65%
	128x128	2048	99.66%
	64x64	512	99.61%

Tabel 4 Pengujian *Resizing* pada *Stego Image*

Ukuran Citra	Jumlah Karakter Sisipan	Ukuran Citra (*Resizing* 75%)	BER
512x512	32768	384x384	99.56%
256x256	8192	192x192	99.65%
128x128	2048	96x96	99.65%
64x64	512	48x48	98.60%

Pengujian *Robustness* menunjukan ketahanan pesan rahasia pada *stego image* setelah mendapatkan serangan. Dari hasil uji *Robustness* di atas dapat dilihat bahwa sistem tidak mampu menjaga keutuhan pesan pada *stego image* yang diberikan serangan *noise* dan geometris. Hal ini dibuktikan dengan BER yang mendekati 100%. Dengan ini dapat dikatakan bahwa sistem sangat rentan terhadap serangan *noise* dan geometris karena hasil ekstraksi pesan setelah diberikan serangan pada *stego image* berubah hampir secara menyeluruh sehingga informasi awal yang disisipkan menjadi hilang atau tidak utuh seperti sedia kala.

C. *Avalanche Effect*

Tabel 5 *Plaintext* Awal sebagai Perbandingan

Plaintext Awal	*Ciphertext*	Bit *Ciphertext*
0000000000	13421342tû	00110001 00110011 00110100 00110010 00110001 00110011 00110100 00110010 01110100 10010110

Tabel 6 Hasil Uji *Avalanche Effect*

Plaintext Uji	*Ciphertext* Uji	Bit *Ciphertext*	Total bit Beda	*Avalanche Effect* (%)	Lokasi bit Beda
0000000001	13421342tù	00110001 00110011 00110100 00110010 00110001 00110011 00110100 00110010 01110100 10010111	1	1.25%	Bit ke-10
0000000010	13421342uû	00110001 00110011 00110100 00110010 00110001 00110011 00110100 00110010 01110101 10010110	1	1.25%	Bit ke-9
0000000100	13421442tû	00110001 00110011 00110100 00110010 00110001 00110100 00110100 00110010 01110100 10010110	1	1.25%	Bit ke-6
0000001000	13421352tû	00110001 00110011 00110100 00110010 00110001 00110011 00110101 00110010 01110100 10010110	1	1.25%	Bit ke-7

0000010000	13422342tû	00110001 00110011 00110100 00110010 00110010 00110011 00110100 00110010 01110100 10010110	1	1.25%	Bit ke-5
0000100000	13421343tû	00110001 00110011 00110100 00110010 00110001 00110011 00110100 00110011 01110100 10010110	1	1.25%	Bit ke-8
0001000000	14421342tû	00110001 00110100 00110100 00110010 00110001 00110011 00110100 00110010 01110100 10010110	1	1.25%	Bit ke-2
0010000000	13521342tû	00110001 00110011 00110101 00110010 00110001 00110011 00110100 00110010 01110100 10010110	1	1.25%	Bit ke-3
0100000000	23421342tû	00110010 00110011 00110100 00110010 00110001 00110011 00110100 00110010 01110100 10010110	1	1.25%	Bit ke-1
1000000000	13431342tû	00110001 00110011 00110100 00110011 00110001 00110011 00110100 00110010 01110100 10010110	1	1.25%	Bit ke-4

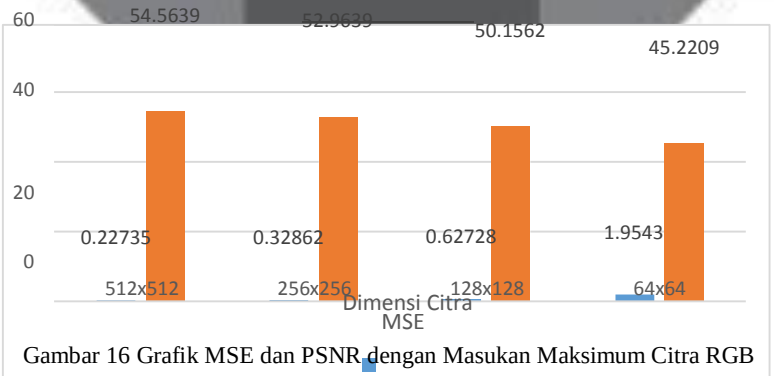
Hasil pengujian tersebut menunjukan bahwa bit *plaintext* awal berhasil teracak dan saling berpindah lokasi satu sama lain, namun total perubahan bit pada *ciphertext* awal dan *ciphertext* uji sama dengan total perubahan *plaintext* awal dan *plaintext* uji. Nilai *avalanche effect* yang dihasilkan untuk 10 karakter pesan hanya sebesar 1.25%. Hal ini diakibatkan oleh penggunaan kunci pada algoritma kriptografi RDC yang bersifat statis atau tidak dapat dimasukan sesuai keinginan pengguna karena kunci dihasilkan berdasarkan panjang *plaintext* yang akan dienkripsi. Dengan ini dapat dikatakan bahwa algoritma enkripsi cukup rentan terhadap serangan karena perubahan 1 bit pada *plaintext* tidak mampu menghasilkan *avalanche effect* sebesar 50%.

D. *Fidelity*

a. Pengujian PSNR

Tabel 8 Nilai PSNR Berdasarkan Panjang Karakter Maksimum RGB

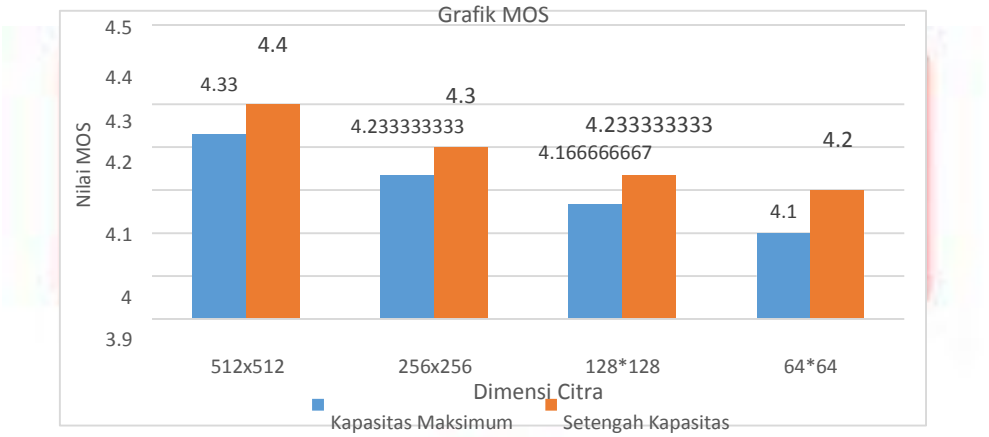
Ukuran Citra	Ukuran Pesan Maksimum	MSE	Nilai PSNR
512x512	98304 karakter	22.735×10^{-2}	54.5639 dB
256x256	24576 karakter	32.862×10^{-2}	52.9639 dB
128x128	6144 karakter	62.728×10^{-2}	50.1562 dB
64x64	1536 karakter	1.9543	45.2209 dB



Gambar 16 Grafik MSE dan PSNR dengan Masukan Maksimum Citra RGB

Dari hasil pengujian dengan menyisipkan maksimum karakter pada citra RGB yang ditunjukan diatas dapat dilihat bahwa citra berdimensi 512x512 memiliki nilai PSNR paling baik, yaitu 54.5639dB. Pada citra uji terkecil dengan dimensi 64x64 diperoleh PSNR dengan nilai 45.2209dB. Hasil ini menunjukan bahwa penggunaan dimensi yang lebih besar akan menghasilkan PSNR yang lebih baik. Berdasarkan pengujian ini dapat disimpulkan bahwa sistem dapat berjalan baik karena mampu menghasilkan nilai PSNR ≥ 30dB dengan masukan maksimum layer RGB.

b. Pengujian MOS



Gambar 17 Grafik MOS

Berdasarkan grafik yang dihasilkan secara subjektif oleh 30 orang responden tersebut dapat kita lihat bahwa ukuran *pixel* mempengaruhi kualitas gambar yang digunakan sebagai media penyimpanan rahasia. Hal ini dibuktikan dengan nilai MOS yang ditunjukan oleh grafik pada diatas.

Grafik tersebut juga menunjukan bahwa kualitas gambar yang dihasilkan setelah proses penyisipan tidak berubah secara signifikan secara kasat mata dikarenakan nilai MOS yang dihasilkan berada diatas 4. Dengan demikian dapat dikatakan bahwa hasil uji MOS pada sistem sangat baik.

5. Kesimpulan

Dari hasil pengujian sistem yang dilakukan pada Tugas Akhir ini, dapat disimpulkan bahwa pengujian *Recovery* sistem dapat menyisipkan dan mengungkap pesan rahasia pada citra uji dengan sangat baik karena memiliki akurasi 100% atau tidak ada bit *error* pada pesan rahasia. Sistem dapat menjaga kualitas citra dengan baik. Hal ini dibuktikan oleh PSNR pada *stego image* yang dimasukan pesan maksimum memiliki nilai ≥ 45.2209dB dan rata-rata nilai MOS yang dihasilkan > 4.

Berdasarkan hasil uji *Robustness* sistem tidak mampu menjaga ketahanan pesan rahasia di dalam *stego image* karena serangan *Gaussian Noise*, *Salt & Pepper Noise*, dan *Resizing* merubah *pixel* citra yang menampung bit pesan rahasia secara signifikan, hal ini dibuktikan dengan nilai BER yang dihasilkan > 99%. Lamanya waktu komputasi steganografi LSB berbanding lurus dengan panjang pesan yang akan disisipkan karena proses penyisipan dilakukan secara satu per satu ke dalam *pixel* gambar. Proses enkripsi algoritma RDC sangat efisien dari segi waktu komputasi karena lamanya proses enkripsi pesan hingga 2000 karakter ≤ 1.00226 detik. Berdasarkan pengujian *Avalanche Effect* algoritma RDC masih rentan terkena serangan karena jumlah bit yang berbeda hanya 1.25%.

Saran

Adapun saran untuk pengembangan Tugas Akhir selanjutnya adalah:

1. Melakukan penyempurnaan agar pesan rahasia tetap terjaga keutuhannya setelah diberikan serangan *noise* dan geometris.
2. Merubah pola rotasi pada algoritma RDC dan penggunaan kunci simetris algoritma RDC yang dapat digunakan untuk merubah seluruh karakter untuk meningkatkan nilai *Avalanche Effect*.
3. Menggunakan media lain sebagai *cover*, seperti *audio* atau *video*.

Daftar Pustaka

[1] D.W. Bender, N.M. Gruhl, A. Lu. 1996. *Techniques for Data Hiding*. IBM Syst.
[2] Anandika, Hari. 2012. *Perancangan dan Analisis Multiple Watermarking pada Citra Digital berbasis Iterative Threshold dan Deteksi Tepi*. Bandung: Institut Teknologi Telkom.
[3] Awcock, G.W. 1996. *Applied Image Processing*. Singapore. McGraw-Hill Book.
[4] Arini, Gies Masita, & Widyawan, Tri Ismardiko. 2012 . *Pengamanan Pesan Steganografi dengan Metode LSB Berlapis Enkripsi dalam PHP*. Jakarta: Universitas Budi Luhur
[5] Schneier, B. 1996. Applied Cryptography: Protocols, Algorthms, and Source Code in C. John Wiley & Sons, Inc.
[6] AJ Menezez, PC van Oorschot, SA Vanstone. 1996. *Handbook of Applied Cryptography*. CRC Press.
[7] Kandele, N., & Tiwari, S. 2013. *New Cryptography Method Using Relative Displacement: RDC Symmetric Key Algorithm*.
[8] Alatas, Putri. 2009. *Teknik Steganografi dengan Metode LSB pada Citra Digital*.
[9] Munir, Rinaldi. 2006. *Kriptografi*. Bandung: Penerbit Informatika.
[10] Asep S, Himawan, Nazori A. 2012. *Aplikasi Steganografi untuk Menyembunyikan Teks dalam Media Image dengan Menggunakan Metode LSB.* Jakarta: Universitas Budi Luhur.
[11] Mulyantini, Agustien. 2012. *Analisis Steganografi pada Citra Digital menggunakan DCT (Discrete Cosine Transform) dan Enkripsi AES.* Bandung: Institut Teknologi Telkom.
[12] Utomo, Tri Prasetyo. 2011. *Steganografi Gambar dengan Metode Least Significant Bit untuk Proteksi Komunikasi pada Media Online*.
[13] Edisuryana, Mukharrom. 2013. *Aplikasi Steganografi pada Citra Berformat Bitmap dengan Menggunakan Metode End Of File*. Semarang: Universitas Dipenogoro.
[14] Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J. & Kalker, T. 2008. *Digital Watermarking and Steganography-Second Edition.* Burlington, MA, USA,
Elsevier Inc.
[15] Jayant P. Bhoge et al. 2014. *Avalanche Effect of AES Algorithm*. Amravati, India: Government College of Engineering.