STEGANOGRAFI BERBASIS *LEAST SIGNIFICANT BIT* (LSB) UNTUK MENYISIPKAN GAMBAR KE DALAM CITRA GAMBAR

ZA'IMATUN NISWATI
zaimatunnis@gmail.com
Program Studi Teknik Informatika, Fakultas Teknik, Matematika dan IPA
Universitas Indraprasta PGRI

Abstrak. Penelitian ini bertujuan untuk menerapkan metode LSB untuk menyisipkan pesan gambar ke gambar grayscale. Hal ini diperlukan karena sering terjadi bahwa pesan gambar dikirim adalah pesan rahasia yang tidak boleh diketahui sembarang orang. Metode LSB bekerja dengan mengganti bit biner gambar kode biner terakhir kode dengan pesan. Keuntungan dari metode ini adalah ukuran gambar yang berisi pesan tidak berubah, sementara di sisi lain kapasitas pesan yang disisipkan terbatas. Alasannya adalah karena penggunaan gambar grayscale citra adalah bentuk digital yang lebih sederhana dari gambar RGB. Matlab adalah aplikasi pemrograman pengolahan citra digital yang menyediakan berbagai alat yang akan mempersingkat waktu penulisan program sehingga peneliti lebih fokus pada hasil dan penelitian inovasi.

Kata kunci: menyisipkan foto, gambar grayscale, least significant bit (lsb), matlab

Abstract. This research has a purpose to implement the LSB method to insert a picture message to the grayscale image. This is necessary because often the case that a picture message sent is a secret message that must not be known to just anyone. LSB method works by replacing the last bit binary code binary code image with a message. The advantage of this method is the size of the image containing the message does not change, while the downside is the capacity of the message to be inserted is limited. The reason is because the use of the image grayscale image is a digital form that is simpler than the RGB image. Matlab is a programming application in digital image processing that provides a variety of tools that will shorten the time of writing the program so that researchers focus more on results and innovation research.

Keywords: insert images, grayscale images, least significant bit (lsb), matlab

PENDAHULUAN

Pengiriman data/pesan dari suatu tempat ke tempat lain banyak terkendala dengan permasalahan keamanan. Apalagi jika data/pesan tersebut merupakan data/pesan yang sangat rahasia, sehingga tidak sembarang orang boleh membaca. Banyak cara yang dapat dilakukan untuk menyembunyikan data/pesan yang akan dikirim. Pertama, menggunakan teknik kriptografi, yakni dengan menyandikan data/pesan dengan menggunakan algoritma tertentu. Tetapi, dengan menyandikan pesan, maka pesan akan nampak sebagai kode-kode aneh yang justru akan membuat penasaran bagi orang yang membacanya, yang akhirnya akan berusaha untuk mengetahui kode-kode aneh tersebut. Teknik lain adalah steganografi dengan menyisipkan pesan yang akan dikirimkan ke media lain, sehingga pesan tersebut akan "tersembunyi" dan yang akan nampak adalah media lain yang digunakan untuk menyisipkan pesan.

Steganografi merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan sesuatu informasi. Steganografi dapat digolongkan sebagai salah

satu bagian dari ilmu komunikasi. Kata steganografi berasal dari bahasa Yunani yang berarti "tulisan tersembunyi". Pada era informasi digital, steganografi merupakan teknik dan seni menyembunyikan informasi dan data digital dibalik informasi digital lain, sehingga informasi digital yang sesungguhnya tidak kelihatan.

Secara teori, semua file umum yang ada di dalam komputer dapat digunakan sebagai media, seperti file gambar berformat JPG, GIF, BMP, atau di dalam musik MP3, atau bahkan di dalam sebuah film dengan format WAV atau AVI. Semua dapat dijadikan tempat bersembunyi, asalkan file tersebut memiliki bit-bit data redundan yang dapat dimodifikasi. Setelah dimodifikasi file media tersebut tidak akan banyak terganggu fungsinya dan kualitasnya tidak akan jauh berbeda dengan aslinya.

Data yang dikirim hasil enkripsi disembunyikan dalam *cover carrier* agar dapat meningkatkan keamanan pada saat transmisi. Banyak metoda steganografi yang melekatkan sejumlah besar informasi rahasia di dalam *pixel* pada *cover image*. Karena perasaan manusia yang tidak sempurna dalam hal visualisasi, keberadaan informasi rahasia yang ditempelkan tersebut dapat saja tidak terlihat. Tetapi informasi rahasia tersebut mungkin saja ditemukan, jika belum ditempatkan secara baik.

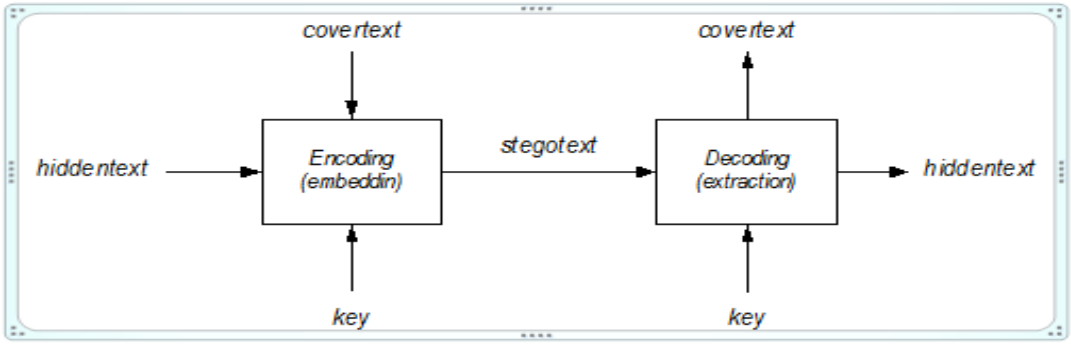
Matlab merupakan aplikasi pemrograman yang telah dikenal dalam pembuatan aplikasi penunjang penelitian. Matlab dalam pengolahan citra digital menyediakan bermacam tools yang akan mempersingkat waktu penulisan program sehingga peneliti lebih berfokus pada hasil dan inovasi penelitian.

Tujuan penelitian ini adalah mengimplementasikan metode LSB untuk menyisipkan pesan gambar ke dalam citra gambar, menggunakan software Matlab R2009b. Metode LSB bekerja dengan mengganti bit terakhir kode biner citra dengan kode biner pesan sebagai nilai derajat keabuan citra pada akhir citra. Kelebihan metode LSB adalah ukuran citra yang mengandung pesan tidak berubah, sedangkan kekurangannya adalah kapasitas pesan yang akan disisipkan terbatas.

TINJAUAN PUSTAKA

Steganografi

Steganografi berasal dari bahasa Yunani, yang berarti tulisan yang tertutup/tersamar ("*covered letter*"). Dalam arti lain dapat dikatakan sebagai cara komunikasi yang menyembunyikan pesan. Data/pesan yang akan dikirim disembunyikan ke media lain. Format media yang bisa dipakai diantaranya adalah: 1) Format image: bmp, jpg, gif., 2) Format audio: wav, mp3, dan 3) c. Format lain: html, pdf, file text.
Bentuk data/pesan tidak berubah, hanya saja karena data/pesan tersebut dikirim dengan disembunyikan dalam media lain, maka yang terlihat adalah media yang dipakai untuk mengirimkan data/pesan tersebut.



Gambar 1. Proses Steganografi

Steganografi merupakan suatu ilmu atau seni dalam menyembunyikan informasi dengan memasukkan informasi tersebut ke dalam pesan lain. Dengan demikian keberadaan informasi tersebut tidak diketahui oleh orang lain (Munir: 2006).

File Gambar

Pada komputer, suatu gambar adalah suatu array dari bilangan yang merepresentasikan intensitas terang pada point yang bervariasi (pixel). Pixel ini menghasilkan *raster data* gambar. Suatu ukuran gambar yang umum adalah 640 x 480 pixel dan 256 warna (atau 8 bit per pixel). Suatu gambar akan berisi kira-kira 300 kilobit data.

Gambar digital disimpan juga secara khusus di dalam file 24-bit atau 8-bit. Gambar 24-bit menyediakan lebih banyak ruang untuk menyembunyikan informasi; bagaimanapun, itu dapat sungguh besar. Semua variasi warna untuk pixel yang diperoleh dari tiga warna dasar: merah, hijau dan biru. Setiap warna dasar direpresentasikan dengan 1 byte; gambar 24-bit menggunakan 3 byte per pixel untuk merepresentasikan suatu nilai warna. 3 byte ini dapat direpresentasikan sebagai nilai hexadecimal, decimal, dan biner. Dalam banyak halaman Web, warna latar belakang direpresentasikan dengan bilangan 6 digit hexadecimal, yang aktualnya tiga ikatan merepresentasikan merah, hijau dan biru. Latar belakang putih akan mempunyai nilai FFFFFF: 100% merah (FF), 100% hijau (FF) dan 100% biru (FF). Nilai decimal-nya 255,255,255 dan nilai biner-nya adalah 11111111, 11111111, 11111111, yang adalah tiga byte yang menghasilkan putih (Munir: 2004).

Definisi latar belakang putih adalah analog dengan definisi warna dari pixel tunggal dalam suatu gambar. Pixel merepresentasikan kontribusi pada ukuran file. Untuk contoh, andaikan kita mempunyai gambar 24-bit luasnya 1,024 pixel dengan ketinggian 768 pixel, yang merupakan resolusi umum untuk grafik beresolusi tinggi. Suatu gambar mempunyai lebih dari dua juta pixel, masing-masing mempunyai definisi yang akan menghasilkan suatu kelebihan file 2 Mbyte. Karena gambar 24-bit masih relative tidak umum pada internet, ukuran seperti ini akan menarik perhatian selama transmisi. Kompresi file akan menguntungkan, jika tidak perlu transmisi file seperti itu.

Kompresi File

Menurut (1996), dua kandungan dari kompresi adalah *lossless* dan *lossy*. Kedua metoda ini menghemat ruang penyimpanan tetapi mempunyai hasil yang berbeda, yang bertentangan dengan penyembunyian informasi. Kompresi *lossless* membiarkan kita merekonstruksi pesan asli yang sama; oleh karena itu, lebih disukai ketika informasi asli harus tetap utuh (seperti dengan gambar steganography). Kompresi *lossless* khusus untuk gambar yang tersimpan sebagai GIF (*Graphic Interchange Format*) dan BMP 8-bit (file bitmap Microsoft Windows dan OS/2).

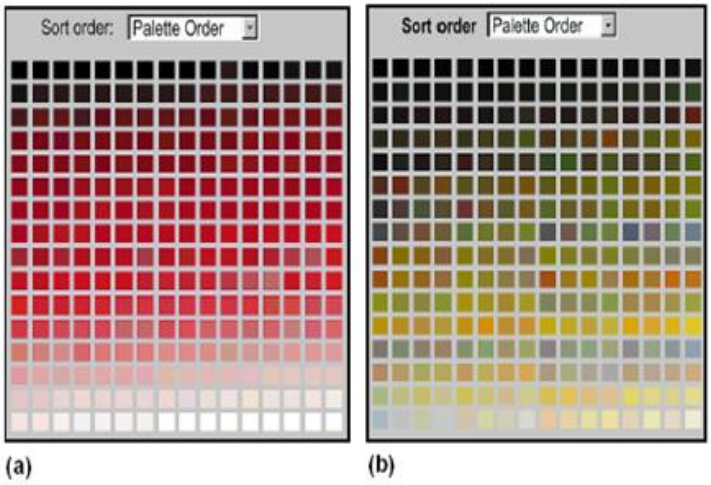
Kompresi *lossy*, pada penanganan lainnya, menghemat ruangan tetapi tidak menjaga integritas gambar aslinya. Metoda ini secara khusus untuk gambar yang tersimpan sebagai JPEG (*Joint Photographic Experts Group*).

Embedding Data

Data embedded, yang tersembunyi dalam suatu gambar membutuhkan dua file. Pertama adalah gambar asli yang belum modifikasi yang akan menangani informasi tersembunyi, yang disebut *cover image*. File kedua adalah informasi pesan yang disembunyikan. Suatu pesan dapat berupa plaintext, chipertext, gambar lain, atau apapun yang dapat ditempelkan ke dalam *bit-stream*. Ketika dikombinasikan, *cover image* dan pesan yang ditempelkan membuat *stego-image*. Suatu *stego-key* (suatu password khusus) juga dapat digunakan secara tersembunyi, pada saat decode selanjutnya dari pesan

(Sukrisno: 2007). Kebanyakan software steganography tidak mendukung atau tidak direkomendasi menggunakan gambar JPEG, tetapi sebagai gantinya direkomendasikan menggunakan gambar *lossless* 24-bit seperti BMP. Alternatif terbaik berikutnya untuk gambar 24-bit adalah 256 warna atau gambar gray scale. Secara umum ditemukan pada Internet atau file GIF (Jonson: 1998).

Dalam gambar 8-bit warna seperti file GIF, setiap pixel direpresentasikan sebagai byte tunggal, dan setiap pixel selalu menunjuk ke tabel indek warna (*palette*) dengan 256-kemungkinan warna. Nilai pixel adalah diantara 0 dan 255. Software secara sederhana menggambarkan indikasi warna pada palette merah, menggambarkan perubahan yang sulit dipisahkan dalam variasi warna: perbedaan visualisasi diantara banyak warna yang sulit. Gambar 2. menunjukkan perubahan warna yang sulit dipisahkan dengan baik.



Gambar 2. Representasi Warna Pallette

Banyak pakar steganography merekomendasikan penggunaan gambar yang meliputi 256 *shade* gray. Gambar gray-scale lebih disukai karena perubahan keteduhan sangat gradual dari byte ke byte, dan lebih sedikit perubahan nilai diantara masukan palette, dimana mereka dapat menyembunyikan informasi lebih baik. Gambar 2 menunjukkan suatu palette gray-scale dari 256 *shade*. Beberapa gambar adalah 4-bit, di buat dengan 16 *shade* dari gray, sesungguhnya gambar ini menawarkan banyaknya variasi yang lebih sedikit.

Penyisipan *Least Significant Bit*

Penyisipan *Least Significant Bit* (LSB) adalah umum, pendekatan yang sederhana untuk menempelkan informasi di dalam suatu file cover. Sayangnya, hal itu sangat peka untuk kejadian yang melalaikan manipulasi gambar. Meng-konvert suatu gambar dari format GIF atau BMP, yang merekonstruksi pesan yang sama dengan aslinya (*lossless compression*) ke JPEG yang *lossy compression*, dan ketika dilakukan kembali akan menghancurkan informasi yang tersembunyi dalam LSB (Maya: 2012).

Untuk menyembunyikan suatu gambar dalam LSB pada setiap byte dari gambar 24-bit, dapat disimpan 3 byte dalam setiap pixel. Gambar 1,024 x 768 mempunyai potensi untuk disembunyikan seluruhnya dari 2,359,296 bit (294,912 byte) pada informasi. Jika pesan tersebut dikompres untuk disembunyikan sebelum ditempelkan, dapat menyembunyikan sejumlah besar dari informasi. Pada pandangan mata manusia, hasil *stego-image* akan terlihat sama dengan gambar cover.

Implementasi LSB

Software steganografi memproses penyisipan LSB dengan membuat informasi yang tersembunyi dapat ditemukan lebih sedikit. Untuk contoh, tool EzStego menyusun palette untuk mengurangi kejadian dari warna indek bersebelahan yang kontrasnya paling banyak sebelum disisipkan pesan. Pendekatan ini bekerja sangat baik dalam gambar gray-scale dan dapat bekerja dengan baik dalam gambar dengan warna yang saling berhubungan (Maya: 2012).

S-Tool, merupakan tool steganography lainnya, yang mengambil pendekatan berbeda dengan memperkirakan cara lekat gambar cover, yang dapat berarti perubahan palette secara radikal. Seperti dengan gambar 24-bit, perubahan LSB pixel dapat membuat warna baru (Warna baru tidak dapat ditambahkan ke gambar 8-bit dalam kaitannya dengan keterbatasan palette). Sebagai gantinya, S-Tool mengurangi jumlah dari warna yang menangani kualitas gambar, sehingga perubahan LSB tidak secara drastis merubah nilai warna.

Masking dan Filtering

Teknik *masking* dan *filtering*, hanya terbatas ke gambar 24-bit dan gray-scale, informasi disembunyikan dengan menandai suatu gambar dengan cara seperti *paper watermark*. Teknik *watermarking* dapat di aplikasikan dengan resiko rusaknya gambar dalam kaitannya dengan *lossy compression,* sebab mereka lebih menyatu ke dalam gambar.

Menurut definisinya, watermark kelihatannya bukanlah steganography. Salah satu perbedaan utama adalah mengenai tujuannya. Steganography tradisional merahasiakan informasi; watermark meluaskan informasi dan menjadikannya suatu attribute dari gambar cover. Watermark digital dapat berupa informasi sebagai copyright, kepemilikan, atau lisensi, seperti yang ditunjukkan dalam Gambar 3. Dalam steganography, objek dari komunikasi adalah pesan yang tersembunyi. Di dalam watermark digital, objek dari komunikasi adalah cover.

Masking lebih *robust* dari pada penyisipan LSB dengan hasil kompresi, *cropping*, dan beberapa pemrosesan gambar. Tehnik masking menempelkan informasi dalam area significant sehingga pesan yang tersembunyi itu lebih bersatu dengan gambar cover dari pada penyembunyian dalam level "noise".

Algoritma dan Transformasi

Manipulasi LSB adalah suatu cara yang cepat dan mudah untuk menyembunyikan informasi tetapi sangat peka untuk perubahan hasil yang kecil dari pemerosesan gambar atau *lossy compression*. Seperti kompresi yang merupakan kunci keuntungan dari gambar JPEG yang mempunyai kelebihan dari format yang lain. Gambar dengan kualitas warna yang tinggi dapat disimpan dalam file yang relative kecil menggunakan metoda kompresi JPEG; sehingga gambar JPEG menjadi lebih berlimpah pada Internet.

Satu tool steganography yang mengintegrasikan algoritma kompresi untuk menyembunyikan informasi adalah Jpeg-Jsteg. Jpeg-Jsteg membuat suatu *stego-image* JPEG dari input suatu pesan yang disembunyikan dan suatu *lossless* gambar cover. Dengan mempertimbangkan *Independent Group* JPEG, software JPEG telah dicoba dengan modifikasi untuk 1-bit steganography dalam file output JFIF, yang mengkomposisikan bagian *lossy* dan *nonlossy*. Software mengkombinasikan pesan dan gambar cover menggunakan algoritma JPEG untuk membuat *stego-image lossy* JPEG.

Encrypt dan *scatter* adalah teknik yang lain dalam menyembunyikan data secara menyeluruh ke gambar. Pesan yang menyebar lebih disukai daripada noise. Penganjur

dari pendekatan ini mengasumsikan bahwa jika pesan bit diekstrak, akan menjadi sia-sia tanpa algoritma dan *stego-key* men-dekodenya. Untuk contoh, tool *White Noise Storm* berdasarkan pada teknologi *spread spectrum* dan *hopping* frekuensi, menyebarkan pesan ke seluruh gambar. Sebagai gantinya, saluran x dari komunikasi yang berubah dengan rumusan dan passkey yang tetap. *White Noise Strom* menyebarkan delapan saluran dengan men-generate bilangan acak dengan ukuran window dan saluran data sebelumnya. Setiap saluran merepresentasikan 1 bit, sehingga setiap window gambar menjaga 1 byte dari informasi dan bit yang tidak digunakan. Rotasi saluran, pertukaran dan penyilangan antar diri mereka ke field permutasi bit yang berbeda. Untuk kejadian, bit 1 boleh ditukar dengan bit 7, atau kedua bit dapat berrotasi satu posisi ke kanan. Aturan untuk pertukaran diatur dengan *stego-key* dan dengan data acak window sebelumnya (sama dengan blok enkripsi DES).

METODE

Penelitian ini mengimplementasikan metode LSB untuk menyisipkan pesan gambar ke dalam citra *grayscale*, menggunakan software Matlab R2009b. Metodologi yang digunakan dalam implementasi ini meliputi: 1) Proses penyisipan pesan dengan metode LSB dapat dituliskan dalam algoritma sebagai berikut: a) Inputkan pesan yang akan disisipkan, b) Ubah pesan menjadi kode-kode biner. Untuk mempermudah dapat terlebih dulu diubah menjadi desimal, kemudian biner, c) Inputkan citra gambar yang akan disisipi pesan, d) Dapatkan nilai derajat keabuan masing-masing piksel, e) Ubah derajat keabuan tersebut menjadi kode-kode biner, f) Ganti bit terakhir kode biner citra dengan bit pesan, g) Ubah kode biner menjadi derajat keabuan citra baru (citra yang sudah disisipi pesan), dan h) Petakan menjadi citra baru. 2) Sedangkan ekstraksi pesan yang sudah disisipkan dengan metode LSB dapat dilakukan dengan algoritma berikut: a) Input image yang sudah mengandung pesan, b) Dapatkan nilai derajat keabuan citra tersebut, c) Ubah nilai derajat keabuan menjadi kode-kode biner, d) Ambil nilai kode biner bit terakhir, dan e) Terjemahkan menjadi karakter.

HASIL DAN PEMBAHASAN

Penyisipan Pesan dan Ekstraksi Pesan dengan Metode Least Significant Bit (LSB).

Pada sebuah citra grayscale 6x6 piksel disisipkan pesan gambar. Kode ASCII dari pesan diberikan, kode ASCII tersebut untuk selanjutnya diubah menjadi 7 bit kode-kode biner. Misalkan matrik tingkat derajat keabuan citra sebagai berikut;

196	10	97	182	101	40
67	200	100	50	90	50
25	150	45	200	75	28
176	56	77	100	25	200
101	34	250	40	100	60
44	66	99	125	190	200

Nilai derajat keabuan masing-masing piksel diubah menjadi biner sehingga menjadi:

11000100	00001010	01100001	10110110	01100101	00101000
01000011	11001000	01100100	00110010	01011010	00110010
00011001	10010110	00101101	11001000	01001011	00011100
10110000	00111000	01001101	01100100	00011001	11001000
01100101	00100010	11111010	00101000	01100100	00111100
00101100	01000010	01100011	01111101	10111110	11001000

Untuk selanjutnya, tiap bit kode biner pesan digunakan untuk menggantikan bit terakhir dari kode biner derajat keabuan citra. Proses penggantian dilakukan terurut, menurut baris ataupun kolom. Pada percobaan ini digunakan kolom.
Setelah proses penggantian maka kode biner untuk matrik citra menjadi:

11000101	00001011	01100001	10110111	01100100	00101000
01000011	11001001	01100101	00110010	01011010	00110010
00011000	10010111	00101101	11001001	01001011	00011100
10110000	00111000	01001101	01100100	00011001	11001000
01100100	00100011	11111011	00101001	01100100	00111100
00101100	01000010	01100010	01111100	10111110	11001000

Matrik biner tersebut dikembalikan lagi menjadi decimal sehingga di dapat matrik berikut:

197	11	97	183	100	40
67	201	101	50	90	50
24	151	45	201	75	28
176	56	77	100	25	200
100	35	251	41	100	60
44	66	98	124	190	200

Matrik ini akan dipetakan kembali ke bentuk citra. Ekstraksi pesan dapat dengan mudah dilakukan dengan mengambil bit terakhir dari kode biner citra.

Banyak cara untuk menyembunyikan informasi di dalam gambar. Untuk menyembunyikan informasi, penyisipan pesan yang langsung dapat mengenkode setiap bit dari informasi dalam gambar atau menempelkan pesan secara selektif dalam area *noisy*, menggambarkan area yang kurang diperhatikan, dimana ada banyak variasi warna natural. Pesan dapat juga terserak secara acak sepanjang gambar. Pola redundansi encoding "wallpapers" menutup gambar dengan pesan.

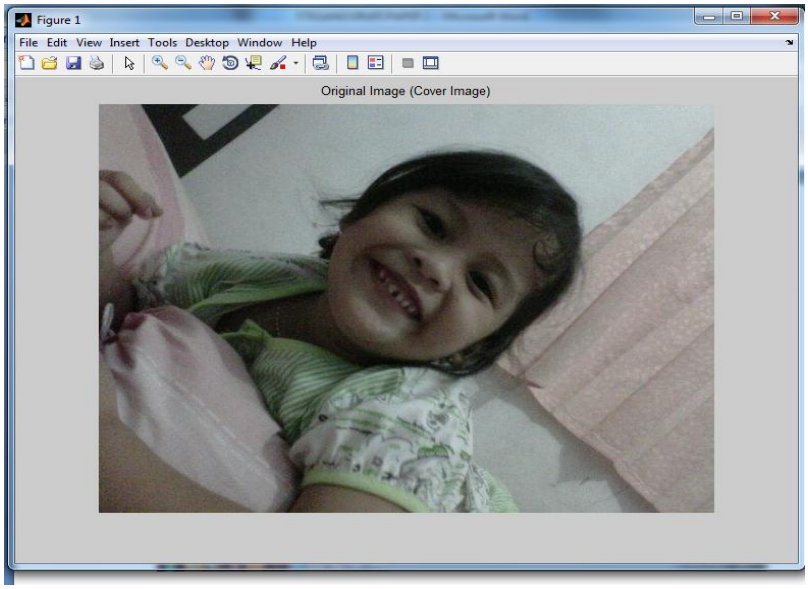
Sejumlah cara yang ada untuk menyembunyikan informasi dalam gambar digital dengan pendekatan yang umum termasuk: 1) penyisipan least significant bit, 2) masking dan filtering, dan 3) algoritma dan transformasi. Setiap teknik-teknik itu dapat diaplikasikan dengan derajat kesuksesan yang bervariasi pada file gambar yang berbeda. Pengirim membuat suatu steganogram menggunakan fungsi *embedding*, dimana fungsi tersebut mempunyai dua parameter sebagai berikut: 1) Media pembawa yang isisnya bersifat random, dan 2) Pesan yang embedded.

Beberapa utilitas steganografi menggunakan kunci rahasia. Ada dua jenis kunci rahasia: kunci steganografi dan kunci kriptografi. Kunci steganografi mengontrol proses embedding dan ekstracting. Sebagai contoh, kunci ini dapat menyebarkan pesan yang ditempelkan ke bagian dari semua tempat dalam media pembawa. Tanpa kunci itu, bagian ini tidak diketahui, dan masing-masing sample yang digunakan untuk mendeteksi penempelan oleh penyerangan statistik adalah dengan pencampuran tempat yang digunakan dan tempat yang tidak digunakan, yang dapat merusak hasilnya. Kunci kriptografi digunakan untuk mengenkripsi pesan sebelum ditempelkan. Kedua aplikasi rahasia ini diperoleh dari algoritma aktual dalam bentuk suatu parameter kunci. Jika kunci itu bersifat rahasia, algoritma steganografi dapat menjadi kunci public. Hal ini dimungkinkan untuk menentukan apakah *bits read* itu dalam kenyataannya menyandikan pesan dari steganogram utama hanya jika pesan itu mempunyai kesesuaian dengan kunci dekripsinya. Enkripsi sebaiknya juga sebagai tambahan utilitas steganografi yang mana tidak secara implisit *encrypt*.

Script Matlab untuk menyisipkan pesan gambar adalah sebagai berikut:

```
%This program hides a message image in the lower
%bit planes of a cover image
%read in cover image filename
covername = input('Enter image file name with extension
(like jennifer.bmp): ', 's');
%read in message image filename
messagename = input('Enter message image file name with
extension: ', 's');
%open cover and message image files
cover = imread(covername);
message = imread(messagename);
%display on screen the two images
figure(1), imshow(cover); title('Original Image (Cover
Image)');
figure(2), imshow(message);title('Image to Hide (Message
Image)');
%change to double to work with addition below
cover=double(cover);
message=double(message);
%imbed = no. of bits of message image to embed in cover
image
imbed=4;
%shift the message image over (8?imbed) bits to right
messageshift=bitshift(message,-(8-imbed));
%show the message image with only embed bits on screen
%must shift from LSBs to MSBs
showmess=uint8(messageshift);
showmess=bitshift(showmess,8-imbed);
figure(3),imshow(showmess);title('4 Bit Image to Hide');
%now zero out imbed bits in cover image
coverzero = cover;
for i=1:imbed
coverzero=bitset(coverzero,i,0);
end
%now add message image and cover image
stego = uint8(coverzero+messageshift);
figure(4),imshow(stego);title('Stego image');
%save files if need to
%4 bit file that was embedded = same as file extracted
imwrite(showmess,'showmesscolor.bmp'); %use bmp to preserve
lower bits
%jpg will get rid of them
%stego file
imwrite(stego,'stegocolor.bmp');
```

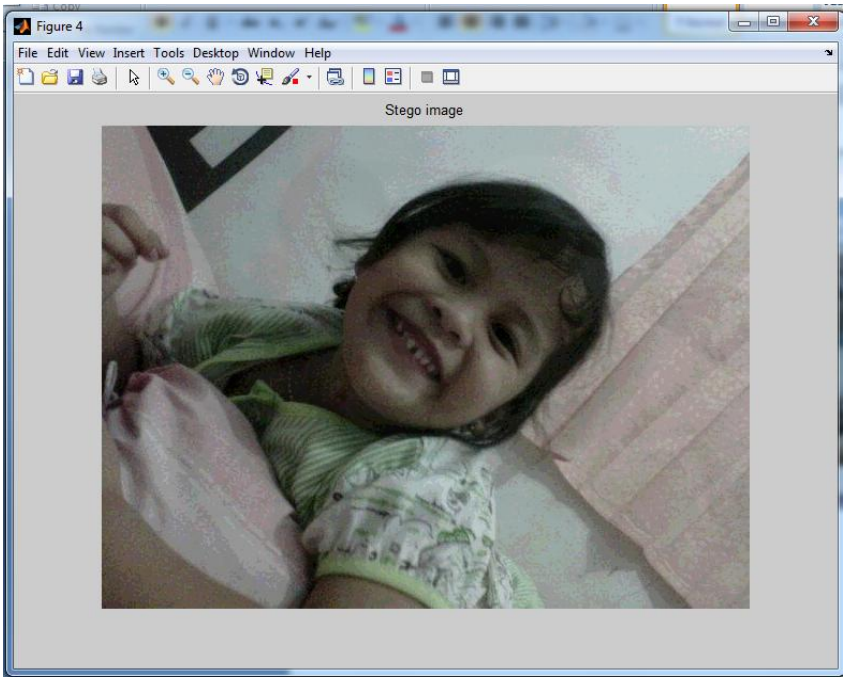
Hasil aplikasi menggunakan Matlab R2009b:



Gambar 3. Gambar pesan yang akan disembunyikan (hiddentext)

Gambar 4. Gambar untuk menyembunyikan pesan (covertext)

Gambar 5. Gambar yang sudah berisi pesan yang disembunyikan (stegotext)



Gambar 6. Gambar yang disembunyikan setelah di ekstraksi

Jika diperhatikan, penggantian bit terakhir tersebut tidak terlalu berpengaruh terhadap derajat keabuan citra. Ada tiga kemungkinan yang terjadi setelah penggantian bit terakhir, yakni: 1) Nilainya derajat keabuan tetap, 2) Nilai derajat keabuan berkurang 1, dan 3) Nilai derajat keabuan bertambah 1.

Perubahan yang sedemikian kecil tersebut tidak mungkin akan dirasakan secara kasat mata, sehingga citra sebelum dan setelah disisipi pesan tidak akan nampak terjadi perubahan. Karena pesan menempati bit terakhir masing masing piksel, tidak dapat dipungkiri lagi bahwa ukuran pesan yang dapat disisipkan terbatas. Oleh karena itu diperlukan pertimbangan dalam pemilihan citra.

PENUTUP

Kesimpulan

Penyisipan pesan/data gambar ke dalam citra gambar dapat dilakukan dengan metode LSB. Metode LSB akan mengganti bit terakhir kode biner masing-masing piksel. Kelebihan dari metode ini adalah ukuran citra tidak berubah/tetap, sehingga tidak mengakibatkan kecurigaan akan adanya pesan rahasia dalam citra. Kekurangan metode ini adalah jumlah karakter pesan yang disisipkan terbatas, sehingga besarnya citra harus menyesuaikan besarnya pesan yang dikirim. Penggunaan matlab dalam pembuatan aplikasi pengolahan citra akan mempersingkat penulisan *source code* dengan penyediaan fasilitas oleh Matlab. Proses penulisan program yang sekiranya dalam bentuk iterasi digantikan oleh fungsi-fungsi dalam Matlab.

Saran

Disarankan untuk melakukan penelitian menyembunyikan pesan suara atau video ke dalam citra video dengan berbagai macam metode.

DAFTAR PUSTAKA

Maya. ***Steganografi LSB.*** (http://maya9luthu.blogsome.com/2006/12/11/steganografi-lsb/), diakses 4 Mei 2012.

Munir, Rinaldi. 2006. **Diktat Kuliah IF5054 Kriptografi.** Bandung: Penerbit ITB.

Munir, Rinaldi. 2004. **Pengolahan Citra Digital Dengan Pendekatan Algoritmik.** Bandung: Informatika.

Neil F. Johnson, Sushil Jajodia. 1998. **Steganalysis of Images Created Using Current Steganography Software.** in David Aucsmith (Ed.): Information Hiding, LNCS 1525, Springer-Verlag Berlin Heidelberg 1998.

Robert Tinsley. 1996. **Steganography and JPEG Compression.** Final Year Project Report, University of Warwick.

Sukrisno. **Steganografi.** (http://mysukris.blogspot.com/2007/04/steganografi-eureka-i-found-it.html), diakses 6 April 2007.

.