

STEGANOGRAFI MENGGUNAKAN METODE LSB DENGAN SOFTWARE MATLAB

Fahri Perdana Prasetyo



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
SYARIF HIDAYATULLAH
JAKARTA
2010 M / 1431 H**

STEGANOGRAFI MENGGUNAKAN METODE LSB DENGAN SOFTWARE MATLAB

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Sains

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh :

Fahri Perdana Prasetyo

104094003025

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI

SYARIF HIDAYATULLAH

JAKARTA

2010 M / 1431 H

PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI UANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA APAPUN

Jakarta, 17 Juni 2010

Fahri Perdana Prasetyo
104094003025

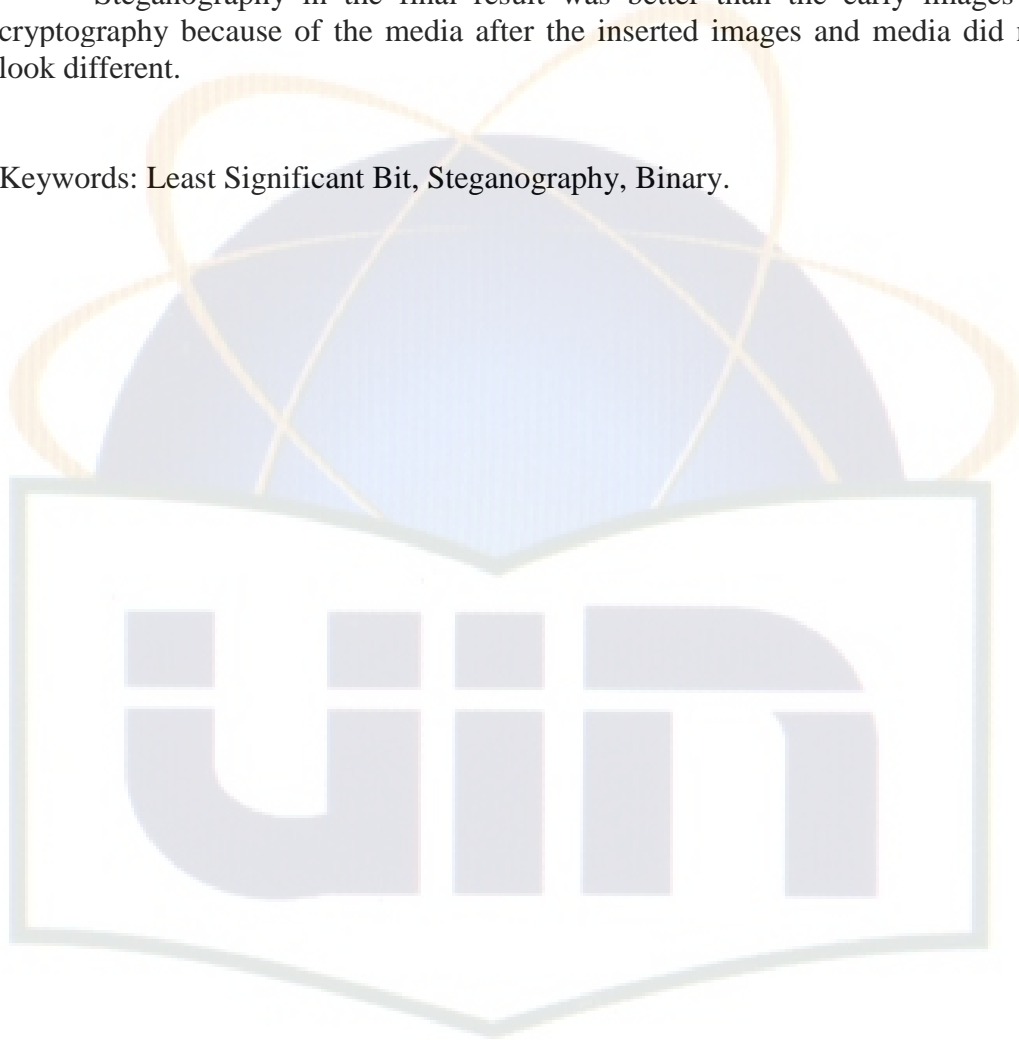


ABSTRACT

Least Significant Bit (LSB) is one method of Steganography is used to hide messages. This method made insertion of the binary message into a binary file used by the media. This method produces images that are visually look exactly with the beginning of a media image.

Steganography in the final result was better than the early images of cryptography because of the media after the inserted images and media did not look different.

Keywords: Least Significant Bit, Steganography, Binary.



ABSTRAK

Least Significant Bit (LSB) merupakan salah satu metode dalam Steganografi yang digunakan untuk menyembunyikan pesan. Metode ini melakukan penyisipan binari pesan kedalam binari *file* media yang digunakan. Metode ini menghasilkan gambar yang secara *visual* terlihat sama persis dengan media gambar awal.

Steganografi dalam hasil akhir lebih baik daripada kriptografi karena media gambar awal dan media gambar setelah disisipkan tidak tampak perbedaan.

Kata Kunci: *Least Significant Bit*, Steganografi, Binari.



KATA PENGANTAR

Alhamdulillah, Penulis panjatkan rasa syukur kehadirat Allah SWT. yang senantiasa memberikan rahmat dan hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi ini. Skripsi ini berjudul “Steganografi Menggunakan Metode LSB Dengan Software Matlab” dan ditulis untuk memenuhi tugas akhir.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

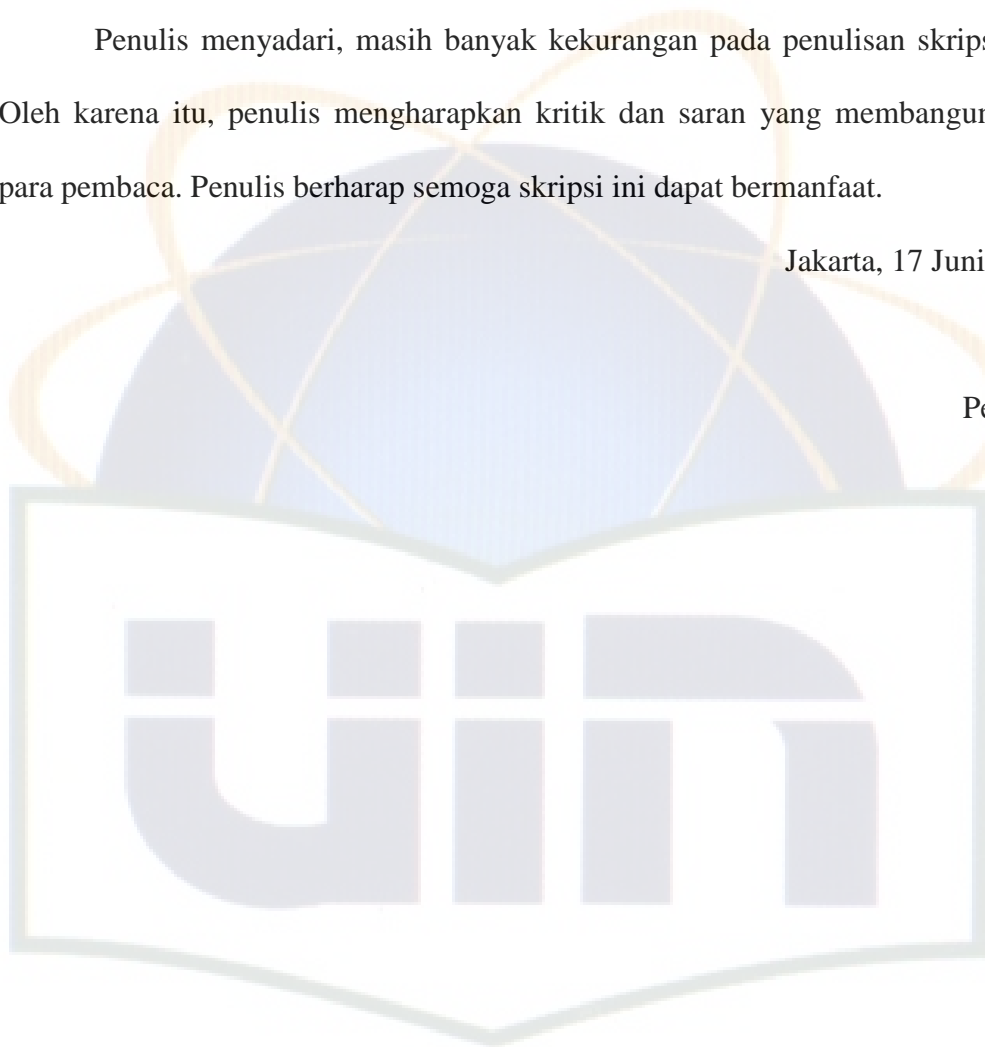
1. Dr. Syopiansyah Jaya Putra, M. Sis, Dekan Fakultas Sains dan Teknologi.
2. Yanne Irene, M. Si, Ketua Program Studi Matematika yang senantiasa memberi bimbingan dan masukan kepada Penulis.
3. Hermawan Setiawan, M. Kom serta Gustina Elfiyanti, M.Si selaku Pembimbing I dan Pembimbing II yang telah sabar menuntun saya sehingga Penulis menuju jalan yang lurus.
4. Taufik Edy Susanto selaku Pembimbing III, Pembimbing yang tidak tercantum namun sangat membantu Penulis menyelesaikan skripsi ini, Serta Dr. Agus Salim, M. Si dan Suma'inna M. Si selaku Penguji I dan Penguji II yang meluangkan waktu menguji Penulis.
5. Mama, adikku Wiro, adikku Aryo yang telah memberikan semangat dan dukungan tanpa henti. Dan Bundaku Lita yang selalu ada disaat Penulis butuh semangat.
6. Sahabat-sahabat terbaikku mahasiswa Matematika 2002, 2003, 2004, dan 2005 akhirnya saya mengikuti jejak kalian.

7. Sahabat-sahabat mahasiswa Bahasa Inggris UHAMKA yang menjadi kampus kedua ku.
8. Serta semua pihak yang secara langsung maupun tidak langsung membantu penulisan skripsi ini.

Penulis menyadari, masih banyak kekurangan pada penulisan skripsi ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari para pembaca. Penulis berharap semoga skripsi ini dapat bermanfaat.

Jakarta, 17 Juni 2010

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
PENGESAHAN UJIAN	ii
PERNYATAAN	iii
PERSEMBAHAN DAN MOTTO	
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	3
1.3 Pembatasan Masalah	4
1.4 Tujuan Penulisan	4
1.5 Manfaat Penulisan	5
Bab II LANDASAN TEORI	6
2.1 Sejarah Steganografi	6
2.2 Kriteria Steganografi	9
2.3 Terminologi dalam Steganografi	9
2.4 LSB	10
2.5 Perbedaan Steganografi dengan Kriptografi	12
2.6 Citra Digital	13
2.7 Biner	15
2.8 Bit	17
2.9 Byte	18
2.10 Pixel	18
2.11 Model Warna Red Green Blue (RGB)	19
2.12 Matriks	20

Bab III PROSES LSB	22
Bab IV HASIL DAN PEMBAHASAN	27
Bab V KESIMPULAN DAN SARAN	41
Kesimpulan	41
Saran	42
LAMPIRAN	43
REFERENSI	51



DAFTAR GAMBAR

Gambar 2.1	<i>Steganography versus Cryptography</i>	13
Gambar 2.2	Ilustrasi Citra Digital	14
Gambar 2.3	Pewarnaan Dalam RGB	15
Gambar 3.1	Algoritma LSB	23
Gambar 3.2	<i>Type</i> Gambar dan Peletakan Gambar	24
Gambar 3.3	Hasil <i>Run</i> Awal	25
Gambar 4.1	Media Awal (Format *TIFF)	27
Gambar 4.2	Cara Merubah <i>Current Directory</i> Agar Sesuai	28
Gambar 4.3	Proses <i>Run</i> Setelah Pesan Di <i>input</i> 1	29
Gambar 4.4	Proses <i>Run</i> Setelah Pesan Di <i>input</i> 2	30
Gambar 4.5	Hasil Gambar Sebelum Disisipkan Pesan	32
Gambar 4.6	Hasil Gambar yang Telah Disisipkan Pesan	33
Gambar 4.7	<i>File</i> yang Digunakan Untuk Proses Pembuktian	34
Gambar 4.8	Perbandingan <i>Visual</i> Media Awal dengan Hasil Akhir	35
Gambar 4.9	Matriks Media Gambar Sebelum Disisipkan Pesan	36
Gambar 4.10	Matriks Media Gambar Setelah Disisipkan Pesan	37
Gambar 4.11	Hasil <i>run</i> program yang mengubah media gambar yang telah disisipkan pesan menjadi pesan awal	40

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia merupakan makhluk sosial yang saling membutuhkan satu sama lain. Dalam hal berkomunikasi misalnya, tiap manusia pasti membutuhkan komunikasi dengan manusia lainnya. Seiring berkembangnya teknologi informasi saat ini manusia dapat berkomunikasi melalui berbagai media informasi digital. Contoh dengan adanya internet sebagai sistem jaringan terluas yang menghubungkan hampir seluruh komputer di dunia, membuat semua komputer dapat dengan mudah untuk saling bertukar data.

Pertukaran informasi melalui internet memiliki banyak kelebihan dibandingkan dengan media komunikasi lainnya, terutama dari segi kecepatannya. Namun informasi yang dikirimkan melalui internet tidak dapat dijamin keamanannya. Penyadapan terhadap informasi rahasia sering terjadi pada media komunikasi ini. Walaupun sebenarnya ada saluran yang aman telah tersedia, tetapi kecepatan koneksi menggunakan saluran yang aman ini biasanya cenderung lambat.

Oleh karena itu, terdapat beberapa usaha untuk menangani masalah keamanan data rahasia yang dikirimkan melalui internet. Diantaranya adalah menggunakan teknik kriptografi (*cryptography*). Dengan teknik *cryptography* pesan asli (*plaintexts*) yang ingin dikirimkan diubah atau dienkripsi dengan suatu kunci (*key*) menjadi suatu informasi acak (*chiphertexts*) yang tidak bermakna. Kunci

hanya diketahui oleh pengirim dan penerima, kemudian dapat digunakan untuk mengembalikan *cipherteks* ke *plainteks* oleh penerima. Sehingga orang lain yang tidak memiliki hak akses terhadap pesan tersebut tidak dapat mengetahui isi pesan sebenarnya, tetapi hanya mengetahui pesan acaknya saja.

Namun karena sifatnya yang acak, timbul suatu kecurigaan terhadap pesan yang dikirim. Untuk mengatasi hal tersebut dapat digunakan teknik lainnya yaitu teknik steganografi (*steganography*).

Steganography merupakan suatu teknik yang memungkinkan para pengguna untuk menyembunyikan (*embedding*) suatu file atau pesan ke dalam pesan lain. Misalkan dalam suatu gambar disisipkan suatu file atau pesan rahasia, tetapi gambar tersebut tidak terlihat file atau pesan rahasia secara kasat mata. Sedangkan apabila diekstrak dengan suatu *software* khusus, maka akan terlihat bahwa terdapat file atau pesan rahasia dalam gambar tersebut. Dibantu oleh kemajuan teknologi yang semakin canggih, hal ini dapat dengan mudah diaplikasikan. Contohnya dengan bantuan *software-software* seperti *steghide*, *mp3stego*, *HideInsidePicture* dan lainnya.

Dengan teknik tersebut dapat memungkinkan juga untuk menyembunyikan informasi hak cipta seperti identitas seorang pengarang, tanggal ciptaan, dan lain-lain, dengan cara menyisipkan atau menyembunyikan informasi tersebut kedalam berbagai macam variasi jenis dokumen besar seperti teks ataupun gambar.

Pada umumnya menyembunyikan pesan kedalam media gambar ada tiga metode yang dapat digunakan, yaitu *Least Significant Bit* (LSB) dan *Most Significant Bit* (MSB), *Masking* dan *Filtering*, *Discrete Cosine Transformation*

(DCT) dan *Wavelet Compression*. Ketiga metode tersebut mempunyai kelebihan dan kekurangannya masing-masing.

LSB merupakan metode yang dianggap sederhana, mudah dimengerti dan masih digunakan sampai sekarang, yaitu dengan mengganti bit rendah atau bit yang paling kanan pada data *piksel* yang menyusun file tersebut. *Masking* atau *Filtering* merupakan suatu metode yang mirip dengan *watermark*, dimana suatu gambar diberi tanda (*marking*) untuk menyembunyikan file atau pesan rahasia. Sedangkan DCT dan *Wavelet Compression* merupakan metode mentransformasi blok-blok piksel yang berurutan dari gambar.

Pada tulisan ini, penulis mengulas tentang Steganografi menggunakan metode *LSB* dengan bantuan Program MATLAB. Pada metode *LSB*, bit yang diganti adalah bit-bit paling kanan pada data piksel yang menyusun *file*. Bit-bit akhir tersebut diganti dengan bit sebuah pesan yang akan disisipkan kedalamnya. Dengan dasar tersebut penulis membuat tulisan dengan judul **STEGANOGRAFI MENGGUNAKAN METODE LSB DENGAN SOFTWARE MATLAB**

1.2 Permasalahan

Permasalahan dirumuskan sebagai berikut.

1. Bagaimana menyembunyikan teks dalam proses *steganography* dengan menggunakan metode LSB.
2. Bagaimana perubahan dalam *file* hasil keluaran dari segi kualitas gambar *file* sebelum dan sesudah disisipkan pesan teks.

1.3 Pembatasan Masalah

Dalam penyusunan skripsi ini, penulis hanya mempelajari teknik-teknik menyembunyikan *file* atau pesan rahasia kedalam media gambar dengan menggunakan Matlab2009 sebagai alat bantu peletakkan bit yang akan diganti oleh bit dari *file* yang akan disamarkan. Skripsi ini hanya akan membahas bagaimana konsep mengganti tiap bit yang ada dalam gambar dan mensubstitusikan dengan bit dari *file* yang akan kita sembunyikan.

Skripsi ini merubah gambar menjadi matriks RGB, kemudian tiap elemen dari masing-masing kolom dan baris matriks tersebut dijadikan deretan binari 8 digit. Setelah itu merubah pesan yang akan disisipkan kedalam gambar tersebut menjadi binari. Setelah semua dalam binari langkah selanjutnya merubah binari terakhir pada media menjadi binari dari pesan. Langkah terakhir merubah kembali deretan binari tersebut menjadi sebuah gambar. Semua dilakukan dengan bantuan Software MATLAB.

1.4 Tujuan Penulisan

Tujuan Penulisan adalah sebagai berikut.

1. Mengetahui perubahan yang terjadi terhadap hasil keluaran dari komponen-komponen *steganography* dalam segi kualitas dari *file* atau pesan rahasia.

2. Mengaplikasikan *steganography* dengan metode LSB menggunakan bantuan Software MATLAB

1.5 Manfaat Penulisan

Manfaat dari penulisan ini yaitu dapat disembunyikannya suatu *file* atau pesan rahasia kedalam media seperti gambar. Dengan menggunakan metode yang sudah ada. Diharapkan akan mampu muncul ide-ide lain dalam menyembunyikan pesan ini.

Adapun *software* umum tentang *steganography* yang kini telah berkembang seperti *mp3stego*, *OutGuest*, *steghide*, *steganography* v1.8, dan lainnya dapat diunduh dan digunakan secara komersial maupun untuk keperluan pembelajaran.

BAB II

LANDASAN TEORI

2.1 Sejarah Steganografi

Steganografi (*steganography*) berasal dari Bahasa Yunani, yaitu “*steganos*” yang artinya menyembunyikan dan “*grapto*” yang artinya tulisan. Jadi Steganografi berarti juga tulisan yang disembunyikan. Steganografi adalah teknik menyembunyikan suatu informasi yang rahasia atau sensitif pada suatu media perantara agar tidak terlihat seperti semestinya.

Secara umum steganografi dapat didefinisikan sebagai ilmu atau seni yang digunakan untuk menyembunyikan pesan rahasia dengan teknik-teknik tertentu sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari pesan rahasia tersebut.

Steganografi merupakan seni penyembunyian pesan ke dalam pesan lainnya sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu di dalam pesan tersebut. Kata steganografi (*steganography*) berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah “menulis tulisan yang tersembunyi atau terselubung. Teknik ini meliputi banyak sekali metoda komunikasi untuk menyembunyikan pesan rahasia. Metoda ini termasuk tinta yang tidak tampak, microdots, pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi spektrum lebar.

Catatan pertama tentang steganografi ditulis oleh seorang sejarawan Yunani, Herodotus, yaitu ketika Histaeus seorang raja kejam Yunani dipenjarakan oleh Raja Darius di Susa pada abad 5 Sebelum Masehi. Histaeus harus mengirim pesan rahasia kepada anak laki-lakinya, Aristagoras, di Militus. Histaeus menulis pesan dengan cara mentato pesan pada kulit kepala seorang budak dan ketika rambut budak itu mulai tumbuh, Histaeus mengutus budak itu ke Militus untuk mengirim pesan di kulit kepalanya tersebut kepada Aristagoras.

Cerita lain tentang steganografi datang juga dari sejarawan Yunani, Herodotus, yaitu dengan cara menulis pesan pada papan kayu yang ditutup dengan lilin. Demeratus, seorang Yunani yang akan mengabarkan berita kepada Sparta bahwa Xerxes bermaksud menyerbu Yunani. Agar tidak diketahui pihak Xerxes, Demaratus menulis pesan dengan cara mengisi tabung kayu dengan lilin dan menulis pesan dengan cara mengukirnya pada bagian bawah kayu, lalu papan kayu tersebut dimasukkan ke dalam tabung kayu, kemudian tabung kayu ditutup kembali dengan lilin. Teknik steganografi yang lain adalah tinta yang tak terlihat. Teknik ini pertama digunakan pada zaman Romawi kuno yaitu dengan menggunakan air sari buah jeruk, urine atau susu sebagai tinta untuk menulis pesan. Cara membacanya adalah dengan dipanaskan di atas nyala lilin, tinta yang sebelumnya tidak terlihat, ketika terkena panas akan berangsur-angsur menjadi gelap, sehingga pesan dapat dibaca. Teknik ini pernah juga digunakan pada Perang Dunia II.

Pada abad 20, steganografi benar-benar mengalami perkembangan. Selama berlangsung perang Boer, Lord Boden Powell (pendiri gerakan kepanduan) yang

bertugas untuk membuat tanda posisi sasaran dari basis artileri tentara Boer, untuk alasan keamanan, Boden Powell menggambar peta-peta posisi musuh pada sayap kupu-kupu agar gambar-gambar peta sasaran tersebut terkamuflase.

Perang Dunia II adalah periode pengembangan teknik-teknik baru steganografi. Pada awal Perang Dunia II walaupun masih digunakan teknik tinta yang tak terlihat, namun teknik-teknik baru mulai dikembangkan seperti menulis pesan rahasia ke dalam kalimat lain yang tidak berhubungan langsung dengan isi pesan rahasia tersebut, kemudian teknik menulis pesan rahasia ke dalam pita koreksi karbon mesin ketik, dan juga teknik menggunakan pin berlubang untuk menandai kalimat terpilih yang digunakan dalam pesan, teknik terakhir adalah microdots yang dikembangkan oleh tentara Jerman pada akhir Perang Dunia II.

Dari contoh-contoh steganografi konvensional tersebut dapat dilihat bahwa semua teknik steganografi konvensional berusaha merahasiakan komunikasi dengan cara menyembunyikan pesan ataupun mengkamufase pesan. Maka sesungguhnya prinsip dasar dalam steganografi lebih dikonsentrasikan pada kerahasiaan komunikasinya bukan pada datanya.

Seiring dengan perkembangan teknologi terutama teknologi komputasi, steganografi merambah juga ke media digital, walaupun steganografi dapat dikatakan mempunyai hubungan erat dengan kriptografi, tetapi kedua metode ini sangat berbeda.

1.2 Kriteria Steganografi

Kriteria steganografi yang bagus yakni sebagai berikut :

1. *Imperceptibility*

Keberadaan pesan tidak dapat dipersepsi oleh inderawi. Jika pesan disisipkan ke dalam sebuah citra, citra yang telah disisipi pesan harus tidak dapat dibedakan dengan citra asli oleh mata. Begitu pula dengan suara, seharusnya tidak terdapat perbedaan antara suara asli dengan suara yang telah disisipi pesan.

2. *Fidelity*

Mutu media penampung (*cover-object*) tidak berubah banyak akibat penyisipan (*embedded*). Perubahan yang terjadi harus tidak dapat dipersepsi oleh inderawi.

3. *Recovery*

Pesan yang disembunyikan harus dapat diungkap kembali. Tujuan steganografi adalah menyembunyikan informasi, maka sewaktu-waktu informasi yang disembunyikan ini harus dapat diambil kembali untuk dapat digunakan lebih lanjut sesuai keperluan.

1.3 Terminologi dalam Steganografi

Ada beberapa terminologi dari steganografi yang memang harus dipahami antara lain sebagai berikut :

1. *Embedded Message* (Hiddentext)

Pesan atau informasi yang disembunyikan. Contohnya dapat berupa teks, gambar, audio, video, dll.

2. *Cover-object* (Coverttext)

Pesan yang digunakan untuk menyembunyikan *embedded message*.

Contohnya dapat berupa teks, gambar, audio, video, dll.

3. *Stego-object* (Stegotext)

Pesan yang sudah berisi pesan *embedded message*.

4. *Stego-key*

Kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari *stegotext*.

1.4 LSB

Cara paling umum untuk menyembunyikan pesan adalah dengan memanfaatkan *Least Significant Bit (LSB)*. Walaupun banyak kekurangan pada metode ini, tetapi kemudahan implementasinya membuat metode ini tetap digunakan sampai sekarang. Metode ini membutuhkan syarat, yaitu jika dilakukan kompresi pada *stego*, harus digunakan format *lossless compression*, karena metode ini menggunakan bit-bit pada setiap piksel pada gambar. Jika digunakan format *lossy compression*, pesan rahasia yang disembunyikan dapat hilang. Jika digunakan gambar 24 bit warna sebagai *cover*, sebuah bit dari masing-masing komponen *Red*, *Green*, dan *Blue*, dapat digunakan sehingga 3 bit dapat disimpan pada setiap piksel. Sebuah gambar 800x600 piksel dapat digunakan untuk menyembunyikan 1.440.000 bit (180.000 *Byte*) data rahasia.

Perubahan *LSB* ini akan terlalu kecil untuk terdeteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Jika digunakan gambar 8 bit

color sebagai *cover*, hanya 1 bit saja dari setiap piksel warna yang dapat dimodifikasi sehingga pemilihan gambar harus dilakukan dengan sangat hati-hati, karena perubahan *LSB* dapat mengakibatkan terjadinya perubahan warna yang ditampilkan pada citra. Akan lebih baik jika gambar berupa gambar *grayscale* karena perubahan warnanya akan lebih sulit dideteksi oleh mata manusia. Proses ekstraksi pesan dapat dengan mudah dilakukan dengan mengekstrak *LSB* dari masing-masing piksel pada *stego* secara berurutan dan menuliskannya ke *output file* yang akan berisi pesan tersebut.

Misalnya pesan yang ingin disisipkan adalah A dalam 3 piksel dan asumsikan tidak ada kompresi. *Raster* data asli untuk 3 piksel (9 *Byte*) menjadi :

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

Nilai biner untuk A adalah 10000011. Sisipkan nilai biner untuk A dalam tiga piksel, maka akan dihasilkan :

(00100111 11101000 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

Bit-bit yang digaris bawah hanya tiga perubahan secara aktual dalam 8 *Byte* yang digunakan. Secara rata-rata, *LSB* membutuhkan hanya setengah bit dalam suatu perubahan suatu gambar.

Contoh lain penyisipan pada gambar 8-bit tidak berfungsi secara optimal karena keterbatasan warnanya. Sebagai bahan perbandingan dapat dijelaskan sebagai berikut :

Suatu *palette* sederhana empat warna dari putih, merah, biru dan hijau mempunyai posisi masukan *palette* yang sesuai secara berturut-turut dari 0(00), 1(01), 2(10), dan 3(11).

Misalkan nilai *raster* dari empat piksel bersebelahan yaitu :

(00 00 10 10)

putih putih biru biru

pesan atau nilai biner yang ingin disisipkan yaitu 1010, maka akan dihasilkan :

(01 00 11 10)

merah putih hijau biru

Kekurangan dari metode modifikasi *LSB* ini adalah bahwa metode ini membutuhkan “tempat penyimpanan” yang relatif besar. Kekurangan lain adalah bahwa *stego* yang dihasilkan tidak dapat dikompres dengan format *lossy compression*.

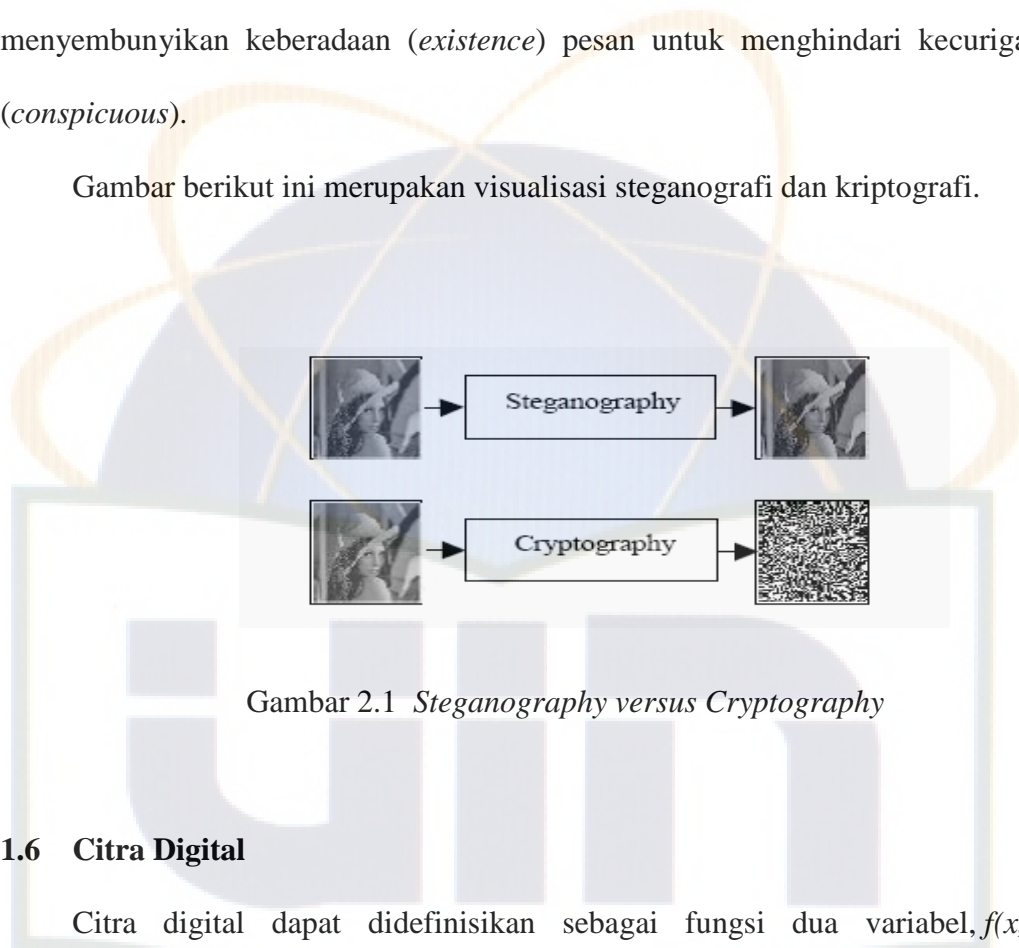
1.5 Perbedaan Steganografi dengan Kriptografi

Steganografi merupakan pelengkap dari kriptografi bukan pengganti. Sebab dari kedua disiplin ilmu tersebut dapat digunakan kedua konsep secara bersamaan ataupun secara terpisah. Seperti halnya pesan yang telah terenkripsi disembunyikan ke dalam suatu media gambar. Proses enkripsi merupakan teknik

dalam ilmu kriptografi sedangkan menyembunyikan pesan yang telah terenkripsi merupakan teknik dalam ilmu steganografi.

Dari segi tujuan kriptografi bertujuan untuk menyembunyikan isi (*content*) pesan agar pesan tidak dapat dibaca. Sedangkan steganografi bertujuan untuk menyembunyikan keberadaan (*existence*) pesan untuk menghindari kecurigaan (*conspicuous*).

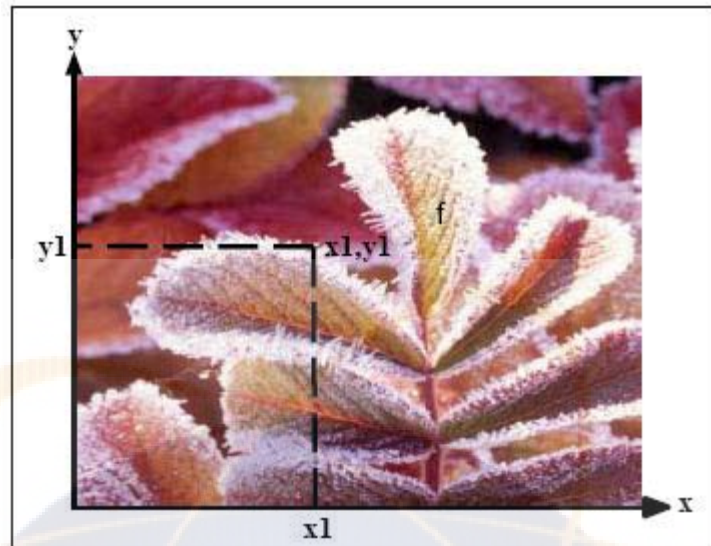
Gambar berikut ini merupakan visualisasi steganografi dan kriptografi.



Gambar 2.1 *Steganography versus Cryptography*

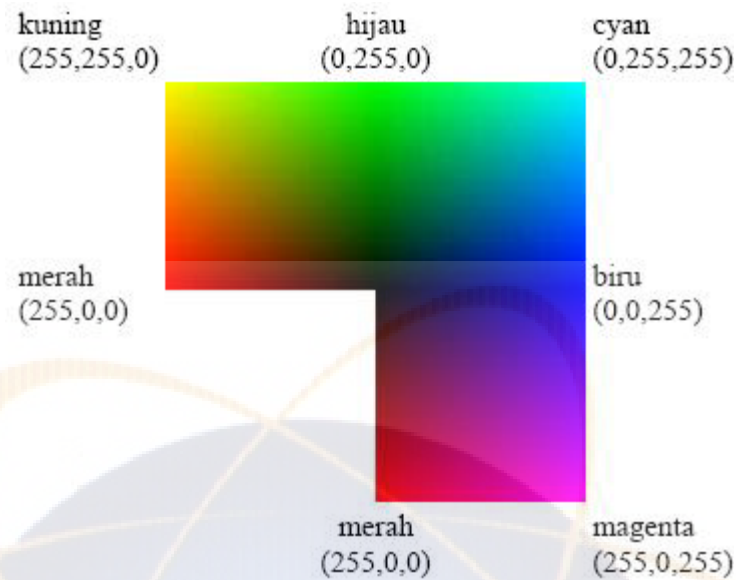
1.6 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada gambar dibawah ini . Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*).



Gambar 2.2 Ilustrasi Citra Digital

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 16777216$ jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-*x*, komponen-*y* dan komponen-*z*. Misalkan sebuah vektor dituliskan sebagai $\mathbf{r} = (x, y, z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen *R*(ed), *G*(reen), *B*(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = *RGB*(30, 75, 255). Putih = *RGB* (255,255,255), sedangkan untuk hitam= *RGB*(0,0,0).




Gambar 2.3 Pewarnaan dalam RGB

1.7 Biner

Sistem bilangan biner atau sistem bilangan basis dua adalah sebuah sistem penulisan angka dengan menggunakan dua simbol yaitu 0 dan 1. Sistem bilangan biner modern ditemukan oleh Gottfried Wilhelm Leibniz pada abad ke-17. Sistem bilangan ini merupakan dasar dari semua sistem bilangan berbasis digital. Dari sistem biner, kita dapat mengkonversinya ke sistem bilangan *Oktal* atau *Hexadesimal*. Sistem ini juga dapat kita sebut dengan istilah bit, atau *Binary Digit*. Pengelompokan biner dalam komputer selalu berjumlah 8, dengan istilah 1 Byte. Dalam istilah komputer, 1 Byte = 8 bit. Kode-kode rancang bangun komputer, seperti ASCII, *American Standard Code for Information Interchange* menggunakan sistem peng-kode-an 1 Byte.

Bilangan desimal yang dinyatakan sebagai bilangan biner akan berbentuk sebagai berikut:



Desimal	Biner (8 bit)
0	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
4	0000 0100
5	0000 0101
6	0000 0110
7	0000 0111
8	0000 1000
9	0000 1001
10	0000 1010
11	0000 1011
12	0000 1100
13	0000 1101
14	0000 1110
15	0000 1111
16	0001 0000

Tabel 2.1 Tabel Biner

$$2^0=1, 2^1=2, 2^2=4, 2^3=8, 2^4=16, 2^5=32, 2^6=64, \text{ dst}$$

Contoh: Mengubah bilangan desimal menjadi biner

Desimal = 10.

Berdasarkan referensi diatas yang mendekati bilangan 10 adalah 8 (2^3), selanjutnya hasil pengurangan $10-8 = 2$ (2^1). sehingga dapat dijabarkan seperti berikut:

$$10 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0).$$

Dari perhitungan di atas bilangan biner dari 10 adalah 1010.

Dapat juga dengan cara lain yaitu $10 : 2 = 5$ sisa 0 (0 akan menjadi angka terakhir dalam bilangan biner), 5 (hasil pembagian pertama) : 2 = 2 sisa 1 (1 akan menjadi angka kedua terakhir dalam bilangan biner), 2 (hasil pembagian kedua): 2 = 1 sisa 0 (0 akan menjadi angka ketiga terakhir dalam bilangan biner), 1 (hasil pembagian ketiga): 2 = 0 sisa 1 (0 akan menjadi angka pertama dalam bilangan biner) karena hasil bagi sudah 0 atau habis, sehingga bilangan biner dari 10 = 1010

Dengan cara yang singkat $10:2=5(0)$, $5:2=2(1)$, $2:2=1(0)$, $1:2=0(1)$ sisa hasil bagi dibaca dari belakang menjadi 1010.

1.8 Bit

Bit merujuk pada sebuah [digit](#) dalam [sistem angka biner](#) (basis 2). Sebagai contoh, angka 1001011 memiliki panjang 7 bit. Digit biner hampir selalu digunakan sebagai [satuan](#) terkecil dalam penyimpanan dan komunikasi informasi di dalam [teori komputasi](#) dan [informasi digital](#). Teori informasi juga sering

menggunakan digit natural, disebut *nit* atau *nat*. Sementara, [komputasi kuantum](#) menggunakan satuan [qubit](#), sebuah potongan informasi dengan kemungkinan informasi tersebut bernilai benar.

Bit juga digunakan sebagai satuan ukuran, yaitu kapasitas informasi dari sebuah digit biner. Lambang yang digunakan adalah bit, dan kadang-kadang (secara tidak resmi) b (contohnya, modem dengan kecepatan 56 kbps atau 56 kilo bit per second/detik). Satuan ini dikenal juga sebagai shannon, dengan lambang Sh.

1.9 Byte

Bit ([Bahasa Inggris: Byte](#)) adalah istilah yang biasa dipergunakan sebagai satuan dari [penyimpanan](#) data dalam [komputer](#). Satu bita terdiri dari delapan [bit](#).

Huruf B digunakan dalam singkatan kepada Byte. (bit menggunakan singkatan b.) seperti kB = kilobita. [Cakram keras](#) (*hard disk*) berkapasitas 40GB secara mudahnya bermaksud cakram keras tersebut mampu menyimpan hingga 40 ribu juta (milyar) bita atau gigabita data.

1.10 Piksel

Piksel adalah unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per [inci](#).

Piksel sendiri berasal dari akronim bahasa Inggris *Picture Element* yang disingkat menjadi *Pixel*. Pada ujung tertinggi skala [resolusi](#), mesin cetak gambar berwarna dapat menghasilkan hasil cetak yang memiliki lebih dari 2.500 titik per

inci dengan pilihan 16 juta warna lebih untuk setiap inci, dalam istilah komputer berarti gambar seluas satu inci persegi yang bisa ditampilkan pada tingkat resolusi tersebut sepadan dengan 150 juta bit informasi.

Monitor atau layar datar yang sering kita temui terdiri dari ribuan piksel yang terbagi dalam baris-baris dan kolom-kolom. Jumlah piksel yang terdapat dalam sebuah monitor dapat kita ketahui dari resolusinya. Resolusi maksimum yang disediakan oleh monitor adalah 1024x768, maka jumlah piksel yang ada dalam layar monitor tersebut adalah 786432 piksel. Semakin tinggi jumlah piksel yang tersedia dalam monitor, semakin tajam gambar yang mampu ditampilkan oleh monitor tersebut.

1.11 Model Warna *Red Green Blue (RGB)*

Model warna *RGB* adalah sebuah model warna tambahan dalam jenis merah, hijau, dan biru muda yang ditambahkan secara bersama dalam berbagai cara untuk memproduksi sebuah kesatuan warna secara luas. Nama dari model ini berasal dari inisial ketiga zat warna primer, yaitu *Red* (merah), *Green* (hijau), dan *Blue* (biru).

Tujuan utama model warna *RGB* adalah untuk menyajikan, dan menampilkan gambar di dalam sistem elektronik, seperti televisi dan komputer, dan digunakan pula pada fotografi konvensional. Sebelum zaman elektronik, model warna *RGB* telah mempunyai suatu teori yang kuat di belakang itu, yang didasarkan persepsi manusia terhadap warna.

Tipe alat yang menggunakan input *RGB* adalah televisi, kamera video, *scanner*, dan kamera digital. Tipe alat yang menggunakan output *RGB* adalah televisi satuan dengan berbagai teknologi (*CRT*, *LCD*, plasma), komputer, dan layar telepon genggam, proyektor video, dan layar besar seperti Jumbotron, dan lain-lain. Warna *printer*, bukanlah *RGB*, tetapi warna *subtractive* (model warna *CMYK*).

1.12 Matriks

Matriks adalah suatu kumpulan besaran (variabel dan konstanta) yang dapat dirujuk melalui indeksnya, yang menyatakan posisinya dalam representasi umum yang digunakan, yaitu sebuah tabel persegipanjang. Matriks merupakan suatu cara visualisasi variabel yang merupakan kumpulan dari angka-angka atau variabel lain, misalnya vektor. Dengan representasi matriks, perhitungan dapat dilakukan dengan lebih terstruktur. Pemanfaatannya misalnya dalam menjelaskan persamaan linier, transformasi koordinat, dan lainnya. Matriks seperti halnya variabel biasa dapat dimanipulasi, seperti dikalikan, dijumlah, dikurangkan dan didekomposisikan.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Penjumlahan dan pengurangan matriks dapat dilakukan dengan mengoperasikan komponen matriks pada letak yang sama, atau dilambangkan dengan

$$a_{ij} \pm b_{ij} = c_{ij}$$

atau dalam representasi dekoratifnya

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \pm \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$$

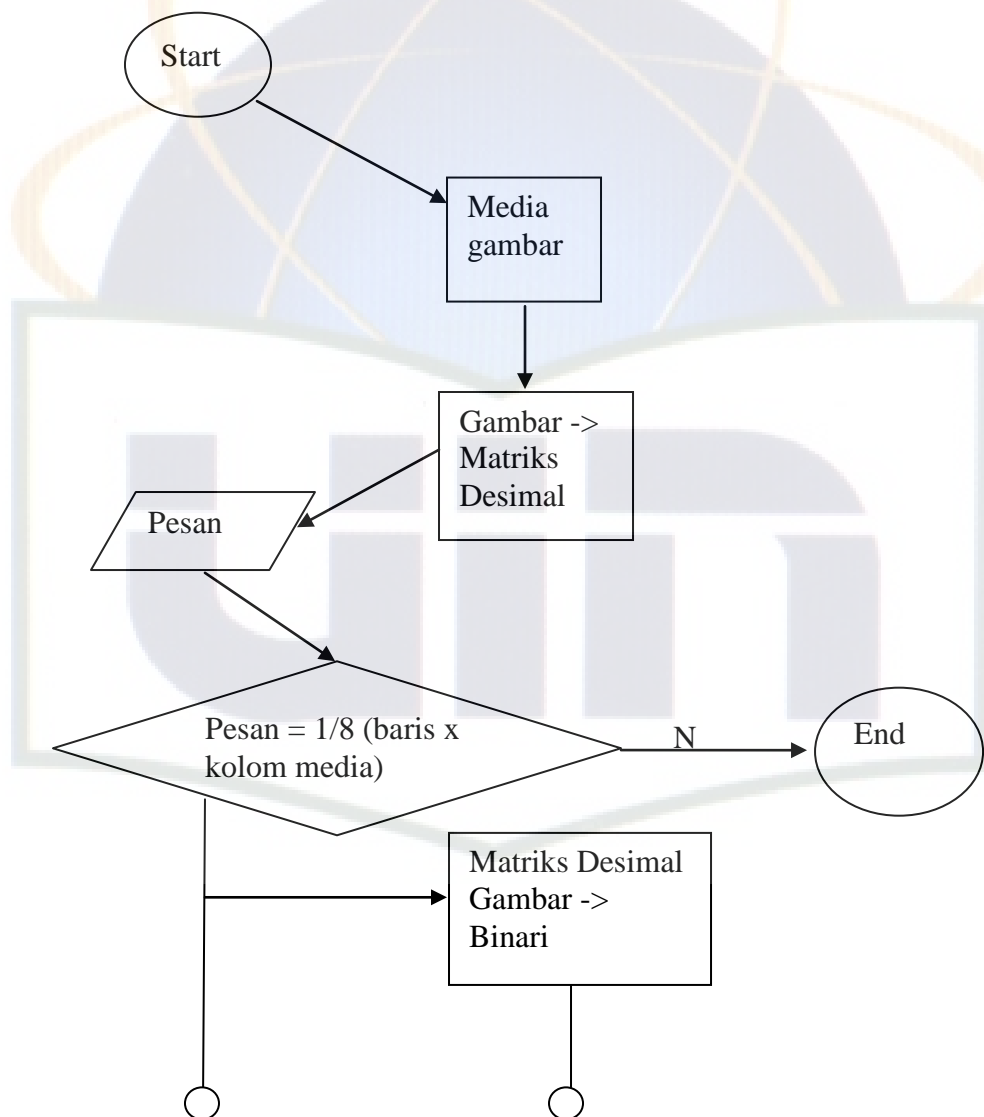
$$\begin{bmatrix} (a_{11} \pm b_{11}) & (a_{12} \pm b_{12}) & (a_{13} \pm b_{13}) \\ (a_{21} \pm b_{21}) & (a_{22} \pm b_{22}) & (a_{23} \pm b_{23}) \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$$

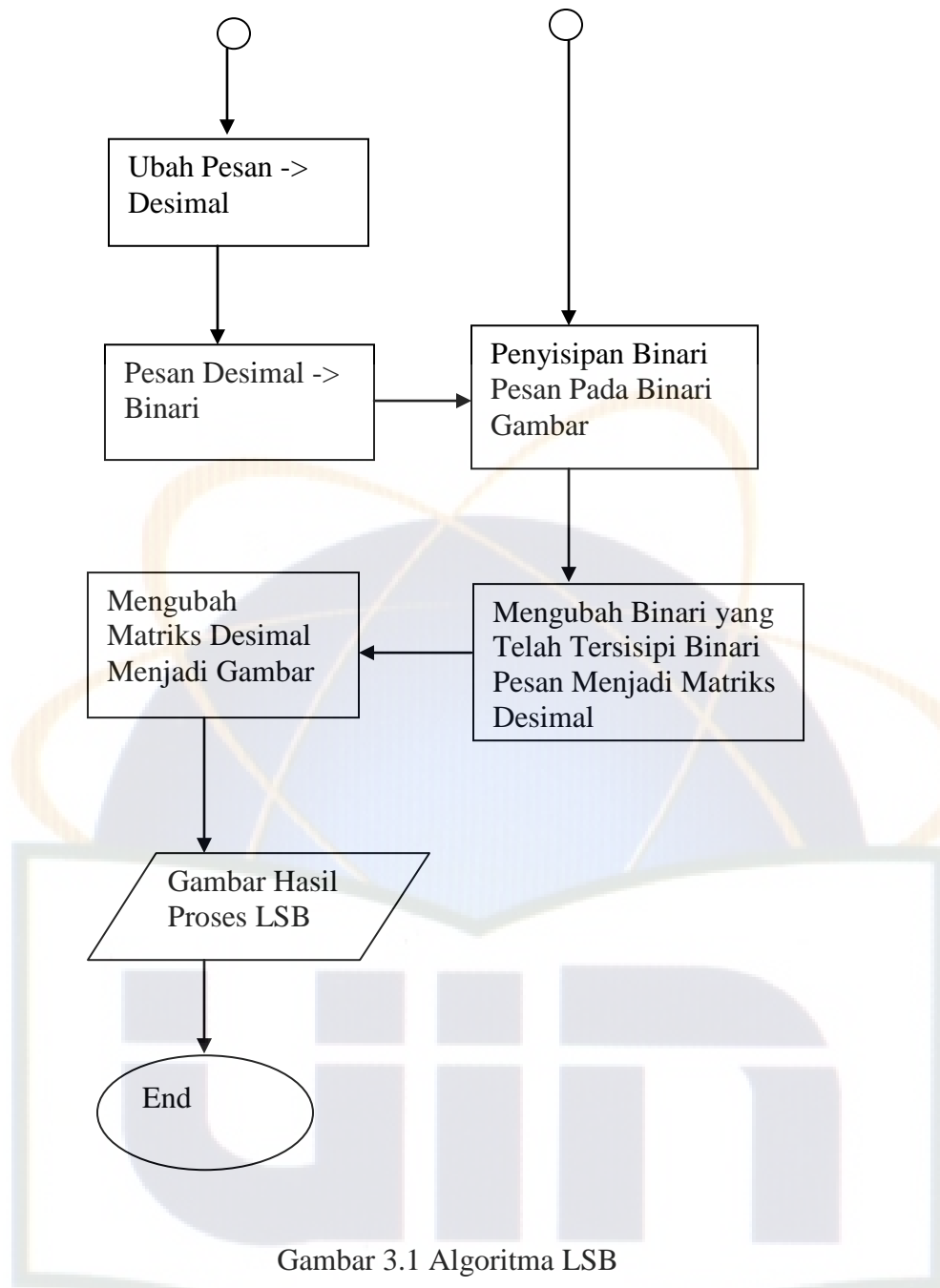
BAB III

PROSES LSB

Pada bab 3 ini penulis akan memaparkan cara kerja dari algoritma yang telah dibuat dengan menggunakan bantuan Matlab2009.

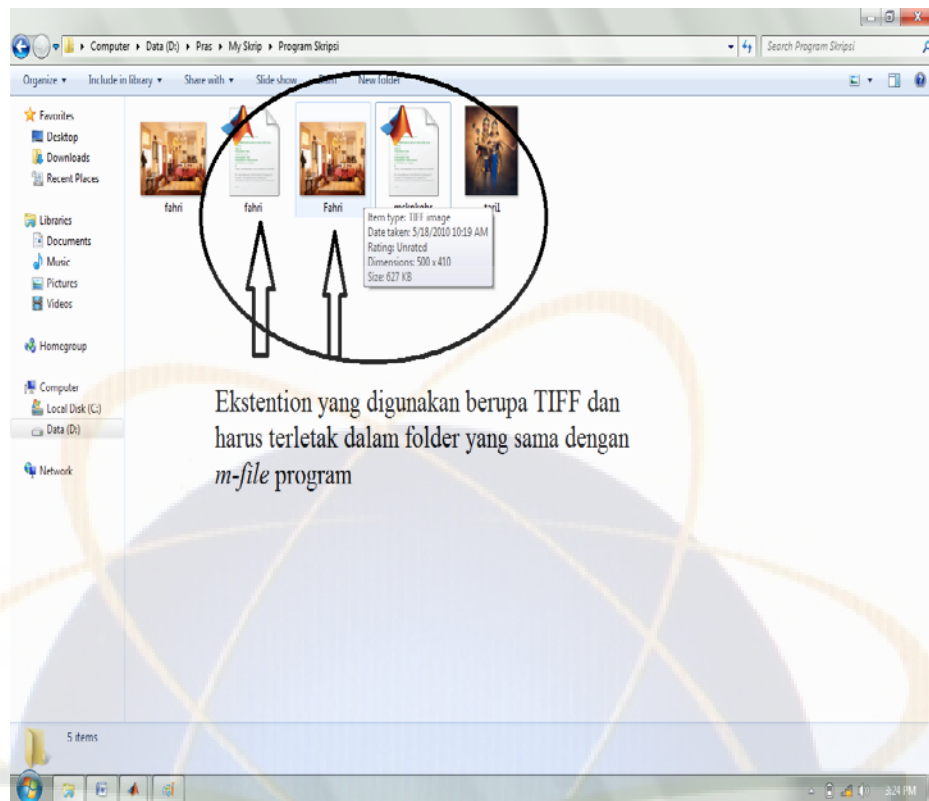
Flowchart dari algoritma metode LSB ini adalah sebagai berikut:





Gambar 3.1 Algoritma LSB

Proses *Least Significant Bit* (LSB) pada skripsi ini adalah sebagai berikut. Langkah pertama pilih *file* gambar yang akan digunakan sebagai media untuk penyisipan. Pada skripsi ini *format file* yang digunakan adalah *file* gambar dengan *ekstention* *TIFF. Letakkan file yang akan digunakan sebagai media tersebut dalam satu *folder* bersama dengan *m-file* matlab LSB.



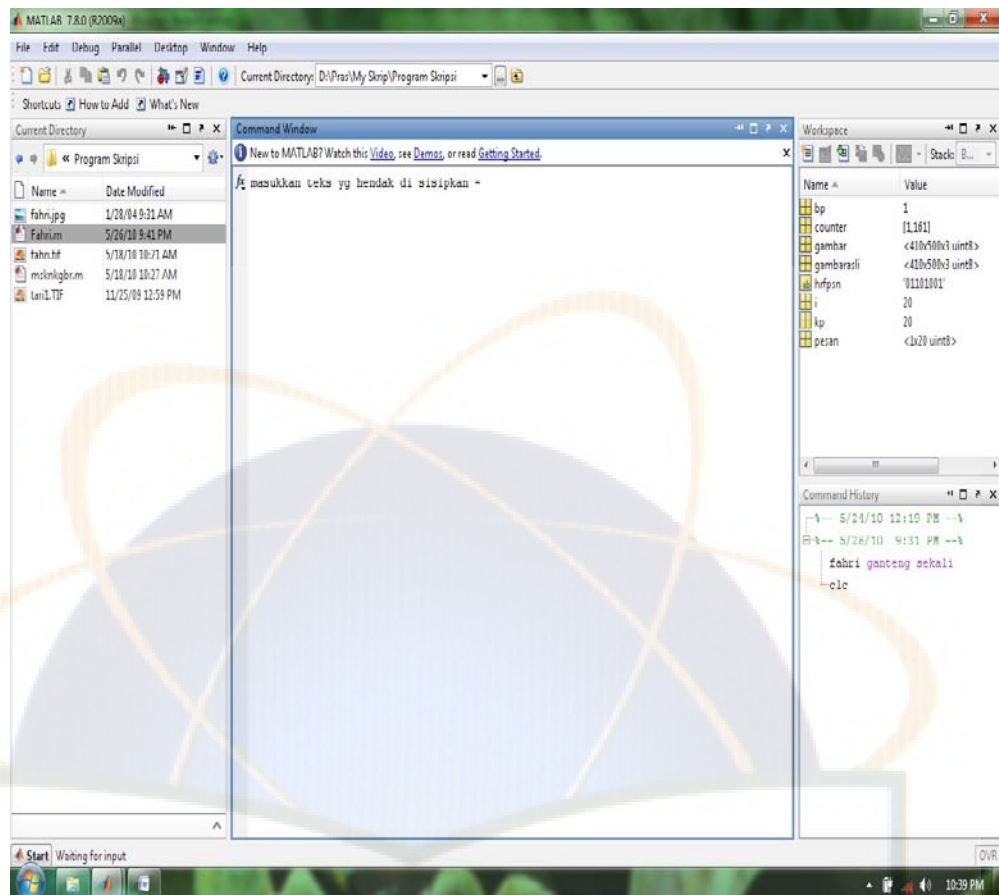
Ekstention yang digunakan berupa TIFF dan harus terletak dalam folder yang sama dengan *m-file* program

Gambar 3.2 *Type* gambar dan peletakan gambar

Nama *file* yang akan digunakan sebagai media tersebut harus sama dengan nama *file* yang terdapat dalam *m-file*.

Media akan dirubah menjadi 3 matriks RGB. Masing-masing elemen pada matriks dalam angka 0-255 yang merepresentasikan indeks warna RGB dalam skala 8-bit. Tiap-tiap angka tersebut selanjutnya dirubah menjadi 8 digit binari yang akan digunakan dalam proes penyisipan.

Langkah selanjutnya jalankan *m-file* yang telah dibuat dengan program Matlab. Pada saat dijalankan program akan meminta pesan yang akan disisipkan kedalam media yang telah tersedia.



Gambar 3.3 Hasil *run* awal

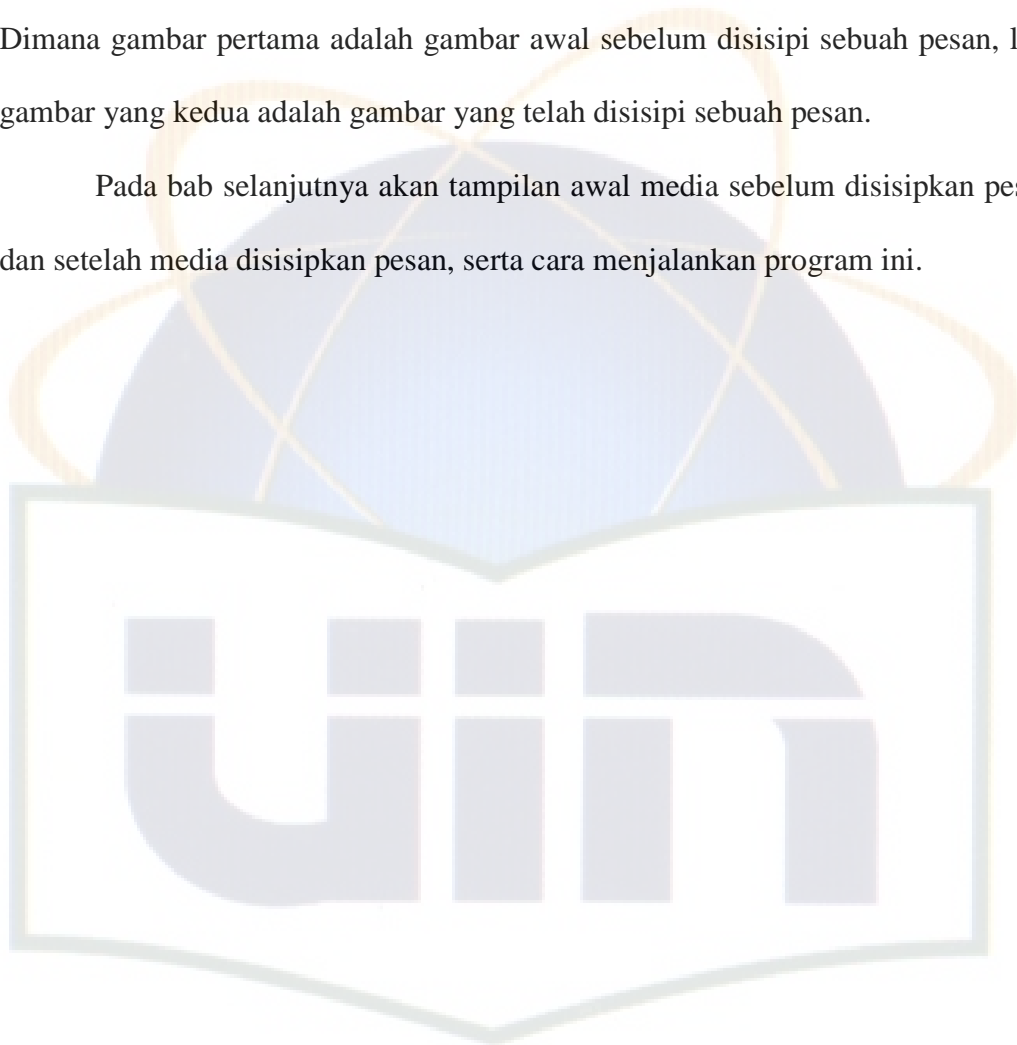
Masukkan pesan yang akan disisipkan. Setelah dimasukan pesan, program ini akan merubah pesan yang telah di input menjadi bilangan ASCII, setelah menjadi bilangan ASCII, program akan merubah bilangan ASCII tersebut menjadi binari-binari yang akan disisipkan kedalam binari-binari terakhir media.

Setelah pesan menjadi deretan binari, langkah selanjutnya adalah mensubtitusikan tiap binari akhir pada media awal dengan tiap binari pada pesan. Dalam program ini dilakukan dalam matriks merah.

Setelah binari-binari akhir pada media tersubtitusikan dengan binari pesan, maka media yang telah tersubtitusi binari-binari akhirnya dirubah kembali

menjadi angka-angka dari 0-255. Proses yang dilakukan ini adalah kebalikan dari langkah awal yang dilakukan pada awal proses. Setelah menjadi angka-angka skala 0-255, langkah selanjutnya adalah merubah kembali angka-angka tersebut menjadi sebuah gambar. Hasil *output* dari program ini adalah 2 buah gambar. Dimana gambar pertama adalah gambar awal sebelum disisipi sebuah pesan, lalu gambar yang kedua adalah gambar yang telah disisipi sebuah pesan.

Pada bab selanjutnya akan tampilan awal media sebelum disisipkan pesan dan setelah media disisipkan pesan, serta cara menjalankan program ini.



BAB IV

HASIL DAN PEMBAHASAN

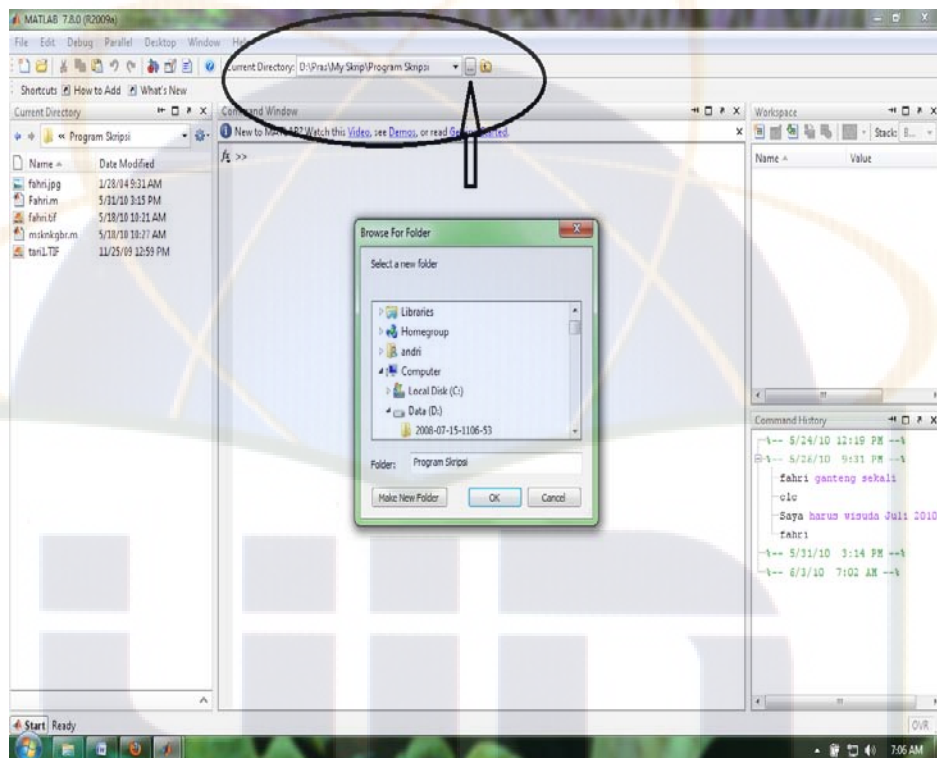
Pada Bab IV ini, penulis akan menunjukkan tampilan dari Matlab dari proses input pesan yang akan disisipkan hingga hasil yang diperoleh.

Gambar yang digunakan pada proses ini adalah gambar berekstension TIFF. Namun tidak menutup kemungkinan jika ingin menggunakan program dengan ekstension lain. Berikut ini adalah media gambar awal yang akan digunakan dalam proses kali ini.



Gambar 4.1 Media awal (format *TIFF)

Pada bab sebelumnya, dikatakan bahwa gambar yang akan digunakan sebagai media harus satu *folder* dengan *m-file* dari program yang akan dijalankan. Langkah awal untuk memulai adalah masuk Matlab2009. Lalu pastikan *current directory* yang tertera sesuai dengan peletakan *m-file* beserta *file* gambar medianya.

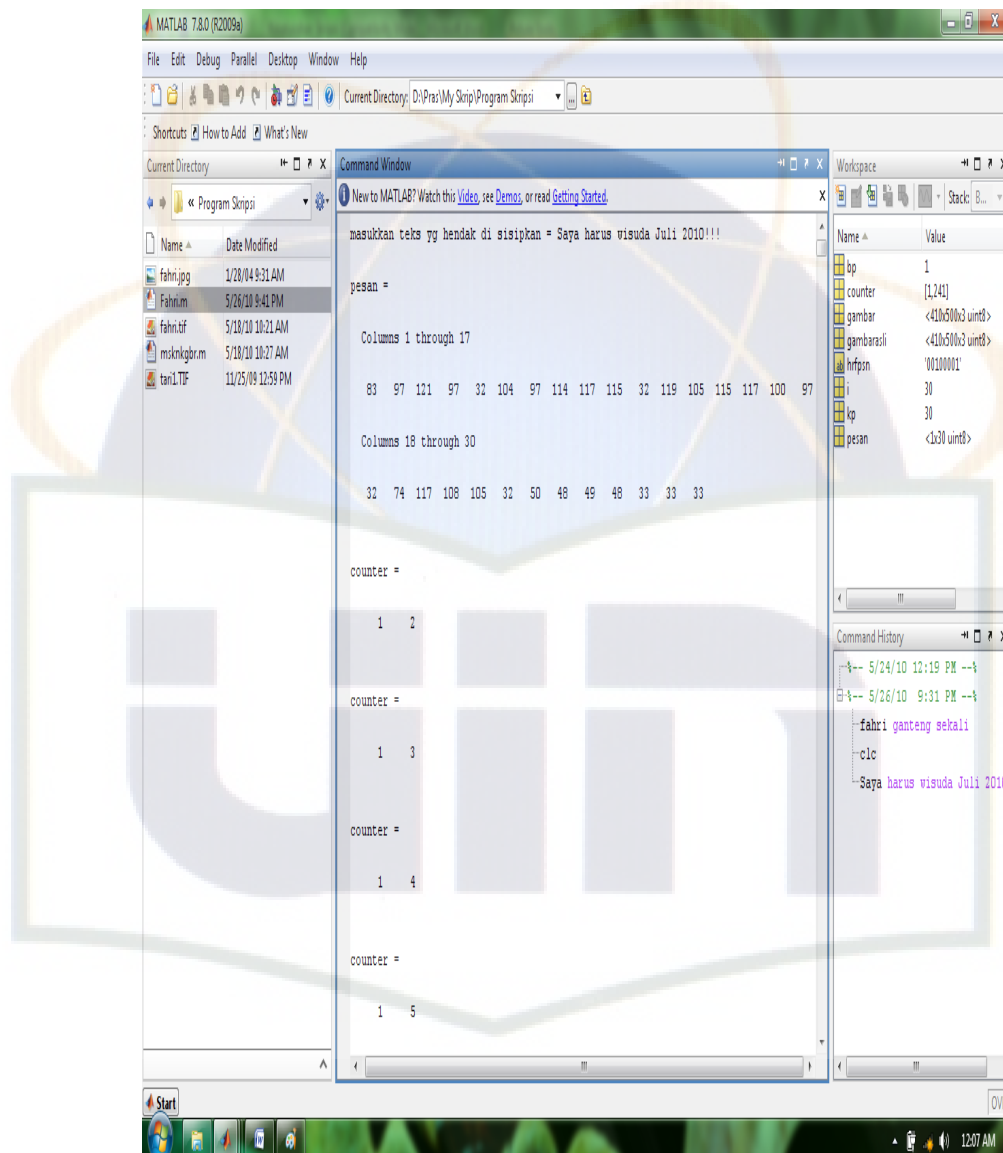


Gambar 4.2 Cara merubah *current directory* agar sesuai

Setelah *current directory* sesuai dengan peletakan *m-file* dan *file* media gambar, akan tampak pada bagian kiri layar *m-file* serta *file* media gambar yang akan digunakan pada skripsi ini.

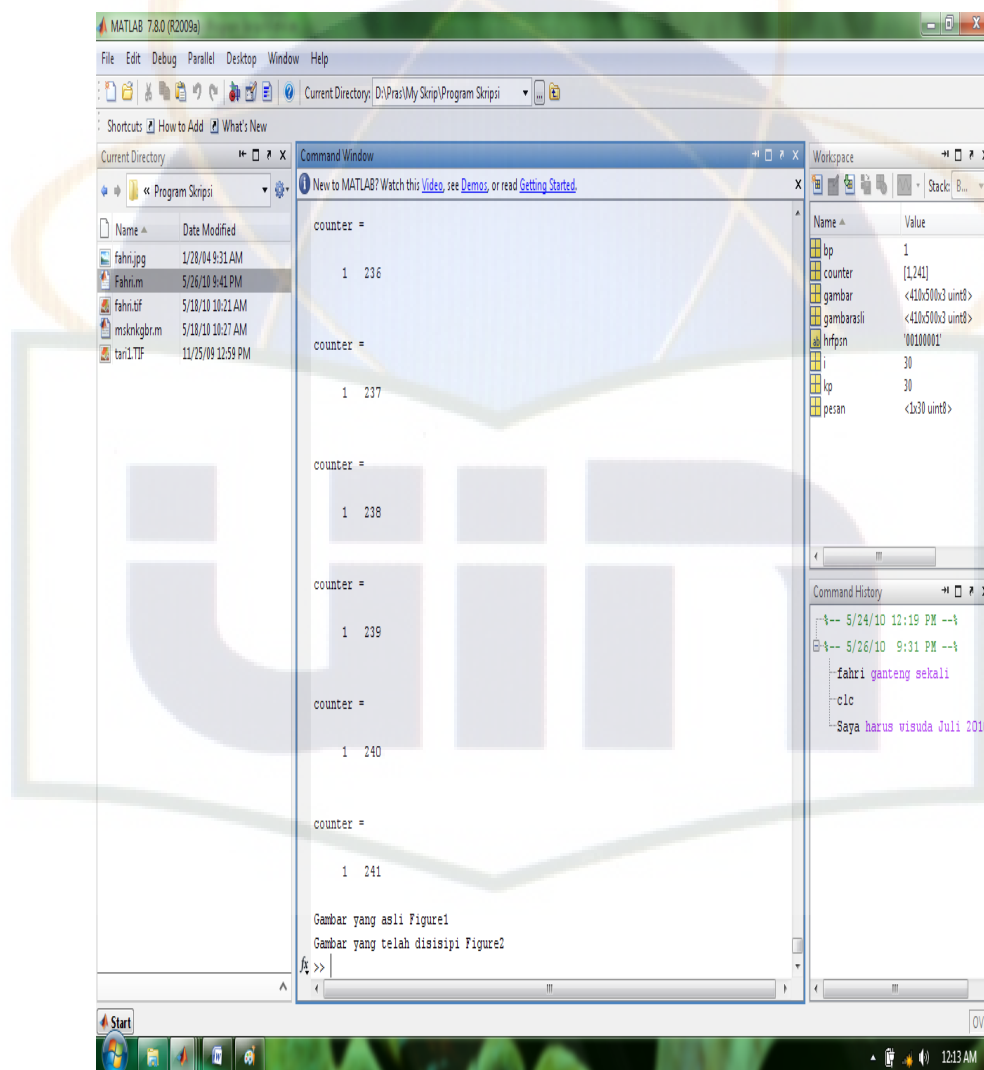
Langkah selanjutnya adalah tekan dua kali pada *m-file*, sehingga bahasa program yang digunakan akan terlihat. Setelah itu jalan kan dengan cara menekan tombol “F5” yang terdapat pada *keyboard*.

Setelah menjalankan bahasa program tersebut, tampilan yang akan terlihat adalah seperti pada Gambar 4.3. Pada gambar 4.3, pesan yang akan disisipkan kedalam media ini adalah: “Saya harus wisuda Juli 2010!!!”. Tampilan proses input pesan adalah sebagai berikut:



Gambar 4.3 Proses run setelah pesan diinput 1.

Deretan angka yang terlihat pada *command window*, yaitu 83 97 121 97 32 104 97 114 117 115 32 119 105 115 117 100 97 32 74 117 108 105 32 50 48 49 48 33 33 33 adalah representasi dari pesan awal yang telah dirubah menjadi bilangan ASCII sebelum dirubah menjadi binari-binari.

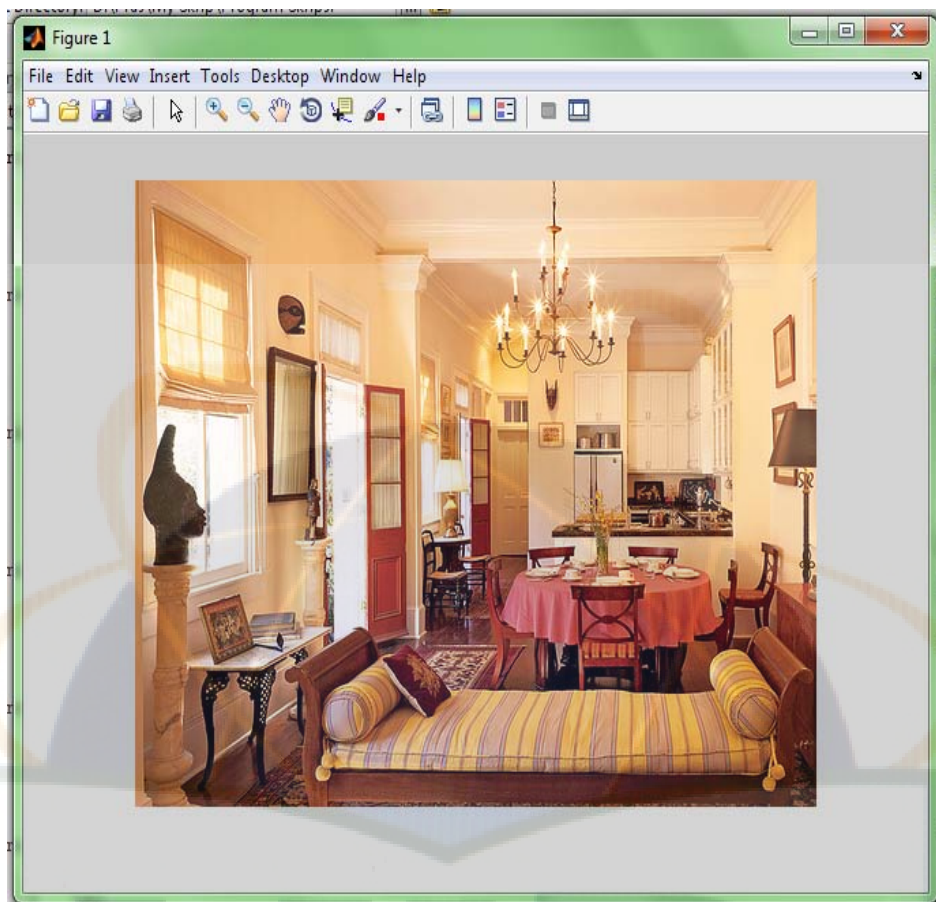


Gambar 4.4 Proses run setelah pesan diinput 2.

Angka 241 itu menunjukkan input telah selesai. Pada proses awal, pesan yang akan disisipkan adalah “Saya harus wisuda Juli 2010!!!”. Pesan tersebut terdiri dari 30 kata. Spasi dan tanda seru tidak diabaikan karena pada table ASCII, spasi dan tanda seru memiliki nilai yang tidak kosong. Jadi tetap dihitung sebagai huruf yang harus dirubah. Angka 241 yang tertera didapatkan dari hasil perkalian 30 kata dengan angka 8. Angka 8 disini karena tiap-tiap kata dikonversikan menjadi binari-binari yang terdiri dari 8 digit. Hasil perkalian 30 dengan 8 adalah 240. Counter yang digunakan itu adalah tempat untuk mensubtitusi binari media dengan binari pesan. 241 yang terlihat pada gambar 4.4 menyatakan 240 binari akhir telah sukses disubtitusikan, karena binari total yang telah disubtitusikan sebanyak 240 maka Counter berhenti pada angka 241.

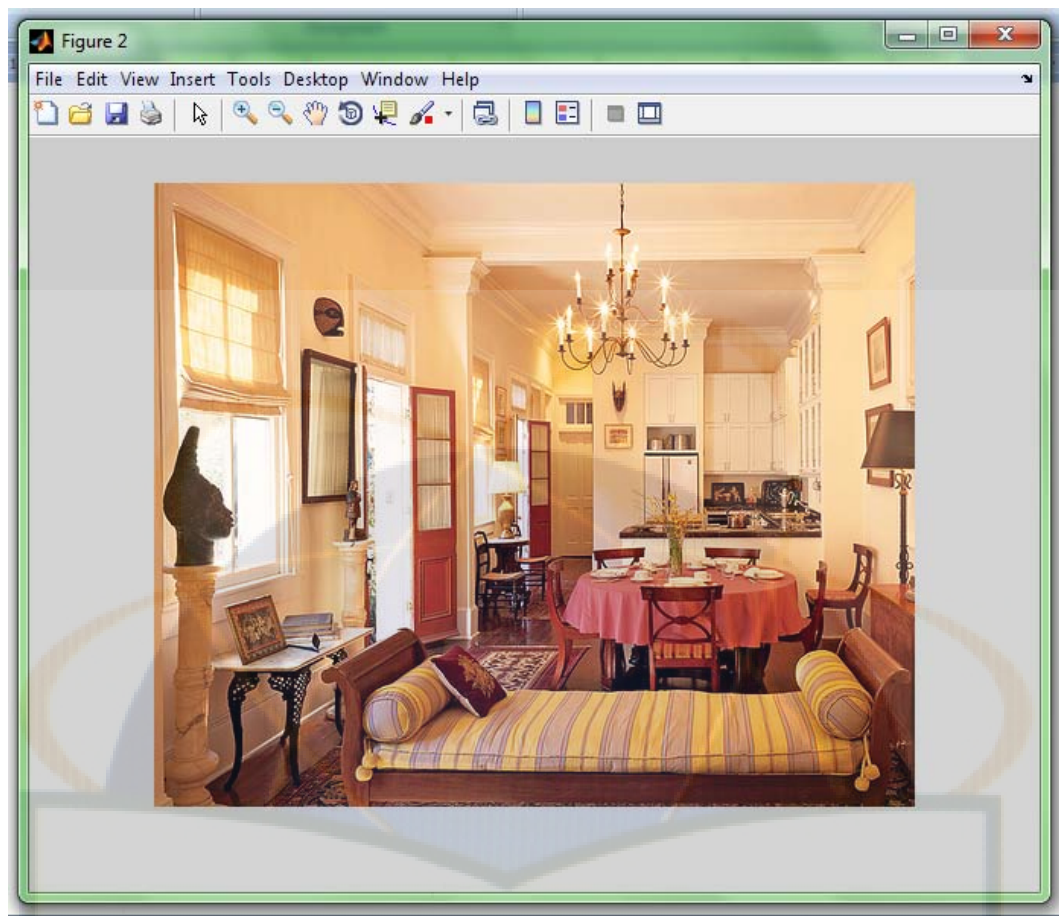
Selanjutnya setelah binari-binari pada *file* media gambar tersisipi, langkah selanjutnya adalah merubah binari-binari tersebut kembali menjadi matriks angka skala 0-255. Pada program ini proses tersebut tidak ditampilkan.

Langkah terakhir pada program ini adalah merubah kembali matriks angka menjadi sebuah gambar. Gambar ini adalah gambar yang telah tersisipi pesan “Saya harus wisuda Juli 2010!!!”. Hasil *run* dari program tersebut menghasilkan 2 buah gambar dalam *figure* 1 dan *figure* 2. Gambar 4.5 adalah *figure* 1, yaitu sebelum gambar disisipkan sebuah pesan. Sedangkan gambar 4.6 adalah hasil tampilan dari program ini atau gambar yang telah disisipkan sebuah pesan.



Gambar 4.5 Hasil gambar sebelum disisipkan pesan

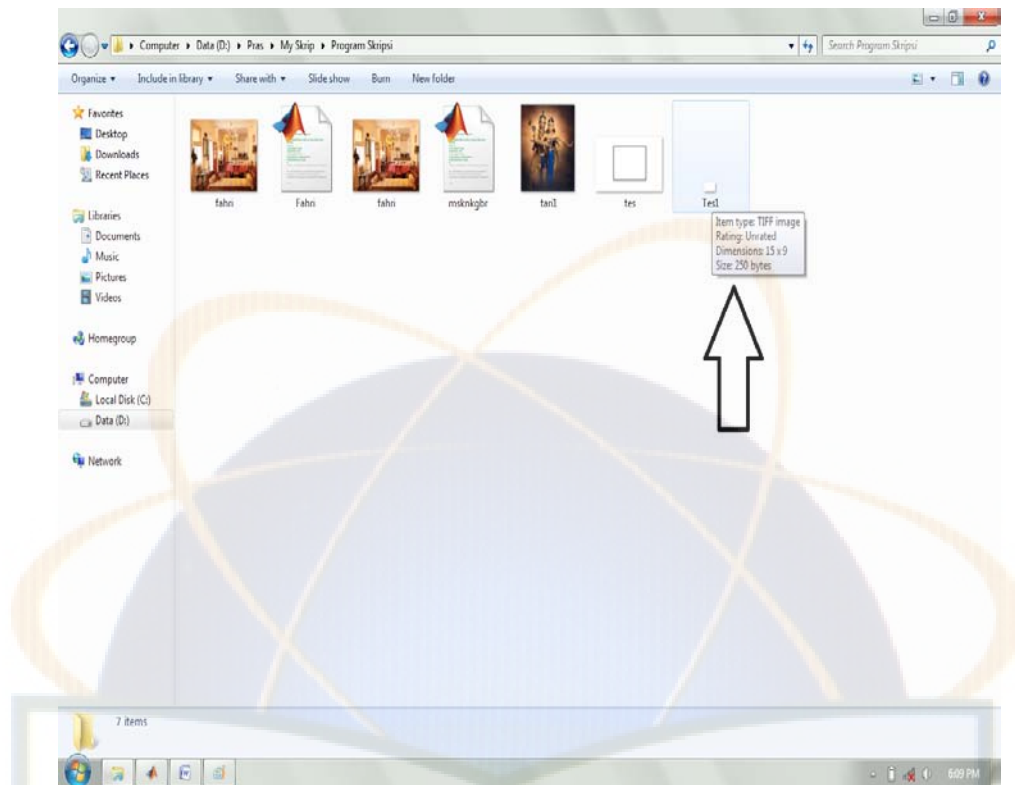
Gambar 4.5 adalah gambar awal sebelum disisipkan sebuah pesan. Gambar ini mempunyai ukuran 500 x 410 pixel. Jadi dalam gambar ini jumlah pesan yang dapat disembunyikan sebanyak $500 \times 410 = 205.000$ kata (termasuk spasi serta tanda baca).



Gambar 4.6 Hasil gambar yang telah disisipkan pesan

Gambar yang dihasilkan disini hanya tampilan saja (*figure 2*), jadi disini tidak biasa dibandingkan ukuran *file* awal dan *file* akhirnya. Yang dapat dibandingkan hanya tampilan visualnya.

Untuk Proses pembuktian bahwa gambar sebelum disisipkan dan setelah disisipkan adalah tidak sama persis, maka dilakukan pemilihan gambar yang ukuran kecil yang berukuran 15 px x 9 px. Maka matriks yang dihasilkan berupa matriks berordo 15 x 9. Pada proses pembuktian ini gambar yang digunakan ada gambar berekstension *TIFF dan berwarna putih semua. Dengan harapan jika metode LSB ini mempunyai kekurangan maka akan terlihat pada gambar hasil.



Gambar 4.7 File yang digunakan untuk proses pembuktian

Langkah awal sama seperti proses yang pada awal bab IV telah disebutkan langkahnya secara rinci. Pada proses pembuktian ini kata kunci yang digunakan adalah “fahri”. Pesan yang berupa huruf dirubah terlebih dahulu menjadi bilangan ASCII, sebelum dirubah menjadi binari. Cara merubah menjadi bilangan ASCII adalah dengan melihat pada table ASCII yang disediakan pada lampiran skripsi ini. Bilangan ASCII dari fahri adalah:

$f = 102, a = 97, h = 104, r = 114, i = 105.$

Selanjutnya angka ASCII tersebut dirubah menjadi angka binari. Hasilnya adalah:

$f = 102 = 0110\ 0110$

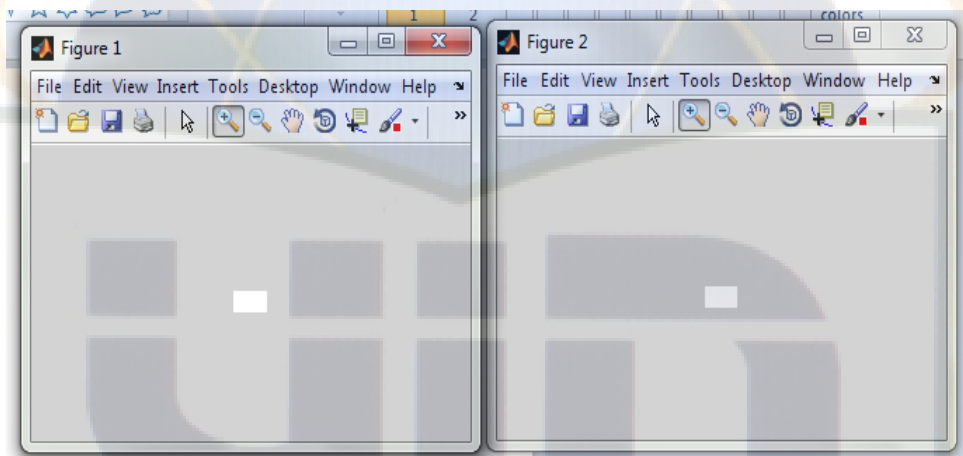
$a = 97 = 0110\ 0001$

$h = 104 = 0110\ 1000$

$r = 114 = 0111\ 0010$

$i = 105 = 0110\ 1001$

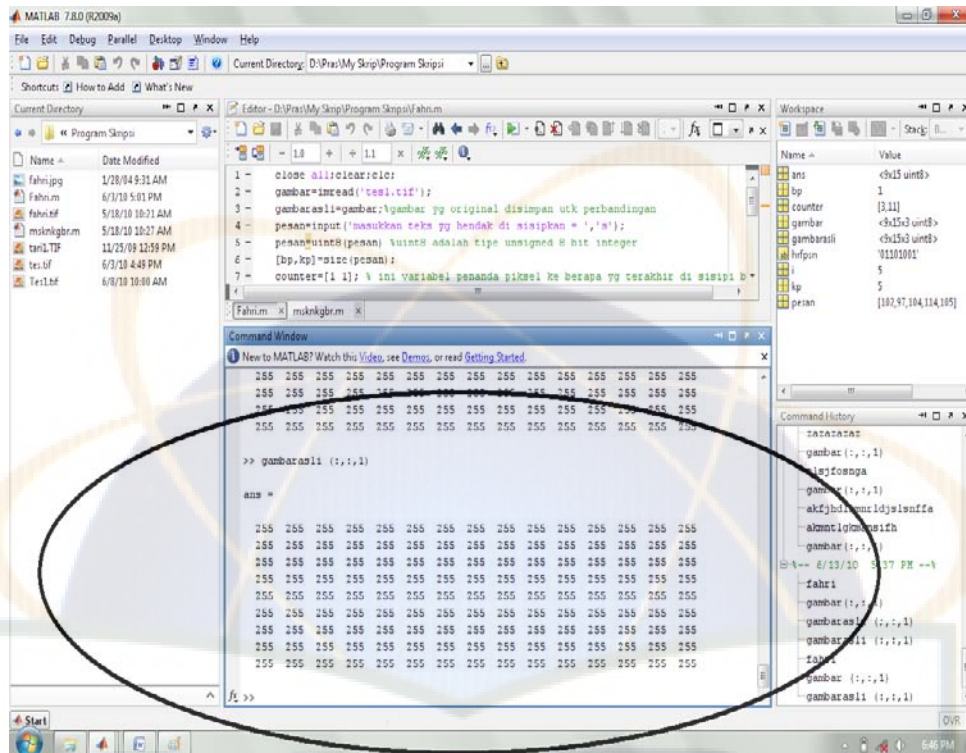
Jika menjalankan program maka terlihat gambar media gambar sebelum disisipkan pesan dan setelah disisipkan pesan tidak terlihat perbedaan secara visual. Pada gambar 4.8 menampilkan media gambar awal sebelum disisipkan pesan dan setelah disisipkan pesan.



Gambar 4.8 Perbandingan *visual* media awal dengan hasil akhir

Pada gambar 4.8 terlihat bahwa tidak ada perubahan secara *visual* dari media awal sebelum disisipkan sebuah pesan dan setelah media awal disisipkan pesan “fahri”. Maka untuk proses pembuktian harus membandingkan matriks media sebelum disisipkan sebuah pesan dengan matriks hasil setelah media disisipka pesan “fahri”. Karena media awal menggunakan gambar putih

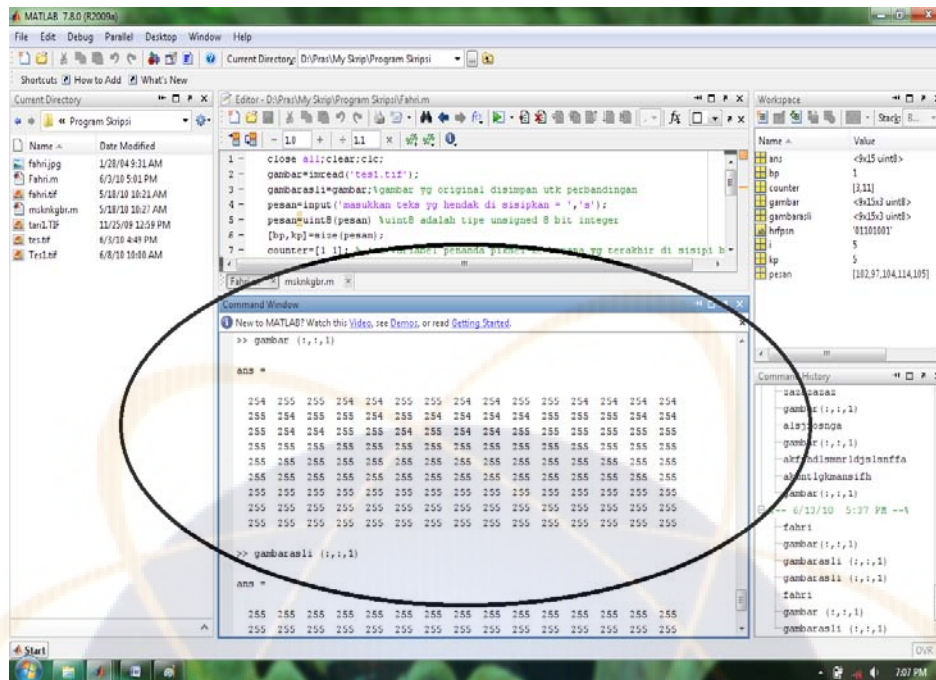
sempurna, maka angka pada matriks adalah 255. Namun untuk membuktikan perhatikan pada matriks yang ditampilkan pada Matlab.



Gambar 4.9 Matriks media gambar sebelum disisipkan pesan

Pada gambar 4.9 terlihat bahwa matriks berordo 15 x 9 dan angka pada matriks secara keseluruhan adalah 255. Karena media yang digunakan adalah putih sempurna yang dalam skala 0-255 putih sempurna mempunyai nilai 255.

Langkah selanjutnya pada proses pembuktian ini adalah tampilkan matriks hasil setelah media disisipkan pesan. Dengan membandingkan kedua matriks tersebut, maka akan terlihat bahwa sesungguhnya terdapat perbedaan pada media gambar sebelum disisipkan pesan dengan media gambar yang telah disisipkan pesan. Gambar 4.10 menampilkan matriks media yang telah disisipkan sebuah pesan.



Gambar 4.10 Matriks media gambar setelah disisipkan pesan

Proses perbandingan nya adalah dengan cara mengambil 8 angka secara urut dari matriks hasil secara berurutan pada baris pertama. Jika baris pertama tidak sampai 8 angka maka dilanjutkan dengan mengambil secara berurutan dari baris kedua kolom pertama dilanjutkan baris kedua kolom kedua dan begitu seterusnya sampai 8 angka.

Selanjutnya bandingkan dengan 8 angka yang didapatkan dari matriks hasil penyisipan dengan 8 angka yang didapatkan dari matriks awal. 8 angka awal yang didapatkan pada matriks hasil adalah 254 255 255 254 254 255 255 254. 8 angka awal yang didapatkan dari matriks awal adalah 255 255 255 255 255 255 255 255. Namun yang dibandingkan disini bukan angka dalam skala 0-255, namun angka-angka yang telah diperoleh tersebut dirubah kedalam bilangan binari. 8 angka awal matriks sebelum disisipkan yang telah dirubah dalam binari

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111. 8 angka awal matriks setelah disisipkan yang telah dirubah dalam binari 11111110 11111111 11111111 11111110 11111110 11111111 11111111 11111110. Perhatikan angka yang dicetak tebal pada 8 angka awal matriks sebelum disisipkan yang telah dirubah dalam binari, setelah itu perhatikan juga angka yang dicetak tebal pada 8 angka awal matriks setelah disisipkan yang telah dirubah dalam binari. Perubahan akan ditampilkan dalam tabel sehingga terlihat hasil huruf yang disisipkan.

Angka disisipkan	Terakhir	Matriks	Sebelum	Angka disisipkan	Terakhir	Matriks	Setelah
		1				0	
		1				1	
		1				1	
		1				0	
		1				0	
		1				1	
		1				1	
		1				0	

Tabel 4.1 Perubahan Binari Matriks Sebelum Disisipkan Dengan Matriks

Setelah Disisipkan

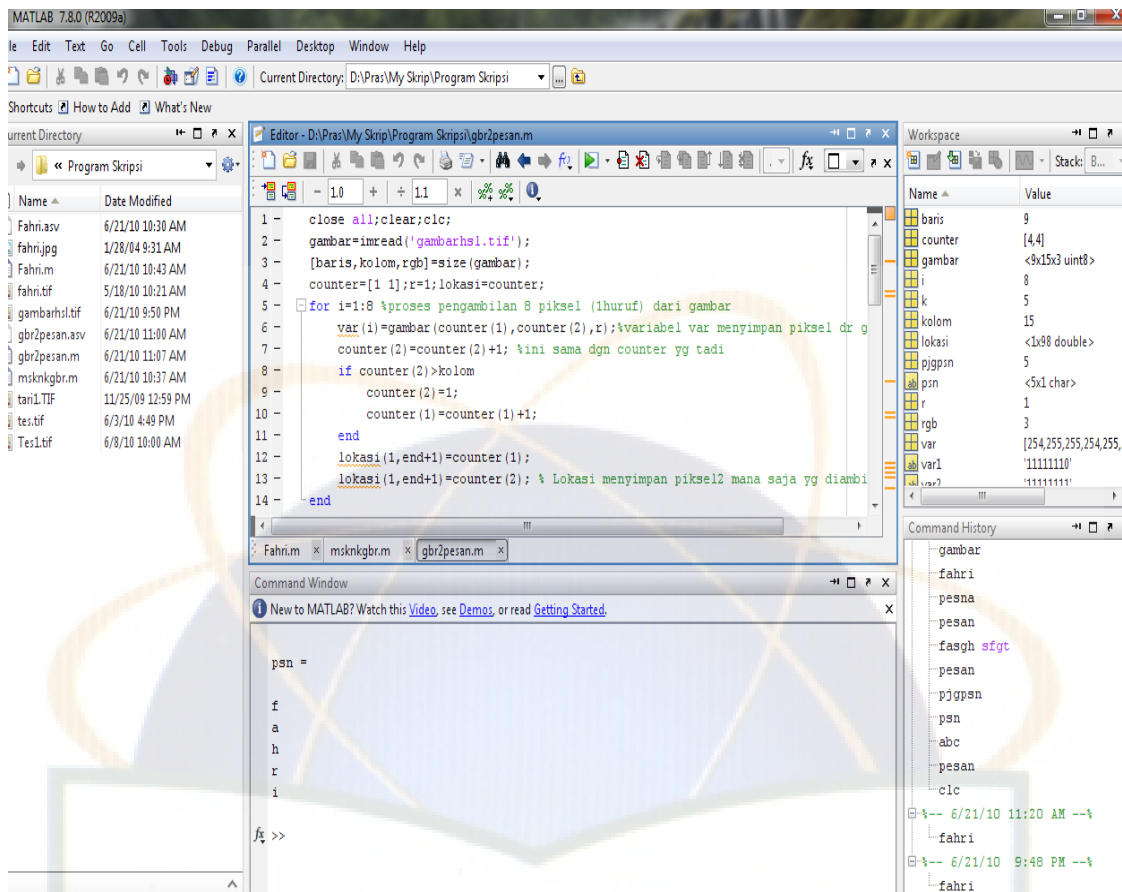
Angka yang terdapat pada kolom Angka Terakhir Matriks Setelah disisipkan pada tabel 4.1 adalah representasi dari 8 digit huruf awal dari pesan

yang disisipkan. Sesuai tabel 4.1 angka yang disisipkan adalah 0110 0110. Setelah mengetahui binari yang telah disisipkan ubah binari-binari tersebut menjadi bilangan skala 0-255 dengan cara:

$$0x2^7 + 1x2^6 + 1x2^5 + 0x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 0x2^0 \\ = 0 + 64 + 32 + 0 + 0 + 4 + 2 + 0 = 102.$$

Angka yang diperoleh adalah 102. Dalam tabel ASCII angka 102 itu berarti huruf “f”, dan ini sesuai dengan huruf awal dari pesan yang disisipkan ke media awal. Untuk langkah selanjutnya gunakan cara yang sama untuk mendapatkan 4 huruf lain yang disisipkan dalam pesan. Karena total huruf yang disisipkan ke dalam media sebanyak 5 huruf.

Untuk mempermudah menjadikan gambar yang telah disisipi menjadi pesan awal, dapat menjalankan program yang telah dibuat. Perhitungan diatas adalah untuk perhitungan manual. Program yang merubah media gambar yang telah disisipi menjadi pesan yang disisipkan pada media awal langsung membaca *file* media gambar yang telah disisipkan, yang pada proses penyisipan pesan media gambar tersebut disimpan dengan nama gambarhasil.TIF. Gambar 4.11 adalah gambar tampilan dari hasil *run* program tersebut.



Gambar 4.11 Hasil *run* program yang mengubah media gambar yang telah disisipkan pesan menjadi pesan awal.

Pada gambar 4.11 menunjukkan bahwa pesan awal yang disembunyikan pada proses penyisipan kali ini adalah “fahri”.

BAB V

KESIMPULAN DAN SARAN

Setelah menjalankan program steganografi dengan LSB dan dengan bantuan *software* Matlab2009 ini, beberapa kesimpulan serta saran yang dapat diperoleh dari pengerjaan skripsi ini adalah sebagai berikut:

5.1. Kesimpulan

Kesimpulan yang diperoleh adalah :

1. Pesan berhasil disisipkan ke dalam media gambar dengan metode LSB menggunakan Software MATLAB 2009.
2. Tampilan media gambar yang digunakan baik sebelum dan sesudah dioperasikan tidak berubah. Perubahan yang terjadi sangat kecil, sehingga perubahan warna yang terjadi tidak dapat dilihat dengan mata telanjang.
3. Pesan awal yang disisipkan dapat diketahui jika melakukan perbandingan hasil matriks media awal sebelum disisipkan dan hasil matriks setelah gambar disisipkan pesan.
4. Pesan awal yang disisipkan dapat diketahui dengan cara menjalankan program yang berfungsi membaca pesan yang disembunyikan pada media yang telah disisipi sebuah pesa.

5.2. Saran

Beberapa saran yang dapat diambil dari skripsi ini adalah:

1. Program ini hanya berlaku untuk penyisipan pesan berupa teks, sehingga untuk menyisipkan *file*, gambar atau *audio* tidak dapat dilakukan dengan program ini.
2. Program ini tidak menampilkan matriks awal sebelum perubahan dan matriks akhir setelah disisipi pesan.
3. Hasil gambar output tidak berupa *file*, hanya tampilan. Jadi tidak dapat dibandingkan ukuran *file* antara sebelum disisipkan dengan *file* gambar setelah disisipkan.
4. Pesan yang akan disisipkan tergantung ukuran gambar dari media. Sebab banyak karakter yang dapat disisipkan adalah seperdelapan dari hasil kali ukuran matriks gambar.

REFERENSI

- [1] **Mangarae, Aelphaeis.** *Steganography FAQ*, Zone-H.Org. 2006.
- [2] <http://patembe.com/2009/07/26/perbedaan-antara-Byte-dan-bit/>
- [3] <http://ismailzone.com/download/cryptography/andino-steganografi.pdf>
- [4] **Hakim A, Muhammad.** *Studi dan Implementasi Steganografi Metode LSB dengan Preprocessing Kompresi data dan Ekspansi Wadah.*
- [5] **Wijaya , Ermadi Satriya. Prayudi, Yudi.** *Konsep Hidden Message Menggunakan Teknik Steganografi Dynamic Cell Spreading.*
- [6] http://www.ittelkom.ac.id/library/index.php?view=article&catid=15:pemrosesan-sinyal&id=344:citra-digital&option=com_content&Itemid=15
- [7] http://id.wikipedia.org/wiki/Sistem_bilangan_biner
- [8] <http://aridaryani87.files.wordpress.com/2009/01/barisan-dan-deret.pdf>
- [9] <http://www.asciitable.com/>
- [10] [http://id.wikipedia.org/wiki/Matriks_\(matematika\)](http://id.wikipedia.org/wiki/Matriks_(matematika))

Lampiran

Program Utama:

```
close all;clear;clc;

gambar=imread('tes1.tif');

gambarasli=gambar;% gambar yg original disimpan utk perbandingan

pesan=input('masukkan teks yg hendak di sisipkan = ','s');

pesan=uint8(pesan); %uint8 adalah tipe unsigned 8 bit integer

[bp,kp]=size(pesan);

pesan2=zeros(bp,kp+1);

pesan2(1,1)=kp;

pesan2(1,2:end)=pesan;

pesan=pesan2;

[bp,kp]=size(pesan);

counter=[1 1]; % ini variabel penanda piksel ke berapa yg terakhir di sisipi bit

for i=1:kp %akan dimasukkan huruf perhuruf pesan ke pixel gbr

    hrfpsn=dec2bin(pesan(i),8);%merubah pesan dari decimal ke binary

    [gambar,counter]=msknkgbr(gambar,counter,hrfpsn);% gambar adlaah gambar,

    counter adalah skrg ada di byte ke berapa di gambar

end

imwrite(gambar, 'gambarhsl.tif');

disp('Gambar yang asli Figure 1');

figure(1)

imshow(gambarasli)
```

```
disp('Gambar yang telah disisipi figure 2');
```

```
figure(2)
```

```
imshow(gambar)
```

```
imshow(gambar)
```

Proses LSB

```
function [gbr,counter]=msknkgbr(gambar,counter,hfrpsn)
```

```
[baris,kolom,rgb]=size(gambar);
```

```
r=1;
```

```
lokasi=counter;
```

```
for i=1:8 %proses pengambilan 8 piksel (1huruf) dari gambar
```

```
    var(i)=gambar(counter(1),counter(2),r);%variabel var menyimpan piksel dr  
gambar
```

```
    counter(2)=counter(2)+1 %ini sama dgn counter yg tadi
```

```
    if counter(2)>kolom
```

```
        counter(2)=1;
```

```
        counter(1)=counter(1)+1;
```

```
    end
```

```
    lokasi(1,end+1)=counter(1);
```

```
    lokasi(1,end+1)=counter(2); % Lokasi menyimpan piksel2 mana saja yg  
diambil
```

```
end
```

```
% merubah piksel2 terambil menjadi binari
```

```

var1=dec2bin(var(1),8);var2=dec2bin(var(2),8);

var3=dec2bin(var(3),8);var4=dec2bin(var(4),8);

var5=dec2bin(var(5),8);var6=dec2bin(var(6),8);

var7=dec2bin(var(7),8);var8=dec2bin(var(8),8);

%Proses penyisipan bit terakhir ada disini

var1(8)=hrfpsn(1); var2(8)=hrfpsn(2);

var3(8)=hrfpsn(3); var4(8)=hrfpsn(4);

var5(8)=hrfpsn(5); var6(8)=hrfpsn(6);

var7(8)=hrfpsn(7); var8(8)=hrfpsn(8);

%Rubah lagi ke decimal setelah disisipi

var1=bin2dec(var1);var2=bin2dec(var2);

var3=bin2dec(var3);var4=bin2dec(var4);

var5=bin2dec(var5);var6=bin2dec(var6);

var7=bin2dec(var7);var8=bin2dec(var8);

%masukkan kembali piksel2 yg sudah disisipi ke gambar

gambar(lokalasi(1),lokalasi(2),r)=var1;

gambar(lokalasi(3),lokalasi(4),r)=var2;

gambar(lokalasi(5),lokalasi(6),r)=var3;

gambar(lokalasi(7),lokalasi(8),r)=var4;

gambar(lokalasi(9),lokalasi(10),r)=var5;

gambar(lokalasi(11),lokalasi(12),r)=var6;

gambar(lokalasi(13),lokalasi(14),r)=var7;

gambar(lokalasi(15),lokalasi(16),r)=var8;

```



```
gbr=gambar;
```

Program Membaca Media Yang Telah Disisipi Sebuah Pesan

```
close all;clear;clc;
```

```
gambar=imread('gambarhsl.tif');
```

```
[baris,kolom,rgb]=size(gambar);
```

```
counter=[1 1];r=1;lokasi=counter;
```

```
for i=1:8 %proses pengambilan 8 piksel (1huruf) dari gambar
```

```
    var(i)=gambar(counter(1),counter(2),r);%variabel var menyimpan piksel dr  
gambar
```

```
    counter(2)=counter(2)+1; %ini sama dgn counter yg tadi
```

```
    if counter(2)>kolom
```

```
        counter(2)=1;
```

```
        counter(1)=counter(1)+1;
```

```
    end
```

```
    lokasi(1,end+1)=counter(1);
```

```
    lokasi(1,end+1)=counter(2); % Lokasi menyimpan piksel2 mana saja yg  
diambil
```

```
end
```

```
% merubah piksel2 terambil menjadi binary
```

```
var1=dec2bin(var(1),8);var2=dec2bin(var(2),8);
```

```
var3=dec2bin(var(3),8);var4=dec2bin(var(4),8);
```

```
var5=dec2bin(var(5),8);var6=dec2bin(var(6),8);
```

```

var7=dec2bin(var(7),8);var8=dec2bin(var(8),8);

%Proses pengambilan bit terakhir menjadi informasi ttg panjang char pesan

%pjgpsn=zeros(1,8,'s');

pjgpsn(1)=var1(8); pjgpsn(2)=var2(8);

pjgpsn(3)=var3(8); pjgpsn(4)=var4(8);

pjgpsn(5)=var5(8); pjgpsn(6)=var6(8);

pjgpsn(7)=var7(8); pjgpsn(8)=var8(8);

pjgpsn=bin2dec(pjgpsn);

%Proses pengambilan teks dari gambar sesuai dgn pjg pesannya
for k=1:pjgpsn
    for i=1:8 %proses pengambilan 8 piksel (1huruf) dari gambar
        var(i)=gambar(counter(1),counter(2),r);% variabel var menyimpan piksel dr
gambar
        counter(2)=counter(2)+1; %ini sama dgn counter yg tadi
        if counter(2)>kolom
            counter(2)=1;
            counter(1)=counter(1)+1;
        end
        lokasi(1,end+1)=counter(1);

        lokasi(1,end+1)=counter(2); % Lokasi menyimpan piksel2 mana saja yg
diambil

    end

    % merubah piksel2 terambil menjadi binary

```

```
var1=dec2bin(var(1),8);var2=dec2bin(var(2),8);  
var3=dec2bin(var(3),8);var4=dec2bin(var(4),8);  
var5=dec2bin(var(5),8);var6=dec2bin(var(6),8);  
var7=dec2bin(var(7),8);var8=dec2bin(var(8),8);  
  
%Proses pengambilan bit terakhir menjadi informasi ttg panjang char pesan  
  
%pjgpsn=zeros(1,8,'s');  
psn(k,1)=var1(8); psn(k,2)=var2(8);  
psn(k,3)=var3(8); psn(k,4)=var4(8);  
psn(k,5)=var5(8); psn(k,6)=var6(8);  
psn(k,7)=var7(8); psn(k,8)=var8(8);  
  
end  
psn=bin2dec(psn); psn=native2unicode(psn)
```

Tabel ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space
1	1	001	SOH (start of heading)	33	21	041	!	!
2	2	002	STX (start of text)	34	22	042	"	"
3	3	003	ETX (end of text)	35	23	043	#	#
4	4	004	EOT (end of transmission)	36	24	044	$	\$
5	5	005	ENQ (enquiry)	37	25	045	%	%
6	6	006	ACK (acknowledge)	38	26	046	&	&
7	7	007	BEL (bell)	39	27	047	'	'
8	8	010	BS (backspace)	40	28	050	((
9	9	011	TAB (horizontal tab)	41	29	051))
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*
11	B	013	VT (vertical tab)	43	2B	053	+	+
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,
13	D	015	CR (carriage return)	45	2D	055	-	-
14	E	016	SO (shift out)	46	2E	056	.	.
15	F	017	SI (shift in)	47	2F	057	/	/
16	10	020	DLE (data link escape)	48	30	060	0	0
17	11	021	DC1 (device control 1)	49	31	061	1	1
18	12	022	DC2 (device control 2)	50	32	062	2	2
19	13	023	DC3 (device control 3)	51	33	063	3	3
20	14	024	DC4 (device control 4)	52	34	064	4	4
21	15	025	NAK (negative acknowledge)	53	35	065	5	5
22	16	026	SYN (synchronous idle)	54	36	066	6	6
23	17	027	ETB (end of trans. block)	55	37	067	7	7
24	18	030	CAN (cancel)	56	38	070	8	8
25	19	031	EM (end of medium)	57	39	071	9	9
26	1A	032	SUB (substitute)	58	3A	072	:	:
27	1B	033	ESC (escape)	59	3B	073	;	;
28	1C	034	FS (file separator)	60	3C	074	<	<
29	1D	035	GS (group separator)	61	3D	075	=	=
30	1E	036	RS (record separator)	62	3E	076	>	>
31	1F	037	US (unit separator)	63	3F	077	?	?

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
64	40	100	@	@	96	60	140	`	`
65	41	101	A	A	97	61	141	a	a
66	42	102	B	B	98	62	142	b	b
67	43	103	C	C	99	63	143	c	c
68	44	104	D	D	100	64	144	d	d
69	45	105	E	E	101	65	145	e	e
70	46	106	F	F	102	66	146	f	f
71	47	107	G	G	103	67	147	g	g
72	48	110	H	H	104	68	150	h	h
73	49	111	I	I	105	69	151	i	i
74	4A	112	J	J	106	6A	152	j	j
75	4B	113	K	K	107	6B	153	k	k
76	4C	114	L	L	108	6C	154	l	l
77	4D	115	M	M	109	6D	155	m	m
78	4E	116	N	N	110	6E	156	n	n
79	4F	117	O	O	111	6F	157	o	o
80	50	120	P	P	112	70	160	p	p
81	51	121	Q	Q	113	71	161	q	q
82	52	122	R	R	114	72	162	r	r
83	53	123	S	S	115	73	163	s	s
84	54	124	T	T	116	74	164	t	t
85	55	125	U	U	117	75	165	u	u
86	56	126	V	V	118	76	166	v	v
87	57	127	W	W	119	77	167	w	w
88	58	130	X	X	120	78	170	x	x
89	59	131	Y	Y	121	79	171	y	y
90	5A	132	Z	Z	122	7A	172	z	z
91	5B	133	[[123	7B	173	{	{
92	5C	134	\	\	124	7C	174	|	
93	5D	135]]	125	7D	175	}	}
94	5E	136	^	^	126	7E	176	~	~
95	5F	137	_	_	127	7F	177		DEL