

TUGAS MAKALAH

Pengamanan Pesan Text Menggunakan Metode Steganografi Least Significant Bit dengan Media Digital Gambar

Mata Kuliah Forensik Digital



Di Buat Oleh:

M. Fatkhur Rohman (1410652017)

FAKULTAS TEKNIK

TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIAH JEMBER

1. Latar Belakang

Metode Least Significant Bit adalah metode penyisipan (steganografi) kedalam media lain yang dilakukan oleh Tri Cahyadi tahun 2012 dengan judul, “Implementasi steganografi LSB dengan vigenere cipher pada citra jpeg”, dengan hasil dari gabungan metode kriptografi vigenere dan steganografi LSB menghasilkan hasil yang cukup baik. Penelitian yang dilakukan oleh Hapsari M pada tahun 2010 dengan judul “Studi steganografi pada image file” dengan hasil LSB pada *.gif adalah algoritma yang sangat efektif untuk digunakan ketika mengembed pesan ke dalam citra grayscale. Penelitian selanjutnya oleh Oster Dwi Merbrial pada tahun 2012 dengan judul “implementasi steganografi untuk menyembunyikan gambardalam audio menggunakan LSB”, dengan hasil dengan metode lsb pesan mampu menggunakan kapasitas maksimal dalam penyembunyian pesan. Dari beberapa penelitian terkait yang dapat dianalisa bahwa metode LSB ini baik untuk metode penyembunyian pesan kedalam suatu media.

2. Pengertian Steganografi

Steganografi berasal dari bahasa Yunani yaitu *Steganos* yang berarti menyembunyikan dan *Graptos* yang berarti tulisan sehingga steganografi adalah tulisan yang disembunyikan. Secara umum steganografi adalah teknik menyisipkan pesan kedalam suatu media. Walaupun steganografi dapat dikatakan mempunyai hubungan erat dengan kriptografi, tetapi metode ini sangat berbeda.

3. Least Significant Bit

file gambar untuk menyembunyikan pesan atau informasi. Informasi atau pesan akan disembunyikan dengan mengacak bit terakhir dari setiap warna. Perubahan bit tersebut diatur sedemikian rupa untuk mengelabui penglihatan manusia sehingga tidak mudah untuk dideteksi. Berdasarkan latar belakang masalah yang telah disebutkan diatas maka judul yang akan diusulkan *Pengamanan Pesan Text Menggunakan Metode Steganografi Least Significant Bit dengan media digital gambar*.

4. Tujuan Penelitian

1. Mengamankan pesan text melalui media gambar sebagai inangnya .
2. Merancang dan mengimplementasikan perangkat lunak steganografi pada gambar digital yang dapat menyembunyikan data berupa text kedalam

gambar digital dengan metode LSB Kriptografi mengacak pesan sehingga tidak dimengerti, sedangkan steganografi menyembunyikan pesan sehingga pesan tidak terlihat. Pesan yang diacak dengan metode kriptografi mungkin akan menimbulkan kecurigaan, namun untuk tidak untuk pesan yang dibuat dengan steganografi.

Secara umum, terdapat dua proses didalam steganografi. Yaitu proses embeding untuk menyembunyikan pesan dan ekstrasi untuk meng ekstrasi pesan yang disembunyikan. Proses-proses tersebut dapat dilihat pada gambar berikut ini:

Sebagai ilustrasi misalkan *cover-object* adalah citra sekumpulan citra berwarna merah seperti yang terlihat pada contoh di bawah ini:

00110011 10100010 11100010 01101111

Dan misalkan pesan rahasia (yang telah dikonversi ke system biner) *embedded message* adalah 0110. Setiap bit dari *watermark* menggantikan posisi LSB dari segmen *pixel-pixel* citra menjadi :

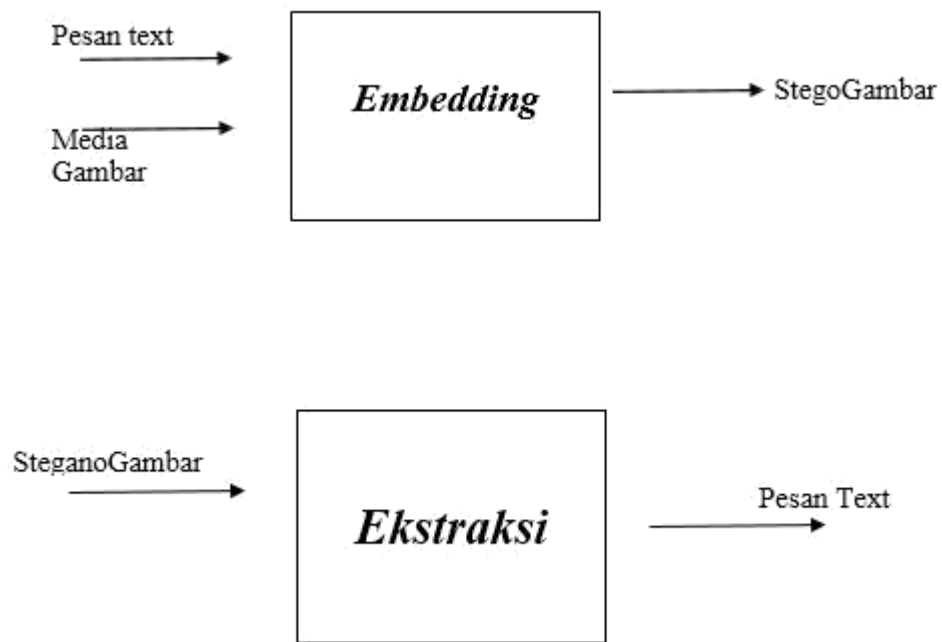
00110010 10100011 11100011 01101110

Dari hasil penanaman atau *embedding* kedalam sekumpulan *pixel* citra berwarna merah tadi diperoleh kembali sekumpulan *pixel* berwarna merah yang telah berubah sedikit pada posisi bit terendah atau LSB dari pixel tersebut. Demikianlah contoh sederhana bagaimana algoritma LSB bekerja untuk menggantikan nilai bit-bit terendah dari setiap *pixel* untuk disisipkan atau digantikan oleh bit baru yang mengandung pesan.

Untuk dapat membuat *hiddentext* tidak dapat dilacak, bit-bit pesan tidak mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit *LSB-nya* dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49. Pembangkitan bilangan acak dilakukan dengan *pseudo-random-number-generator (PRNG)* yang berlaku sebagai kunci stegano.

Pada citra 8-bit yang berukuran 256 x 256 *pixel* terdapat 65536 *pixel*, setiap *pixel* berukuran 1 *byte* sehingga kita hanya dapat menyisipkan 1 bit pada setiap *pixel*. Pada citra 24-bit yang berukuran 256 x 256 *pixel*, satu *pixel* berukuran 3 *byte* (atau 1 *byte* untuk setiap komponen R, G, B), sehingga kita bisa menyisipkan pesan sebanyak $65536 \times 3 \text{ bit} = 196608 \text{ bit}$ atau $196608/8 = 24576 \text{ byte}$. Pesan yang

disembunyikan di dalam citra dapat diungkap kembali dengan mengekstraksinya. Posisi *byte* yang menyimpan bit pesan dapat diketahui dari bilangan acak yang dibangkitkan oleh *PRNG*. Jika kunci yang digunakan pada waktu ekstraksi sama dengan kunci pada waktu penyisipan, maka bilangan acak yang dibangkitkan juga sama. Dengan demikian, bit-bit data rahasia yang bertaburan di dalam citra dapat dikumpulkan kembali. Berikut ini ilustrasi proses yang terjadi steganografi secara umum:



Keterangan :

—————→ : input/output

menunjukkan proses penyembunyian pesan dimana dibagian pertama, dilakukan proses embedding pesan text yang hendak disembunyikan secara rahasia kedalam media gambar sebagai penyimpanan, sehingga dihasilkan media dengan data tersembunyi didalamnya (StegoGambar).

Contoh Kasus:

Persamaan Utama dari LSB ini adalah kebanyakan teknik steganografi, ekstraksi pesan tidak akan mengembalikan media (gambar) awal persis sama dengan media setelah dilakukan ekstraksi bahkan sebagian besar mengalami kehilangan. Karena saat penyimpanan pesan tidak dilakukan pencatatan kondisi awal dari media yang digunakan untuk penyimpanan pesan.

$$S(i,j) = C(i,j) - 1, \text{ if } \text{LSB}(C(i,j)) = 1 \text{ and } m = 0$$

$$S(i,j) = C(i,j), \text{ if } \text{LSB}(C(i,j)) = m$$

$$S(i,j) = C(i,j) + 1, \text{ if } \text{LSB}(C(i,j)) = 0 \text{ and } m = 1$$

Dimana, $\text{LSB}(i,j)$ adalah cover-image dan $C(i,j)$ adalah pesan text yang disembunyikan. $S(i,j)$ adalah stego-image.

Contoh kasus, untuk menyembunyikan pesan ke dalam gambar dengan ukuran 3 pixel (24-bit RGB). Bit dari 3 pixel ini adalah :

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

Program dapat menyembunyikan huruf "J" yang mana posisinya berada no 74 dalam tabel ASCII ya itu "01001010", dan akan disisipkan menjadi :

(11101010 11101001 11001010)

(01100110 11001011 11101000)

(11001001 00100100 11101001)

Bit berubah setelah di embeding didalam stego-image, selanjutnya gambar siap dikirim.

5. Metode Stenografi

1. Metode Penyisipan


Proses penyisipan menggunakan Metode LSB mempunyai langkah langkah sebagai berikut, pertama adalah sistem meng-load file gambar RGB (media gambar digital) dari smartphone, kemudian gambar diekstrak per pixel sampai menjadi satuan terkecil(bit) misal 11100010, selanjutnya menghitung kapasitas gambar untuk media, besarnya kapasitas media untuk cover stego adalah 3 kali jumlah pixel dari gambar stego. Untuk mengacak bit yang berisi pesan pada cover image stegano dibutuhkan pengacak yang terstruktur atau generator kunci stegano yang disebut *Pseudo-random*, yang disini di implementasikan *pseudo-number-sequence* salah satu dari metode *pseudo random*. Selanjutnya akan dibaca semua bit yang ada di gambar, selagi dibaca di siapkan text yang

akan sisipkan pada media gambar. Setelah semuanya siap, pesan text akan sisipkan ke media gambar dengan metode LSB. Kemudian akan di transformasikan kembali menjadi gambar RGB. Setelah gambar stego maupun di distribusikan. image terbentuk gambar siap ditampilkan

2. Metode Ekstraksi

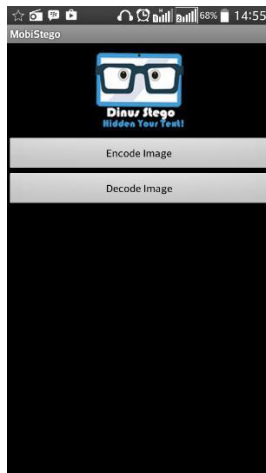
proses ekstraksi menggunakan Metode LSB dan Deskripsi pesan menggunakan Algoritma LSB yang mempunyai langkah langkah sebagai berikut, pertama inputan berupa gambar yang *disupport* atau di dukung oleh perangkat lunak. Gambar akan dibaca dan dirubah menjadi susunan terkecil(bit) misal 11100010, kemudian di cek kunci dengan metode pseudo random.

6. Eksperimen dilakukan dengan pembuatan program android dengan menggunakan ECLIPSE Android Developer Tools. Setelah Stego-Gambar diperoleh, kemudian kualitas Stego-Gambar diukur menggunakan PNSR. Dari nilai PNSR ini akan diketahui apakah Gambar yang berisi informasi pesan tersebut *robust* atau tidak. Dari proses penyisipan pesan ke dalam file citra tentunya akan ada perbedaan kualitas citra sebelum dan sesudah proses penyisipan apakah ada kuncinya? Jika tidak maka akan gagal yakni tidak ada pesan yang di tampilkan, jika ada maka akan dibangkitkan pseudo random untuk membaca seluruh bit-bit gambar sehingga didapatkan bit pesan yang berisi karakter. Bit bit pesan di baca sesuai urutan dari pseudo random kemudian bit bit pesan di ekstraksi menjadi karakter dan di tampilkan. pesan, untuk mengetahui seberapa besar penurunan kualitas citra maka akan dilakukan perhitungan nilai PSNR seperti yang telah dijelaskan pada bab sebelumnya. Citra pengujian memiliki ukuran yang bervariasi, yang diharapkan dapat menunjukkan kemampuan aplikasi steganography yang dibuat terhadap berbagai macam ukuran citra uji.

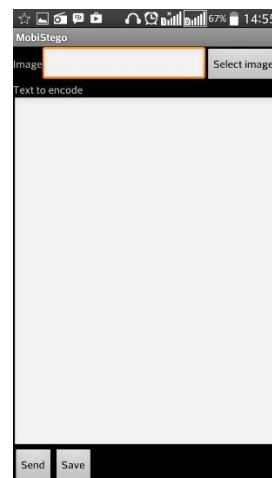
Nama Citra	Gambar Citra	Ukuran Citra (Pixel x Pixel)	Ukuran Citra awal (Kb)
Lena.jpg		329 x 361	255 Kilobyte

7. Proses Penyisipan Pesan

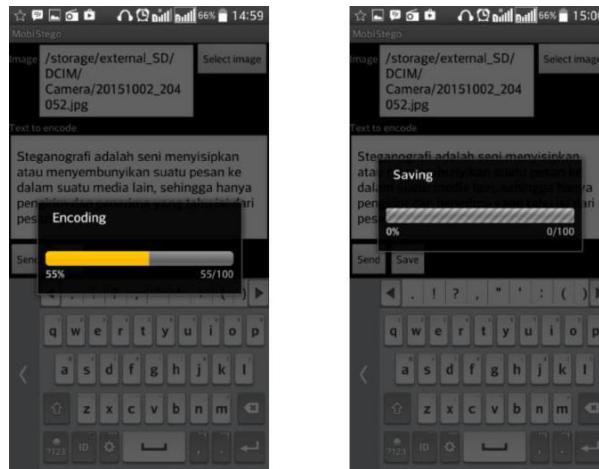
Untuk memulai proses penyisipan pesan, pengguna harus membuat berkas gambar cover (JPG) sebagai media penampung pesan (teks) dengan cara mengambil gambar menggunakan kamera



smartphone atau gambar yang sudah ada di galeri. Setelah meng-load gambar selanjutnya memasukkan pesan text yang ingin disembunyikan, pada text-area yang sudah tersedia.



Setelah memilih file gambar dan mengisi teks, maka proses selanjutnya adalah proses encoding.



Setelah menu encoding akan muncul pilihan menu “save” atau “send”. Pada metode Least Significant Bit, Informasi yang akan di *embedding* dimasukkan ke Bit/ bit. Bit ini akan di masukkan pada bit RGB gambar.