PENGAMANAN INFORMASI TEKS DI CITRA DIGITAL MENGGUNAKAN METODE LSB DAN ALGORITMA RC4

Taufiq Nur Ihsan[1], Solichul Huda[2]
Jurusan Tehnik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Nakula I no 5-11 Semarang, Kode Pos 50131, Telp. (024)3515261, 3520165 Fax:3569684
E-Mail: ikhsan.shincan@gmail.com[1], solichul.huda@dsn.dinus.ac.id[2]

Abstrak

Dalam berbagai bidang pekerjaan informasi dapat diperjual-belikan. Untuk mengindari penyalahgunaan, maka informasi tersebut perlu diamankan. Pengamanan informasi dapat dilakukan dengan berbagai macam cara. Informasi berupa teks dapat diamankan dengan cara menyisipkannya kedalam file lain ataupun disandikan sehingga tidak dapat dibaca. Penelitian ini mengimplementasikan metode Least Significant Bit dan algoritma RC4 untuk mengamankan iformasi tersebut. Metode ini diterapkan agar file tidak mengalami perubahan kapasitas dikarenakan jumlah pesan yang disisipkan, cara kerja metode ini adalah mengganti bit terakhir setiap byte citra dengan bit pesan. Algoritma RC4 digunakan untuk menyandikan pesan terlebih dahulu sebelum disisipkan agar informasi menjadi lebih aman. Algoritma ini digunakan karena akan membuat kunci baru sepanjang 256byte, sehingga meskipun diberikan kunci yang sedikit, maka mempunyai keamanan yang baik. Hasil analisa uji beda statistik nilai PSNR terhadap citra hasil stegano dengan menggunakan enkripsi dan tanpa enkripsi menyatakan bahwa memiliki perbedaan nilai yang signifikan, tetapi perbedaan ini tetap tidak bisa dibedakan dengan indra penglihatan, dikarenakan nilai PSNR diatas 30dB dan diperlihatkan dengan hasil histogram yang memiliki nilai rata-rata sama. Terdapat perbedaan waktu eksekusi program antara yang menggunakan enkripsi dan tanpa enkripsi,hal ini disebabkan karena source code yang dikerjakan semakin banyak untuk menyandikan pesan terlebih dahulu, sehingga mempengaruhi waktu eksekusi.

Kata kunci: *kriptografi, steganografi, LSB, RC4, java*

Abstract

In the job information can be bought and sold. To avoid that, that information needs to be secured. Security of information can be done with just about anything. Information in the form of a text can be secured by means of paste into other files or encrypted so it can not be read. This study implements the method of Least Significant Bit and the RC4 algorithm to secure the iformasi. This method is applied so that the file has not changed because the capacity of the number of messages inserted, the workings of this method is to replace the last bits of each byte image with one bit of the message. RC4 algorithm used to encrypt a message before inserted so that the information becomes more secure. This algorithm is used because it would create a new key along 256byte, so although given the key bit, it has a good security. Statistical analysis of the results of different test image PSNR against stegano by using encryption and no encryption generate significant value differences, but these differences are indistinguishable to the eye, because the value of PSNR above 30dB and the results are shown in the histogram that have the same average value. There are differences between the program execution time using encryption and without encryption, this is because the source code that is executed getting longer to encode a message in advance, thus affecting the execution time

Keywords: *Chryptography, Steganography, LSB, RC4, java*

1. PENDAHULUAN

1.1 Latar Belakang

Dalam bidang perkerjaan jurnalistik surat kabar, informasi merupakan suatu hal yang sangat penting. Informasi yang digunakan dapat berupa gambar dan tulisan. Informasi berupa tulisan merupakan yang lebih penting daripada gambar, karena gambar digunakan hanya untuk memberi pandangan tentang informasi yang ditulis. Informasi yang diperoleh dapat diperjual belikan kepada pihak lain diluar instansi yang bersangkutan, sehingga perlu pengamanan untuk menghidari dari penyalahgunaan.

Pengaman yang digunakan yaitu menggunakan metode steganografi dan kriptografi, sehingga pesan akan disandikan yang kemudian akan dimasukkan kedalam gambar, sehingga tidak diketahui adanya informasi penting yang dibutuhkan.

Metode steganografi yaitu pengaman informasi dengan menyisipkan pesan kedalam sebuah gambar, sedangkan kriptografi yaitu menyandikan pesan yang akan disisipkan kedalam sehingga mendapatkan keamanan yang lebih baik [1].

Pada penelitian ini menggunakan metode Least Significant Bit yang memiliki keunggulan yaitu menghasilkan ukuran file yang tetap dibandingkan dengan metode End Of File, ukuran file yang dihasilkan tidak mengalami perbesaran dari yang aslinya [2]. Untuk menjaga keamaan informasi, maka digunakan algoritma RC4 yang memiliki kemanaan yang lebih baik dengan menggunakan kunci 256bit[4]. Dibandingkan algoritma kriptografi klasik yang dapat menggunakan kunci pendek sehingga kurang memiliki keamaanan yang baik.

1.2 Tujuan

Penelitian ini bertujuan untuk dapat mengamankan informasi berupa teks yang akan disisipkan pada citra digital serta dapat mengetahui perbedaan ukuran file dan kualitas citra yang dihasilkan dari metode Least Significant Bit dan algoritma RC4 dalam mengamankan informasi berupa teks.

2. TINJAUAN PUSTAKA

2.1 Data Teks

Teks adalah sekumpulan dari angka dan huru dan simbol-simbol lainnya, karakter-karakter tersebut dapat diubah menjadi bit-bit kecil sehingga dapat diproses lebih lanjut.

2.2 Citra Digital

Citra digital adalah gambar dua dimensi yang dihasilkan dari proses digitalisasi yang bersifat analog ke digital. Citra digital tersusun dalam bentuk kotek, setiap kotak memiliki koordinat (x,y).

2.3 Steganografi Least Significant Bit

Metode ini bekerja dengan cara mengganti bit terakhir citra denga pesan yang akan disisipkan. Satu bit pesan akan menggantikan 1 bit terakhir citra, sehingga perubahan 1 bit citra tidak akan menghasilkan perbedaan warna yang signifikan.

Dalam citra JPG dan PNG terdapat 3 byte penyusun warna, setiap byte terdapat 8 bit, sehingga citra yang mempunyai resolusi 1024 x 768 dapat

menampung data sebesar 2.359.296 bit atau 295.912 karakter.

2.4 Algoritma RC4

Algoritma RC4 merupakan salah satu algoritma kriptografi yang menggunakan kunci simetris. RC4 menggunakan kunci sepanjang 256 byte, meskipun kunci yang digunakan kurang dari 256 karakter, maka kunci otomatis akan di buat menjadi 256byte dengan cara menyalin kunci hingga mencapat 256byte. Setelah didapat kunci 256bye, kemudian kunci tersebut akan di generate kembali untuk di EXOR kan dengan pesan.

RC4 memproses data dalam ukuran byte (1byte=8bit). Algoritma ini menggunakan dua buah array s-box untuk melakukan permutasi, untuk mendapatkan kunci baru.

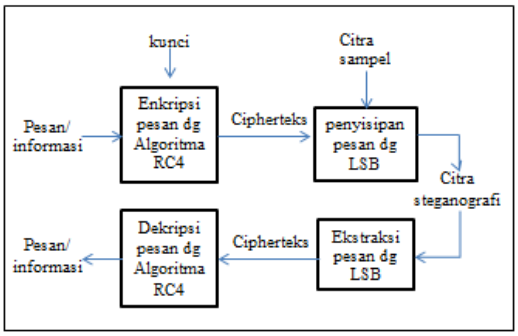
3. METODE

3.1 Jenis Data

Data yang digunakan dalam penelitian ini adala data citra digital dan teks. Terdapat 20 sampel citra digital yang akan digunakan dalam penelitian ini, pada citra tersebut akan disisipkan sebuah pesan dan diperoleh kapasitas file dan nilai PSNR, dari 20 sampel citra akan didapatkan perubahan yang terjadi ketika disisipkan pesan dengan jumlah yang berbeda-beda.

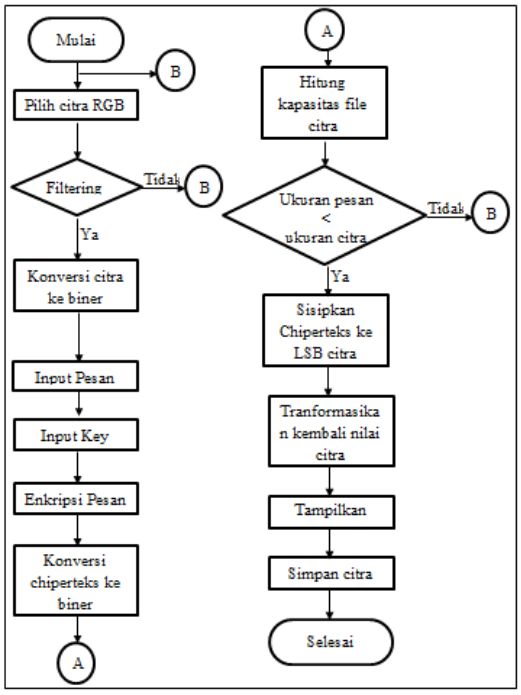
Pesan yang disisipkan sebanyak 20 sampel yaitu 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000. Jumlah pesan yang berbeda bertujuan untuk mengetahui seberapa pengaruhkan jumlah pesan yang disisipkan dengan kualitas citra yang dihasilkan.

3.2 Metode Penelitian

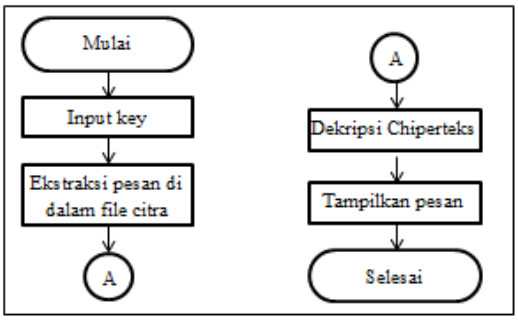


Gambar 1. Metode yang diusulkan

Dari metode yang diusulkan maka dapat dibuat perancangan penelitian dapat mengetahui fungsi yang dijalankan dari metode tersebut. Untuk dapat lebih jelas, metode dapat dijabarkan sebagai berikut:



Gambar 2. Metode encoding pesan



Gambar 3. Metode decoding pesan

3.3 Eksperimen

Untuk membuktikan hasil penelitian, maka dilakukan uji coba dengan menguji hasil metode yang digunakan. Pengujian dilakukan dengan memberikan 20 sampel citra yang akan disisipkan dengan 1 sampel pesan sebanyak 20 kali.

Pengujian yang dilakukan meliputi pengujian nilai PSNR citra, pengujian kapasitas file citra, dan pengujian waktu eksekusi program.

4. ANALISIS DAN PEMBAHASAN

Dari hasil penelitian yang dilakukan, penyisipan pesan dilakukan dengan dua cara, yaitu tanpa enkripsi dan dengan enkripsi. Cara ini dilakukan untuk mengetahui perbedaan dari penambahan keamanan yang telah dilakukan dengan cara mengenkripsi pesan yang akan disisipkan.

4.1 Citra Sampel.

Citra digital dari penelitian ini menggunakan 20 sampel foto yang didapat dari hasil pemotretan menggunakan kamera nikon D2HS yang mempunyai resolusi 2464 x 1632piksel dengan format (*png). Format (*.png) digunakan karena memiliki 24 bit penyusun warna, sehingga mengahsilkan kualitas gambar yang baik.

4.2 Proses Enkripsi dan Dekripsi

Sebelum pesan disisipkan kedalam citra sampel, maka terlebih dahulu pesan tersebut disandikan menggunakan algoritma RC4. Diberikan pesan berisi “kriptografisteganografi” dan kunci yang digunakan adalah “skripsi”.

Yang pertama dilakukan adalah membuat array s-box dengan isi bilangan 0-255. Selanjutkan membuat array s-box kunci dengan cara menyalin kunci yang digunakan sehingga memiliki panjang 256byte yang kemudian disimpan di dalam s-box kunci.

Setelah didapatkan kedua array s-box maka dilakukan permutasi antara array s[i] dengan array s[j]. permutasi dilakukan dengan terlebih dahulu mengubah isi array K menjadi bentuk ascii sehingga bisa dilakukan penjumlahan. Permutasi dilakukan dengan cara

j=0

For i=0 to 255

j=(j+S[i]+K[i]) mod 256

isi S[i] ditukar dengan isi S[j]

Setelah array S dilakukan permutasi, maka selanjutnya membangkitkan aliran kunci yang digunakan untuk enkripsi dan dekripsi pesan.

i = j = 0

i = (i + 1) mod 256

j = (j + S[i]) mod 256

isi S[i] dan S[j] ditukar

t = (S[i] + S[j])mod 256

K= S[t]

Setelah mendapatkan kunci aliran K, maka untuk memperoleh cipherteks dilakukan XOR antara K dengan P[ixd] den sedangkan untuk memperoleh plainteks kembali maka dilakukan XOR antara K dengan C[idx].

Setelah dilakukan XOR antara plainteks dengan K, maka didapatkan hasil cipherteks berikut “ úú`– N¤Ù1JŒ}Ç6U¤Á;Ïw•Ñu ”

4.3 Penyisipan Pesan

Setelah pesan berhasil disandikan, kemudia pesan akan disisipkan kedalam citra digital. Sebelum pesan disisipkan kedalam citra terlebih dahulu dihitung panjang pesan yang sudah di ubah kedalam byte dan ditambah dengan byte penanda dengan panjang citra. 32byte citra sampel yang pertama sisipkan jumlah panjang pesan sehingga memudahkan dalam proses decoding. Setelah pesan menunjukkan dapat ditampung dalam citra maka kemudian diambil 1 per 1 bit pesan dan kemudian bit citra akan diganti dengan bit pesan. Peletakkan bit pesan kedalam byte citra sebagai berikut:

Di hitung jumlah byte pesan adalah N kemudian dilakukan perulangan di setiap byte pesan, pesan akan disisipkan setalah 32byte pertama, karena 32byte pertama digunakan sebagai penanda jumlah pesan.

For (i=0; i<N; i++)

Dilakukan pemilihan bit pesan yang akan digunakan untuk mengganti bit terkhir citra.

```
Add = byte pesan ke i
for(bit=7; bit>=0; offset++)
{
        b = add >>> bit & 0x1
        citra[offset] = ((byte)
(citra[offset] & 0xFE | b))
        bit - -
}
Return citra
```

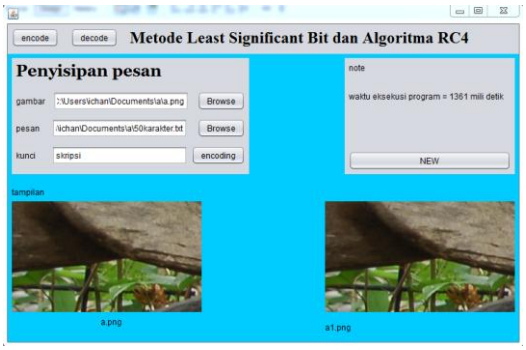
b merupakan nilai byte pesan yang ke “bit” diambil 1 bit saja dan kemudian digunakan untuk mengganti byte citra yang ke offset dengan memilih bit yang terakhir. Setelah pesan selesai disisipkan, kemudian citra sampel akan buat kedalam file citra stegano yang baru dengan menggunakan ekstensi (*.png). format (*.png) dipilih karena memiliki 4 byte penyusun warna yang terdiri dari Red, Green, Blue, dan Alpha, sehingga apabila digunakan untuk menyisipkan citra png tidak kehilangan nilai alpha nya.

Gambar 4. Citra hasil steganografi

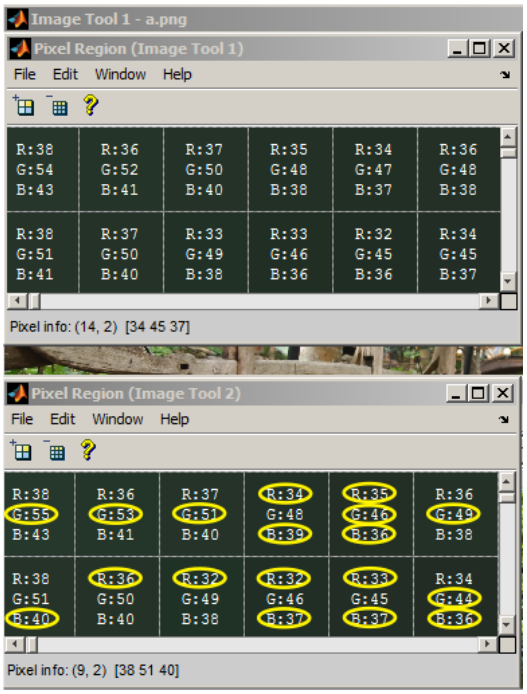
4.4 Simulasi dan Analisis

Simulasi untuk penyisipan pesan dilakukan dengan menggunakan citra sampel a.png yang akan disisipkan

dengan pesan yang berbeda- beda. Berikut ini adalah perbandingan citra sampel dan citra hasil stegano dengan jumlah pesan yang berbeda.



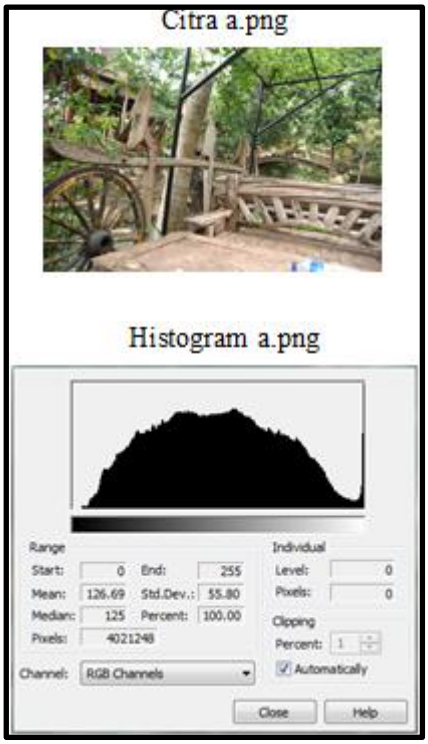
Gambar 5. Tampilan penyisipan pesan



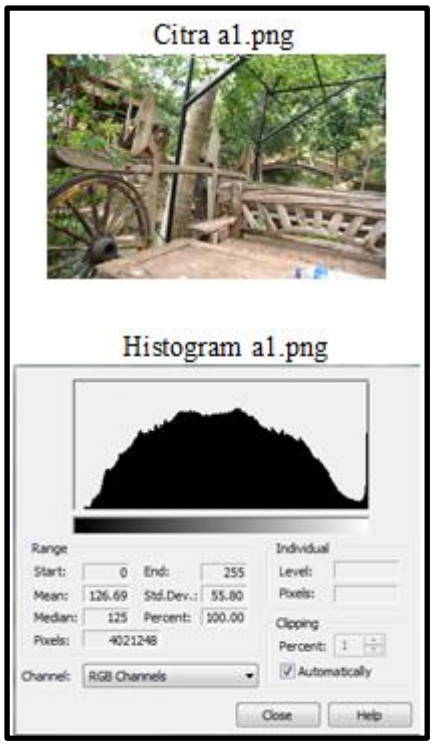
Gambar 6. Perbedaan nilai RGB citra hasil stegano

Diambil nilai RGB dari 12 piksel citra a.png dan a20.png. Dari a.png didapatkan hasil penjumlahan nilai RGB sebanyak 1473 yang kemudian dibagi banyak byte yang sudah diambil nilainya, yaitu 3 byte setiap piksel sehingga jumlah bytenya menjadi 36. Nilai rata-rata RGB a.png yaitu 1473 : 36 menjadi 40,91

Kemudian diambil nilai RGB dari citra a20.png. jumlah nilai yang didapatkan yaitu 1473, yang kemudian dihitung rata-rata nilai RGB yaitu 1473: 36 menjadi 40,91. Sehingga dapat disimpulkan bahwa rata-rata nilai RGB hasil stegano tidak mengalami perubahan sehingga tidak dapat dibedakan oleh indera penglihatan manusia. Penghitungan rata-rata nilai RGB keseluruhan dapat dilihat dari histogram, yang menampilkan grafik nilai RGB dan mengetahui rata-rata nilai RGB.



Gambar 7. Hasil histogram citra asli a.png



Gambar 8. Hasil histogram a1.png dengan 50 byte pesan

Berdasarkan gambar 7 dan 8 dapat dilihat bahwa histogram antara citra asli dengan citra hasil steganografi tidak mengalami perubahan, yaitu nilai mean histogram adalah 126.69 dan nilai median 125. Hasil histogram ini menandakan hasil steganografi yang dibuat tidak mengalami perubahan warna yang signifikan sehingga tidak dapat dibedakan dengan indera penglihatan.

Perbedaan citra sampel asli dengan citra hasil stegano dapat dibedakan dengan mengghitung nilai MSE dan PNSRs citra asli dan citra stegano. Perbedaan ini dapat dilihat dikarenakan penyisipan pesan kedalam citra digital dapat merubah kualitas citra. Maka dengan ini hasil citra steganografi (LSB) dapat diuji dengan menghitung nilai PSNR.

Untuk mendapatkan informasi yang sudah disisipkan, maka perlu dilakukan proses decoding. Pesan hasil decoding dapat disimpan kembali dalam bentuk (*.txt).

Gambar 9. Tampilan decoding pesan

4.5 Pengujian

4.5.1 Uji MSE dan PSNR Dengan Enkripsi

Pengujian niali MSE dan PSNR dilakukan dengan menggunakan MATLAB. Dilakukan perhitungan dengan menggunakan 1 sampel citra yaitu (a.png) dengan 20 sampel pesan, sehingga diperoleh hasil perhitungan sebagai berikut:

Tabel 1: Uji nilai MSE dan PSNR dengan enkripsi

Pesan (karakter)	MSE	PSNR (dB)
50	2.47E-05	161.5328
100	4.87E-05	154.7258

150	7.16E-05	150.8621
200	9.36E-05	148.1795
250	1.23E-04	145.4587
300	1.49E-04	143.4988
350	1.75E-04	141.9028
400	2.03E-04	140.4208
450	2.28E-04	139.2877
500	2.56E-04	138.1241
550	2.82E-04	137.1407
600	1.49E-04	143.4988
650	1.74E-04	141.9873
700	3.53E-04	134.9186
750	3.77E-04	134.2479
800	3.99E-04	133.6713
850	4.27E-04	132.9922
900	4.54E-04	132.3948
950	4.76E-04	131.9061
1000	5.04E-04	131.3405

Dari tabel 1 diperoleh nilai MSE dan PSNR untuk citra sampel a.png dengan menggunakan enkripsi. Nilai MSE dan PSNR menunjukkan perbedaan citra sampel dan citra hasil stegano. Standart nilai MSE adalah mendekati 0 sedangkan nilai PSNR adalah 30 dB, semakin kecil nilai MSE maka memiliki tingkat kesamaan yang baik, sedangkan semakin tinggi nilai PSNR maka memiliki tingkat kesamaan yang baik pula

4.5.2 Uji MSE dan PSNR Tanpa Enkripsi

Tabel 2: Uji nilai MSE dan PSNR tanpa enkripsi

Pesan (karakter)	MSE	PSNR (dB)
50	1.88E-05	164.2457
100	3.66E-05	157.5886
150	5.07E-05	154.3124
200	7.03E-05	151.0444
250	8.59E-05	149.0546
300	1.02E-04	147.3337
350	1.19E-04	145.8206
400	1.37E-04	144.3836
450	1.53E-04	143.2379
500	1.69E-04	142.2627
550	1.85E-04	141.3429
600	1.02E-04	147.3337
650	1.18E-04	145.8952
700	2.34E-04	139.0351
750	2.50E-04	138.3403
800	2.71E-04	137.5502
850	2.82E-04	137.1376
900	3.06E-04	136.3504
950	3.20E-04	135.8987
1000	3.38E-04	135.3425

Dari pengujian nilai MSE dan PSNR citra stegano tanpa enkripsi menunjukan bahwa citra stegano mengalami penurunan tingkat kesamaan yang ditunjukkan dengan semakin besarnya nilai MSE dari nilai standart yang

mendekati 0 dan semakin kecilnya nilai PSNR yang mendekati 30dB

4.5.3 Analisa Uji Beda Statistik

Pada pengujian sebelumnya telah didapat nilai PSNR citra sampel (a.png) yang menggunakan enkripsi dan tanpa enkripsi. Untuk mengetahui apakah ada perbedaan antara citra stegano yang mengunakan enkrip dan tanpa enkripsi, maka digunakan pengujian perbedaan dua rata-rata dari sampel berkorelasi. Untuk menguji signifikan atau tidaknya perbedaan dua rata-rata sampel dapat menggunakan rumu uji t sebagai berikut:

$$t = \frac{\sum D}{\sqrt{\frac{n\sum D^2 - (\sum D)^2}{n-1}}}$$

$$t = \frac{-75{,}42}{\sqrt{\frac{20.288{,}25 - (-75{,}42)^2}{20-1}}}$$

$$t = \frac{-75{,}42}{\sqrt{\frac{5765 - 5688{,}1764}{19}}}$$

$$t = \frac{-75{,}42}{2{,}01}$$

t = -37,522

Pengujian dilakukan pada taraf signifikan α = 0,05 dan derajat kebebasan dk =(n_1+n_2)-2=38, maka dari daftar distribusi t dengan peluang 1-α = 0,95 dan dk = 38 diperoleh $t_{0,95}(38)$ = 2.025

Beradasarkan perhitungan penilitian diperoleh t_{hitung} = -37,522, jadi $t_{hitung} < t_{tabel}$ yaitu = -37,522 < -2,025 dilihat dari kedudukan t_{hitung} dan t_{tabel} sehingga H_0 ditolak dan H_1 diterima dengan taraf signifikan α = 0,05. Dapat disimpulkan bahwa terdapat perbedaan yang signifikan antara nilai PSNR citra stegano yang menggunakan enkripsi dan tanpa enkripsi.

Untuk memperkuat hasil pengujian , maka dilakukan kembali pengujian nilai PSNR dengan menggunakan 20 citra sampel lainnya dan 20 sampel pesan. Pengujian dilakukan untuk mengetahui seberapa besar perbedaan yang terjadi dari nilai PSNR yang menggunakan enkripsi dan tanpa enkripsi. Dari 20 sampel citra didapatkan hasil uji beda korelasi sebagai berikut:

Tabel 3. Hasil uji beda korelasi nilai PSNR

citra	t_{hitung}	perbedaan
a.png	-37,522	Ada
b.png	-1.453	Tidak ada
c.png	43.225	Ada
d.png	72.839	Ada
e.png	52.808	Ada
f.png	6.971	Ada
g.png	70.971	Ada
h.png	40.826	Ada
i.png	93.666	Ada
j.png	31.318	Ada
k.png	54.167	Ada
l.png	-0.999	Tidak ada
m.png	41.486	Ada
n.png	0.140	Tidak ada
o.png	55.939	Ada
p.png	0.435	Tidak ada
q.png	59.903	Ada
r.png	36.065	Ada
s.png	42.882	Ada
t.png	14.458	Ada

Dari tabel 3 dapat diketahui perbedaan antara citra hasil stegano yang

menggunakan enkripsi dan tanpa enkripsi. Dari 20 pengujian beda statistik terhadap nilai psnr. Memperoleh hasil 4 sampel tidak memiliki perbedaan dan 16 sampel terdapat perbedaan yang signifikan antara citra yang menggunakan enkripsi dan tanpa enkripsi. dilihat dari perbandingan t_{tabel} dengan t_{hitung} dari 20 citra sampel yang digunakan. Dimana $t_{hitung} < t_{tabel}$ yang dilihat dari nilai t_{hitung} melebihi batas nilai yang ditentukan yaitu antara -2.025 sampai 2.025.

4.5.4 Waktu Eksekusi dan Kapasitas File

Untuk menguji keefektifan algoritma kriptografi dan metode steganografiyang digunakan, dapat dilhat dari waktu eksekusi program, dan kapasitas file citra stegano yang sudah dibuat. Sehingga dapat diketahui seberapa efektifkah algoritma dan metode yang digunakan. Pengujian dilakukan dengan menggunakan enkripsi dan tanpa enkripsi yang menggunakan 20 sampel citra dan 20 sampel pesan. Pengujian dilakukan dengan citra sampel a.png terdapat dalam tabel 4

Tabel 4. Uji waktu dan kapasitas file

Pesan (karakter)	Dengan enkripsi		Tanpa enkripsi	
	Waktu eksekusi (ms)	Kapasitas file (KB)	Waktu eksekusi (ms)	Kapasitas file (KB)
50	3467	10.144	3691	10.144
100	3647	10.144	4209	10.144
150	3084	10.144	3244	10.144
200	3553	10.144	3033	10.144
250	3207	10.144	3220	10.144
300	3298	10.144	3099	10.144
350	3751	10.144	3451	10.144
400	3351	10.144	3110	10.144
450	3571	10.144	3880	10.144
500	3426	10.144	3962	10.144
550	3763	10.144	3200	10.144
600	3084	10.144	3134	10.144
650	3392	10.144	3154	10.144
700	3249	10.144	3235	10.144
750	4021	10.144	3155	10.144
800	3142	10.144	3455	10.144
850	3333	10.144	3446	10.144
900	3139	10.144	3347	10.144
950	3434	10.144	4123	10.144
1000	3247	10.144	3236	10.144

Dari tabel 4 dapat dilihat bahwa waktu eksekusi program dapat berjalan dengan cepat dengan waktu kurang dari 5 detik. Waktu ini memperlihatkan bahwa semakin cepat waktu eksekusi program maka semakin efektif algoritma dan metode yang digunakan. Pengujian dilakukan dengan menggunakan leptop asus A43SD yang memiliki spesifikasi prosesor corei5 2.5GHz dan ram 6gb dengan pengaturan kerja prosesor maksimum 60% yang dapat menjalankan program dengan cepat. Hasil pengujian waktu eksekusi dapat berbeda apabila leptop yang digunakan memiliki spesifikasi yang lebih rendah ataupun lebih tinggi.

Dan dilihat dari kapasitas file citra hasil stegano yang dihasilkan, tidak memiliki perbedaan kapasitas file antara citra

yang disisipkan pesan sebanyak 50 karakter hingga 1000 karakter. Sehingga dapat disimpulkan bahwa seberapa banyak pesan yang disisipkan tidak akan mempengaruhi kapasitas file citra dikarenakan tidak dilakukannya perubahan ukuran citra hasil steganografi.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan maka dapat diambil kesimpulan seperti berikut:

1. Pengamanan informasi berupa teks dengan menggunakan metode Least Significant Bit dan algoritma RC4 dapat berjalan dengan baik.
2. Berdasarkan uji beda rata-rata statistik nilai PNSR yang dihasilkan citra stegano yang menggunakan enkripsi mengalami perbedaan yang signifikan dengan citra stegano tanpa enkripsi.
3. Waktu eksekusi program mengalami perbedaan antara sampel yang menggunakan enkripsi dan tanpa enkripsi
4. Ukuran kapasitas file yang dihasilkan tidak mengalami perubahan di setiap pesan yang disisipkan.
5. Iformasi yang disisipkan menjadi lebih aman, karena pesan terlebih dahulu disandikan sebelum disisipkan.

5.2 Saran

Sedangkan sran yang dapat diberikan pada penelitian ini adalah sebagai berikut:

1. Penelitian ini dapat dilanjutkan dengan memperbaiki algoritma yang digunakan, meskipun dapat menghasilkan pesan yang diharapkan, masih terdapat kesalahan dalam dekripsi.
2. Penelitian ini dapat dilanjutkan dengan menambahkan algoritma kriptografi yang lainnya agar menjadi lebih aman.

DAFTAR PUSTAKA

[1] Prayudi, Y.; Kuncoro, P.S., "IMPLEMENTASI STEGANOGRAFI MENGGUNAKAN TEKNIK ADAPTIVE MINIMUM ERROR LEAST SIGNIFICANT BIT REPLACEMENT (AMELSBR)," in *Seminar Nasional Aplikasi Teknologi Informasi*, Yogyakarta, 2005, pp. 1-6.

[2] Sembiring, S., "Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan Teks Pada Gambar dengan Metode End Of File," *Pelita Informatika Budi Darma*, vol. IV, no. 2, pp. 45-51, agustus 2013.

[3] Cahyadi , T., "Implementasi Steganografi LSB dengan Enkripsi Vigenere Cipher pada Citra JPEG," *TRANSIENT*, vol. 1, no. 4, Desember 2012.

[4] Setyaningsih, E., "Implementasi System Sandi Stream Cipher untuk Pengamanan Data Image," in *Seminar Nasional Teknologi dan Komputasi*, Bangkalan, 2013, pp. 84-91.

[5] Krisnawati, "Metode Least Significant Bit (LSB) dan End Of File (EOF) untuk Menyisipkan Teks ke Dalam Citra Grayscale," in *Seminar Nasional Informatika*, Yogyakarta, 2008, pp. 39-44.

[6] Munir, R., "Penyembunyian Data Secara Aman di Dalam Citra Berwarna dengan Metode LSB Jamak Berbasis Chaos," in *Seminar Nasional Ilmu Komputer dan Aplikasinya*, Bandung, 2007, pp. 1-5.

[7] Sutiono, A.P., "Algoritma RC4 Sebagai Perkembangan Metode Kriptografi," Institut Teknologi Bandung, Bandung, Makalah 2010.

[8] Lestari, D.; Riyanto, M.Z., "Suatu Algoritma Kriptografi Stream Cipher Berdasarkan Fungsi Chaos," in *Kontribusi Pendidikan Matematika dan Matematika dalam Membangun*, Yogyakarta, 2012, pp. 33-40

.

[9] Ariyanto , Y., "Algoritma RC4 dalam Proteksi Transmisi dan Hasil Query untuk ORDBMS POSTGRESQL," *Jurnal Informatika*, vol. 10, no. 1, pp. 53-59, Mei 2009.

[10] Allam Mousa and Ahmad Hamad, "Evaluation of the RC4 Algorithm for Data Encryption," *International Journal of Computer Science & Applications*, vol. 3, no. 2, pp. 44-56, june 2006.

[11] Shilpa Gupta, Geeta Gujral, and Neha Aggarwal, "Enhanced Least Significant Bit algorithm For Image Steganography," *IJCEM International Journal of Computational Engineering & Management*, vol. 15, no. 4, pp. 40-42, July 2012.

[12] Prof. Dr. Sugiyono , *Statistika untuk penelitian*. BANDUNG, Jawa Barat: ALFABETA, 2011.