

Steganografi Citra Menggunakan Metode *Least Significant Bit (LSB)* Dan *Linear Congruential Generator (LCG)*

Andre Hernandes^{*1}, Hartini², Dewi Sartika³

^{1,3}Program Studi Informatika Universitas Indo Global Mandiri, Palembang

²Program Studi Teknik komputer AMIK Sigma Palembang

e-mail: ^{*1}andrehernandes4@gmail.com, ²arpi.hatini.my@gmail.com,

³dewi.sartika@uigm.ac.id

Abstrak

Steganografi merupakan sebuah cara yang digunakan untuk menyembunyi-kan pesan rahasia dari orang yang tidak berhak mengetahuinya. Pada penelitian ini penulis menggunakan salah satu metode steganografi yaitu metode *least significant bit (LSB)* untuk menyisipkan bit-bit pesan rahasia berupa teks kedalam citra digital RGB berekstensi file bitmap, dengan cara menggabungkan metode *LSB* dan *linear congruential generator (LCG)* untuk membangkitkan bilangan acak dari posisi pixel yang akan disisipkan pesan rahasia. Hasil dari penelitian ini penulis berhasil membangun aplikasi steganografi dengan bahasa pemrograman java dan menguji kualitas stego image yang menghasilkan nilai rata-rata *Peak Signal to Noise Ratio (PSNR)* yang mencapai 51 dB. Dari penilaian ini, disimpulkan bahwa stego image yang dihasilkan dalam kualitas baik dan tidak mengalami perubahan yang signifikan.

Kata kunci—*LCG, LSB, PSNR, Steganografi*

Abstract

Steganography is a method used to hide secret messages from people who are not entitled to know it. In this study the author uses one of the steganography methods, the *least significant bit method (LSB)* to insert bits of secret message in the form of text into digital images RGB bitmap file extension, by combining the *LSB* method and *linear congruential generator (LCG)* to generate random numbers from the position of the pixel that will be inserted a secret message. The results of this study the authors managed to build steganography applications with the Java programming language and test the quality of stego image that produces an average value *Peak Signal to Noise Ratio (PSNR)* that reach 51 dB. From this assessment, it was concluded that stego image produced in good quality and did not experience significant changes.

Keywords—*LCG, LSB, PSNR, Steganography*

1. PENDAHULUAN

Informasi merupakan bagian yang sangat penting sekarang ini. Seiring dengan kemudahan dalam mendapatkan informasi sehingga ini menjadi ancaman terhadap keamanan informasi yang dibutuhkan semakin besar, terutama untuk informasi yang dirahasiakan. Permasalahan tersebut dapat diatasi dengan teknik penyembunyian data. Teknik penyembunyian data yang terkenal adalah steganografi. Kata steganografi berasal dari bahasa Yunani, yaitu dari kata *Stegos* yang berarti tersembunyi (*covered*) dan *Grphein* yang berarti tulisan (*writing*). Steganografi di dunia modern biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam suatu arsip citra digital, audio, atau video. Steganografi adalah suatu

teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya [1].

Teknik steganografi ini telah banyak digunakan dalam strategi peperangan dan pengiriman sandi rahasia sejak jaman Yunani. Penguasa Yunani dalam mengirimkan pesan rahasia menggunakan kepala budak atau prajurit sebagai media. Dalam hal ini, rambut sudah dibotaki, lalu pesan ditulis pada kulit kepala budak. Ketika rambut budak tumbuh, budak tersebut diutus untuk membawa pesan rahasia dikepalanya. Cerita lain tentang steganografi datang juga dari sejarawan Yunani, Herodotus, yaitu dengan cara menulis pesan pada kayu yang ditutup dengan lilin. Demeratus, seorang Yunani yang akan mengabarkan berita kepada Sparta bahwa Xerxes bermaksud menyerbu Yunani. Agar tidak diketahui pihak Xerxes, Demeratus menulis pesan dengan cara mengisi tabung kayu dengan lilin dan menulis pesan dengan mengukirnya pada bagian bawah kayu, lalu papan kayu tersebut dimasukkan ke dalam tabung kayu, kemudian tabung kayu ditutup dengan lilin [2].

Media yang digunakan umumnya merupakan suatu media yang berbeda dengan media pembawa informasi rahasia. Dalam penelitian ini media penampung yang digunakan adalah citra digital image dan media yang disembunyikan data berupa *file* teks, dimana disinilah fungsi dari teknik steganografi yaitu sebagai teknik penyamaran menggunakan media lain yang berbeda sehingga informasi rahasia dalam media awal tidak terlihat secara jelas. Steganografi biasanya sering disalah artikan dengan kriptografi karenanya keduanya sama-sama bertujuan untuk melindungi informasi yang berharga. Perbedaan yang mendasar antara keduanya yaitu steganografi berhubungan dengan informasi tersembunyi sehingga tampak seperti tidak ada informasi tersembunyi sama sekali. Jika seseorang mengamati obyek yang menyimpan informasi tersembunyi tersebut, maka dia tidak akan menyangka bahwa terdapat pesan rahasia dalam obyek tersebut, dan karenanya dia tidak akan berusaha memecahkan informasi dari obyek tersebut sedangkan kriptografi informasi tampak jelas sehingga seseorang tertarik untuk memecahkan informasi yang tersembunyi.

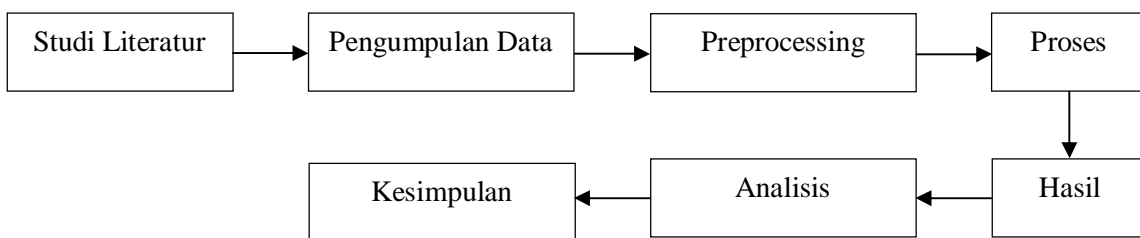
Sebagai pertimbangan dalam penelitian ini akan dicantumkan penelitian terdahulu yang dilakukan oleh Haikal Nando Winata dan Raja Nasrul Fuad dengan judul konsep penyandian *file* jpeg dengan menggunakan metode LSB dapat ditarik kesimpulan bahwa aplikasi steganografi dengan implementasi metode LSB dapat digunakan dengan baik dalam melakukan penyembunyian pesan (teks) [3]. Penelitian serupa juga pernah dilakukan oleh Endang Ratnawati Djuwitaningrum dan Melisa Apriyani dengan judul *Text Message Steganograph Using Least Significant Bit Method and Linear Congruential Generator Algorithm* telah di presentasikan penyisipan pesan teks ke dalam sebuah citra warna 24 bit menggunakan metode LSB 2 bit dan algoritma LCG, dan dalam mengimplementasikan algoritma LCG untuk membangkitkan bilangan acak semu, nilai konstanta m sebaiknya tidak sama dengan jumlah *pixel cover image*, sehingga nilai ini yang juga merupakan nilai *stego key* akan lebih tahan terhadap serangan *brute force attack* [4].

Semakin pentingnya nilai dari sebuah informasi, maka semakin berkembang pula metode-metode yang dapat digunakan untuk melakukan steganografi, mulai dari teknik *transformation*, *redundant pattern encoding*, *spread spectrum method* dan *least significant bit* (LSB). Pada penelitian ini *Least significant bit* dipilih sebagai metode yang akan digunakan, hal ini dikarenakan metode LSB merupakan metode yang paling sederhana dengan menyisipkan informasi ke dalam bit rendah atau paling kanan pada data pixel yang menyusun file tersebut dan format yang mempunyai nilai *bits redundancy tinggi*. *Bit redundancy* adalah bit yang dapat dirubah tanpa merubah banyak karakteristik *file* secara keseluruhan, sehingga meminimalisir perubahan yang signifikan pada *pixel file* citra yang telah disisipkan informasi rahasia. Penggunaan LSB tidaklah cukup untuk menyembunyikan informasi rahasia ini dikarenakan banyaknya *tools open source* yang dapat mengekstrak pesan dan informasi yang disembunyikan

pada media citra digital. Sehingga pada penelitian ini untuk menambah keamanan dari data yang akan disisipkan pada citra maka ditambah dengan menggunakan metode pembangkit bilangan acak *Pseudorandom Number Generator* (PRNG) salah satu metode yang digunakan adalah *Linear Congruential Generator* (LCG) untuk membangkitkan bilangan acak sebagai penentu posisi pixel yang akan disisipkan pesan rahasia. Berdasarkan permasalahan diatas penulis merumuskan judul “Steganografi Citra Menggunakan Metode *Least Significant Bit* (LSB) dan *Linear Congruential Generator* (LCG)”.

2. METODE PENELITIAN

2.1 Tahapan Penelitian



Gambar 1. Tahapan Penelitian

2.1.1 Pengumpulan Data

Berikut teknik pengumpulan data yang dilakukan didalam penelitian ini:



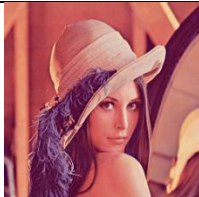

a. Studi literatur

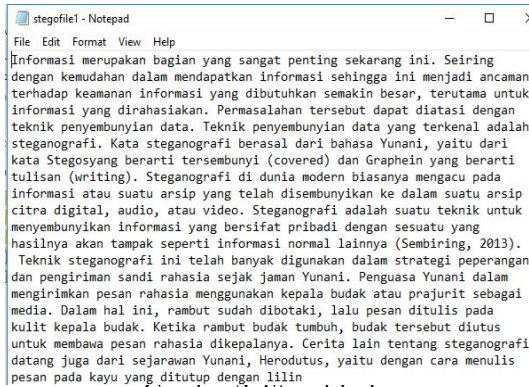
Tahap pengumpulan data berdasarkan sumber-sumber literatur seperti jurnal, buku-buku, artikel, paper, makalah yang berhubungan dengan pembuatan aplikasi steganografi citra.

b. Data Sekunder

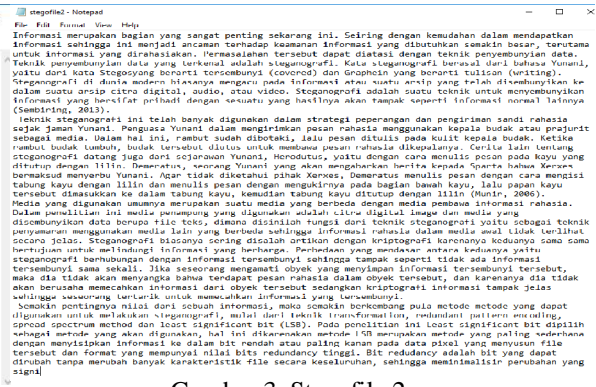
Data yang digunakan dalam penelitian ini menggunakan data sekunder berupa 4 buah citra RGB (*Red, Green, Blue*) 24 bit berformat *.bmp (bitmap) dengan 3 ukuran citra yang digunakan yaitu 100 x 100 pixel, 500 x 500 pixel dan 1000 x 1000 pixel. Pengumpulan data sekunder menggunakan instrumen penelitian internet, jadi data citra yang digunakan ini didapatkan dari web di internet melalui mesin pencarian google dengan kata kunci “24 bit bitmap image”.

Tabel 1. Citra *Bitmap* 24 bit RGB

 Baboon.bmp	 Tiger.bmp
 Lenna.bmp	 Pepper.bmp



Gambar 2. Stegotitle 1



Gambar 3. Stegotitle 2

2.1.2 Metode Pengembangan Aplikasi

Metode pengembangan perangkat lunak yang di pakai adalah RUP (*Rational Unified Process*). (Rosa A.S dan M.shalahuddin, 2015) [5] *Rational Unified Process* atau dikenal juga dengan proses *iterative* dan *Incremental* merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara *iterative* (berulang) dan *incremental* (bertahan dengan progress menaik). RUP memiliki 4 (empat) buah tahap atau fase yang dapat dilakukan pula secara iteratif yaitu *inception, elaboration, construction, transition*.



Sumber: (Rosa A.S dan M.Shalahuddin, 2015) [5]

Gambar 4. Metode RUP

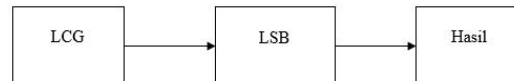
2.2 Deskripsi Umum Aplikasi

Preprocessing

Proses penyisipan pada penelitian ini dimulai dengan preprocessing citra digunakan untuk menormalisasi citra (*resize*) menjadi ukuran yang telah ditentukan pada batasan masalah penelitian menjadi 100 x 100 pixel, 500 x 500 pixel dan 1000 x 1000 pixel dengan menggunakan *tools Paint* pada *dekstop windows*.

Alur Proses

Alur proses keseluruhan aplikasi steganografi metode LSB dan LCG dimulai dengan cara memasukkan citra penampung (*cover image*) dan memasukkan pesan rahasia yang akan disisipkan. Proses dimulai dengan mengacak dengan metode LCG untuk menentukan posisi *pixel* yang akan disisipkan pesan rahasia yang kemudian akan disisipkan dengan metode LSB pada tiap 1 bit RGB terakhir *pixel* yang telah ditentukan. Hasil citra yang disisipkan pesan rahasia ini sebut dengan *stego image*.



Gambar 5. Alur Proses

2.3 Metode Linear Congruential Generator

Algoritma *Linear Congruential Generator* (LCG) [6], digunakan untuk menghasilkan bilangan acak semu, yang diberikan dalam bentuk persamaan sebagai berikut:

$$X_i = (a X_{i-1} + b) \bmod m \quad (1)$$

Keterangan:

- X_{i-1} = bilangan acak sebelumnya
- X_i = bilangan acak ke- i
- a = konstanta pengali
- b = konstanta kenaikan (penambah)
- m = konstanta modulus

Persamaan 1 memiliki nilai awal X_0 sebagai kunci pembangkit atau sering juga disebut umpan (*seed*). X_0 merupakan bilangan bulat lebih besar atau sama dengan nol dan lebih kecil dari m . LCG mempunyai periode tidak lebih besar dari m dan akan mempunyai periode penuh jika memenuhi syarat sebagai berikut [2]:

1. b relatif prima terhadap m .
2. $(a - 1)$ dapat dibagi dengan semua faktor prima dari m .
3. $(a - 1)$ adalah kelipatan 4 jika m adalah kelipatan 4.
4. $m > \max(a, b, X_0)$.
5. $a > 0, b > 0$

Algoritma LCG mempunyai periode yang tidak lebih besar dari *modulus* (m), *modulus* ini merupakan ambang batas maksimum dalam pengacakan bilangan. Semakin besar ukuran citra yang akan disisipi pesan, semakin besar pula periodik LCG yang dapat dibangkitkan, sehingga semakin banyak pula pesan yang bisa disisipkan.

Tujuan LCG pada citra adalah untuk mengacak posisi penyisipan pada *pixel* yang akan disisipkan pesan rahasia (*Secret Message*). Citra yang dimasukkan akan terlebih dahulu diubah kedalam matriks, kemudian menghitung jumlah *pixel* yang dibutuhkan sesuai dengan banyaknya pesan rahasia yang akan disisipkan. Pengacakan menggunakan metode LCG (*Linear Congruential Generator*) untuk menentukan posisi *pixel* dalam melakukan proses penyisipan pesan. Berikut *Pseudocode* LCG untuk membangkitkan bilangan acak:

```
Program Pengacakan_LCG;  
DEKLARASI  
i,a,b,m : integer;  
x[i] : integer;  
ALGORITMA  
{  
    For i ← 0 to m do i++  
        x[i] = (a*(x[i]-1)+b)mod(m)  
    End for  
}
```

Pada ilustrasi ini ditentukan bahwa nilai konstanta $m= 36$, $a = 13$, $X_0= 1$ dan $b = 35$ yang memenuhi syarat periode penuh. Dengan mensubstitusi nilai konstanta $m =$ jumlah seluruh *pixel*, konstanta a dan b dihitung agar dapat menghasilkan bilangan acak sebanyak *modulus* (m).

Setelah itu dimasukkan kedalam persamaan $X_i = (13x_{i-1} + 35) \bmod 36$ maka akan didapat angka acak yang dapat terlihat pada Tabel 2:

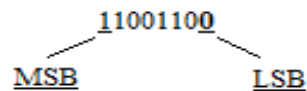
Tabel 2. Perhitungan Angka Semu Acak

i	X_i	12	25	25	24
0	1	13	0	26	23
1	12	14	35	27	10
2	11	15	22	28	21
3	34	16	33	29	20
4	9	17	32	30	7
5	8	18	19	31	18
6	31	19	30	32	17
7	6	20	29	33	4
8	5	21	16	34	15
9	28	22	27	35	14
10	3	23	26	36	1
11	2	24	13		

Dari Tabel 2 perhitungan angka semu acak yang dihasilkan, bilangan acak berulang lagi pada $i = 36$. Sehingga *modulus* (m) sebagai batas maksimum dalam pengacakan angka semu acak.

2.4 Metode Least Significant Bit

Salah satu metode untuk menyembunyikan informasi adalah *Least Significant Bit* (LSB) pada data citra. Metode penyisipan ini didasarkan pada kenyataan bahwa bit paling signifikan dalam foto dapat dianggap *noise* acak, dan akibatnya seseorang menjadi tidak responsif terhadap perubahan pada gambar. *Least significant bit* (LSB) adalah jenis yang paling umum digunakan dalam skema penyisipan yang digunakan saat ini dalam steganografi digital. Metode ini mungkin adalah cara termudah menyembunyikan informasi dalam gambar dan namun sangat efektif. Menyisip pesan rahasia yang tersembunyi dengan mengubah sedikit yang signifikan dalam lapisan tertentu dari *file* gambar (Ahmed Laskar and Hemachandran, 2012).



Gambar 6. Posisi MSB dan LSB

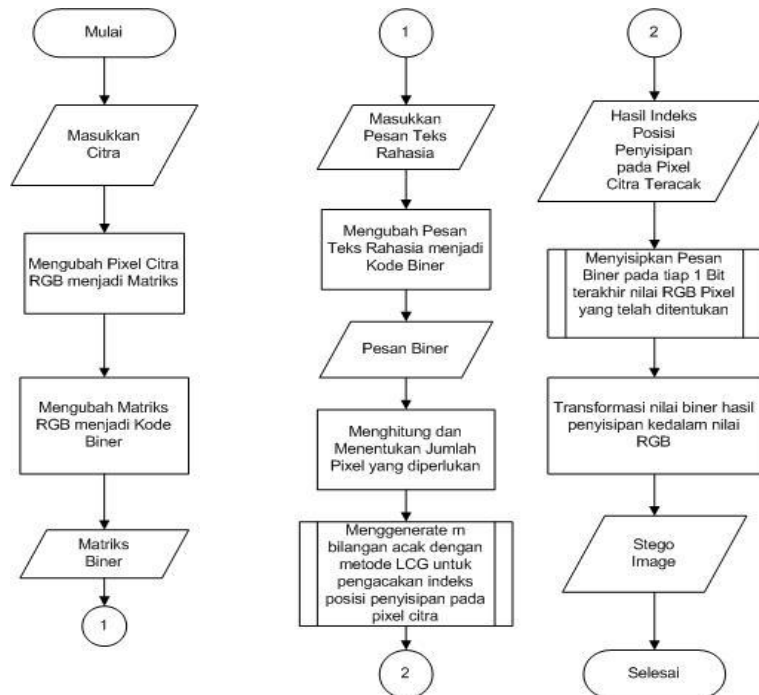
Setelah proses LCG atau pengacakan selanjutnya proses penyisipan pesan rahasia (*Secret Message*) menggunakan metode LSB (*Least Significant Bit*) yang menyisipkan pesan rahasia kedalam 1 bit terakhir tiap-tiap *pixel* RGB. Bit-bit dari pesan rahasia terlebih dahulu diubah kedalam bentuk kode ASCII atau biner dan matriks *pixel* yang telah ditentukan juga diubah kedalam bentuk kode ASCII atau biner, barulah proses penyisipan dimulai dengan menyisipkan bit-bit pesan rahasia kedalam 1 bit terakhir tiap *pixel* RGB yang telah ditentukan.

Berikut algoritma LSB untuk proses penyisipan:

1. Inisialisasi awal: proses penyisipan hanya menyisipkan pesan ke dalam tiap 1 bit terakhir (LSB) pada citra RGB, ukuran citra \geq Pesan yang disisipkan
Output: Stego image
2. Masukkan Citra RGB
3. Ubah Citra RGB menjadi Matriks kemudian ubah Matriks ke biner
4. Masukkan Pesan teks \leq Citra RGB
5. Ubah Pesan teks menjadi biner

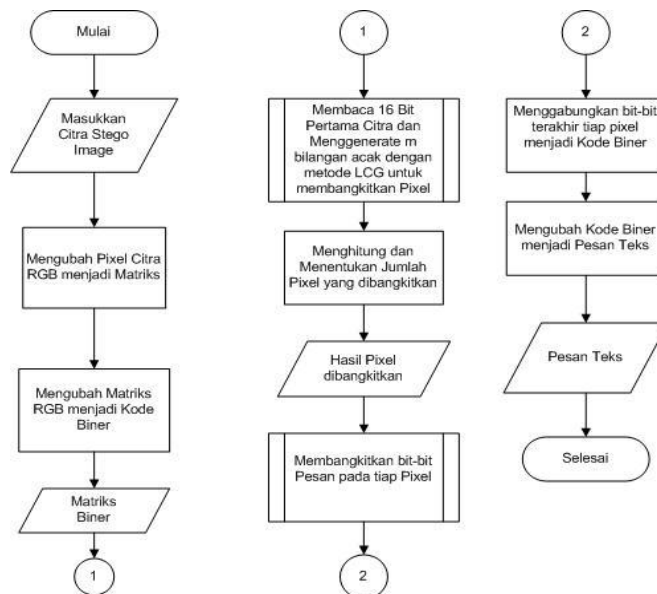
6. Sisipkan setiap 1 bit pesan teks untuk dimasukkan kedalam 1 bit terakhir pada setiap nilai citra RGB
7. Mentransformasikan nilai biner hasil proses penyisipan kedalam nilai citra RGB
8. Simpan citra yang sudah disisipkan pesan menjadi *Stego image*

2.5 Flowchart Penyisipan



Gambar 7. Flowchart Penyisipan

2.6 Flowchart Ekstraksi



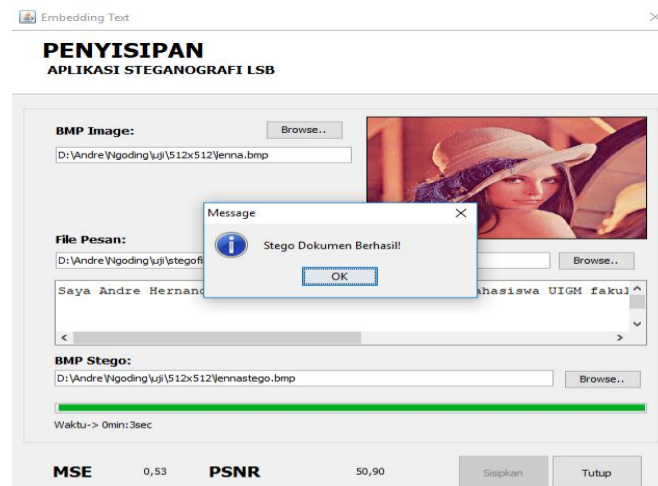
Gambar 8. Flowchart Ekstraksi

2.7 Implementasi



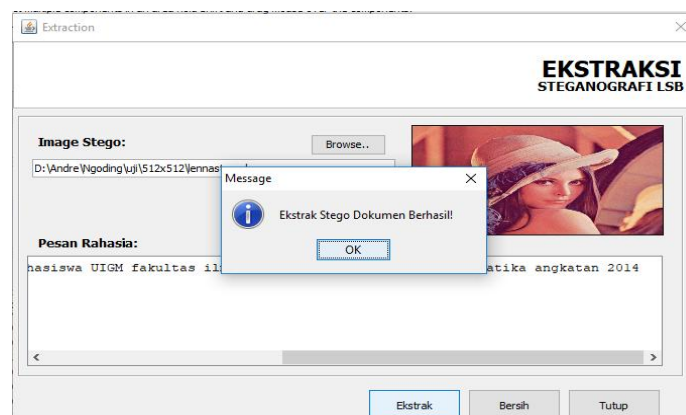
Gambar 9. Implementasi Menu Utama

Tampilan Menu Utama terdiri dari 4 buah tombol Penyisipan, Ekstraksi, Petunjuk dan Keluar



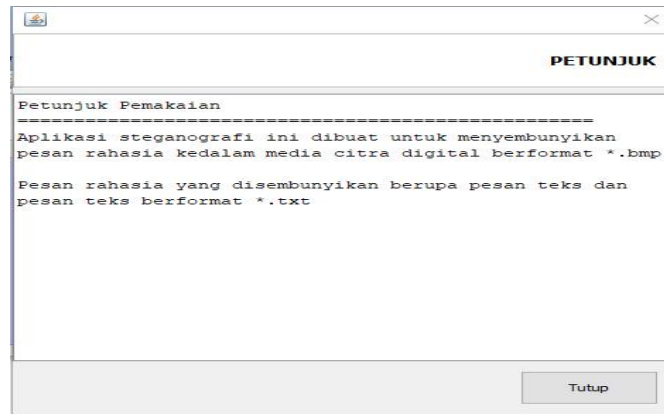
Gambar 10. Implementasi Menu Penyisipan

Pada menu penyisipan masukkan citra penampung berformat .bmp, pesan teks berformat .txt yang akan disembunyikan dan lokasi penyimpanan *file* hasil *stego image* kemudian pilih sisipkan. Setelah diproses akan tampil lama waktu proses penyisipan, MSE dan PSNR.



Gambar 11. Implementasi Menu Ekstraksi

Pada menu ekstraksi masukkan *stego image* berformat .bmp yang akan diekstraksi kemudian pilih ekstrak. Setelah itu akan tampil pesan rahasia yang tersembunyi didalam *stego image*.



Gambar 12. Implementasi Menu Petunjuk

Pada menu petunjuk berisi petunjuk penggunaan aplikasi.

3. HASIL DAN PEMBAHASAN

Citra yang telah disisipkan sebanyak 24 citra bitmap dengan ukuran 100x100, 500x500, 1000x1000 *pixel* dan berhasil menyisipkan pesan teks berformat .txt dengan nama *file* stegofile1.txt dan stegofile2.txt. Pengujian dilakukan dengan membandingkan ukuran *file*, pengujian *recovery* untuk mengukur pesan yang disembunyikan harus dapat diungkapkan kembali dan menghasilkan pesan yang sama dengan yang disisipkan dan pengujian PSNR untuk membandingkan kualitas citra sebelum dan sesudah disisipkan pesan apakah terdapat perubahan (*noise*) yang signifikan pada *pixel* citra setelah dilakukan proses penyisipan bit-bit pesan.

3.1 Pengujian

Pengujian yang penulis lakukan yaitu pengujian sistem menggunakan *Block-Box testing*, pengujian perbandingan ukuran *file*, pengujian *recovery* dan pengujian PSNR.

3.1.1 Pengujian Sistem

Black-Box Testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

3.1.2 Pengujian Perbandingan Ukuran *File*

Pengujian perbandingan antara ukuran *file cover image* dan *stego image* dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Ukuran *File*

No.	<i>Cover Image</i>	Pesan Rahasia	Ukuran <i>File Cover Image</i>	Ukuran <i>File Stego Image</i>
1.	Baboon100.bmp	Stegofile1.txt	30,054 byte	30,054 byte
2.	Baboon500.bmp	Stegofile1.txt	750,054 byte	750,054 byte
3.	Baboon1000.bmp	Stegofile1.txt	3,000,054 byte	3,000,054 byte
4.	Lenna100.bmp	Stegofile1.txt	30,054 byte	30,054 byte
5.	Lenna500.bmp	Stegofile1.txt	750,054 byte	750,054 byte

6.	Lenna1000.bmp	Stegofile1.txt	3,000,054 byte	3,000,054 byte
7.	Pepper100.bmp	Stegofile1.txt	30,054 byte	30,054 byte
8.	Pepper500.bmp	Stegofile1.txt	750,054 byte	750,054 byte
9.	Pepper1000.bmp	Stegofile1.txt	3,000,054 byte	3,000,054 byte
10.	Tiger100.bmp	Stegofile1.txt	30,054 byte	30,054 byte
11.	Tiger500.bmp	Stegofile1.txt	750,054 byte	750,054 byte
12.	Tiger1000.bmp	Stegofile1.txt	3,000,054 byte	3,000,054 byte
13.	Baboon100.bmp	Stegofile2.txt	30,054 byte	30,054 byte
14.	Baboon500.bmp	Stegofile2.txt	750,054 byte	750,054 byte
15.	Baboon1000.bmp	Stegofile2.txt	3,000,054 byte	3,000,054 byte
16.	Lenna100.bmp	Stegofile2.txt	30,054 byte	30,054 byte
17.	Lenna500.bmp	Stegofile2.txt	750,054 byte	750,054 byte
18.	Lenna1000.bmp	Stegofile2.txt	3,000,054 byte	3,000,054 byte
19.	Pepper100.bmp	Stegofile2.txt	30,054 byte	30,054 byte
20.	Pepper500.bmp	Stegofile2.txt	750,054 byte	750,054 byte
21.	Pepper1000.bmp	Stegofile2.txt	3,000,054 byte	3,000,054 byte
22.	Tiger100.bmp	Stegofile2.txt	30,054 byte	30,054 byte
23.	Tiger500.bmp	Stegofile2.txt	750,054 byte	750,054 byte
24.	Tiger1000.bmp	Stegofile2.txt	3,000,054 byte	3,000,054 byte

Berdasarkan Tabel 3 diperoleh hasil perbandingan antara *file cover image* dan *file stego image* (*file* yang telah disisipkan pesan) bahwa antara ukuran *file coverimage* dan ukuran *file stego image* tidak terdapat perubahan yang signifikan.

3.1.3 Pengujian Recovery

Pengujian terhadap *recovey* bertujuan untuk pesan yang disembunyikan harus dapat diungkapkan kembali, Karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu pesan rahasia didalam *stego image* harus dapat diambil kembali untuk digunakan lebih lanjut.

$$\text{Akurasi} = \frac{\text{jumlah kata benar diekstraksi}}{\text{jumlah kata yang disisipkan}} \times 100\% \quad (2)$$

1. $\text{Presentase Baboon100stegofile1} = \frac{184}{184} \times 100\% = 100\%$
2. $\text{Presentase Baboon500stegofile1} = \frac{184}{184} \times 100\% = 100\%$
3. $\text{Presentase Baboon1000stegofile1} = \frac{184}{184} \times 100\% = 100\%$
4. $\text{Presentase Lenna100stegofile1} = \frac{184}{184} \times 100\% = 100\%$
5. $\text{Presentase Lenna500stegofile1} = \frac{184}{184} \times 100\% = 100\%$
6. $\text{Presentase Lenna1000stegofile1} = \frac{184}{184} \times 100\% = 100\%$
7. $\text{Presentase Pepper100stegofile1} = \frac{184}{184} \times 100\% = 100\%$
8. $\text{Presentase Pepper500stegofile1} = \frac{184}{184} \times 100\% = 100\%$
9. $\text{Presentase Pepper1000stegofile1} = \frac{184}{184} \times 100\% = 100\%$
10. $\text{Presentase Tiger100stegofile1} = \frac{184}{184} \times 100\% = 100\%$
11. $\text{Presentase Tiger500stegofile1} = \frac{184}{184} \times 100\% = 100\%$

12. $Presentase\ Tiger1000stegofile1 = \frac{184}{184} \times 100\% = 100\%$
13. $Presentase\ Baboon100stegofile2 = \frac{476}{476} \times 100\% = 100\%$
14. $Presentase\ Baboon500stegofile2 = \frac{476}{476} \times 100\% = 100\%$
15. $Presentase\ Baboon1000stegofile2 = \frac{476}{476} \times 100\% = 100\%$
16. $Presentase\ Lenna100stegofile2 = \frac{476}{476} \times 100\% = 100\%$
17. $Presentase\ Lenna500stegofile2 = \frac{476}{476} \times 100\% = 100\%$
18. $Presentase\ Lenna1000stegofile2 = \frac{476}{476} \times 100\% = 100\%$
19. $Presentase\ Pepper100stegofile2 = \frac{476}{476} \times 100\% = 100\%$
20. $Presentase\ Pepper500stegofile2 = \frac{476}{476} \times 100\% = 100\%$
21. $Presentase\ Pepper1000stegofile2 = \frac{476}{476} \times 100\% = 100\%$
22. $Presentase\ Tiger100stegofile2 = \frac{476}{476} \times 100\% = 100\%$
23. $Presentase\ Tiger500stegofile2 = \frac{476}{476} \times 100\% = 100\%$
24. $Presentase\ Tiger1000stegofile2 = \frac{476}{476} \times 100\% = 100\%$

Berdasarkan hasil pengujian akurasi dapat ditarik kesimpulan bahwa dari 24 citra yang diujikan tingkat keberhasilan aplikasi ini mampu mengekstraksi pesan yang sama dengan pesan yang disisipkan dengan tingkat keberhasillanya adalah 100%.

3.1.4 Pengujian PSNR

Setelah citra mengalami proses penyisipan, untuk mengetahui kualitas citra dilakukan perhitungan *Peak Signal to Noise Ratio* (PSNR). PSNR bertujuan untuk membandingkan kualitas citra sebelum dan sesudah disisipkan apakah terdapat perubahan (*noise*) yang signifikan pada *pixel* citra setelah dilakukan proses penyisipan bit-bit pesan. Untuk menentukan PSNR, terlebih dahulu ditentukan nilai rata-rata kuadrat *error Mean Square Error* (MSE) antara citra asli dan citra sesudah disisipkan (*stego image*) dengan menggunakan persamaan berikut:

1. Mean Square Error (MSE)

[4] MSE adalah nilai rata-rata kuadrat *error* antara citra asli (*cover image*) dengan citra hasil penyisipan (*stego image*). Perhitungan MSE menggunakan persamaan berikut:

$$MSE = \frac{1}{MN} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - K(i,j))^2 \quad (3)$$

Dimana:

- MSE = Nilai *Mean Square Error* antara *cover image* dengan *stego image*
- m = panjang citra tersebut (dalam *pixel*)
- n = lebar citra tersebut (dalam *pixel*)
- (i,j) = koordinat *pixel*
- I = *cover image*
- K = *stego image*

2. Peak to Signal Noise Ratio (PSNR)

PSNR sering dinyatakan dalam skala logaritmik, dalam decibel (dB). Menurut [7] nilai PSNR dibawah 30 dB mengindikasikan kualitas yang relatif rendah, dimana distorsi yang disebabkan penyisipan terlihat jelas. Sedangkan kualitas *stego image* yang tinggi berada pada

nilai 40 dB dan di atasnya. Menurut [8] citra tidak mengalami perubahan kualitas yang signifikan dengan nilai PSNR diatas 30 dB.

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad (4)$$

Berikut ini tabel hasil pengujian PSNR:

Tabel 4. Hasil Pengujian PSNR

<i>Output File</i>	<i>Ukuran File</i>	<i>Size File</i>	Nilai PSNR
Baboon100stegofile1.bmp	30.054 byte	3 x (100 x 100) Pixel	51,12
Baboon500stegofile1.bmp	750,054 byte	3 x (500 x 500) Pixel	51,12
Baboon1000stegofile1.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,13
Lenna100stegofile1.bmp	30.054 byte	3 x (100 x 100) Pixel	51,17
Lenna500stegofile1.bmp	750,054 byte	3 x (500 x 500) Pixel	51,20
Lenna1000stegofile1.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,20
Pepper100stegofile1.bmp	30.054 byte	3 x (100 x 100) Pixel	51,18
Pepper500stegofile1.bmp	750,054 byte	3 x (500 x 500) Pixel	51,16
Pepper1000stegofile1.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,10
Tiger100stegofile1.bmp	30.054 byte	3 x (100 x 100) Pixel	51,15
Tiger500stegofile1.bmp	750,054 byte	3 x (500 x 500) Pixel	51,15
Tiger1000stegofile1.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,04
Baboon100stegofile2.bmp	30.054 byte	3 x (100 x 100) Pixel	51,12
Baboon500stegofile2.bmp	750,054 byte	3 x (500 x 500) Pixel	51,13
Baboon1000stegofile2.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,19
Lenna100stegofile2.bmp	30.054 byte	3 x (100 x 100) Pixel	51,14
Lenna500stegofile2.bmp	750,054 byte	3 x (500 x 500) Pixel	51,16
Lenna1000stegofile2.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,19
Pepper100stegofile2.bmp	30.054 byte	3 x (100 x 100) Pixel	51,14
Pepper500stegofile2.bmp	750,054 byte	3 x (500 x 500) Pixel	51,11
Pepper1000stegofile2.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,11
Tiger100stegofile2.bmp	30.054 byte	3 x (100 x 100) Pixel	51,14
Tiger500stegofile2.bmp	750,054 byte	3 x (500 x 500) Pixel	51,15
Tiger1000stegofile2.bmp	3,000,054 byte	3 x (1000 x 1000) Pixel	51,07

Berdasarkan Tabel 4 hasil uji perbandingan penilaian kualitas citra digital yang dilakukan dengan cara menghitung nilai *Peak Signal To Noise Rational* (PSNR). Perbandingan, *24 stego image* mendapatkan nilai PSNR yang tinggi berada diatas 51 dB, disimpulkan bahwa citra berkualitas baik dan tidak mengalami perubahan kualitas citra yang signifikan, dikarenakan hanya mengubah bit terendah pada pixel.

4. KESIMPULAN

Berdasarkan hasil penelitian tentang steganografi citra menggunakan metode *Least Significant Bit* (LSB) dan *Linear Congruential Generator* (LCG) untuk melakukan penyisipan pesan rahasia kedalam citra maka dapat ditarik kesimpulan:

1. Berhasil menerapkan metode *Least Significant Bit* (LSB) untuk penyisipan pesan teks dan metode *Linear Congruential Generator* (LCG) sebagai pengacakan indeks posisi *pixel* tempat disisipkannya pesan teks pada aplikasi steganografi.
2. Aplikasi steganografi ini membuktikan bahwa metode LSB dan LCG yang digunakan menghasilkan *stego image* dengan kualitas yang baik dan dapat dibuktikan dengan hasil pengujian nilai PSNR yang mencapai 51 dB.

5. SARAN

Dengan melihat hasil yang dicapai pada penelitian ini ada beberapa hal yang penulis sarankan untuk pengembangan aplikasi selanjutnya yaitu:

1. Diharapkan untuk pengembangan selanjutnya dapat menggunakan media citra digital lainnya sebagai penampung dan dapat menyimpan pesan dengan berbagai macam format lainnya.
2. Diharapkan dapat dikembangkan lebih lanjut agar *stego image* tetap tangguh menghadapi rotasi, *resize*, *cropping* dan manipulasi data lainnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada kedua orang tuaku dan kedua dosen pembimbingku yang telah memberikan dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] S. Sembiring. 2013, "*Perancangan Aplikasi Steganografi untuk Menyisipkan Pesan Teks pada Gambar Dengan Metode End of File*," *Pelita Informatik, Budi Darma*, Vol. IV.
- [2] R. Munir. 2006, *Pengolahan Citra Digital*, Informatika, Bandung.
- [3] H. Nando Winata and R. Nasrul Fuad, 2017. "*Konses Penyandian File JPEG Menggunakan Metode LSB*," *Jurnal Nasional Informatika dan Teknologi. Jaringan*, Vol. 1, No. 2.
- [4] E. R. Djuwitaningrum and M. Apriyani, 2016. "*Teknik Steganografi Pesan Teks Menggunakan Metode Least Significant Bit dan Algoritma Linear Congruential Generator (Text Message Steganography Using Least Significant Bit Method and Linear Congruential Generator Algorithm)*," Vol. IV, No. November, pp. 79–85.
- [5] Rosa A.S dan M.Shalahuddin, 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Informatika, Bandung.
- [6] S. . Park and K. . Miller, 1988. "*Random Number Generators: Goods One Are Hard To Find*, *Communication of The ACM*," Vol. 31, No. 10, pp. 1192–1201.

- [7] A. Cheddad, J. Condell, K. Curran, and P. M. Kevvit, 2010. “*Digital Image Steganography Survey and Analysis of Current Methods*,” Vol. 90.
- [8] D. Sartika, “*Pengembangan Perangkat Lunak Penyembunyian Pesan Terenkripsi Menggunakan Algoritma Mars pada Citra Digital Dengan Metode Adaptif*,” *Jurnal Ilmiah Informatika Global*, vol. 7, No. 1, pp. 1–6, 2016.