



TUGAS AKHIR – SM141501

**PENYISIPAN PESAN PADA CITRA DIGITAL
MENGGUNAKAN METODE LEAST
SIGNIFICANT BIT**

MUHAMMAD AZLANSYAH
NRP 06111140000119

Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT

DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT – SM141501

**EMBEDING MESSAGE TO DIGITAL IMAGE
USING LEAST SIGNIFICANT BIT**

MUHAMMAD AZLANSYAH
NRP 06111140000119

Supervisor
Dr. Budi setiyono, S.Si, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics, Computation, and Data Science
Institut Teknologi Sepuluh Nopember
Surabaya 2018

LEMBAR PENGESAHAN

PENYISIPAN PESAN PADA CITRA
DIGITAL MENGGUNAKAN METODE LEAST
SIGNIFICANT BIT

EMBEDING MESSAGE TO DIGITAL IMAGE USING
LEAST SIGNIFICANT BIT

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

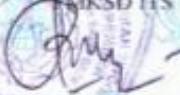
Oleh :
MUHAMMAD AZLANSYAH
NRP. 06111140000119

Menyetujui,
Dosen Penumbung.


Dr. Budi Setiyono, S.Si, MT
NIP. 19720207 199702 1 001

Mengetahui,

Ketua Departemen Matematika
FMKSD ITS


Dr. Imam Mukilash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, 03 Agustus 2018

PENYISIPAN PESAN PADA CITRA DIGITAL MENGGUNAKAN METODE LEAST SIGNIFICANT BIT

Nama : Muhammad Azlansyah
NRP : 06111140000119
Jurusan : Matematika FMKSD-ITS
Pembimbing : Dr. Budi Setiyono, S.Si, MT

Abstrak

Berbagai macam teknik untuk melindungi informasi yang dirahasiakan telah banyak dilakukan. Steganografi adalah salah satu teknik yang digunakan dalam penyembunyian pesan ke dalam sebuah media sedemikian sehingga manusia sulit menyadari keberadaan pesan tersebut. Pada tugas akhir ini teknik steganografi yang digunakan metode *Least Significant Bit* (LSB). Metode *Least Significant Bit* (LSB) adalah metode menyembunyikan pesan pada bit terakhir citra digital sehingga tidak terjadi perubahan secara kasat mata. Pada metode *Least Significant Bit* (LSB) dibutuhkan 3 input berupa citra, pesan, dan kunci. Pertama kunci dan pesan dirubah kedalam bentuk ASCII, setelah itu dirubah kedalam bentuk biner. Kemudian diambil channel merah (red) dari citra yang dirubah kedalam bentuk biner. Selanjutnya pesan dan kunci disisipkan pada citra tersebut. Terakhir citra yang telah disisipkan pesan disimpan dengan nama *Stego-Image*. Untuk uji coba penulis menggunakan 3 citra dengan karakteristik yang berbeda, yaitu: citra langit yang cenderung homogen, citra wajah yang sedikit komplek, dan citra pemandangan yang lebih komplek. Disamping itu uji coba juga dilakukan pada 4 tipe citra yang berbeda, yaitu bmp, jpeg, png, dan tiff. Hasil yang dicapai pada tugas akhir ini, citra yang disisipkan pesan tidak mengalami perubahan bentuk secara kasat mata, nilai rata-rata PSNR dari keempat tipe citra tersebut untuk penyisipan teks 250 kata adalah sebagai berikut: citra langit 68.22975 dB, citra wajah 72.228575dB, citra pemandangan

74.322525 dB dan pesan yang disisipkan dapat dikembalikan seperti semula saat proses ekstraksi.

Kata kunci : Steganografi, Metode Least Significant Bit (LSB), citra digital.

EMBEDING MESSAGE TO DIGITAL IMAGE USING LEAST SIGNIFICANT BIT

Name	:	Muhammad Azlansyah
NRP	:	06111140000119
Department	:	Mathematics
Supervisor	:	Dr. Budi Setiyono, S.Si, MT

Abstrak

Various techniques to protect undisclosed information have been widely practiced. Steganography is one of the techniques used in concealment of messages into a media so that humans find it difficult to realize the existence of the message. In this final project steganography technique used Least Significant Bit (LSB) method. The Least Significant Bit (LSB) method is a method of hiding messages on the last bit of a digital image so that no visible changes occur. In Least Significant Bit (LSB) method required 3 input image, message, and key. First the key and the message is changed into ASCII form, after which it is converted into binary form. Then taken the red channel (red) from the image that is converted into binary form. The message and key are then inserted in the image. Last image that has been inserted message saved with the name Stego-Image. To test the writer using 3 images with different characteristics: sky image that tends to homogeneous, complex face image, and image of a more complex landscape. Besides that, trials are also performed on 4 different types of images, namely bmp, jpeg, png, and tiff. The results achieved in this final project, the image inserted image does not change visually, the average value of PSNR of the four types of images for the insertion of 250 word text is as follows: sky image 68.22975 dB, face image 72.228575 dB, image 74.322525 dB views

and inserted messages can be restored as they were during the extraction process.

Keywords: Steganography, LSB, Text, Digital Image.

KATA PENGANTAR

Puji syukur penulis panjatkan pada kehadirat Allah Swt. Karena hanya dengan karunia rahmat, bimbingan, serta anugrah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul **“PENYISIPAN PESAN PADA CITRA DIGITAL MENGGUNAKAN METODE LEAST SIGNIFICANT BIT”** Tugas Akhir ini dapat diselesaikan dengan baik berkat kerjasama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis mengucapkan terimakasih kepada:

1. Dr. Budi Setiyono, S.Si, MT selaku dosen pembimbing yang senantiasa membimbing dengan sabar dan memberikan kritik dan saran dalam penyusunan Tugas Akhir ini.
 2. Dr. Imam Mukhlash, S.Si,MT selaku Ketua Departemen Matematika.
 3. Sunarsini, S.Si, M.Si selaku Dosen Wali.
 4. Drs. Komar Baihaqi, M.Si, Drs. Bandung Arry S., MI.Komp, Dr. Chairul Imron, MI.Komp dan Dr. Dwi Ratna Sulistyaningrum, S.Si, MT selaku dosen penguji Tugas Akhir ini.
 5. Dr. Didik Khusnul Arif S.Si, M.Si selaku Koordinator Tugas Akhir.
 6. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
- Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga Tugas Akhir ini bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2018

Penulis

“Halaman ini sengaja dikosongkan”

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN	xix
 BAB I. PENDAHULUAN	
1.1 LatarBelakang	1
1.2 RumusanMasalah	3
1.3 BatasanMasalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 SistematikaPenulisanTugasAkhir	4
 BAB II. TINJAUAN PUSTAKA	
2.1 Steganografi	7
2.1.1 Sejarah Steganografi.....	9
2.1.2 Manfaat Steganografi.....	10
2.2 Citra Digital.....	10
2.2.1 Resolusi Citra	11
2.2.2 Citra Warna (8 bit).....	11
2.2.3 Citra Warna (16 bit).....	11
2.2.4 Citra Warna (24 bit).....	11
2.3 Metode <i>Least Siginificant Bit</i> (LSB).....	12
2.4 Perhitungan Nilai MSE dan PSNR	13
 BAB III. METODOLOGI	
i Studi Literatur	15
ii Mengkaji Metode yang Dipakai.....	15

iii	Implementasi Algoritma Steganografi Menggunakan LSB	15
iv	Pengujian Modul – Modul Steganografi.....	15
v	Penulisan Tugas Akhir.....	16
vi	Diagram Alir.....	16
BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM		
4.1	Analisis Kebutuhan.....	19
4.1.1	Analisis Kebutuhan user.....	19
4.1.2	Analisis Kebutuhan Sistem.....	19
4.2	Deskripsi Metode.....	19
4.2.1	Deskripsi Metode Penyisipan	20
4.2.2	Deskripsi Metode Ekstraksi.....	24
4.3	Perancangan Perangkat Lunak.....	25
4.3.1	Perancangan Gambaran Umum Sistem	25
4.4	Pemrograman	29
4.4.1	Implementasi Masukkan Citra.....	29
4.4.2	Implementasi Penyisipan Pesan.....	30
4.4.3	Implementasi Ekstraksi Pesan	33
BAB V. PENGUJIAN DAN PEMBAHASAN		
5.1	Data UjiCoba	37
5.2	PengujianKualitatif	38
5.3	Pengujian Penyisipan dan Ekstraksi Pesan	43
5.4	Perbandingan Ukuran File	44
5.5	Pengujian Nilai <i>PSNR</i> (<i>Peak Signal to Noise Ratio</i>).....	45
BAB VI. KESIMPULAN DAN SARAN		
6.1	Kesimpulan	49
6.2	Saran	49
DAFTAR PUSTAKA	51
LAMPIRAN	53

DAFTAR GAMBAR

Tabel 2.1	Proses <i>sampling</i> dan kuantisasi	9
Tabel 3.1	Diagram alir penyisipan pesan	16
Tabel 3.2	Diagram alir ekstraksi pesan	17
Tabel 4.1	Citra sebelum disisipkan pesan	19
Tabel 4.2	Citra sesudah disisipkan pesan	22
Tabel 4.3	<i>Use Case</i> diagram perangkat lunak Steganografi	25
Tabel 4.4	<i>Activity</i> diagram penyisipan pesan ..	26
Tabel 4.5	<i>Activity</i> diagram ekstraksi pesan	27
Tabel 4.6	Tampilan antar muka pengambilan masukkan citra	29
Tabel 4.7	Tampilan antar muka masukkan kunci dan pesan	31
Tabel 4.8	Tampilan antar muka penyimpanan <i>StegoImage</i>	32
Tabel 4.9	Tampilan antar muka pengambilan <i>StegoImage</i>	34
Tabel 5.0	Tampilan antar muka masukkan kunci dan pesan yang disisipkan	34

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Tabel 4.1	Tabel Kebutuhan Sistem	18
Tabel 5.1	Data citra yang digunakan	35
Tabel 5.2	Citra Langit Uji Pertama	36
Tabel 5.3	Citra Wajah Uji Pertama	37
Tabel 5.4	Citra Pemandangan Uji Pertama	37
Tabel 5.5	Citra Langit Uji Kedua	38
Tabel 5.6	Citra Wajah Uji Kedua	38
Tabel 5.7	Citra Pemandangan Uji Kedua	39
Tabel 5.8	Citra Pemandangan Uji Ketiga	40
Tabel 5.9	Citra Langit Uji Ketiga	41
Tabel 5.10	Citra Wajah Uji Ketiga	41
Tabel 5.11	Hasil pengujian sistem pada proses penyisipan dan ekstraksi untuk semua format citra	42
Tabel 5.12	Hasil perbandingan ukuran citra asli dan <i>StegoImage</i> pada pengujian pertama	42
Tabel 5.13	Hasil perbandingan ukuran citra asli dan <i>StegoImage</i> pada pengujian kedua	43
Tabel 5.14	Hasil perbandingan ukuran citra asli dan <i>StegoImage</i> pada pengujian ketiga	43
Tabel 5.15	Hasil pengujian nilai <i>PSNR</i> pada pengujian Pertama	44

Tabel 5.16	Hasil pengujian nilai <i>PSNR</i> pada pengujian Kedua.....	45
Tabel 5.17	Hasil pengujian nilai <i>PSNR</i> pada pengujian ketiga.....	45

DAFTAR LAMPIRAN

A. Source Code Halaman Awal	51
B. Source Code Proses Penyisipan Pesan	56
C. Source Code Proses Ekstraksi Pesan	67
D. Tampilan Program	76

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

Pada bab ini dijelaskan latar belakang masalah yang dibahas pada tugas akhir, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

1.1 Latar Belakang

Di dunia modern seperti sekarang ini, internet dan komputer telah banyak digunakan di banyak tempat sebagai alat komunikasi. Dengan adanya internet, kita dapat berkomunikasi secara mudah dari berbagai tempat, dimanapun, dan kapanpun. Akan tetapi, berkomunikasi jarak jauh mempunyai resiko yang besar. Kita tidak mengetahui apakah pesan kita benar-benar sampai ke tangan orang yang kita tuju kemungkinan lain, data akan diretas oleh oknum untuk kepentingan yang tidak seharusnya. Hal inilah yang menuntut adanya pengamanan data agar tidak sampai dicuri oleh pihak lain. Oleh karena itu, pengguna teknologi semakin ramai mengembangkan suatu sistem pengamanan terhadap data yang biasa disebut kriptografi. Metode pengamanan kriptografi adalah dengan merubah pesan (*Plaintext*) kebentuk kode yang disandikan (*Chipertext*) kebentuk simbol, maka mudah diduga bahwa simbol tersebut adalah pesan yang disandikan. Penerapan awalnya pada masa pemerintahan Julius Ceasar yang dikenal *Caesar cipher*. Namun, hal itu sudah dapat diselesaikan sehingga dibutuhkan teknik baru hingga saat ini dengan algoritma terbaru yaitu RSA.

Dalam kriptografi muncul istilah steganografi, yaitu suatu teknik menyisipkan pesan ke dalam suatu media. Walaupun steganografi masih berkaitan dengan kriptografi, tapi teknik ini sangat berbeda. Hal ini memberikan keuntungan dengan mengurangi kecurigaan terhadap media yang kita gunakan bahwa media tersebut membawa pesan rahasia.

Steganografi merupakan ilmu yang mempelajari tentang seni menyembunyikan pesan atau informasi. Steganografi dapat digolongkan sebagai salah satu bagian dari ilmu komunikasi. 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. Seni dan ilmu ini telah diterapkan sejak dahulu oleh orang Yunani kuno yang menyembunyikan pesan dengan cara membuat tato di kepala pembawa berita yang dibotaki dan menunggu sampai rambutnya tumbuh.

Teknik steganografi lainnya adalah dengan menggunakan *invisible ink* (tinta yang tidak tampak). Tulisan yang ditulis dengan menggunakan *invisible ink* ini hanya dapat dibaca jika kertas tersebut diletakkan di atas lampu atau diarahkan ke matahari. Ketika perang dunia pertama, orang Jerman menyembunyikan pesan dalam bentuk *microdot*, yaitu titik-titik yang kecil. Agen dapat membuat foto kemudian mengecilkannya sampai sekecil titik di tulisan dalam buku. Buku ini kemudian bisa dibawa-bawa tanpa ada yang curiga bahwa tanda titik di dalam tulisan dibuku itu berisi pesan ataupun citra.

Dalam steganografi ada beberapa metode yang dapat digunakan salah satunya adalah metode *Least Significant Bit*(LSB). Metode *LSB* merupakan teknik penyisipan pesan dalam steganografi dimana penyisipan pesan dilakukan dengan mengganti deretan bit-bit data yang paling belakang dalam segmen citra dengan deretan bit-bit pesan yang akan disisipkan.

Ada beberapa penelitian sebelumnya yang berhubungan dengan metode *LSB* (*Least Significant Bit*) yaitu “**Penyembunyian Informasi (steganography) Gambar Menggunakan Metode LSB (Least Significant Bit)**” oleh Irfan. Pada penelitian ini pesan yang disisipkan berupa citra berformat

BMP dan menggunakan *cover image* berformat **JPEG**, “Penyisipan Media Teks dan Citra Menggunakan Teknik Steganografi pada Media Pembawa Citra Digital” oleh **Alim Muadzani, Oky Dwi Nurhayati, Ike Pertiwi Windasari**. Pada penelitian ini pesan yang disisipkan berupa pesan teks dan *cover image* yang digunakan hanya citra berformat **Portable Network Graphic (PNG)**.

Adapun contoh penerapan steganografi pada kehidupan sehari-hari yaitu “Aplikasi Pengamanan Dokumen Digital Menggunakan Algoritma Kriptografi *Advanced Encryption Standard* (AES-128), Kompresi Huffman dan Steganografi *End Of File* (EOF) BerbasisD Pada CV. Karya Perdana” oleh Wawan Budianto, Safrina Amini, Pipin Farida Ariyani. “Penerapan Steganografi pada *Corporate Internet Reporting (CIR)*” oleh Imamah. “Penerapan Steganografi Metode *End Of File*(EOF) dan Enkripsi Metode *Data Encryption Standard*(DES) Pada Aplikasi Pengamanan Data Gambar JOGJACK Factory Oulet” oleh Yayuk Anggraini, Dolly Virgian Shaka Yudha Sakti.

1.2 Rumusan Masalah

Adapun rumusan masalah dari Tugas Akhir ini adalah :

1. Bagaimana menyisipkan pesan teks ke dalam citra ?
2. Bagaimana mengimplementasikan metode LSB untuk menyisipkan pesan teks ke dalam citra ?

1.3 Batasan Masalah

Batasan Masalah yang akan dibahas dalam penelitian tugas akhir ini adalah sebagai berikut :

1. Data yang disisipkan berupa pesan teks.
2. Media Penyimpanan hanya pada citra digital.
3. Citra yang digunakan berupa citra warna 16 bit.

1.4 Tujuan

Sedangkan tujuan dari penelitian tugas akhir ini adalah sebagai berikut :

1. Mengkaji teknik menyisipkan pesan teks pada citra.
2. Mengimplementasikan metode *LSB* untuk menyisipkan pesan pada citra.

1.5 Manfaat

Manfaat dari penelitian ini adalah dapat menjadi bahan pertimbangan bagi seseorang dalam mengamankan informasi yang akan dikirimkan.

1.6 Sistematika Penulisan

Sistematika dari penulisan tugas akhir ini terbagi dalam 6 (enam) bab. Berikut penjelasan dari masing-masing bab:

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat penulisan tugas akhir serta sistematika penulisan tugas akhir.

BAB II TINJAUAN PUSTAKA

Menjelaskan tentang teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi Pengertian Citra Digital, Steganografi dan Metode *LSB* itu sendiri.

BAB III METODOLOGI PENELITIAN

Menjelaskan metodologi atau urutan pelaksanaan yang dilakukan dalam menyelesaikan tugas akhir, meliputi studi literatur, pengembangan perangkat lunak hingga penulisan laporan tugas akhir. Pada pengembangan perangkat lunak dilakukan analisis kebutuhan, perancangan perangkat lunak, pemrograman dan pengujian.

BAB IV ANALISA DAN PERANCANGAN

Bab ini membahas hasil analisa dan perancangan yang meliputi pembahasan mengenai deskripsi metode, analisa perangkat lunak dan perancangan perangkat lunak.

BAB V PENGUJIAN DAN PEMBAHASAN HASIL

Bab ini menampilkan hasil uji coba, perbandingan uji kualitatif berdasarkan pengamatan visual dari citra yang disisipi pesan (*Stego-Image*) dan citra asli, dan perbandingan ukuran file antara citra asli dan citra yang telah disisipi pesan (*Stego-Image*).

BAB VI PENUTUP

Bab ini merupakan penutup, berisi tentang kesimpulan yang dapat diambil berdasarkan data yang ada dan saran yang selayaknya dilakukan bila tugas akhir ini dilanjutkan.

“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan tinjauan pustaka yang menjadi dasar materi dalam penyusunan tugas akhir serta menunjang metode yang digunakan dalam pembahasan tugas akhir ini.

2.1 Steganografi

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Sebaliknya, kriptografi menyamarkan arti dari suatu pesan, tapi tidak menyembunyikan bahwa ada suatu pesan. pesan steganografi muncul dengan bentuk lain seperti , artikel, daftar belanjaan, atau pesan-pesan lainnya. Pesan yang tertulis ini merupakan tulisan yang menyelubungi atau menutupi.

Istilah steganografi berasal dari bahasa Yunani yaitu *Steganós* yang berartimenyembunyikan dan *Graptos* yang artinya tulisan, sehingga secara keseluruhanartinya adalah tulisan yang disembunyikan. Secara umum steganografi adalah ilmudan seni menyembunyikan pesan rahasia sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia.

Keuntungan penggunaan steganografi adalah memungkinkan pengiriman pesan secara rahasia tanpa diketahui bahwa pesan sedang dikirim. Ini membuat pihak ketiga tidak menyadari keberadaan pesan. Sebaliknya, penggunaan kriptografi akan menarik kecurigaan pihak ketiga bahwa ada sesuatu yang disembunyikan dalam pesan yang sedang dikirim. Steganografi juga memiliki kelemahan.

Tidak seperti kriptografi, steganografi memerlukan banyak ruang untuk dapat menyembunyikan beberapa bit pesan. Akan tetapi, kelemahan ini sedikit demi sedikit dapat diatasi seiring dengan perkembangan teknik-teknik dalam melakukan steganografi.

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya, kebanyakan pesan disembunyikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah yang terlihat tidak berbahaya. Perubahan ini bergantung pada kunci (sama pada kriptografi) dan pesan untuk disembunyikan. Orang yang menerima kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan.

Dalam menyembunyikan pesan, ada beberapa kriteria yang harus dipenuhi, diantaranya adalah:

1. ***Imperceptibility***. Keberadaan pesan tidak dapat dipersepsi oleh indrawi. 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, telinga haruslah mendapatkan perbedaan antara suara asli dan suara yang telah disisipi pesan.

2. ***Fidelity***. Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan yang terjadi harus tidak dapat dipersepsi oleh indra manusia.

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.

4. ***Robustness***. Data yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi yang dilakukan pada penampung, seperti pengubahan kontras, penajaman, pemampatan, rotasi, perbesaran citra, pemotongan (*cropping*), enkripsi, dan sebagainya. Bila pada dilakukan operasi operasi pengolahan tersebut, maka data yang disembunyikan seharusnya tidak rusak (tetap *valid* jika diekstraksi kembali).

2.1.1 Sejarah Steganografi

Catatan tertua mengenai penggunaan steganografi tercatat pada masa Yunani kuno. Pada saat itu, penguasa Yunani, Histiaeus, sedang ditawan oleh Raja Darius di Susa. Histiaeus ingin mengirim pesan rahasia kepada menantunya, Aristagoras, di Miletus.

Untuk itu, Histiaeus mencukur habis rambut budaknya dan menatokan pesan rahasia yang ingin dikirim di kepala budak tersebut. Setelah rambut budak tadi tumbuh cukup lebat, barulah ia dikirim ke Miletus. Cerita lain masih juga berasal dari zaman Yunani kuno. Medium tulisan pada saat itu adalah papan yang dilapisi lilin dan tulisan ditulisi di papan tersebut. Demeratus, perlu memberitahu Sparta bahwa Xerxes bermaksud untuk menginvasi Yunani. Agar pesan yang dikirimnya tidak diketahui keberadaannya, Demeratus melapisi lagi papan tulisannya dengan lilin. Papan tulisan yang terlihat masih kosong inilah yang dikirim ke Sparta. Tinta yang tidak nampak merupakan salah satu metode yang populer dalam bidang steganografi.

Bangsa Romawi telah menggunakan tinta yang tidak nampak ini untuk menulis pesan di antara baris-baris pesan yang ditulis dengan tinta biasa. Tinta yang tidak nampak ini dapat terbuat dari sari jeruk atau susu. Ketika dipanaskan, warna tinta yang tidak tampak akan menjadi gelap dan tulisannya akan menjadi dapat terbaca. Tinta yang tidak tampak ini juga digunakan dalam Perang Dunia II. Steganografi terus berkembang selama abad kelima belas dan keenam belas. Pada masa itu, banyak penulis buku yang enggan mencantumkan namanya karena takut akan kekuatan penguasa pada saat itu. Pengembangan lebih jauh lagi mengenai steganografi terjadi pada tahun 1883 dengan dipublikasikannya kriptografi militer oleh Auguste Kerckhoffs.

Meskipun sebagian besar berbicara mengenai kriptografi, Kerckhoffs menjabarkan beberapa deskripsi yang patut dicatat ketika merancang sebuah sistem steganografi. Lebih jauh lagi, Les Filigranes, yang ditulis oleh Charle Briquet di tahun 1907,

merupakan sebuah kamus sejarah dari *watermark*, salah II-4 satu wujud pengaplikasian steganografi. Dengan adanya komputer, steganografi memperoleh kemajuan yang sangat pesat. Penyembunyian pesan memasuki era baru berkat adanya komputer.

2.1.2 Manfaat Steganografi

Steganografi dapat digunakan untuk menyembunyikan informasi rahasia ke dalam media lain, untuk melindunginya dari pencurian dan dari orang-orang yang tidak berhak untuk mengetahuinya. Steganografi juga dapat digunakan untuk pengiriman pesan secara rahasia tanpa diketahui bahwa pesan sedang dikirim. Ini membuat pihak ketiga tidak menyadari keberadaan pesan.

Di sisi lain steganografi juga bisa digunakan sebagai sarana kejahatan. Steganografi dapat digunakan untuk mencuri data yang disembunyikan pada data lain sehingga dapat dikirim ke pihak lain tanpa ada yang curiga. Steganografi juga dapat digunakan oleh para teroris untuk saling berkomunikasi dengan yang lain.

2.2 Citra Digital

Citra digital dapat didefinisikan sebagai sebuah fungsi 2 dimensi, $f(x,y)$ dimana x dan y adalah koordinat bidang datar dan harga fungsi f disetiap pasangan koordinat (x,y) disebut intensitas/level keabuan (*gray level*) dari gambar dititik itu. Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar/ *pixel/ piksel/ pels/ picture element*) menyatakan tingkat keabuan pada titik tersebut. Matriks pada citra digital berukuran M (baris/tinggi) x N (kolom/lebar).

2.2.1 Resolusi Citra

Resolusi citra adalah ukuran detail terkecil pada suatu citra. Dalam pengolahan citra digital ada dua klasifikasi paling umum yang sering digunakan dalam penelitian yakni resolusi spasial dan tingkat bit.

Resolusi spasial mengacu pada detail informasi yang dapat dilihat pada citra. Semakin tinggi resolusi sebuah citra akan semakin detail citra tersebut. Resolusi spasial pada dasarnya menggambarkan jumlah piksel dalam citra. Misal citra dengan ukuran 320 x 210 merepresentasikan citra dengan jumlah piksel sebanyak 320 pada kolomnya dan 210 pada barisnya. Tingkat bit mengacu pada jumlah bit yang digunakan untuk menentukan warna pada setiap piksel suatu citra.

2.2.2 Citra Warna (8 bit)

Citra berwarna, atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap pixel dari citra hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 255 warna. Ada 2 jenis citra warna 8 bit. pertama, citra 8 bit menggunakan palet warna 255 dengan setiap palet memiliki pemetaan nilai RGB tertentu. Kedua, setiap *pixel* memiliki format 8 bit dengan pemetaan bit 5-7 bernilai red, bit 2-4 bernilai green dan bit 0 dan 1 bernilai blue.

2.2.3 Citra Warna (16 bit)

Citra warna 16 bit biasanya disebut sebagai citra highcolor dengan setiap pixelnya diwakili dengan 2 byte memory. Citra 16 bit memiliki warna 65.536 warna. Dalam formasi bitnya, nilai red dan blue mengambil tempat di 5 bit kanan dan kiri. Komponen green memiliki 5 bit ditambah 1 bit ekstra.

2.2.4 Citra Warna (24 bit)

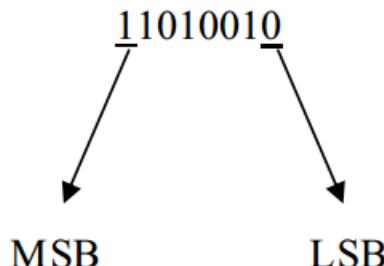
Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga total variasi warna ada 16.777.216. setiap poin informasi

pixel disimpan kedalam 8 byte data. 8 bit pertama menyimpan nilai blue, kemudian diikuti dengan nilai green pada 8 bit kedua dan pada 8 bit terakhir merupakan warna red.

2.3 Metode Least Significant Bit (LSB)

Metode *Least Significant Bit (LSB)* adalah teknik penyisipan pesan dalam steganografi dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit pesan yang akan disisipkan. Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit yang paling berarti (*Most Significant Bit* atau MSB) dan bit yang paling kurang berarti (*Least Significant Bit* atau LSB).

Contoh sebuah susunan bit pada sebuah *byte* :



Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti. Lagi pula, mata manusia tidak dapat membedakan perubahan yang kecil.

Adapun kelebihan dan kekurangan dari metode *Least Significant Bit* yaitu ukuran citra tidak berubah atau tetap, sehingga tidak mengakibatkan kecurigaan akan adanya pesan

rahasia dalam citra. Sedangkan kekurangannya adalah jumlah karakter pesan atau data yang akan disisipkan sangat terbatas, sehingga besarnya citra harus menyesuaikan besarnya pesan yang dikirim.

2.4 Perhitungan Nilai MSE dan PSNR

Untuk mengukur kualitas citra steganografi diperlukan suatu pengujian secara obyektif. Pengujian secara obyektif dilakukan dengan menghitung nilai PSNR.

Peak Signal to Ratio (PSNR) adalah perbandingan antara nilai maksimum sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR diukur dalam satuan desibel (dB). Pada penelitian ini, PSNR digunakan untuk mengetahui perbandingan kualitas citra asli dengan citra yang telah disisipkan pesan (*Stego-Image*). Untuk menentukan nilai PSNR, terlebih dahulu ditentukan nilai MSE (*Mean Square Error*). MSE adalah nilai error rata-rata antara citra asli dengan *Stego-Image*. Secara matematis MSE dirumuskan Sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x, y) - I'(x, y)]^2 \quad 2.1$$

Dimana:

MSE = Nilai *Mean Square Error* citra steganografi

M = Panjang citra (dalam *pixel*)

N = Lebar citra (dalam *pixel*)

I(x,y) = Nilai piksel dari *Stego-Image*

I'(x,y) = Nilai piksel dari citra asli

Setelah diperoleh nilai MSE maka nilai dari PSNR dapat dihitung dengan menggunakan rumus sebagai berikut :

$$PSNR = 10 \cdot \log \left(\frac{MAX_i^2}{MSE} \right) \quad 2.2$$

Dimana :

MSE = Nilai *Mean Square Error*

MAXi = Nilai Maksimum citra yang digunakan

Kualitas *Stego-Image* dikatakan relatif rendah apabila nilai PSNR dibawah 30 dB, sebaliknya kualitas *Stego-Image* dikatakan relatif tinggi apabila nilai PSNR diatas 30 dB.

BAB III

METODOLOGI

Bab ini menguraikan metode yang digunakan dalam penelitian secara rinci. Metodologi penelitian yang digunakan berguna sebagai acuan sehingga penelitian dapat berjalan sistematis.

i. Studi Literatur

Pada tahap ini akan dilakukan identifikasi masalah sekaligus mencari referensi yang berkaitan dengan steganografi dan metode LSB. Referensi yang diambil dapat berupa jurnal, buku, tugas akhir, maupun artikel yang terdapat di internet.

ii. Mengkaji Metode yang Dipakai

Pada tahap ini akan dilakukan pengkajian terhadap algoritma dari metode *Least Significant Bit(LSB)* dan Steganografi.

iii. Implementasi Algoritma Steganografi Menggunakan LSB

Setelah mengkaji metode-metode yang dipakai akan dilakukan implementasi algoritma kedalam bentuk program.

iv. Pengujian Modul – Modul Steganografi

Pada tahap ini dilakukan pengujian terhadap perangkat lunak yang telah dikembangkan. Pengujian yang dilakukan adalah perbandingan uji kualitatif berdasarkan pengamatan visual dari citra yang disisipi pesan (*StegoImage*) dan citra asli dan perbandingan ukuran file antara citra asli.

Dalam implementasi dan pengujian ini akan digunakan:

1. Perangkat lunak dan perangkat keras.

Lingkungan implementasi sistem ada dua, yaitu lingkungan perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

2. Coding

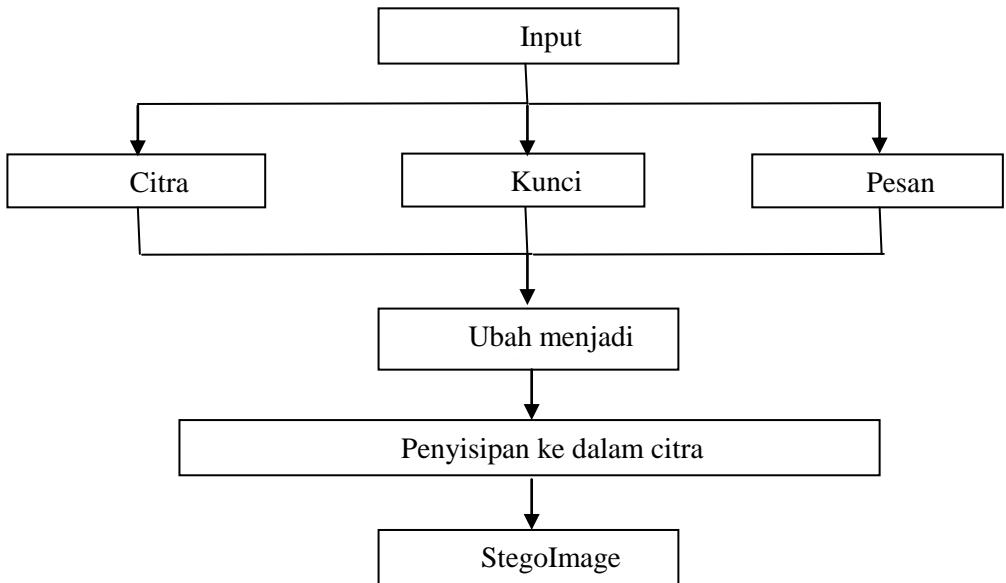
Pembuatan coding program dilakukan menggunakan program *MATLAB*. Kumpulan dari semua program yang telah diintegrasikan perlu dites kembali untuk melihat apakah suatu program dapat menerima masukan data dengan baik, dapat memprosesnya dengan baik dan dapat memberikan *output* kepada program lainnya.

v. Penulisan tugas akhir

Pada tahap terakhir ini akan dibuat laporan dari hasil penelitian yang telah dilakukan pada tahap-tahap sebelumnya.

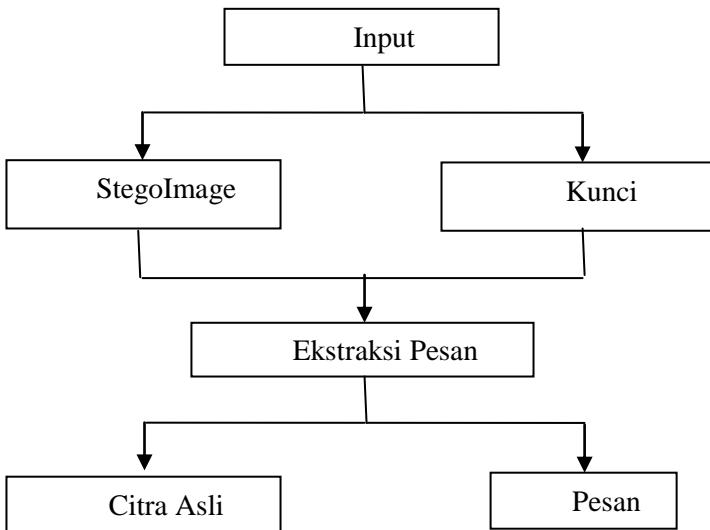
vi. Diagram Alir

Pada tahap ini disusun diagram alir sebagaimana ditunjukkan pada Gambar 3.1 sebagai langkah-langkah yang dilakukan untuk mencapai tujuan dari penelitian.



Gambar 3.1 Diagram alir Penyisipan Pesan

Pada gambar 3.1 merupakan diagram alir proses penyisipan teks ke dalam citra. User menginputkan pesan, kunci dan citra. Kemudian pesan, kunci dan piksel citra dirubah menjadi kode biner. Selanjutnya pesan dan kunci disisipkan ke bit akhir citra dan disimpan sebagai *StegoImage*.



Gambar 3.2 Diagram alir Ekstraksi Pesan

Pada gambar 3.2 merupakan diagram alir proses ekstraksi pesan. User menginputkan citra yang telah disisipkan pesan (*StegoImage*) dan Kunci. Selanjutnya dilakukan proses ekstraksi pesan sehingga didapat citra asli dan data teks yang disisipkan.

BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini membahas hasil analisa dan perancangan yang meliputi pembahasan mengenai deskripsi metode, analisa perangkat lunak dan perancangan perangkat lunak.

4.1 Analisa Kebutuhan

4.1.1 Analisa Kebutuhan User

Pada perangkat lunak yang dirancang ini harus memenuhi beberapa kebutuhan *user* sebagai pengguna perangkat lunak:

1. Perangkat lunak yang dirancang mempunyai tampilan yang familiar bagi pengguna.
2. Data kunci dan pesan tidak akan tersimpan secara permanen di dalam sistem untuk alasan keamanan data.

4.1.2 Analisa Kebutuhan Sistem

Perangkat lunak ini dikembangkan menggunakan software *Jcreator* baik dari desain antar muka, sistem penyisipan dan ekstraksi pesan. Untuk perangkat keras dijelaskan pada tabel dibawah ini.

Tabel 4.1 Tabel Kebutuhan

Perangkat Keras	Prosesor	: AMD A4-3330MX APU with Radeon(tm) HD Graphics 2.30 GHz
	RAM	: 1720 MB
	Sistem Operasi	: Windows 7 Ultimate

4.2 Deskripsi Metode

Steganografi dengan menggunakan metode *Least Significant Bit(LSB)* membutuhkan 3 (tiga) data masukan untuk memulai proses, yaitu berkas citra digital, kunci, dan pesan rahasia.

4.2.1 Deskripsi Metode Penyisipan

Untuk memperjelas metode ini, maka akan diberikan contoh sebagai berikut:

Misalkan terdapat pesan “jul” dan kunci “R” yang memiliki kode ASCII sebagai berikut:

106	117	108
j	u	l

82
R

Kemudian pesan dan kunci dirubah menjadi kode biner menjadi

01101010	01110101	01101110
j	u	l

01010010
R

Misalkan citra yang akan digunakan sebagai wadah adalah citra “langit.jpg”



Gambar 4.1Citra sebelum disisipkan pesan

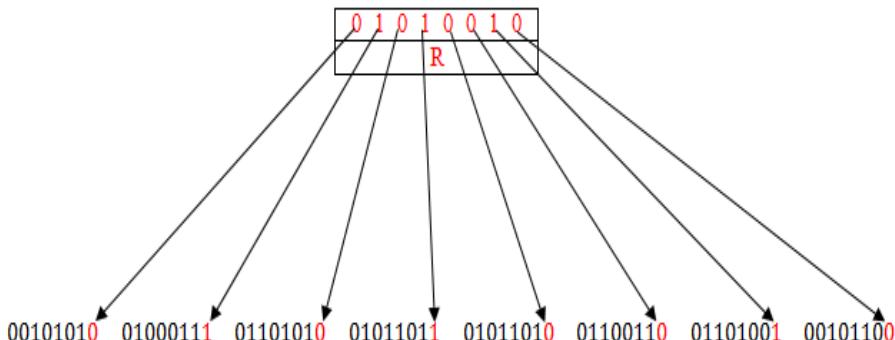
Dan misalkan diambil channel red (merah) untuk tempat menyisipkan pesan memiliki matriks 10×4 sebagai berikut:

$$\begin{pmatrix} 41 & 21 & 90 & 31 & 71 & 75 & 91 & 90 \\ 65 & 51 & 81 & 66 & 74 & 79 & 95 & 93 \\ 50 & 87 & 96 & 78 & 92 & 71 & 40 & 61 \\ 65 & 43 & 76 & 82 & 99 & 78 & 72 & 39 \\ 65 & 85 & 55 & 49 & 95 & 70 & 75 & 35 \\ 73 & 67 & 50 & 47 & 69 & 77 & 45 & 34 \end{pmatrix}$$

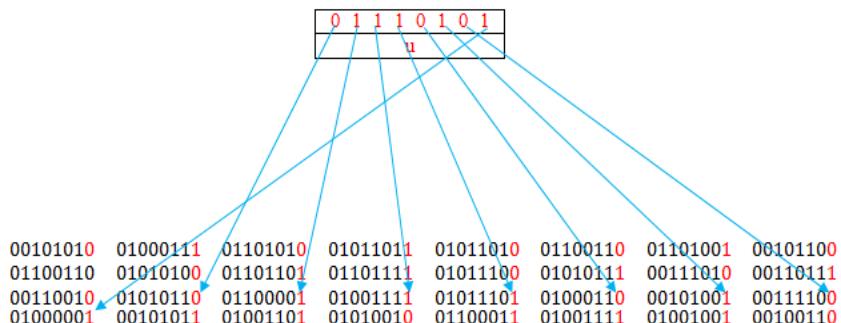
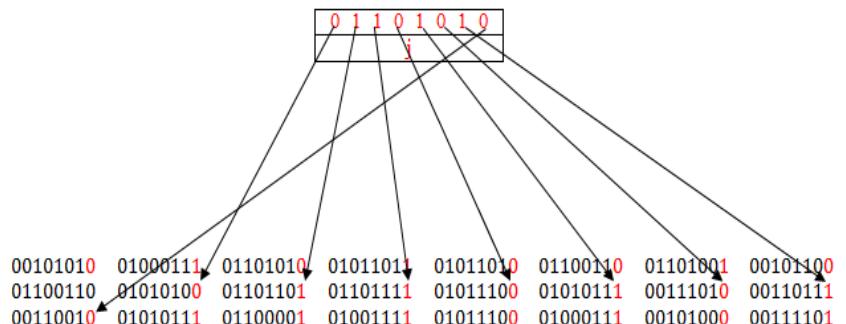
Kemudian matriks di atas dirubah menjadi bentuk biner

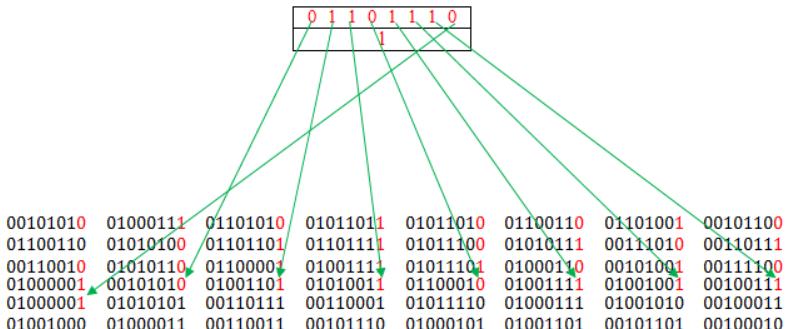
00101011	01000111	01101010	01011010	01011010	01100110	01101001	00101101
01100110	01010101	01101100	01101110	01011101	01010111	00111010	00110111
00110010	01010111	01100000	01001110	01011100	01000111	00101000	00111101
01000001	00101011	01001100	01010010	01100011	01001110	01001000	00100111
01000001	01010101	00110111	00110001	01011111	01000110	01001011	00100011
01001001	01000011	00110010	00101111	01000101	01001101	00101101	00100010

Selanjutnya kunci yang telah dirubah menjadi bentuk biner disisipkan ke elemen awal matriks citra yang telah dirubah menjadi kode biner



Selanjutnya penyisipan pesan yang telah dirubah menjadi bentuk matriks dimulai satu elemen setelah elemen terakhir kunci disisipkan





Kemudian matriks di atas dirubah kembali menjadi bentuk desimal

$$\begin{pmatrix} 40 & 21 & 90 & 32 & 71 & 75 & 91 & 90 \\ 65 & 50 & 82 & 67 & 73 & 79 & 95 & 93 \\ 50 & 86 & 97 & 80 & 93 & 71 & 40 & 60 \\ 64 & 43 & 77 & 82 & 99 & 78 & 73 & 38 \\ 64 & 85 & 55 & 49 & 94 & 71 & 74 & 35 \\ 72 & 67 & 51 & 46 & 69 & 77 & 45 & 34 \end{pmatrix}$$

Terakhir simpan sebagai *StegoImage*.



Gambar 4.2Citra sesudah disisipkan pesan

4.2.2 Deskripsi Metode Ekstraksi

Pada proses ekstraksi, citra yang diinputkan adalah citra yang telah disisipkan pesan (*StegoImage*)



Gambar 4.3Citra sesudah disisipkan pesan

$$\begin{pmatrix} 40 & 21 & 90 & 32 & 71 & 75 & 91 & 90 \\ 65 & 50 & 82 & 67 & 73 & 79 & 95 & 93 \\ 50 & 86 & 97 & 80 & 93 & 71 & 40 & 60 \\ 64 & 43 & 77 & 82 & 99 & 78 & 73 & 38 \\ 64 & 85 & 55 & 49 & 94 & 71 & 74 & 35 \\ 72 & 67 & 51 & 46 & 69 & 77 & 45 & 34 \end{pmatrix}$$

Matriks di atas merupakan matriks dari *StegoImage*, kemudian matriks dirubah menjadi bentuk biner menjadi:

00101010	01000111	01101010	01011011	01011010	01100110	01101001	00101100
01100110	01010100	01101101	01101111	01011100	01010111	00111010	00110111
00110010	01010110	01100001	01001111	01011101	01000110	00101001	00111100
01000001	00101010	01001101	01010011	01100010	01001111	01001001	00100111
01000001	01010101	00110111	00110001	01011110	01000111	01001010	00100011
01001000	01000011	00110011	00101110	01000101	01001101	00101101	00100010

Lalu inputkan kunci

82
R

Kemudian kunci dirubah menjadi bentuk biner

01010010
R

Selanjutnya sistem akan menyocokkan kunci yang diinputkan dengan kunci yang ada dalam piksel citra. Apabila kunci sesuai maka sistem mengambil bit-bit pesan yang disisipkan. Terakhir, sistem mengubah bit-bit pesan menjadi teks dan menampilkannya pada perangkat lunak.

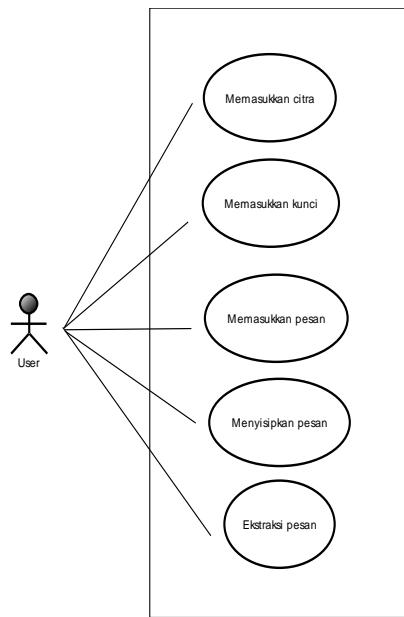
4.3 Perancangan Perangkat Lunak

Pada subbab ini menjelaskan proses perancangan perangkat lunak meliputi perancangan gambaran umum sistem yang disajikan dalam bentuk *Use Case Diagram* dan *Activity Diagram*, perancangan proses algoritma dan perancangan antar muka sistem.

4.3.1 Perancangan Gambaran Umum Sistem

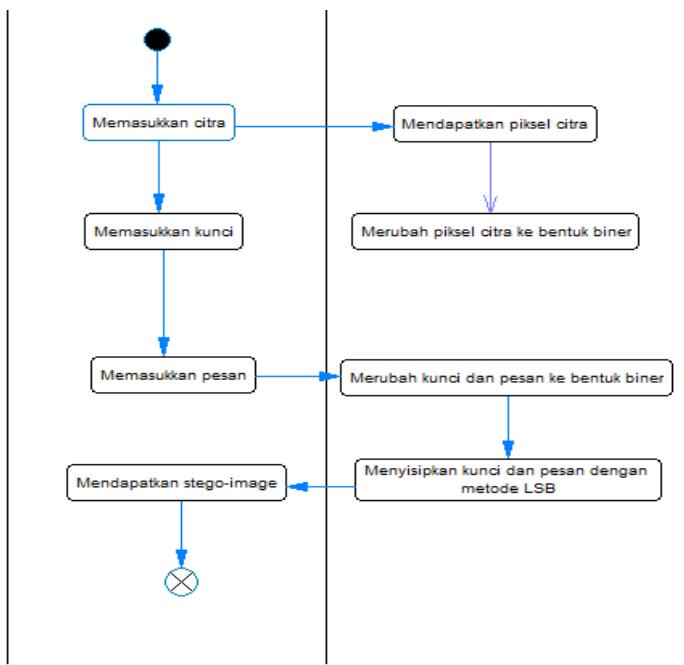
Gambaran Gambaran awal dari perangkat lunak ini menunjukkan hal – hal yang dapat dilakukan oleh *user* yang disajikan dalam *Use Case Diagram*. Dari *Use Case Diagram* dapat kita simpulkan bahwa *user* dapat melakukan 5 hal:

1. Menentukan citra masukan.
2. Menentukan kunci.
3. Menentukan pesan.
4. Melakukan proses penyisipan pesan.
5. Melakukan proses ekstraksi pesan.



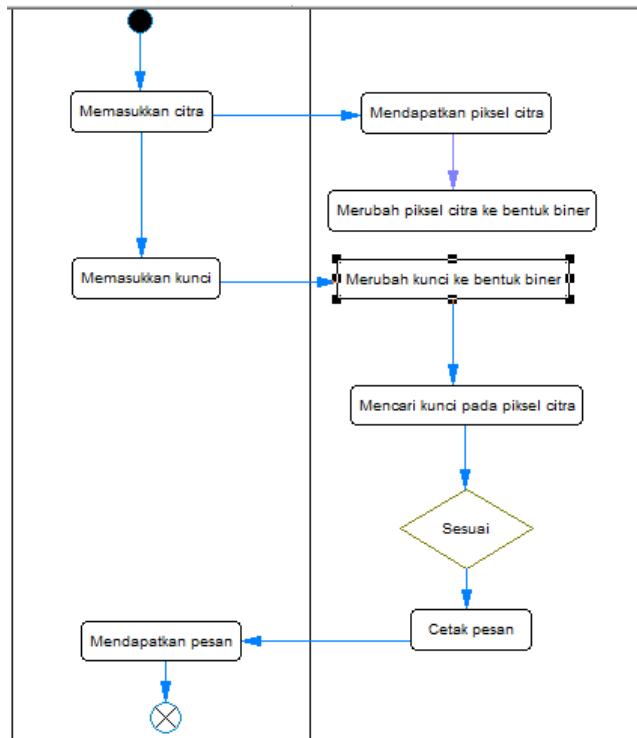
Gambar 4.3 Use Case Diagram perangkat lunak steganografi

Kemudian untuk proses penyisipan pesan dan ekstraksi pesan dijelaskan pada Activity Diagram.



Gambar 4.4ActivityDiagram penyisipan pesan

Pada proses ini citra masukkan berupa citra berwarna. Kemudian sistem merubah piksel citra ke bentuk biner. Kemudian masukkan kunci dan pesan yang akan disisipkan. Kunci dan pesan diubah ke bentuk biner. Tahap selanjutnya menyisipkan kunci dan pesan pada bit akhir citra dan disimpan sebagai *StegoImage*.



Gambar 4.5ActivityDiagram ekstraksi pesan

Dari diagram proses ekstraksi,citra masukan pesan tersebut berupa citra yang mengandung pesan didalamnya (*StegoImage*). Lalu *user* memasukkan sebuah kunci yang nantinya akan dirubah menjadi kode biner. Sistem kemudian menyocokkan kunci dengan kunci yang ada dalam piksel citra. Apabila kunci sesuai maka sistem akan merubah piksel citra yang terdapat kunci dan pesan menjadi teks. Terakhir, sistem hanya mengambil pesan dan menampilkannya pada perangkat lunak.

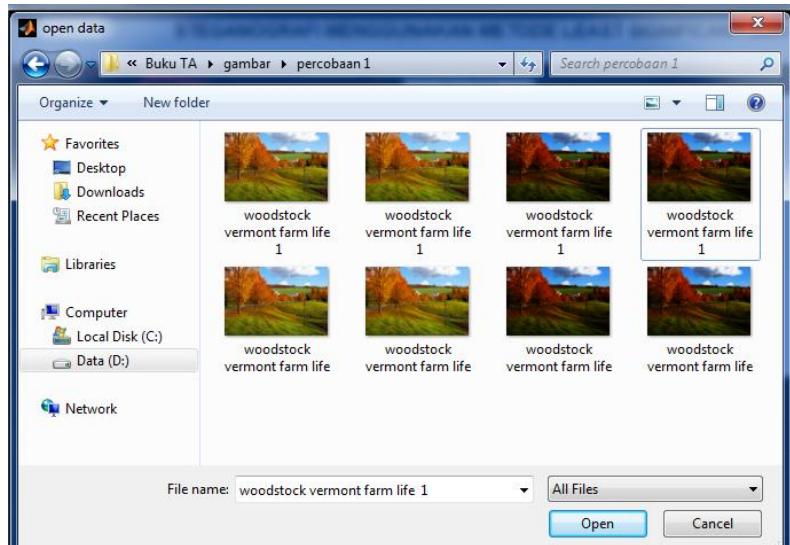
4.4 Pemrograman

4.4.1 Implementasi Masukkan Citra

Masukkan pada perangkat lunak ini berupa citra warna 16 bit. Proses tersebut diimplementasikan dalam *source code* berikut:

```
[filename,path]=uigetfile([{ '*.bmp' };{ '*.jpg' };{ '*.tiff' };{ '*.png' }], 'open data');
if isequal(filename,0)
    return
end
% baca citra sesuai direktori
img=imread(fullfile(path,filename));
% tampilkan citra
axes(handles.axes1)
imshow(img)
% simpan data img dlm figure
setappdata(handles.figure1,'img',img)
```

Kode program lebih lengkap dapat dilihat pada lampiran B.



Gambar 4.6 Tampilan antar muka pengambilan masukkan

4.4.2 Implementasi Penyisipan Pesan

Pada proses ini dilakukan penyisipan pesan setelah *user* memasukkan citra masukkan. Selanjutnya *user* memasukkan kunci dan pesan pada perangkat lunak dan proses penyisipan bisa dilakukan. Proses penyisipan diimplementasikan dalam *sourcecode* berikut:

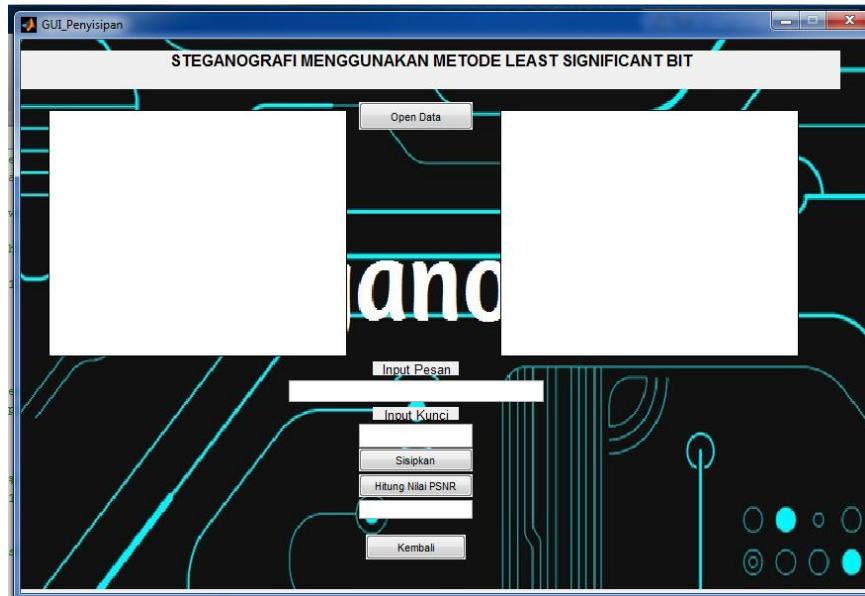
```
k=0;
for i = 1 : size(c,2) % lebar citra
    for j = 1 : size(c,1) % panjang citra
        k = k + 1;
        ig=c(j,i); % baca nilai pixel citra
baris ke i dan kolom ke j
        bg(k,:)=dec2bin(ig,8); % ubah ke
biner
        if (k <= size(b,1)) % sisipkan
sebanyak nilai biner (text)
```

```

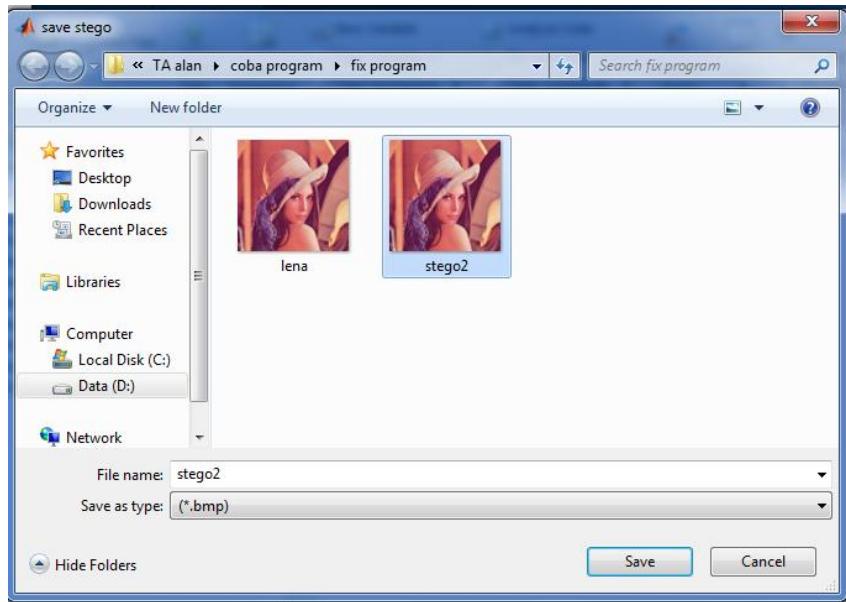
            bg(:,end)=num2str(b(k));           %
sisipkan dibagian akhir array
        end
        % setelah penyisipan ubah kembali
ke nilai desimal (pixel)
        d=(bg(k,:));
        hg(k,:)=bin2dec(d);
    end
end
s=reshape(hg,size(c)); % ubah ukuran seperti
citra awal
s=uint8(s); % transpose dan ubah ke nilai 8
bit
ky=size(b,1)/8; % ukuran panjang nilai biner
imgs(:,:,1)=s; % ubah chanel r (merah) dengan
nilai hasil penyisipan
imgs(end,end,1)=ky; % simpan jumlah nilai
biner
imgs(end,end-1,1)=1; % simpan jumlah panjang
kata
% tampilkan di axes 2
axes(handles.axes2)
imshow(imgs)
% simpan hasil penyisisipan
[filename,path]=uiputfile([{'*.bmp'}; {'*.jpg'}
}; {'*.tiff'}; {'*.png'}], 'save stego');
if isequal(filename,0)
    return
end
if strcmp(filename(:,end-2:end), 'bmp') || %
strcmp(filename(:,end-2:end), 'iff')
    imwrite(imgs, fullfile(path, filename));
else
    imwrite(imgs, fullfile(path, filename),
'Mode', 'lossless');
end

```

Kode program lebih lengkap dapat dilihat pada lampiran B.



Gambar 4.7Tampilan antar muka masukkan kunci dan pesan



Gambar 4.8 Tampilan antar muka penyimpanan *StegoImage*

4.4.3 Implementasi Ekstraksi Pesan

Pada proses ekstraksi pesan ini citra masukkan berupa citra yang mengandung pesan (*StegoImage*). Proses ekstraksi diimplementasikan dalam *source code* berikut :

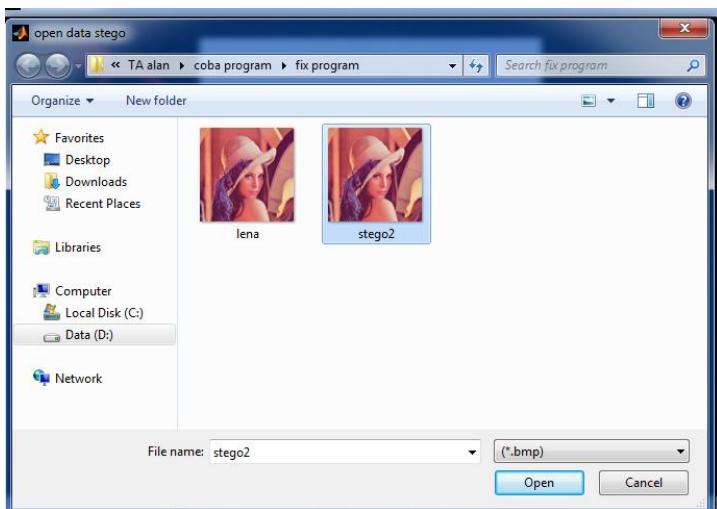
```
% ekstraksi
for h=1:ky
    si=s(h,:); % baca pixel yang disiipkan
    db=dec2bin(si,8); % ubah ke biner
    ds(h,:)=str2num(db(:,end)); % ubah ke
numerik
end
% ubah ke biner string
for k = 1:size(ds,1);
    ib=ds(k);
    if(ib == 1)
        es(k) = '1';
```

```

        else
            es(k) = '0';
        end
    end
    % ubah kedalam ukuran biner awal
    tx=reshape(es,[8,1]);
    % ubah nilai biner ke desimal kemudian di
    ubah ke text
    for i=1:size(tx,2)
        thisString=char(tx(:,i))'; % baca nilai
        biner string
        thisChar(:,i) = char(bin2dec((thisString))); % ubah dari biner
        ke desimal kumudian ke text (string)
    end
    % cek key benar atau salah
    C = strsplit(thisChar);
    key=get(handles.edit1,'string');
    if ~strcmp(C(:,1),key);
        warndlg('key salah')
        set(handles.edit2,'string',[])
        return
    end
    % jika benar tampilkan pesan ke edit 2
    Ct=[];
    for j=2:size(C,2)
        Ci=cell2mat((C(:,j)));
        Ct=[Ct,' ',Ci];
    end
    set(handles.edit2,'string',Ct

```

Kode program lebih lengkap dapat dilihat pada lampiran C.



Gambar 4.9Tampilan antar muka pengambilan *StegoImage*.



Gambar 5.0Tampilan antar muka masukkan kunci dan pesan yang disisipkan.

“Halaman ini sengaja dikosongkan”

BAB V

PENGUJIAN DAN PEMBAHASAN

Bab ini menampilkan hasil uji coba, perbandingan uji kualitatif, dan perbandingan ukuran file antara citra asli dan citra yang telah disisipi pesan. Hasil pengujian ini digunakan dalam perumusan kesimpulan dan saran.

5.1 Data Uji Coba

Uji coba pada perangkat lunak dalam tugas akhir ini dilakukan terhadap 3 citra yang berbeda serta menggunakan 3 jenis kunci dan pesan yang berbeda.

Tabel 5.1 Data citra yang digunakan

No	Nama	Resolusi	Gambar
1	Woodstock vermont.jpg	736x 490	
2	Langit Biru.jpg	350 x 229	
3	Bean.jpg	650 x 365	

5.2 Pengujian Kualitatif

Pengujian kualitatif dilakukan berdasarkan pengamatan visual dari citra yang disisipi pesan (*Stegoimage*). Dalam penelitian ini pengujian kualitatif dibagi menjadi 4 kriteria, yaitu:

1. Baik Sekali : *StegoImage* tidak dapat dibedakan dengan citra asli.
2. Baik : *StegoImage* dapat dibedakan dengan citra asli dengan pengamatanya yang teliti.
3. Jelek : *StegoImage* yang disisipi dapat dibedakan dengan citra asli dengan mudah.
4. Jelek Sekali : Perbedaan nyata terlihat jelas.

- **Uji Pertama**

Pada uji pertama ini menggunakan kunci “ITS” pesan yang disisipkan adalah “fakultas matematika, komputasi, dan sains data”, panjang dari pesan yang disisipkan untuk uji pertama ini adalah 6 kata.

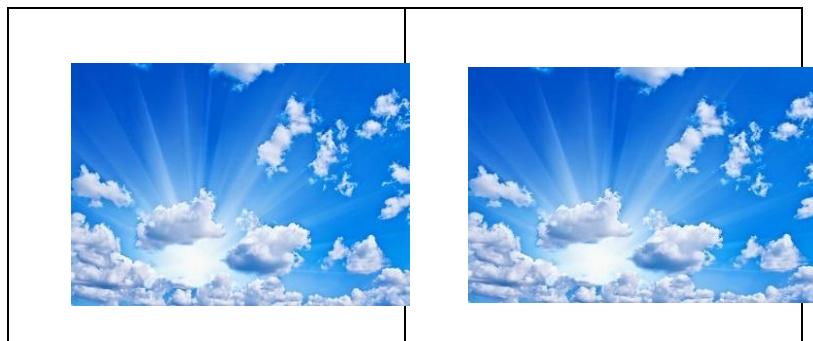


Table 5.2 Citra Langit Uji Pertama
Gambar (a) Citra Asli Dan Gambar (b) *StegoImage*



Table 5.3 Citra Wajah Uji Pertama
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*



Table 5.4 Citra Pemandangan Uji Pertama
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*

- **UjiKedua**

Pada uji kedua ini menggunakan kunci “Ketika” pesan yang disisipkan adalah “ketikaembunpagi di rerumputan daun – daun serentakmembekuputihbagaisalju. Ketika embunan kasih yang kudapatkan hatipun tersanjung sampai kelangit biru meski embun

memutih nan kujumpa, sejuk terasa di raga ini meski kita bersua hanya dimaya, engkau sejukan jiwa ini ketika embun pagi di rerumputan daun – daun serentak membeku putih bagai salju. Ketika embunan kasih yang kudapatkan hatipun tersanjung sampai ke langit biru meski embun memutih nan kujumpa, sejuk terasa di raga ini meski kita bersua hanya dimaya , engkau sejukan jiwa ini”, panjang dari pesan yang disisipkan untuk uji kedua ini adalah 86 kata.

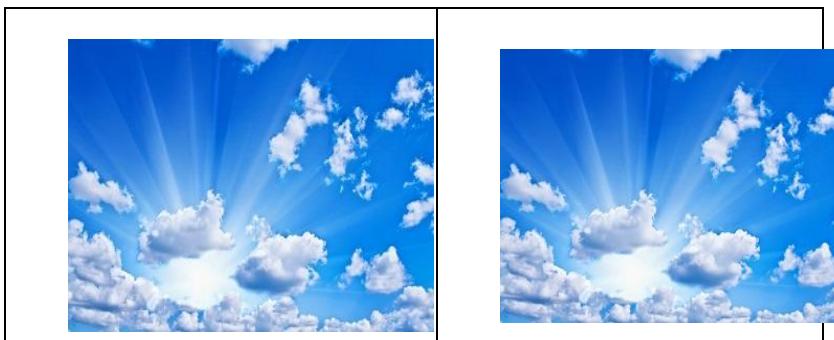


Table 5.5 Citra Langit Uji Kedua
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*



Table 5.6 Citra Wajah Uji Kedua
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*



Table 5.7 Citra Pemandangan Uji Kedua
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*

- **UjiKetiga**

Pada uji ketiga ini menggunakan kunci “Abstrak”pesan yang disisipkan adalah “Berbagai macam teknik untuk melindungi informasi yang dirahasiakan telah banyak dilakukan. Steganografi adalah salah satu teknik yang digunakan dalam penyembunyian pesan ke dalam sebuah media sedemikian sehingga manusia sulit menyadari keberadaan pesan tersebut. Pada tugas akhir ini teknik steganografi yang digunakan metode *Least Significant Bit* (LSB). Metode *Least Significant Bit* (LSB) adalah metode menyembunyikan pesan pada bit terakhir citra digital sehingga tidak terjadi perubahan secara kasat mata. Pada metode *Least Significant Bit* (LSB) dibutuhkan 3 input berupa citra, pesan, dan kunci. Pertama kunci dan pesan dirubah kedalam bentuk ASCII, setelah itu dirubah kedalam bentuk biner. Kemudian diambil channel merah (red) dari citra yang dirubah kedalam bentuk biner. Selanjutnya pesan dan kunci disisipkan pada citra tersebut. Terakhir citra yang telah disisipkan pesan disimpan dengan nama *Stego-Image*. Untuk uji coba paenulis menggunakan 3 citra dengan karakteristik yang berbeda, yaitu: citra langit yang cenderung homogen, citra wajah yang sedikit komplek, dan citra pemandangan yang lebih komplek. Disamping

itu uji coba juga dilakukan pada 4 tipe citra yang berbeda, yaitu bmp, jpeg, png, dan tiff. Hasil yang dicapai pada tugas akhir ini, citra yang disisipkan pesan tidak mengalami perubahan bentuk secara kasat mata,nilai rata-rata PSNR dari keempat tipe citra tersebut untuk penyisipan teks 250 kata adalah sebagai berikut: citra langit 68.22975 dB, citra wajah 72.228575 dB, citra pemandangan 74.322525 dB dan pesan yang disisipkan dapat dikembalikan seperti semula saat proses ekstraksi.”, panjang dari pesan yang disisipkan untuk uji ketiga ini adalah 228 kata.



Table 5.8 Citra Pemandangan Uji Ketiga
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*



Table 5.9 Citra Langit Uji Kedua
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*



Table 5.10 Citra Wajah Uji Kedua
Gambar (a) CitraAsli Dan Gambar (b) *StegoImage*

5.3 Pengujian Penyisipan dan Ekstraksi Pesan

Pada subbab ini dilakukan proses penyisipan dan ekstraksi pesan untuk menguji apakah kedua hal tersebut dapat dilakukan untuk semua format citra. Hasil dari pengujian tersebut disajikan pada table di bawah ini :

Tabel 5.11 Hasil pengujian sistem pada proses penyisipan dan ekstraksi pada semua format citra

Fomat Citra	Penyisipan	Ekstraksi
Png	Berhasil	Berhasil
Bmp	Berhasil	Berhasil
Jpg	Berhasil	Berhasil
Tiff	Berhasil	Berhasil

5.4 Perbandingan Ukuran File

Pada subbab ini dilakukan perbandingan ukuran file antara *StegoImage* dengan citra asli. Panjang kunci dan pesan mengikuti aturan pengujian sebelumnya. Hasil perbandingan ukuran file tersebut disajikan pada table di bawahini :

Tabel 5.12 Hasil perbandingan ukuran citra asli dan *StegoImage* pada pengujian Pertama

No.	Nama	Format	Citra Asli	<i>StegoImage</i>
1	woodstock vermont	Bmp	1.03MB	1.03MB
2	woodstock vermont	Jpg	101 KB	785 KB
3	woodstock vermont	Tiff	1.03MB	1.03MB
4	woodstock vermont	Png	718 KB	624 KB
5	Bean	Bmp	695KB	695KB
6	Bean	Jpg	31.2 KB	374 KB
7	Bean	Tiff	695 KB	700 KB
8	Bean	Png	298 KB	306 KB
9	Langit Biru	Bmp	235 KB	235 KB
10	Langit Biru	Jpg	31.4KB	133 KB
11	Langit Biru	Tiff	235KB	236 KB
12	Langit Biru	Png	127KB	128KB

Tabel 5.13 Hasil perbandingan ukuran citra asli dan *StegoImage* pada pengujian kedua

No.	Nama	Format	Citra Asli	<i>StegoImage</i>
1	woodstock vermont	Bmp	1.03 MB	1.03 MB
2	woodstock vermont	Jpg	101 KB	785 KB
3	woodstock vermont	Tiff	1.03MB	1.03MB
4	woodstock vermont	Png	718 KB	624 KB
5	Bean	Bmp	197 KB	197KB
6	Bean	Jpg	85.6 KB	85.6 KB
7	Bean	Tiff	88 KB	88 KB
8	Bean	Png	100 KB	100 KB
9	Langit Biru	Bmp	722 KB	722 KB
10	Langit Biru	Jpg	15.5KB	15.5 KB
11	Langit Biru	Tiff	721KB	721KB
12	Langit Biru	Png	191KB	191 KB

Tabel 5.14 Hasil perbandingan ukuran citra asli dan *StegoImage* pada pengujian ketiga

No.	Nama	Format	Citra asli	<i>Stegoimage</i>
1	woodstock vermont	Bmp	1.03 MB	1.03 MB
2	woodstock vermont	Jpg	101 KB	785 KB
3	woodstock vermont	Tiff	1.03MB	1.03MB
4	woodstock vermont	Png	718 KB	624 KB
5	Bean	Bmp	197 KB	197KB

6	Bean	Jpg	85.6 KB	85.6 KB
7	Bean	Tiff	88 KB	88 KB
8	Bean	Png	100 KB	100 KB
9	Langit Biru	Bmp	722 KB	722 KB
10	Langit Biru	Jpg	15.5KB	15.5 KB
11	Langit Biru	Tiff	721KB	721KB
12	Langit Biru	Png	191KB	191 KB

5.5 Pengujian Nilai *PSNR* (*Peak Signal to Noise Ratio*)

Pengujian PNSR (*Peak Signal to Noise Ratio*) digunakan untuk mengukur kualitas citra yang dihasilkan. Metode PNSR adalah ukuran perbandingan antara nilai piksel citra awal dengan nilai piksel pada *StegoImage* yang dihasilkan. Hasil dari pengujian nilai *PSNR* disajikan pada table di bawah ini :

Tabel 5.15 Hasil pengujian nilai *PSNR* pada pengujian Pertama

No.	Nama	Format	<i>Stegoimage</i>	<i>PSNR</i>
1	stego vermont 1	Bmp	1.03 MB	80.7642 dB
2	stego vermont 2	Jpg	785 KB	80.7716 dB
3	stego vermont 3	Tiff	1.03MB	81.7241 dB
4	stego vermont 4	Png	624 KB	81.7241 dB
5	stego Bean 1	Bmp	197 KB	78.8726 dB
6	stego Bean 2	Jpg	85.6 KB	78.8726 dB
7	stego Bean 3	Tiff	88 KB	78.7162 dB
8	stego Bean 4	Png	100 KB	78.7162 dB

9	stego Langit Biru 1	Bmp	722 KB	74.2547 dB
10	stego Langit Biru 2	Jpg	15.5KB	74.2621 dB
11	stego Langit Biru 3	Tiff	721KB	83.1329 dB
12	stego Langit Biru 4	Png	191KB	83.1329 dB

Tabel 5.16 Hasil pengujian nilai *PSNR* pada pengujian Kedua

No.	Nama	Format	<i>Stegoimage</i>	<i>PSNR</i>
1	stego vermont 1	Bmp	1.03 MB	77.8809 dB
2	stego vermont 2	Jpg	785 KB	77.8961 dB
3	stego vermont 3	Tiff	1.03MB	77.6092 dB
4	stego vermont 4	Png	624 KB	77.6092 dB
5	stego Bean 1	Bmp	197 KB	75.9648 dB
6	stego Bean 2	Jpg	85.6 KB	75.9648 dB
7	stego Bean 3	Tiff	88 KB	75.1251 dB
8	stego Bean 4	Png	100 KB	75.1251 dB
9	stego Langit Biru 1	Bmp	722 KB	71.3568dB
10	stego Langit Biru 2	Jpg	15.5KB	71.3567 dB
11	stego Langit Biru 3	Tiff	721KB	71.8211 dB
12	stego Langit Biru 4	Png	191KB	71.7211 dB

Tabel 5.17 Hasil pengujian nilai *PSNR* pada pengujian ketiga

No.	Nama	Format	<i>Stegoimage</i>	<i>PSNR</i>
1	stego vermont 1	Bmp	1.03 MB	74.3684 dB
2	stego vermont 2	Jpg	785 KB	74.3819 dB
3	stego vermont 3	Tiff	1.03MB	74.2699 dB
4	stego vermont 4	Png	624 KB	74.2699 dB
5	stego Bean 1	Bmp	197 KB	72.5819 dB
6	stego Bean 2	Jpg	85.6 KB	72.587 dB
7	stego Bean 3	Tiff	88 KB	71.8727 dB
8	stego Bean 4	Png	100 KB	71.8727 dB
9	stego Langit Biru 1	Bmp	722 KB	68.0512dB
10	stego Langit Biru 2	Jpg	15.5KB	68.037dB
11	stego Langit Biru 3	Tiff	721KB	68.4154dB
12	stego Langit Biru 4	Png	191KB	68.4154dB

Berdasarkan pengujian *PSNR* didapatkan bahwa nilai *PSNR* di atas 30dB, berarti kualitas antara citra asli dengan *StegoImage* tidak mengalami perubahan yang signifikan. Jadi keberadaan dari file yang tersembunyi tidak mudah terdeteksi oleh indra penglihatan manusia.

BAB VI

PENUTUP

Pada bab ini diberikan kesimpulan sebagai hasil dari analisa model yang telah diperoleh dan saran sebagai pertimbangan dalam pengembangan lebih lanjut.

6.1 Kesimpulan

Berdasarkan analisa dari hasil pengujian perangkat lunak steganografi menggunakan metode *Least Significant Bit(LSB)* ini, penulis dapat menarik kesimpulan sebagai berikut :

1. Hasil dari penerapan untuk penyisipan pesan rahasia pada gambar berjalan dengan baik. Pesan yang disisipkan pada citra dapat diperoleh kembali secara utuh.
2. Citra yang telah disisipkan pesan (*StegoImage*) tidak mengalami perubahan yang signifikan dengan citra asli, sehingga secara kasat mata tidak dapat diketahui bahwa terdapat pesan rahasia pada citra tersebut.
3. Dari uji coba diperoleh hasil bahwa citra yang lebih kompleks mempunyai hasil yang lebih baik, terbukti dengan memiliki rata-rata nilai PSNR 74.322525 dB.

6.2 Saran

Berdasarkan hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Pada penelitian ini penulis hanya menggunakan media citra digital sebagai media penampung, diharapkan untuk penelitian selanjutnya menggunakan media audio, video, dan lain sebagainya.
2. Program masih menyisipkan pesan dalam bentuk *plaintext* pada penelitian berikutnya diharapkan pesan tersebut terenkripsi.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

- [1] Cole, Eric. (2003). “*Hiding in Plain sight : Steganography and the Art of Covert Communication*”. **Wiley Publishing, Inc.**
- [2] Rakmat, Basuki., dan Fairuzabadi, Muhammad. 2010. “Steganografi menggunakan Metode Least Significant Bit dengan Kombinasi Algoritma Criptografi Vignere dan RC4”
- [3] D. Chopra, P. Gupta, G. S. B.C , A. Gupta. 2012, “Lsb Based Digital Image Watermarking For Gray Scale Image,” *Jurnal of Computer Engineering (IOSRJCE)*, vol. 6, no. 1, pp. 36-41.
- [4] Kruus, Peter, Caroline Scace, Michael Heyman, dan Mathew Mundy. (2002)“*A Survey of Steganographic Techniques for Citra Files*”. **Advanced Security Research Journal – Network Associates Laboratories, Network Associates, Inc.**
- [5] Irfan. 2013. “Penyembunyian Informasi (steganography) Gambar Menggunakan Metode LSB (*Least Significant Bit*)”.
- [6] Alim Muadzani, Oky Dwi Nurhayati, Ike Pertwi Windasari. 2016. “Penyisipan Media Teks dan Citra Menggunakan Teknik Steganografi pada Media Pembawa Citra Digital”. **Semarang: Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro.**
- [7] Ineke Pakereng, Yos Richard Beeh, Sonny Endrawan. (2010). “Perbandingan Steganografi Metode *Spread Spectrum* dan *Least Significant Bit (LSB)*”.
- [8] Yeni Setiani. (2008).”**PEMBUATAN APLIKASI STEGANOGRAFI MENGGUNAKAN MATLAB**”.
- [9] Hermawanti, Fajar Astuti. (2013).”**Pengolahan Citra Digital : Konsep&Teori**”. ANDI. Yogyakarta.
- [10] Munir, Rinaldi. 2013. “Bahan Kuliah IF4020 Criptografi”.

“Halaman ini sengaja dikosongkan”

LAMPIRAN A

Source Code HalamanAwal

```
function varargout = tampilan1(varargin)
% TAMPILAN1 MATLAB code for tampilan1.fig
%             TAMPILAN1, by itself, creates a new
TAMPILAN1 or raises the existing
%           singleton*.
%
%           H = TAMPILAN1 returns the handle to a
new TAMPILAN1 or the handle to
%           the existing singleton*.
%
%
TAMPILAN1('CALLBACK', hObject, eventData, handles, .
..) calls the local
%           function named CALLBACK in TAMPILAN1.M
with the given input arguments.
%
%           TAMPILAN1('Property','Value',...)
creates a new TAMPILAN1 or raises the
%           existing singleton*. Starting from
the left, property value pairs are
%           applied to the GUI before
tampilan1_OpeningFcn gets called. An
%           unrecognized property name or invalid
value makes property application
%           stop. All inputs are passed to
tampilan1_OpeningFcn via varargin.
%
%           *See GUI Options on GUIDE's Tools
menu. Choose "GUI allows only one
%           instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response
to help tampilan1
```

```

% Last Modified by GUIDE v2.5 20-Feb-2018
17:13:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State        = struct('gui_Name',
filename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @tampilan1_OpeningFcn, ...
'gui_OutputFcn', @tampilan1_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback
    = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}]
    = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before tampilan1 is made
visible.
function      tampilan1_OpeningFcn(hObject,
 eventdata, handles, varargin)
    % This function has no output args, see
OutputFcn.
    % hObject    handle to figure
    % eventdata   reserved - to be defined in a
future version of MATLAB
    % handles    structure with handles and user
data (see GUIDATA)

```

```

% varargin      command line arguments to
tampilan1 (see VARARGIN)

% Choose default command line output for
tampilan1
handles.output = hObject;
guidata(hObject, handles);
hback = %
axes('units','normalized','position',[0 0 1 1]);
uistack(hback,'bottom'); % menciptakan axes
untuk tempat menampilkan gambar
% menampilkan background
[back map]=imread('maxresdefault.jpg');
image(back)
colormap(map)
background=imread('maxresdefault.jpg');
% set(handles.panl,'CData',background);
% handlevisibility off agar axes tidak
terlihat
% dan gambar background saja yang muncul.
set(hback,'handlevisibility','off','visible',
'off')
% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned
to the command line.
function varargout =
tampilan1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output
args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Get default command line output from
handles structure

```

```
varargout{1} = handles.output;

% -----
% Executes on button press in Penyisipan.
function Penyisipan_Callback(hObject,
 eventdata, handles)
% hObject    handle to Penyisipan (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
close(tampilan1)
GUI_Penyisipan

% -----
% Executes on button press in Ekstraksi.
function Ekstraksi_Callback(hObject,
 eventdata, handles)
% hObject    handle to Ekstraksi (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
close(tampilan1)
GUI_Ekstraksi

% --- Executes on button press in Keluar.
function Keluar_Callback(hObject, eventdata,
handles)
% hObject    handle to Keluar (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
close(tampilan1)
```

```
% -----  
-----  
function About_Callback(hObject, eventdata,  
handles)  
    % hObject    handle to About (see GCBO)  
    % eventdata   reserved - to be defined in a  
future version of MATLAB  
    % handles    structure with handles and user  
data (see GUIDATA)  
    close(tampilan1)  
Info  
  
% -----  
-----  
function Exit_Callback(hObject, eventdata,  
handles)  
    % hObject    handle to Exit (see GCBO)  
    % eventdata   reserved - to be defined in a  
future version of MATLAB  
    % handles    structure with handles and user  
data (see GUIDATA)  
close(tampilan1)
```

“Halaman ini sengaja dikosongkan”

LAMPIRAN B

Source Code Proses Penyisispan Pesan

```
function varargout = GUI_Penyisispan(varargin)
%     GUI_PENYISIPAN      MATLAB      code      for
GUI_Penyisispan.fig
    %         GUI_PENYISIPAN, by itself, creates a
new GUI_PENYISIPAN or raises the existing
    %         singleton*.
%
%         H = GUI_PENYISIPAN returns the handle
to a new GUI_PENYISIPAN or the handle to
    %         the existing singleton*.
%
%
GUI_PENYISIPAN('CALLBACK', hObject, eventData, handles,...) calls the local
    %                 function named CALLBACK in
GUI_PENYISIPAN.M with the given input arguments.
%
%         GUI_PENYISIPAN('Property','Value',...
creates a new GUI_PENYISIPAN or raises the
    %         existing singleton*. Starting from
the left, property value pairs are
    %                 applied to the GUI before
GUI_Penyisispan_OpeningFcn gets called. An
    %         unrecognized property name or invalid
value makes property application
    %                 stop. All inputs are passed to
GUI_Penyisispan_OpeningFcn via varargin.
%
%             *See GUI Options on GUIDE's Tools
menu. Choose "GUI allows only one
    %         instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```

% Edit the above text to modify the response
to help GUI_Penyisipan

% Last Modified by GUIDE v2.5 10-Jan-2018
10:09:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @GUI_Penyisipan_OpeningFcn,
...
    'gui_OutputFcn', @GUI_Penyisipan_OutputFcn,
...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_Penyisipan is
made visible.
function GUI_Penyisipan_OpeningFcn(hObject,
 eventdata, handles, varargin)
% This function has no output args, see
OutputFcn.
% hObject    handle to figure

```

```

    % eventdata reserved - to be defined in a
future version of MATLAB
    % handles structure with handles and user
data (see GUIDATA)
    % varargin command line arguments to
GUI_Penyisipan (see VARARGIN)

    % Choose default command line output for
GUI_Penyisipan
handles.output = hObject;
guidata(hObject, handles);
hback =
axes('units','normalized','position',[0 0 1 1]);
uistack(hback,'bottom'); % menciptakan axes
untuk tempat menampilkan gambar
% menampilkan background
[back map]=imread('z.jpg');
image(back)
colormap(map)
background=imread('z.jpg');
% set(handles.pan1,'CData',background);
% handlevisibility off agar axes tidak
terlihat
% dan gambar background saja yang muncul.
set(hback,'handlevisibility','off','visible',
'off')
% Update handles structure
guidata(hObject, handles);

    % UIWAIT makes GUI_Penyisipan wait for user
response (see UIRESUME)
    % uiwait(handles.figure1);

    % --- Outputs from this function are returned
to the command line.
function varargout =
GUI_Penyisipan_OutputFcn(hObject, eventdata,
handles)

```

```

% varargout cell array for returning output
args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Get default command line output from
handles structure
varargout{1} = handles.output;

% --- Executes on button press in
pushbutton1.
function pushbutton1_Callback(hObject,
 eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)
% buka citra awal
[filename,path]=uigetfile([{'*.bmp'};{'*.jpg'}
};{'*.tiff'};{'*.png'}],'open data');
if isequal(filename,0) % jika pilih cancel
maka kembali ke awal
return
end
% baca citra sesuai direktori
img=imread(fullfile(path,filename));
% tampilkan citra
axes(handles.axes1)
imshow(img)
% simpan data img dlm figure
setappdata(handles.figure1,'img',img)

```

```

function edit1_Callback(hObject, eventdata,
handles)
    % hObject    handle to edit1 (see GCBO)
    % eventdata   reserved - to be defined in a
future version of MATLAB
    % handles    structure with handles and user
data (see GUIDATA)

    % Hints:    get(hObject,'String')    returns
contents of edit1 as text
    %           str2double(get(hObject,'String'))
returns contents of edit1 as a double


% --- Executes during object creation, after
setting all properties.
function edit1_CreateFcn(hObject, eventdata,
handles)
    % hObject    handle to edit1 (see GCBO)
    % eventdata   reserved - to be defined in a
future version of MATLAB
    % handles    empty - handles not created
until after all CreateFcns called

    % Hint: edit controls usually have a white
background on Windows.
    % See ISPC and COMPUTER.
    if ispc                                         &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end


function edit2_Callback(hObject, eventdata,
handles)
    % hObject    handle to edit2 (see GCBO)

```

```

    % eventdata reserved - to be defined in a
    future version of MATLAB
    % handles structure with handles and user
    data (see GUIDATA)

        % Hints: get(hObject,'String') returns
    contents of edit2 as text
        % str2double(get(hObject,'String'))
    returns contents of edit2 as a double


        % --- Executes during object creation, after
    setting all properties.
    function edit2_CreateFcn(hObject, eventdata,
handles)
    % hObject handle to edit2 (see GCBO)
    % eventdata reserved - to be defined in a
    future version of MATLAB
    % handles empty - handles not created
    until after all CreateFcns called

        % Hint: edit controls usually have a white
    background on Windows.
        % See ISPC and COMPUTER.
        if ispc                                         &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

        % --- Executes on button press in
pushbutton2.
    function pushbutton2_Callback(hObject,
 eventdata, handles)
    % hObject handle to pushbutton2 (see GCBO)
    % eventdata reserved - to be defined in a
    future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)

img=getappdata(handles.figure1,'img');           %
citra awal
imgs=img;
c=imgs(:,:,1); % ambil data citra chanel r
(merah)
key=get(handles.edit2,'string');    % masukan
kunci
% cek jumlah kata kunci (tidak boleh lebih
dari 1)
C = strspliit(key);
if size(C,2)>1
    warndlg('tidak lebih dari 1 kata')
    return
end
pesan = get(handles.edit1,'string');    %
masukan pesan
text=[key, ' ', pesan]; % semua text yang di
sisipkan
%text = strtrim(text);
l=length(text); % panjang text yang
disisipkan
AsciiCode = uint8(text); % ubah ke 8 bit
binaryString =
transpose(dec2bin(AsciiCode,8)); % transpose dan
ubah ke biner
binaryString = binaryString(:); % ubah ke
dalam 1 array
N = length(binaryString); % jumlah nilai
biner
% ubah nilai biner string ke numerik
b = zeros(N,1); % nilai awal pembentukan
biner (string) ke numerik

```

```

for kk = 1:N
if(binaryString(kk) == '1')
    b(kk) = 1;
else
    b(kk) = 0;
end
end
% lakukan penyisipan
k=0;
for i = 1 : size(c,2) % lebar citra
for j = 1 : size(c,1) % panjang citra
    k = k + 1;
    ig=c(j,i); % baca nilai pixel citra
    baris ke i dan kolom ke j
    bg(k,:)=dec2bin(ig,8); % ubah ke biner
    if (k <= size(b,1)) % sisipkan sebanyak nilai biner (text)
        bg(:,end)=num2str(b(k)); % sisipkan dibagian akhir array
    end
    % setelah penyisipan ubah kembali ke nilai desimal (pixel)
    d=(bg(k,:));
    hg(k,:)=bin2dec(d);
end
end

s=reshape(hg,size(c)); % ubah ukuran seperti citra awal
s=uint8(s); % transpose dan ubah ke nilai 8 bit
ky=size(b,1)/8; % ukuran panjang nilai biner
imgs(:,:,1)=s; % ubah chanel r (merah) dengan nilai hasil penyisipan
imgs(end,end,1)=ky; % simpan jumlah nilai biner
imgs(end,end-1,1)=1; % simpan jumlah panjang kata

```

```

% tampilkan di axes 2
axes(handles.axes2)
imshow(imgs)
% simpan hasil penysisisipan dalam bmp
[filename,path]=uiputfile([{'*.bmp'}; {'*.jpg'}
}; {'*.tiff'}; {'*.png'}], 'save stego');
if isequal(filename,0)
return
end
if strcmp(filename(:,end-2:end), 'bmp') || %|
strcmp(filename(:,end-2:end), 'iff')
    imwrite(imgs,fullfile(path,filename));
else
    imwrite(imgs,fullfile(path,filename),
'Mode','lossless');
end
% simpan dalam data figure
setappdata(handles.figure1,'imgs',imgs)

% --- Executes on button press in
pushbutton3.
function pushbutton3_Callback(hObject,
 eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)
% hitung nilai PSNR
X=getappdata(handles.figure1,'img'); % citra
awal
Xs=getappdata(handles.figure1,'imgs'); % %
citra penyisipan

% PSNR (dB)
D = abs(X-Xs).^2;
mse = sum(D(:))/numel(X);
psnr = 10*log10(255*255/mse);

```

```

    set(handles.edit3,'string',[num2str(psnr),'  

dB']) % tampilan hasil pada edit3

    function edit3_Callback(hObject, eventdata,  

handles)  

    % hObject      handle to edit3 (see GCBO)  

    % eventdata    reserved - to be defined in a  

future version of MATLAB  

    % handles      structure with handles and user  

data (see GUIDATA)

    % Hints: get(hObject,'String') returns  

contents of edit3 as text  

    % str2double(get(hObject,'String'))  

returns contents of edit3 as a doubl

    % --- Executes during object creation, after  

setting all properties.  

    function edit3_CreateFcn(hObject, eventdata,  

handles)  

    % hObject      handle to edit3 (see GCBO)  

    % eventdata    reserved - to be defined in a  

future version of MATLAB  

    % handles      empty - handles not created  

until after all CreateFcns called

    % Hint: edit controls usually have a white  

background on Windows.  

    % See ISPC and COMPUTER.
    if ispc  

&&  

isequal(get(hObject,'BackgroundColor'),  

get(0,'defaultUicontrolBackgroundColor'))  

    set(hObject,'BackgroundColor','white');
end

```

```
% --- Executes on button press in Kembali.
function Kembali_Callback(hObject, eventdata,
handles)
% hObject    handle to Kembali (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
close(GUI_Penyisipan)
tampilan1
```

“Halaman ini sengaja dikosongkan”

LAMPIRAN C

Source Code Proses Ekstraksi Pesan

```
function varargout = GUI_Ekstraksi(varargin)
%   GUI_EKSTRAKSI      MATLAB      code      for
GUI_Ekstraksi.fig
    %           GUI_EKSTRAKSI, by itself, creates a
new GUI_EKSTRAKSI or raises the existing
    %           singleton*.
%
%
%           H = GUI_EKSTRAKSI returns the handle
to a new GUI_EKSTRAKSI or the handle to
    %           the existing singleton*.
%
%
%
GUI_EKSTRAKSI('CALLBACK', hObject, eventData, handles,...) calls the local
    %           function named CALLBACK in
GUI_EKSTRAKSI.M with the given input arguments.
%
%
%           GUI_EKSTRAKSI('Property','Value',...
creates a new GUI_EKSTRAKSI or raises the
    %           existing singleton*. Starting from
the left, property value pairs are
    %           applied to the GUI before
GUI_Ekstraksi_OpeningFcn gets called. An
    %           unrecognized property name or invalid
value makes property application
    %           stop. All inputs are passed to
GUI_Ekstraksi_OpeningFcn via varargin.
%
%
%           *See GUI Options on GUIDE's Tools
menu. Choose "GUI allows only one
    %           instance to run (singleton)".
%
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```

% Edit the above text to modify the response
to help GUI_Ekstraksi

% Last Modified by GUIDE v2.5 10-Jan-2018
10:11:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @GUI_Ekstraksi_OpeningFcn,
...
'gui_OutputFcn', @GUI_Ekstraksi_OutputFcn,
...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_Ekstraksi is
made visible.
function GUI_Ekstraksi_OpeningFcn(hObject,
 eventdata, handles, varargin)
% This function has no output args, see
OutputFcn.
% hObject    handle to figure

```

```

    % eventdata reserved - to be defined in a
future version of MATLAB
    % handles structure with handles and user
data (see GUIDATA)
    % varargin command line arguments to
GUI_Ekstraksi (see VARARGIN)

    % Choose default command line output for
GUI_Ekstraksi
handles.output = hObject;
guidata(hObject, handles);
hback =
axes('units','normalized','position',[0 0 1 1]);
uistack(hback,'bottom'); % menciptakan axes
untuk tempat menampilkan gambar
% menampilkan background
[back map]=imread('security.jpg');
image(back)
colormap(map)
background=imread('security.jpg');
% set(handles.pan1,'CData',background);
% handlevisibility off agar axes tidak
terlihat
% dan gambar background saja yang muncul.
set(hback,'handlevisibility','off','visible',
'off')
% Update handles structure
guidata(hObject, handles);

    % UIWAIT makes GUI_Ekstraksi wait for user
response (see UIRESUME)
    % uiwait(handles.figure1);

    % --- Outputs from this function are returned
to the command line.
function varargout =
GUI_Ekstraksi_OutputFcn(hObject, eventdata,
handles)

```

```
% varargout cell array for returning output
args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Get default command line output from
handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata,
handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of edit1 as text
% str2double(get(hObject,'String'))
returns contents of edit1 as a double

% --- Executes during object creation, after
setting all properties.
function edit1_CreateFcn(hObject, eventdata,
handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created
until after all CreateFcns called
```

```

% Hint: edit controls usually have a white
background on Windows.
% See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in
pushbutton1.

function pushbutton1_Callback(hObject,
 eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)
% buka citra hasil pengisian
[filename,path]=uigetfile({{'.bmp'}},{'.jpg'}
};{'.tiff'});{'.png'}],'open data stego');
if isequal(filename,0) % jika pilih cancel
maka kembali ke awal
    return
end
% baca citra sesuai direktori
imgs=imread(fullfile(path,filename));
% tampilkan citra
axes(handles.axes1)
imshow(imgs)
% simpan dalam data figure
setappdata(handles.figure1,'imgs',imgs)

% --- Executes on button press in
pushbutton2.

function pushbutton2_Callback(hObject,
 eventdata, handles)

```

```

% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)
% load citra hasil penyisipan
s=getappdata(handles.figure1,'imgs');
s=s(:,:,1); % ambil data citra pada hanya
chanel r (red)
ky=double(s(end,end,:))*8; % baca jumlah data
bit yang disisipkan
l=double(s(end,end-1,:));% baca panjang
karakter kata yang disisipkan
s=double(s); % ubah ke desimal
s=s; % lakukan transpose
s=s(:); % ubah ke dalam 1 array / vektor

% ekstraksi
for h=1:ky
    si=s(h,:);% baca pixel yang disisipkan
    db=dec2bin(si,8); % ubah ke biner
    ds(h,:)=str2num(db(:,end)); % ubah ke
numerik
end

% ubah ke biner string
for k = 1:size(ds,1);
    ib=ds(k);
    if(ib == 1)
        es(k) = '1';
    else
        es(k) = '0';
    end
end

% ubah kedalam ukuran biner awal
tx=reshape(es,[8,1]);
% ubah nilai biner ke desimal kemudian di
ubah ke text

```

```

for i=1:size(tx,2)
    thisString=char(tx(:,i))'; % baca nilai
biner string
    thisChar(:,i) = char(bin2dec((thisString))); % ubah dari biner
ke desimal kumudian ke text (string)
end

% cek key benar atau salah
C = strsplit(thisChar);
key=get(handles.edit1,'string');
if ~strcmp(C(:,1),key);
    warndlg('key salah')
    set(handles.edit2,'string',[])
return
end

% jika benar tampilkan pesan ke edit 2
Ct=[];
for j=2:size(C,2)
    Ci=cell2mat((C(:,j)));
    Ct=[Ct,' ',Ci];
end
set(handles.edit2,'string',Ct)

function edit2_Callback(hObject, eventdata,
handles)
    % hObject    handle to edit2 (see GCBO)
    % eventdata   reserved - to be defined in a
future version of MATLAB
    % handles    structure with handles and user
data (see GUIDATA)

    % Hints: get(hObject,'String')    returns
contents of edit2 as text
    %           str2double(get(hObject,'String'))
returns contents of edit2 as a double

```

```

% --- Executes during object creation, after
setting all properties.
function edit2_CreateFcn(hObject, eventdata,
handles)
    % hObject      handle to edit2 (see GCBO)
    % eventdata     reserved - to be defined in a
future version of MATLAB
    % handles       empty - handles not created
until after all CreateFcns called

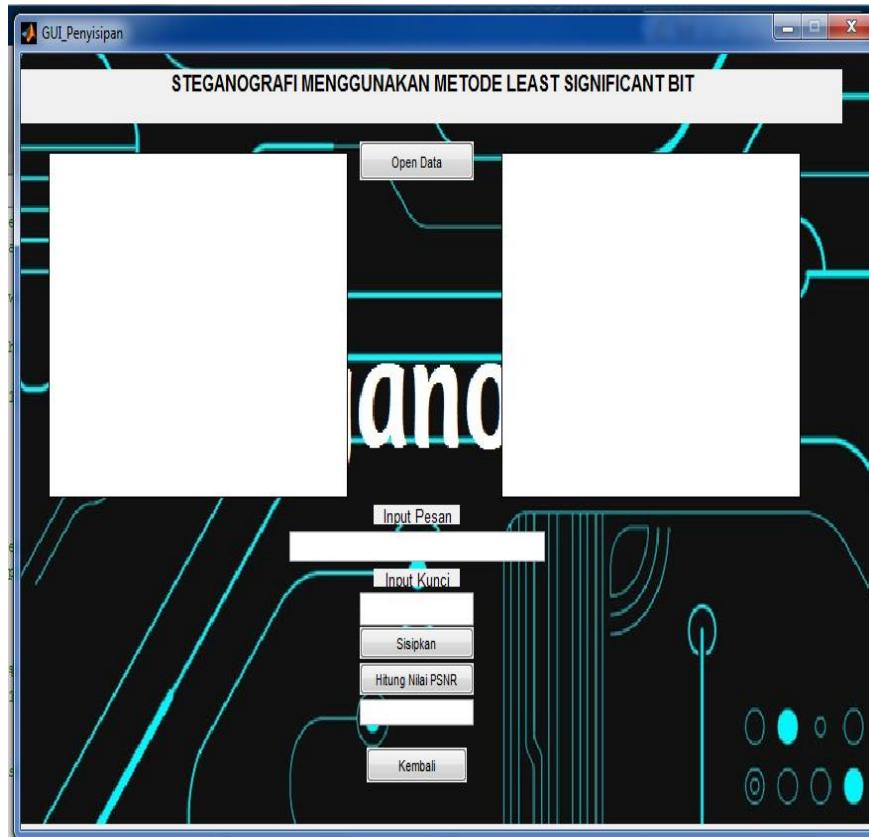
    % Hint: edit controls usually have a white
background on Windows.
    %         See ISPC and COMPUTER.
    if ispc                                         &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

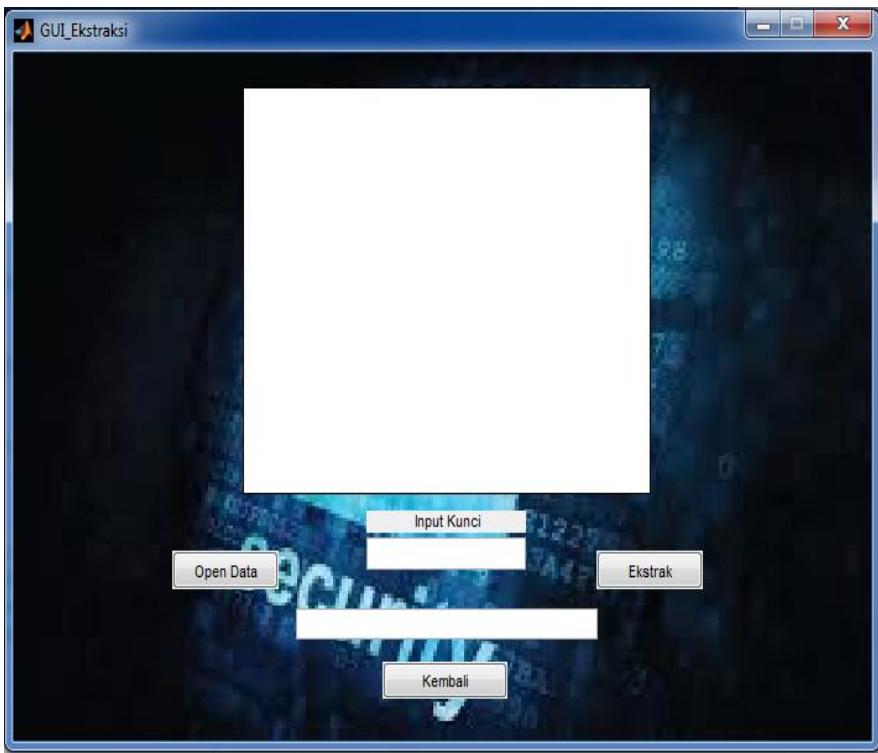
% --- Executes on button press in Kembali.
function Kembali_Callback(hObject, eventdata,
handles)
    % hObject      handle to Kembali (see GCBO)
    % eventdata     reserved - to be defined in a
future version of MATLAB
    % handles       structure with handles and user
data (see GUIDATA)
    close(GUI_Ekstraksi)
    tampilan1

```

LAMPIRAN D







BIODATA PENULIS



Muhammad Azlansyah yang mempunyai nama panggilan Alan lahir di Surabaya, 25Maret 1993. Penulis menempuh pendidikan di SD Khadijah 3 Surabaya, SMP Negeri 20 Surabaya, dan SMA KhadijahSurabaya. Penulis yang mempunyai hobi olahraga bulutangkis dan bersepeda ini diterima di Jurusan Matematika Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2011. Di Jurusan Matematika ITS ini, Penulis mengambil rumpun mata kuliah Ilmu komputer.

Apabila ingin memberikan kritik, saran, dan pertanyaan mengenai Tugas Akhir ini, bisa melalui email mike09jhoni@gmail.com .