REPOSITOR, Vol. 2, No. 11, November 2020, Pp. 1503-1512
ISSN : 2714-7975
E-ISSN : 2716-1382

Steganografi Dalam File Citra Menggunakan Fungsi Hash dan Metode MLSB

Alfian yuniarto[1], Eko budi cahyono[2], Agus Eko Minarno[3]
[123]Universitas Muhammadiyah Malang
alfianyuniarto@.webmail.umm.ac.id[*1], ekobudi@umm.ac.id[2], agoes.minarno@gmail.com[3]

Abstrak

Semakin berkembangnya jaringan internet dapat mempermudah proses pertukaran informasi. Ketika suatu informasi bersifat rahasia ditransmisikan lewat jaringan komputer/internet, maka diperlukan jaminan bahwa informasi tersebut hanya dapat diakses oleh pihak yang berkepentingan. Pernaslahan tersebut dapat di selesaikan dengan salah satu cara yaitu penyembunyian pesan dalam pengiriman adalah merubah data menjadi yang tidak dimengerti dengan cara di sisipkan ke dalam pixel yang dinamakan teknik steganografi. Pada Teknik steganografi ini salah satu metode yang dapat di gunakan adalah metode Modified Least Significant Bit (MLSB). MLSB merupakan pengembangan dari metode LSB yaitu dengan memodifikasi data dengan cara merubah dari 8 bit menjadi 5 bit. Selain metode MLSB tersebut kita juga dapat menggunakan fungsi hash MD5 sebagai tambahan pengamanan. Dengan menggunakan fungsi hash MD5 serta dengan menggunakan metode MLSB (modified least significat bit) ini kualitas dari file citra yang di hasilkan menunjukkan nilai baik karena dari rata - rata nilai dari gambar sampel dengan nilai MSE 0.0037 serta PSNR di atas 60db.

Kata kunci *: Steganografi, Modified Least Significant Bit , MD5, PRNG.*

Abstract

Nowadays, people can be easy to acces the internet as changing information. People who wants to transfer information (undercover) through the internet, then the internet needs a guarantee that can only be accesed by profesional person. In sending hiden messages, we can changes the data that is not understood by inserting into pixels which is called by steganography techniques. Therefore, this steganography techniques used the modified least significant bit (MLSB) method. MLSB is the development of the LSB method. LSB method is modifying data be changing from 8 bit to 5 bit. In addition, we can also use the hash MD5 function as an ancillary security. Using both hash MD5 function and MLSB method showed a good result from quality of images file which hasthe average value of sample images with MSE 0,0037 balues and PSNR above 60dB.

Keywords *: Steganography, Modified Least Significant Bit, Message-Digest algortihm, PRNG.*

1. Pendahuluan

Semakin berkembangnya jaringan internet dapat mempermudah proses pertukaran informasi. Ketika suatu informasi bersifat rahasia ditransmisikan lewat jaringan komputer/internet, maka diperlukan jaminan bahwa informasi tersebut hanya dapat diakses oleh pihak yang berkepentingan [1]. Dari kasus tersebut maka dalam proses pertukaran suatu informasi harus menjadi topik penelitian yang harus di kaji lebih mendalam. Dalam bidang kemanan suatu data ada banyak cara yang di lakukan, Salah satunya adalah penyembunyian pesan dalam pengiriman adalah merubah data menjadi yang tidak dimengerti dengan cara di sisipkan ke dalam pixel yang dinamakan teknik steganografi [2].

Di dalam teknik stegaografi salah satunya adalah teknik *Least Significant Bit (LSB)*, yaitu data akan di sisipkan ke media dalam bit terakhir dari pixel[3]. Seiring dengan perkembangan teknolongi maka banyak yang telah mengembangkan metode ini. Salah satunya metode LSB ini di kembangkan menjadi *Modified Least Significant Bit* (*MLSB)* untuk meningkatkan dalam segi kemanan dan kapasitas. MLSB merupakan pengembangan dari metode LSB yaitu dengan

memodifikasi data dengan cara merubah dari 8 bit menjadi 5 bit, yang nantinya akan di lakukan proses penyisipan ke dalam file citra[4].

Ada beberapa metode yang di gunakan untuk menjaga keamanan password salah satunya mengunakan fungsi hash.Dalam isitilah umum, tujuan utama fungsi hash adalah integritas data. Sementara itu penjelasan dari fungsi tersebut merupakan sebuah fungsi yang menerima inputan berbentuk string yang memiliki panjang sembarang dan akan di lakukan proses konversi sehingga akan menghasilkan output dengan panjang tetap (berukuran lebih kecil dari ukuran awal[6].

Berdasarkan dari berberapa uraian referensi di atas maka penulis menerapkan skema dengan menerapakan fungsi hash MD5 yang nantinya di gunakan pada pengamanan password yang di gunakan. Setelah di dapatkan nilai hash dari password maka proses selanjutnya dilakukan proses pengacakan pada hasil proses hashing MD5 tersebut menggunakan PRNG *(pseudo-random-number-generator*). Penulis nantiya juga menggunakan algoritma MLSB (*Metode Modified Least Significant Bit*) yang nantinya diyakini lebih bagus untuk proses pengamanan suatu data serta tidak akan menghasilkan banyak eror pada image, terbukti dengan hasil dari pengujian sebelumnya [8] yang menghasilkan PSNR di atas 40dB dan dengan adanya proses hashing pada password serta proses pengacakan yang di nilai lebih baik dalam proses pegamanan data.

2. Metode Penelitian

Dalam metode penelitian ini akan menjelaskan bagaimana skema yang di akan di rancang berdasarkan beberapa algoritma yang akan di gunakan yang nantinya bertujuan bagaimana kecocokan algoritma penulis untuk proses enkripsi dan dekripsi serta memenuhi aspek keamanan pada gambar.

2.1. Analisa Metode yang digunakan

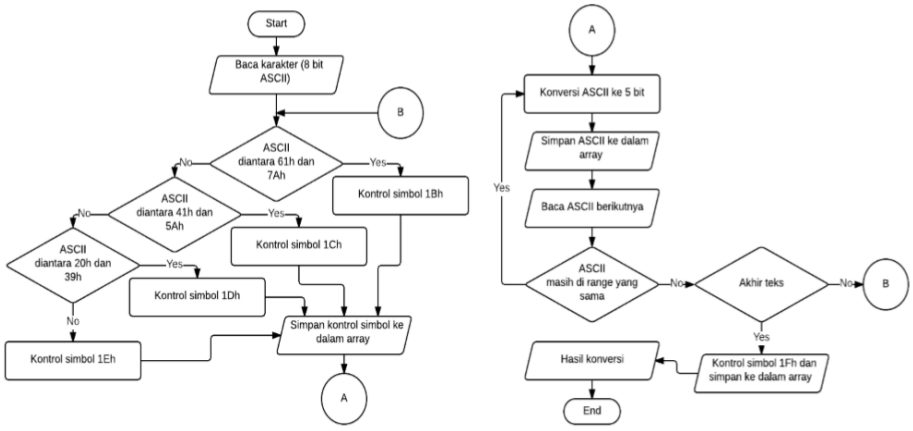
Pada tahap ini berisi tentang analisa bagaimana skema yang di akan di rancang berdasarkan beberapa referensi yang telah di jelaskan pada bab sebelumnya yang nantinya bertujuan bagaimana kecocokan algoritma penulis untuk proses enkripsi dan dekripsi serta memenuhi aspek keamanan pada gambar.

2.1.1 Fungsi Hash MD5

Hash atau proses hashing ini adalah proses perubahan suatu data menjadi nilai yang lain dengan Panjang tertentu, namun pada fungsi hash MD5 sendiri akan menerima inputan sembarang dan akan berubah menjadi message digest yang panjangnya tetap yaitu menjadi 128 bit. Penulis menggunakan algoritma hash MD5 ini karena pada dasarnya adalah fungsi satu arah dan hal tersebut cocok di gunakan untuk pengamanan suatu password yang akan di gunakan oleh penulis dalam penelitian ini dan di gunakan juga untuk proses selanjutnya.

2.1.2 Penggunaan Algoritma MLSB

Pada algoritma ini bilangan yang awalnya terdiri dari 8 bit seperti pada LSB kemudian di lakukan konversi menjadi bilangan 5 bit. Yang selanjutnya hasil konversi tersebut di sisipkan menggunakan metode LSB atau tetap di sisipkan di akhir bit. Algoritma ini dinilai bekerja dengan baik dalam hal penyembunyian suatu pesan, karena pada penerapannya pada stego image dan cover image akan terlihat sama pada penglihatan manusia. Berikut merupakan flowchart algoritma *Modified Least Significant Bit* (MLSB).



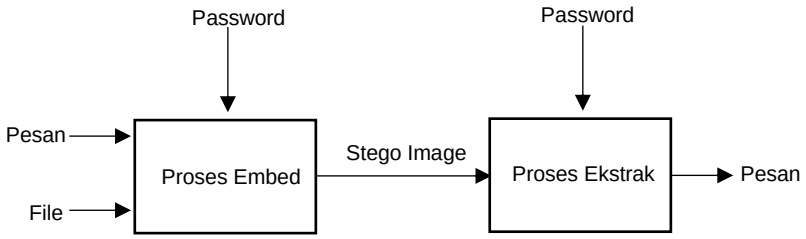
Gambar 2.1 flowchart algoritma *Modified Least Significant Bit*

Berikut merupakan flowchart dari *Modified Least Significant Bit* (MLSB) sebagaimana dapat dilihat dari gambar 2.1

1. Small Letter (61h Ŕ 7Ah) Jika tipe karakter adalah small letter (a - z) , maka harus menempatkan 5 bit diawal yaitu 1Bh, kemudian konversikan menjadi 5 bit.
2. Capital Letter (41h - 5Ah) Jika tipe karakter adalah capital letter (A - Z), maka harus menempatkan 5 bit diawal yaitu 1Ch, kemudian konversi menjadi 5 bit.
3. Space Pada karakter spasi, maka harus menempatkan 5 bit yaitu 1Dh sebagai ganti dari 20h (Kode ASCII dari spasi) .
4. Number (20h - 39h) Pada tipe karakter yang berupa angka, harus menempatkan 1Eh diawal.
5. End text.

2.2 Rancangan Umum Sistem

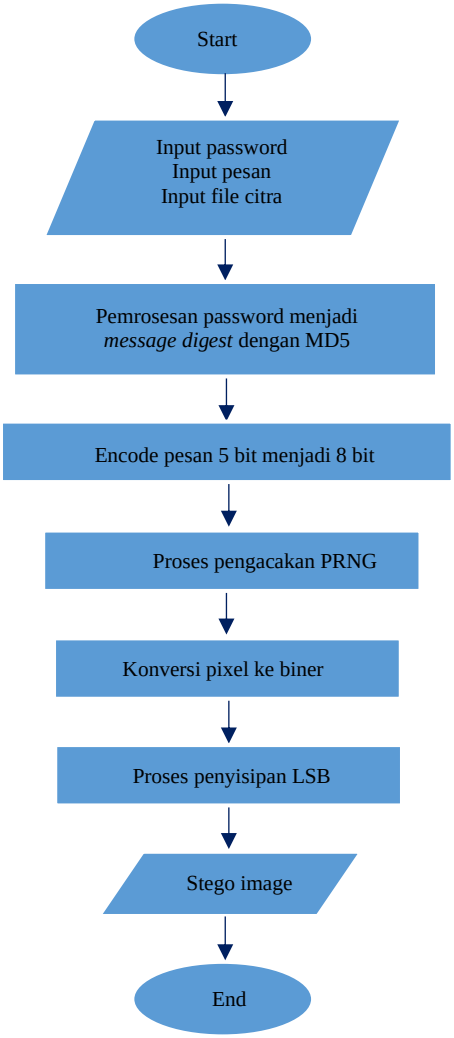
Dalam tahap ini berisi tentang bagaimana rancangan sebuah program steganografi yang bertujuan untuk menyisipkan suatu teks atau karakter dengan menggunakan algoritma dan dengan menggunakan fungsi hash pada media citra berdasarkan dari hasil Analisa algortima sebelumnya. Berikut merupakan skema secara umum yang penulis gunakan.



Gambar 2.2 Skema Umum Penyisipan Pesan

Berdasarkan pada skenario yang telah di gambarkan pada Gambar 3.1 pada awalnya inputan terdiri dari dua inputan yaitu pesan dan file citra yang di gunakan sebagai penampung pesan tersebut, setelah itu maka ada inputan lagi yaitu inputan password. Password yang di maksudkan pada proses di atas adalah sebagai kunci pada saat proses embed. Selanjutnya setelah proses embed tersebut maka di dapatkan hasil yaitu stego image adalah file citra yang telah di sisipi suatu pesan.

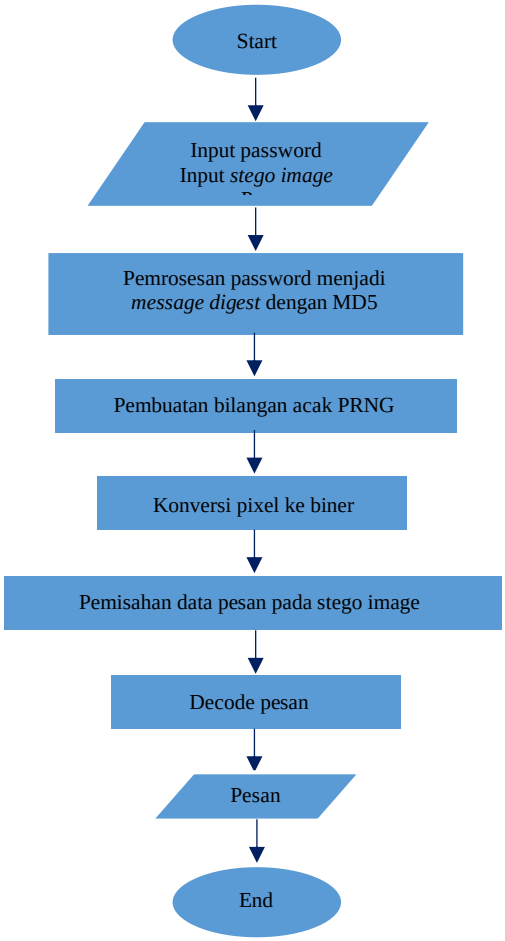
2.3 Flowchart Embed Pesan



Gambar 2.3 Flowchart Penyisipan Pesan

1. Tahap pertama pada proses ini setelah di lakukannya pemrosesan untuk password adalah proses penginputan file citra yang di gunakan sebagai citra penampung dan pesan yang akan di sisipkan itu sendiri. Pesan yang di sisipkan berupa karakter yang nantinya di lakukan konversi dengan menggunakan tabel ASCII. Sementara itu file citra akan di koversi ke dalam bilangan bit sehingga dapat di jadikan sebagai citra penampung.
2. Dalam tahap ini pesan yang sudah di dapatkan akan di lakukan proses encode dengan menggunakan algoritma *MLSB*. Di dalam algoritma ini nilai dari karakter ASCII yang pada awalnya memiliki nilai 8 bit (255 desimal) di kodekan sehingga pesan tersebut memiliki nilai 5 bit (31 desimal). proses tersebut di hasilkan berdasarkan tabel Control Symbol. .
3. Dalam proses ini byte password yang telah di dapatkan selanjutnya untuk di gunakan pada proses pengacakan menggunakan *pseudo random number generator (PRNG)*. Nilai ASCII dari setiap karakter akan di lakukan proses looping dan akan mejadi parameter untuk membentuk nilai random. Dari nilai random tersebut maka akan dijadikan sebuah alamat pada file citra.
4. Pada tahap ini yang merupakan tahap inti dari proses penyisipan pesan. Setelah proses encode pesan menjadi 5 bit selesai dan membentuk bilangan acak,maka tahap setelah itu adalah menyisipkan byte pesan ke dalam byte citra secara acak berdasarkan bilangan acak sebelumnya dengan menggunakan metode *least significant bit (LSB)*.
5. Setelah proses penyisipan selesai maka byte citra yang telah di sisipi pesan maka akan di koversi lagi kedalam pixel sehingga akan menjadi stego image.

2.4 Flowchart Ekstrak pesan



Gambar 2.4 Flowchart Ekstrak pesan

1. Untuk proses pertama di dalam Gambar 3.5 adalah menginputkan stego image dan tentunya password yang nantinya di gunakan sebagai kunci dimana untuk membuka pesan yang telah di sisipkan. Dalam stego image diatas nantinya akan di konversi ke dalam bilangan biner untuk dapat di lakukan proses pemisahan pesan.
2. Pada tahap selanjutnya adalah proses pengolahan password menggunakan MD5 sehingga menghasilkan *message digest*. Karena fungsi tersebut satu arah, maka alur proses tersebut di lakukan sama seperti yang telah di jelaskan pada gambar 3.3.
3. Di dalam tahap ketiga ini byte password yang telah di dapatkan akan di gunakan untuk proses selanjutnya yaitu pengacakan menggunakan metode *pseudo random number generator* (PRNG). Hasil proses ASCII untuk setiap karakter akan di lakukan proses looping dan akan menjadi sebuah parameter dan membentuk sebuah nilai random. Kemudian, password yang di gunakan pada proses embed dan ekstrak harus sama untuk menghasilkan nilai random yang sama. Nilai random tersebut untuk di jadikan sebagai alamat untuk mendapatkan nilai bit-bit pesan pada stego image.
4. Tahap selanjutnya bit pesan pada file citra di peroleh dengan mengambil 1 bit terakhir dari byte file citra secara urut berdasar pada alamat dari nilai random yang telah di dapatkan pada proses sebelumya. Setiap 5 bit citra yang di dapatkan, di lakukan proses kembali sehingga menjadi 8 bit. Di dalam proses itu dilakukan berulang ulang sehingga akan di dapatkan sebuah pesan yang di embed.
5. Dalam tahap ini nilai byte pesan untuk setiap karakter sudah di dapatkan, namun masih dalam encode dengan panjang 5 bit. Untuk proses selanjutnya pesan akan di lakukan proses decode yang bertujuan yaitu mengembalikan nilai ASCII setiap karakter mengunakan metode MLSB.

2.5 Landasan pengujian Hasil citra steganografi

Kualitas hasil citra yang baik adalah tujuan utama dari kesusksesan algoritma pada steganografi ini, cara yang digunakan untuk kualitas dari citra steganografi adalah dengan menggunakan rumus *Means Square Error (MSE)* dan *Peak Signal To Noise Ratio (PSNR).* pengertian dari MSE sendiri merupakan rata-rata kuadrat nilai dari kesalahan antara citra asli dengan citra yang telah di sisipi pesan. *Peak Signal to Noise Ratio* (PSNR) adalah perbandingan antara nilai maksimum dari sinyal. Satuan dari PSNR ini adalah decibel (dB). sebelum menentukan nilai PSNR, maka hal yang pertama di lakukan adalah menentukan nilai dari *Means Square Error (MSE).* Berikut merupakan rumus untuk menentukan nilai dari *MSE* dan *PSNR* :

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[I(i,j) - K(i,j) \right]^2 \quad (1)$$

$$PSNR = 20 \times log \frac{255}{\sqrt{MSE}} \quad (2)$$

Dapat dinilai apabila semakin rendah nilai MSE maka akan baik kualitas citra. Setelah di dapatkan nilai dari MSE kemudian maka akan di lakukan pengitungan untukk memperoleh nilai dari PSNR. Kebalikan dari nilai MSE, nilai PSNR ini apabila semakin besar nilai dari PSNR maka akan baik pula nilai file citra steganografi. Citra yang di nilai baik dari penghitungan PSNR ini apabila citra tersebut mempunyai nilai di atas 30dB – 40dB.

3. Hasil dan Pembahasan

Dalam tahap ini penulis akan menggunakan beberapa sampel untuk file citra yang di gunakan dalam proses penyisispan teks, dari beberapa gambar tersebut gambar yang di gunakan penulis bersifat grayscale atau hitam putih. Sedangkan beberapa gambar tersebut semuanya berformat *png*, bmp maupun jpg. berikut merupakan proses bagaimana pengujian embed pesan seperti yang telah di rancang pada bab sebelumnya.

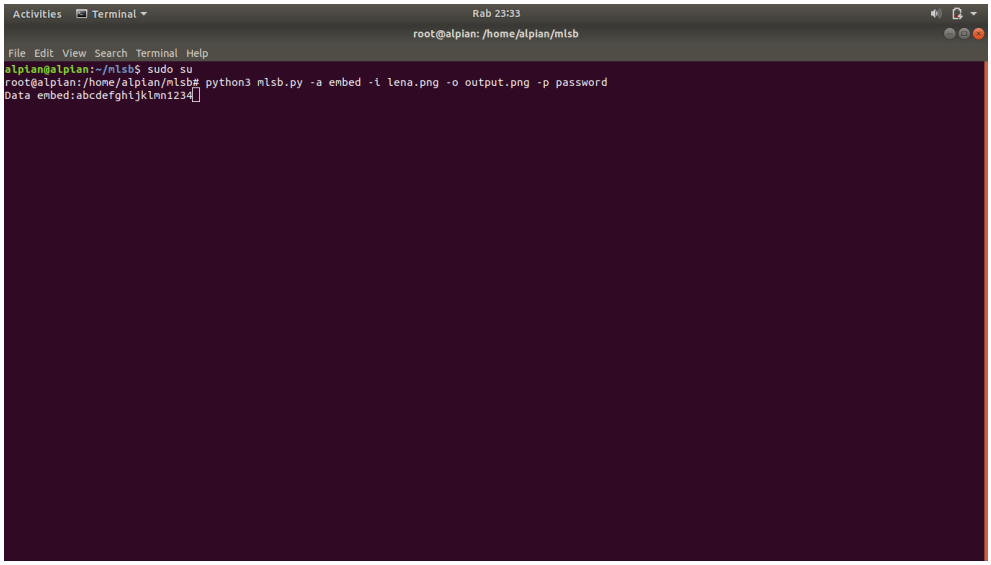
Hasil yang di peroleh dari proses pengujian program yang telah di buat seperti yang telah terpaparkan pada gambar di bawah untuk proses embed dan extract pesan. Adapun langkah - langkah yang di lakukan adalah sebagai berikut:

3.1 Proses embed

Untuk proses embed ini seperti di jelaskan pada gambar berikut. Pada tahap ini penulis mengambil 1 dari 5 gambar sampel yang di gunakan sebagai image yaitu “lena.png”, password yang penulis gunakan adalah “password” serta gambar outputan akan berubah menjadi “output.png” kemudia data yang di sisipkan adalah”abcdefghijklmn1234”. berikut merupakan contoh gambar yang akan di lakukan proses embed.

Gambar 3.1 Gambar Sampel Lena.png

Untuk lebih jelasnya bagaimana proses embed maka akan di jelaskan pada gambar di bawah. Gambar di bawah merupakan bagaimana cara mengetikkan perintah untuk proses penyisipan pesan atau di namakan proses embed.



Gambar 3.2 Perintah Embed Pesan

Seperti yang telah di paparkan pada gambar di atas maka cara yang di lakukan untuk proses embed ini ada 4 options atau intputan yang di lakukan, inputan tersebut adalah pertama menginputkan karakter -a yang maksudnya adalah bertujuan untuk action atau memilih embed/extract. Selanjutnya adalah -i adalah untuk mengiputkan gambar yang di gunakan sebagai cover image pesan, Sementara itu -o adalah untuk output gambar keluaran atau gambar yang telah di sisipi pesan, dan -p sendiri adalah password yang di gunakan. Setelah proses tersebut di lakukan maka akan keluar perintah inputkan karakter yang akan di embed.

Setelah di lakukan perintah proses embed di lakukan maka akan di dapatkan stego image atau gambar yang telah di sisipi pesan. Berikut merupakan output dari gambar yang telah di sisipi pesan yang berisi “abcdefghijklmn1234”.

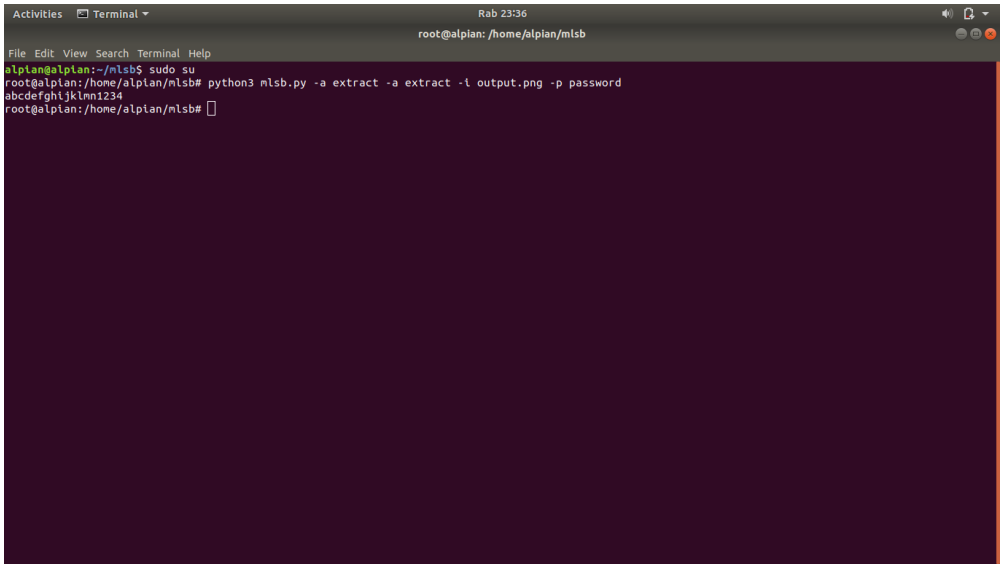
Gambar 3.3 Gambar Hasil Stego Image Output.png

Gambar di atas adalah gambar yang telah menjadi stego image. Sekilas gambar tersebut terlihat sama dengan gambar awal sebelum proses penyisipan pesan terjadi. Untuk mengetahui

bagaimana kualitas gambar awal dengan yang telah di sisipi pesan maka akan di hitung nilai MSE dan PSNR yang akan di jelaskan pada pembahasan selanjutnya.

3.2 Proses ekstrak

Untuk selanjutnya adalah proses mengembalikan pesan teks agar bisa di dapatkan kembali atau di sebut dengan proses extract pesan, langkah - langkah dari proses ekstrakpesan sendiri seperti yang di jelaskan pada gambar berikut



Gambar 3.4 Proses Ekstrak Pesan

Sedikit berbeda dari proses embed, proses ekstrak ini sendiri terdiri dari tiga inputan yaitu action atau pemilihan proses embed/extract, inputan gambar yang telah berisi karakter yang telah di inputkan beserta password yang bertujuan untuk membuka pesan itu sendiri. Dari gambar di atas pesan di dapatkan adalah “abcdefghijklmn1234” setelah di ketikkan perintah ekstrak.

3.3 Hasil Penghitungan MSE dan PSNR

Tabel 3.1 *Hasil penghitugan MSE dan PSNR*

Cover Image	pixel	Jumlah karakter	MSE	PSNR(dB)
Lena.png	512 x 512	100	0.00334167480469	72.8911617665
		1000	0.0365753173828	62.4989225746
		10000	0.150226593018	56.3633354286
Zelda.bmp	512 x 512	100	0.0036506652832	72.5070834504
		1000	0.037410736084	62.4008410752
		10000	0.150260925293	56.3623430201
Clown.bmp	512 x 512	100	0.00358200073242	72.5895469055
		1000	0.0372848510742	62.4154794813
		10000	0.154541015625	56.2403659882
Helmet.png	1600 x 1200	100	0.0004703125	81.4069383926
		1000	0.0050484375	71.0992337683
		10000	0.02095625	64.917667901
Mouse.png	1600 x 1200	100	0.0004875	81.2510574083
		1000	0.00501875	71.1248479825

		10000	0.0207	64.9711001541
Shape.png	1600 x 1200	100	0.000509375	81.0604273478
		1000	0.00500625	71.1356782744
		10000	0.0206453125	64.9825889986
B.png	4608 x 3072	100	0.000006506178	89.9975441245
		1000	0.00068537394205	79.7715277235
		10000	0.00280274285211	73.6549710722
G.png	4608 x 3072	100	0.00000652737087	89.9834207142
		1000	0.00068495008680	79.7742143582
		10000	0.00276014539931	73.7214840038
M.png	4608 x 3072	100	0.00000686645507	89.7634777772
		1000	0.00067647298177	79.8282990524
		10000	0.00281715393066	73.6326978317

Pada tahap ini setelah di lakukan pengujian terhadap beberapa sampel file citra dan beberapa karakter yang telah di inputkan, kemudian di lakukan pengujian seberapa besar kualitas dari file citra dari sebelum di sisipi pesan dan setelah di lakukan proses pesnyisipan pesan. Pengujian yang di lakukan adalah dengan menentukan nilai dari MSE dan PSNR seperti yang telah di jelaskan pada bab sebelumnya. Berikut merupakan tabel pengujian nilai MSE dan PSNR setelah di lakukan pengujian.

Dari keempat gambar yang berada di tabel 4.1 semuanya telah di lakukan proses embed dengan jumlah karakter yang berbeda yaitu terdiri 10, 50 dan 100 karakter pesan. Pengujian ini di lakukan bertujuan untuk melihat noise pada gambar yang telah di sisispi pesan. Pada tabel di attas dapat dilihat bahwa rata - rata MSE mendekati 0.0 yang berarti kerusakan pada gambar setelah melalui proses embed tidaklah significant.

Sedangkan nilai dari PSNR sendiri rata rata menujukkan di atas 70 yang berati kualitas gambar setelah proses embed dinilai baik, dalam arti noise yang di hasilkan tidaklah besar, Kualitas citra dapat ditarik kesimpulan cukup baik jika nilai PSNR memiliki nilai yaitu di atas 30dB – 40dB [9]. Dalam perhitungan secara matematis dapat di lihat bahwa semakin kecil nilai MSE maka akan besar pula nilai dari PSNR yang berarti sedikit pula terjadi noise pada gambar.

Selain dari pengujian di ata penulis juga melakukan seberapa besar karakter yang bisa di sisipkan dalam penggunaan fungsi hash dan algoritma MLSB ini. Berikut merupakan tabel guna untuk menghitung seberapa besar karakter yang bisa di sisipkan pada ukuran gambar yang berbeda.

Tabel 3.2 *Jumlah maksimal karakter pesan*

Cover Image	Ukuran Pixel	Maksimal Karakter	MSE	PSNR
Lena32.png	32 x 32	136	1.30079125	46.988760267
Lena64.png	64 x 64	433	1.062011718	47.896510518
Lena128.png	128 x 128	1327	0.791198730	49.147947913

Dari tabel di atas dapat di mengerti bahwa dengan gambar yang berukuran 32 x 32 maka dapat menampung sebanyak 136 karakter. Sedangkan dengan ukuran pixel 64 x 64 maka dapat di sisipkan karater sebanyak 433 karakter. Serta dengan gambar dengan ukuran 128 x 128 maka karakter yang dapat di tampung sebanyak 1327 karakter dan dengan nilai PSNR sebesar 49.147947913. dari beberapa pembahasan tersebut dapat di tarik kesimpulan bahwa jika semakin besar ukuran suatu gambar maka semakin bagus pula untuk proses penyembunyian suatu pesan.

4. Kesimpulan

4.1 Kesimpulan

Dari hasil analisa, perancangan, implementasi serta pada tahap pengujian maka dapat di simpulkan oleh penulis sebagai berikut :

1. Dengan menggunakan fungsi hash MD5 serta dengan menggunakan metode MLSB (*modified least significat bi*t) ini kualitas dari file citra yang di hasilkan menunjukkan nilai baik karena dari rata - rata nilai dari gambar sampel dengan nilai *MSE* mendekati 0.0 serta *PSNR* di atas 70db.
2. Inputan mencapai 10000 karakter masih tidak terlalu berpegaruh pada kualitas citra terbukti dengan dengan rata - rata nilai PSNR di atas 56dB.
3. Citra dengan ukuran 32x32 maka dapat menampung sebanyak 136 karakter. Sedangkan dengan ukuran pixel 64x64 maka dapat di sisipkan karater sebanyak 433 karakter. Serta dengan gambar dengan ukuran 128x128 maka karakter yang dapat di tampung sebanyak 1327 karakter dan dengan nilai PSNR sebesar 49.147947913.
4. Dapat ditarik kesimpulan bahwa semakin banyak karakter yang di sisipkan maka nilai *MSE* semakin besar maupun *PSNR* akan semakin kecil sehingga kualitas citra akan berkurang.

4.2 Saran

Adapun saran yang dapat di berikan untuk bahan penelitian selanjutnya adalah sebagai berikut :

1. Dapat menyembunyikan pesan teks menggunakan file citra yang bersifat RGB, video maupun audio dengan metode MLSB (*modified least significat bi*t).
2. Dengan menambahkan metode lain untuk algoritma MLSB (*modified least significat bi*t) ini seperti penggunaan fungsi hash SHA-1, SHA-2 maupun teknik enkripsi lain seperti kriptografi RSA.

Referensi

[1] M. A. I. Pakereng *et al.*, “Perancangan dan Implementasi Aplikasi CryptoSteganography untuk Pengamanan Pengiriman Gambar Menggunakan Vernam Cipher , RSA dan LSB Embedding Artikel Ilmiah Perancangan dan Implementasi Aplikasi Crypto-Steganography untuk Pengamanan Pengiriman Gambar Me,” no. 672008054, 2014.

[2] J. I. Sari, Sulindawaty, and H. T. Sihotang, “Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma HILL Cipher Dan Metode Least Significant BIT (LSB),” *J. Mantik Penusa*, vol. 1, no. 2, pp. 1–8, 2017.

[3] C. Paper and S. N. Indonesia, “Teknik Steganography Dengan Metode Least Significant Bit (Lsb),” no. September 2015, 2016.

[4] M. Z. Abu, “Modified Least Significant Bit (MLSB),” *Ccsnet*, vol. 4, no. 1, pp. 60–67, 2011.

[5] Inayatullah, “Analisis Penerapan Algoritma MD5 Untuk Pengamanan Password,” *J. Algoritm.*, vol. 3, no. 3, pp. 1–5, 2007.

[6] A. Kumar and R. Sharma, “International Journal of Advanced Research in A Secure Image Steganography Based on RSA Algorithm and Hash-LSB Technique,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 7, pp. 363–372, 2013.

[7] A. A. Lubis, N. P. Wong, I. Arfiandi, V. I. Damanik, and A. Maulana, “ Steganografi pada Citra dengan Metode MLSB dan Enkripsi Triple Transposition Vigenere Cipher,” *Steganografi pada Citra dengan Metod. MLSB dan Enkripsi Triple Transposition Vigenere Cipher*, vol. 16, no. 2, pp. 125–134, 2015.

[8] A. Sulistyanto, T. Informatika, F. T. Informasi, and U. B. Luhur, “Digital Watermarking Pada Citra Menggunakan Metode Modified Least Significant Bit (MLSB) Dengan Penyebaran Pesan Secara Acak Menggunakan Metode Linear Congruential Generator (LCG): Studi Laboratorium ICT Terpadu Universitas BUDI LUHUR.”

[9] J. Sembiring, “Analisis Algoritma Sha-512 Dan Watermarking Dengan Metode Least Significant Bit Pada Data Citra,” *Semin. Nas. Sist. Inf. Indones.*, pp. 2–4, 2013.

[10] M. M. Emam, A. A. Aly, and F. A. Omara, “An Improved Image Steganography Method Based on LSB Technique with Random Pixel Selection,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 3, pp. 361–366, 2016.