IMPLEMENTASI METODE *IMAGE SCALING* UNTUK MENINGKATKAN KAPASITAS PENYIMPANAN INFORMASI PADA STEGANOGRAFI LSB

Skripsi

diajukan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Pendidikan Program Studi Pendidikan Teknik Informatika dan Komputer

Oleh

Bagus Pria Ambada NIM. 5302411180

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2017

PERSETUJUAN PEMBIMBING

Yang bertanda tangan di bawah ini

Nama	: Dr. Hari Wibawanto, M.T.
NIP	: 196501071991021001
Pangkat / Golongan	: IV/A
Jabatan Akademik	: Lektor Kepala
Sebagai Pembimbing	

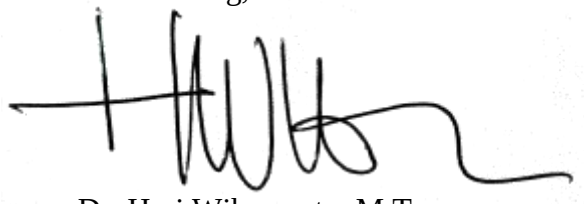
Melaporkan bahwa penyusunan Skripsi / Tugas Akhir oleh mahasiswa:

Nama	: Bagus Pria Ambada
NIM	: 5302411180
Program Studi	: Pendidikan Teknik Informatika dan Komputer
Topik	: IMPLEMENTASI METODE *IMAGE SCALING* UNTUK MENINGKATKAN KAPASITAS PENYIMPANAN INFORMASI PADA STEGANOGRAFI LSB

Telah selesai dan siap untuk diujikan

Semarang, November 2016

Pembimbing,



Dr. Hari Wibawanto, M.T.

NIP. 196501071991021001

PENGESAHAN

Skripsi dengan judul Implementasi Metode *Image Scaling* Untuk Meningkatkan Kapasitas Penyimpanan Informasi Pada Steganografi LSB telah dipertahankan didepan sidang panitia Ujian Skripsi Fakultas Teknik UNNES pada tanggal 22 November 2016.

Oleh

Nama	: Bagus Pria Ambada
NIM	: 5302411180
Program Studi	: Pendidikan Teknik Informatika dan Komputer (S1)

Panitia :

Ketua Panitia	Sekretaris
Dr.-Ing. Dhidik Prastiyanto S.T., M.T.	Ir. Ulfah Mediaty Arief M.T.
NIP. 197805312005011002	NIP. 196605051998022001

Penguji I	Penguji II	Penguji III/ Pembimbing
Dr.-Ing. Dhidik Prastiyanto S.T., M.T.	Anggraini Mulwinda S.T., M.Eng.	Dr. Hari Wibawanto, M.T.
NIP. 197805312005011002	NIP. 197812262005012002	NIP. 196501071991021001

Mengetahui :
Dekan Fakultas Teknik UNNES

Dr. Nur Qudus, M.T.
NIP. 196911301994031001

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini asli dibuat oleh penulis dan tidak terdapat campur tangan orang lain. Adapun tulisan yang terdapat dalam tulisan ini semua adalah gagasan penulis kecuali tulisan yang dicantumkan sebagai acuan dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka. Apabila di kemudian hari terbukti terdapat plagiat, maka penulis siap di beri sanksi sesuai dengan perundang-undangan.

Semarang, Januari 2017

Bagus Pria Ambada
NIM.5302411180

MOTTO & PERSEMBAHAN

"*Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya…*"
(Al Baqarah : 286)

Dengan mengucap syukur alhamdulillah kepada Allah SWT, penulis mempersembahkan skripsi ini kepada :

1. Ibu dan Bapak tercinta yang telah mendidik dan membesarkan saya serta tidak henti-hentinya mendukung dan mendoakan saya.
2. Program studi tercinta yaitu PTIK UNNES karena di tempat ini tempat penulis menimba ilmu, menambah wawasan dan meningkatkan ketrampilan.
3. teman-teman yang selalu mendukung secara langsung maupun tidak langsung.

ABSTRAK

Ambada, Bagus Pria. 2016. **"Implementasi Metode *Image Scaling* Untuk Meningkatkan Kapasitas Penyimpanan Informasi Pada Steganografi LSB".** Skripsi. Jurusan Teknik Elektro : Fakultas Teknik. Universitas Negeri Semarang.

Pembimbing : Dr. Hari Wibawanto, M.T.

Kata Kunci : *Image Scaling*, Steganografi, LSB, *Least Significant Bit*.

Steganografi merupakan seni dan ilmu menyembunyikan pesan pada media *cover.* Metode yang sering digunakan adalah LSB (*Least Significant Bit*). LSB dapat meminimalkan perubahan kualitas citra sebagai media *cover*. Penyisipan pada bit terakhir memiliki keterbatasan, yaitu media *cover* hanya dapat menampung karakter $\frac{1}{8}$ dari total ukuran media *cover.* Tujuan dari penelitian ini adalah menerapkan metode *image scaling* pada steganografi LSB untuk meningkatkan kapasitas penyimpanan pesan.

Metode pengembangan program yang dilakukan dalam penelitian ini adalah *waterfall* dengan tahapan analisis kebutuhan, pemodelan, pengkodean, pengujian. Pengujian yang dilakukan adalah uji *black box,* uji responden melalui uji HVS (*Human Visual System*), serta uji *stego image.*

Hasil dari penelitian menunjukkan aplikasi steganografi LSB (*Least Significant Bit*) dapat digunakan untuk menyisipkan pesan teks ke dalam citra digital atau biasa disebut *cover image* serta dapat meningkatkan kapasitas penyimpanan pesan. Berdasarkan pengujian kepada responden, rata-rata melihat bahwa citra digital sebelum dan sesudah proses steganografi memilki kualitas bagus dan kesamaannya antara 90-100%. Hal itu diperkuat dengan pengujian parameter nilai MSE dan PSNR. Pengujian terhadap 4 citra uji menghasilkan MSE yang rendah, yaitu 0,5 sedangkan PSNR yang didapatkan adalah 61,99 dB. Angka tersebut mengindikasikan bahwa *stego image* sebagai pembawa pesan berkualitas baik dan minim adanya *noise.* Sehingga keberadaan pesan dalam citra digital tidak mudah diketahui.

KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul **"IMPLEMENTASI METODE *IMAGE SCALING* UNTUK MENINGKATKAN KAPASITAS PENYIMPANAN INFORMASI PADA STEGANOGRAFI LSB".**

Keberhasilan penulis dalam menyelesaikan skripsi ini tidak lepas dari bimbingan dan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan rasa terima kasih kepada :

1. Rektor Universitas Negeri Semarang yang telah memberikan kesempatan untuk menyelesaikan studi di UNNES.
2. Dekan Fakultas Teknik Universitas Negeri Semarang yang telah memberi kelancaran administrasi dalam penyusunan skripsi ini.
3. Ketua dan Sekertaris Jurusan Teknik Elektro UNNES yang telah memberikan berbagai kemudahan dalam penyusunan skripsi ini.
4. Dr. Hari Wibawanto, M.T. selaku dosen Pembimbing yang telah memberikan bimbingan, bantuan, kritik dan saran, serta motivasi dalam penyusunan skripsi.
5. Bapak dan Ibu Dosen Jurusan Pendidikan Teknik Informatika dan Komputer yang telah memberikan bekal ilmu pengetahuan kepada penulis sehingga dapat menyelesaikan penyusunan skripsi ini.
6. Semua pihak yang tidak dapat disebutkan satu-persatu yang telah membantu baik secara langsung maupun tidak langsung dalam penyusunan skripsi ini.

Semarang, Januari 2017

penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi didukung adanya media internet sebagai salah satu media penyampaian informasi. Internet semakin dibutuhkan karena manusia semakin memerlukan kemudahan serta cepatnya proses pengiriman maupun penerimaan informasi. Praktek penggunaan internet tidak terlepas dari penggunaan macam-macam data digital baik itu berupa gambar, teks, suara maupun video.

Informasi yang dikirimkan melalui internet dapat berupa informasi rahasia. Sifat internet pada dasarnya adalah global, sehingga meskipun terdapat informasi yang bersifat rahasia, bukan tidak mungkin terdapat orang-orang yang sebenarnya tidak berkepentingan untuk mengetahui informasi rahasia tersebut memaksa untuk mengetahui isi dari informasi tersebut. Untuk mengatasi hal tersebut berbagai cara telah dikembangkan, diantaranya kriptografi dan steganografi (Prasetyo, 2013).

Kriptografi merupakan seni dan ilmu mengamankan pesan (Schneier, 1996). Kriptografi bekerja dengan mengubah bentuk data awal menjadi bentuk lain yang tidak terbaca (enkripsi). Seorang kriptanalis dapat memecahkannya meskipun membutuhkan waktu yang lama guna mendapatkan informasi didalamnya.

Steganografi adalah seni dan ilmu menyembunyikan pesan pada *media cover* sehingga tidak terlihat keberadaan pesan tersebut (Provos dan Honeyman,

2003:32). Pada konsepnya, steganografi merupakan penyisipan pesan ke dalam *media cover*. *Media cover* merupakan media tempat disisipkannya informasi. *Media cover* dapat berupa citra digital, suara, maupun video. Namun, dalam prakteknya, penyisipan tersebut dapat berpengaruh terhadap kualitas *media cover* dikarenakan akan mengubah derajat intensitas warna dari piksel citra yang disisipi. Untuk meminimalkan perubahan kualitas pada *media cover*, dapat dilakukan dengan cara penyisipan pada bit terakhir (*least significant bit*) *media cover*. Dengan cara tersebut, perubahan kualitas *media cover* tidak akan tampak kasat mata (Chan dan Cheng, 2003). Penyisipan pada bit terakhir memiliki keterbatasan, yaitu *media cover* hanya dapat menampung karakter $\frac{1}{8}$ dari total ukuran *media cover*.

Steganografi dengan metode LSB dapat dilakukan dengan menggunakan program Matlab versi R2015b sebagai software yang mampu menyediakan fungsi-fungsi utama yang digunakan untuk menyisipkan pesan kedalam *media cover*. Diantara fungsi-fungsi yang diperlukan adalah fungsi yang digunakan untuk mengubah data string menjadi data biner, mengubah bilangan integer menjadi bilangan biner 8 bit, mengubah string menjadi bilangan integer (decimal ASCII).

Berdasarkan latar belakang tersebut tersebut, penulis akan mengembangkan aplikasi implementasi metode *image scaling* untuk meningkatkan kapasitas penyimpanan informasi *media cover* pada steganografi LSB.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, terdapat beberapa masalah yang dapat diidentifikasi :

1. Manusia perlu keamanan informasi.
2. Keberadaan informasi rahasia perlu disamarkan.
3. Steganografi LSB memiliki keterbatasan kapasitas penyimpanan.

1.3 Pembatasan Masalah

Batasan-batasan masalah yang digunakan dalam penelitian ini adalah :

1. Aplikasi ini dibuat dengan menggunakan aplikasi Matlab versi R2015b dan hanya dapat dijalankan melalui Matlab.
2. *Media cover* yang digunakan memiliki kedalaman warna 24 bit dengan format bitmap.
3. Data rahasia yang disembunyikan berupa pesan plainteks.
4. Steganografi yang digunakan adalah teknik LSB (*Least Significant Bit*).
5. Menggunakan teknik *Image scalling* untuk memperbesar ukuran *media cover*.

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah di atas maka muncul rumusan masalah sebagai berikut :

1. Bagaimana membuat aplikasi steganografi untuk menjaga keamanan informasi?

2. Apakah aplikasi steganografi yang dikembangkan mampu menyembunyikan keberadaan pesan rahasia?
3. Apakah metode *image scaling* dapat meningkatkan kapasitas penyimpanan karakter *media cover*?

1.5 Tujuan Penelitian

1. Membuat aplikasi untuk menjaga kerahasiaan data.
2. Menyembunyikan keberadaan pesan rahasia kedalam *media cover* serta dapat membangkitkan kembali pesan yang telah disisipkan.
3. Meningkatkan kapasitas penyimpanan *media cover*.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini diantaranya :

1. Memberikan masukan, sumbangan pemikiran serta wawasan keilmuan bagi mahasiswa tentang steganografi pada citra digital utamanya untuk menyisipkan pesan kedalam citra digital.
2. Memberikan informasi mengenai metode yang dapat dikombinasikan dengan steganografi LSB guna meningkatkan efektifitas steganografi.

1.7 Sistematika Penulisan Skripsi

Untuk mempermudah dalam memahami keseluruhan isi dari penelitian skripsi, maka disusun dalam 3 bagian, yaitu:

1. Bagian Awal Skripsi, terdiri dari :

a. Halaman Judul.
b. Halaman Pengesahan.
c. Motto dan Persembahan.
d. Kata Pengantar.
e. Abstrak.
f. Daftar Isi.
g. Daftar Tabel.
h. Daftar Gambar.
i. Daftar Lampiran.

2. Bagian Isi Skripsi, terdiri dari :
   a. BAB I Pendahuluan, berisi tentang :
      1) Latar Belakang.
      2) Identifikasi Masalah.
      3) Pembatasan Masalah.
      4) Perumusan Masalah.
      5) Tujuan Penelitian.
      6) Manfaat Penelitian.
      7) Sistematika Penulisan Skripsi.
   b. BAB II Landasan Teori.
   c. BAB III Metode Penelitian.
   d. BAB IV Hasil Penelitian dan Pembahasan.
   e. BAB V Simpulan dan Saran.
3. Bagian Akhir Skripsi, terdiri dari :

a. Daftar Pustaka.

b. Lampiran.

BAB II

LANDASAN TEORI

2.1 Deskripsi Teoritik

2.1.1 Steganografi

Steganografi berasal dari bahasa Yunani, yang berarti tulisan yang tersamarkan (*covered letter*). Selain itu dapat juga diambil pengertian sebagai komunikasi yang dilakukan dengan cara menyembunyikan pesan. Pesan yang dikirim disembunyikan/disamarkan ke media lain yang biasa disebut "*media cover"*(Krisnawati,2008:39). Media yang sering digunakan untuk menyembunyikan pesan diantaranya adalah : citra, suara, dan video. Pada dasarnya, steganografi tidak mengubah bentuk pesan yang dikirimkan melainkan disembunyikan/disamarkan dalam *media cover* yang membawa pesan tersebut.

Secara umum, steganografi membutuhkan dua syarat utama, yaitu *media cover* dan pesan rahasia (Rakhmat dan Fairuzabadi,2010:5-6). Proses penyembunyian pesan ke dalam media disebut penyisipan (*embedding*), sedangkan proses sebaliknya, yaitu pengambilan pesan dari media disebut ekstraksi (Prasetyo, 2013:8).

2.1.1.1 Sejarah Steganografi

Sejarah steganografi diawali sekitar 404 SM dengan contoh yang paling terkenal adalah "*Histories of Herodotus"*. Mengisahkan Histiæus mencukur kepala budak terpercayanya dan mentatonya dengan pesan yang hanya akan

terlihat jika rambut palsu yang dipakainya terjatuh. Hal itu dilakukan untuk penyelidikan pemberontakan di Persia. Metode ini masih digunakan oleh mata-mata Jerman pada awal abad ke-20. Herodotus juga mengisahkan bagaimana Demeratus memperingatkan Yunani tentang datangnya Xerxes, raja Persia. Dalam ceritanya disebutkan bahwa pesan peringatan tersebut dituliskan di sebuah tablet dan dilumuri dengan lilin. Secara kasat mata, tablet tersebut terlihat seperti tablet biasa yang tidak ada pesan apapun tertulis di atasnya. Cara yang terakhir digunakan adalah dengan menggunakan tinta tak terlihat untuk menuliskan pesan, cara ini digunakan oleh mata-mata Jerman dalam perang dunia (Katzenbeisser dan Petitcolas, 2002:3).

Sekitar 4000 tahun yang lalu, di Kota Memet Khufu, Mesir. Ide membuat pesan rahasia pertama dimulai dengan digunakannya *hieroglyphic* yang berarti menulis dengan menggunakan karakter-karakter berbentuk gambar. Hal ini dilakukan bangsa mesir kuno dahulu untuk menceritakan kehidupan majikannya. Hal tersebut dianggap sebagai steganografi pertama (Ariyus, 2009).

2.1.1.2 Kriteria Steganografi

Kriteria yang harus diperhatikan dalam steganografi adalah sebagai berikut :

1. *Fidelity*. Tidak banyak mengubah kualitas media penampung. Setelah penyisipan pesan rahasia, media hasil steganografi masih terlihat dengan baik. Pesan yang disamarkan ke dalam media penampung tidak mudah terlihat dengan kasat mata.

2. *Robustness.* Pesan yang disembunyikan harus tahan terhadap manipulasi yang dilakukan terhadap media penampung.
3. *Recovery.* Pesan yang disembunyikan harus dapat diekstrak kembali (*recovery*). Karena tujuan steganografi adalah menyembunyikan pesan, maka sewaktu-waktu pesan tersebut harus dapat diambil kembali dari media penampungnya (Munir, 2004:4-5).

2.1.1.3 Teknik Steganografi

Menurut Ariyus (2009), terdapat tujuh teknik dasar yang digunakan dalam steganografi, yaitu :

a. *Injection,* Merupakan teknik yang digunakan dengan cara menysisipkan secara langsung pesan rahasia ke suatu media. Kekurangan dari teknik ini adalah ukuran media setelah disisipi akan menjadi lebih besar dari ukuran sebelumnya, sehingga mudah terdeteksi. Teknik ini sering disebut teknik *embedding.*
b. Substitusi , data normal digantikan dengan data rahasia. Hasil akhir dari teknik ini tidak banyak mengubah ukuran media asli. Namun, teknik ini dapat menurunkan kualitas media yang disisipi.
c. *Transform Domain,* merupakan teknik yang sangat efektif. Pada dasarnya, transformasi domain disembunyikan pada *transform space*.
d. *Spread Spectrum,* sebuah teknik pengtransmisian menggunakan *pseudonoise code,* yang independen terhadap data informasi sebagai

modulator bentuk gelombang untuk menyebarkan energi sinyal dalam sebuah jalur komunikasi (*bandwidth*) yang lebih besar daripada sinyal jalur komunikasi informasi. Oleh penerima, sinyal dikumpulkan kembali menggunakan replika *pseudo-noise code* tersinkronisasi.

e. *Statistical Method*, teknik ini disebut juga skema *steganographic* 1 bit. Skema tersebut menanamkan satu bit informasi pada media tumpangan dan mengubah statistik walaupun hanya 1 bit. Perubahan statistik ditunjukkan dengan indikasi 1 dan jika tidak ada perubahan, terlihat indikasi 0. Sistem ini bekerja berdasarkan kemampuan penerima dalam membedakan antara informasi yang dimodifikasi dan yang belum.

f. *Distortion*, metode ini menimbulkan perubahan atas benda yang ditumpangi oleh data rahasia.

g. *Cover Generation*, metode ini lebih unik daripada metode lainnya karena *cover object* dipilih untuk menyembunyikan pesan. Contoh dari metode ini adalah *Spam Mimic*.

2.1.1.4 Kegunaan Steganografi

Menurut Prasetyo (2013) kegunaan utama dari steganografi adalah menyembunyikan keberadaan pesan. Beberapa penerapan penggunaan steganografi diantaranya adalah :

1. *Tamper-proofing* steganografi digunakan sebagai alat untuk mengidentifikasi dan menunjukkan data *host* telah mengalami perubahan dari aslinya.
2. *Feature location* steganografi digunakan sebagai alat untuk mengidentifikasi isi dari data digital pada lokasi-lokasi tertentu, seperti contohnya penamaan objek tertentu dari beberapa objek yang lain pada suatu citra digital.
3. *Annotation*/*caption* steganografi digunakan sebagai keterangan tentang data digital itu sendiri. Contohnya *geotagging*, atau *properties* pada suatu citra digital.
4. *Copyright-Labeling* steganografi digunakan untuk menyembunyikan label hak cipta pada data digital sebagai bukti otentik kepemilikan karya digital tersebut.

2.1.2 LSB (*Least Significant Bit*)

Terdapat banyak teknik menyembunyikan pesan ke dalam *media cover*. Salah satu teknik yang sederhana yaitu LSB (*Least Significant Bit*) (Chan dan Cheng, 2003:6). Teknik ini pada dasarnya adalah metode steganografi dengan memanipulasi bit-bit terakhir pada *byte* sebuah file citra digital. LSB akan lebih efektif jika menggunakan citra digital dengan tipe *lossless compression,* hal ini dikarenakan pada tipe tersebut perbedaan bit pada *byte* citra digital tidak terlihat jelas. Sehingga citra hasil steganografi akan lebih tahan jika dilakukan kompresi. Berbeda ketika menggunakan citra digital dengan tipe *lossy compression,* jika

dilakukan kompresi maka pesan yang tersimpan dalam citra akan rusak atau hilang.

Secara umum, dalam teknik ini dikenal dua istilah yang digunakan untuk citra digital, yaitu :

a. *Cover image* artinya citra digital yang digunakan untuk menyimpan pesan rahasia atau disebut juga *media cover*.
b. *Stego image* artinya citra digital yang dihasilkan setelah proses steganografi/penyisipan pesan.

Dalam proses LSB terdapat dua jenis metode yaitu *LSB Replacement* dan *LSB Matching. LSB Replacement* dilakukan dengan cara mengganti bit-bit terakhir pada *byte-byte* citra digital, dengan bit-bit dari pesan yang akan disisipkan. Sedangkan *LSB Matching* bekerja dengan cara membandingkan setiap bit dari pesan yang akan disisipkan dengan bit terakhir dari *byte* citra digital yang bersesuaian. Jika bit tersebut cocok (bernilai sama) maka tidak akan diubah nilainya, jika tidak cocok (bernilai berbeda) maka nilai bit terakhir dari *byte* tersebut akan diubah nilainya (Syakura,2010:2).

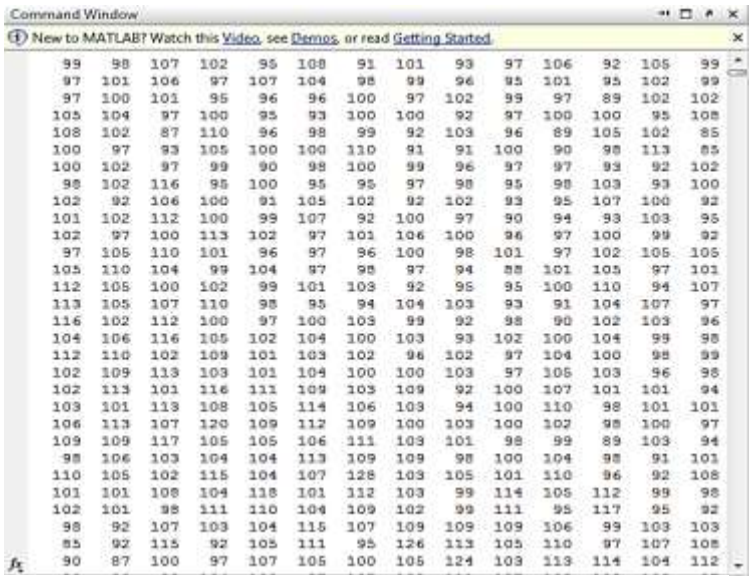
Pada penelitian ini menggunakan LSB dengan metode *LSB Replacement* karena tekniknya yang lebih sederhana jika dibandingkan dengan *matching.* Sebagai ilustrasi awal, teknik *LSB Replacement* dapat dijelaskan sebagai berikut :

Gambar 2.1 Lenna (*Sumber :*

http://sipi.usc.edu/database/database.php?volume=misc)

Pada berkas bitmap 24 bit, setiap pixel (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111 sebagai contoh gambar 2.1. Dengan demikian, pada setiap pixel berkas bitmap 24 bit kita dapat menyisipkan 3 bit data.

Berikut ini adalah potongan nilai matriks dari citra Lena di atas.



Gambar 2.2 Potongan nilai matriks dari citra Lenna

Misalkan diambil satu piksel dari citra tersebut, maka didapatkan nilai RGB sebagai berikut : [99,98,107] nilai tersebut adalah nilai yang mewakili warna satu piksel pada citra. Hasil konversi nilai desimal RGB ke dalam biner maka didapatkan :

99 = 0110001**1**

98 = 0110001**0**

107 = 0110101**1**

Nilai biner yang ditulis merah tersebut yang dinamakan LSB pada bilangan biner. Misalkan biner data yang akan disisipkan ke dalam gambar adalah ‘ 010 ‘ maka :

1. Nilai biner sebelum penyisipan

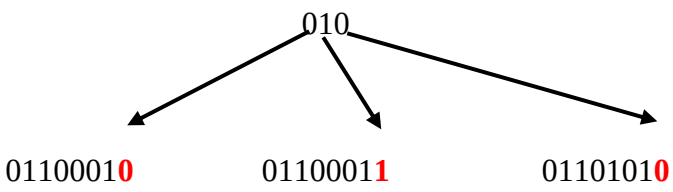
   0110001**1** 0110001**0** 0110101**1**

2. Nilai biner yang akan disisipkan

   010

3. Proses *replacement*

   0110001**1** 0110001**0** 0110101**1**

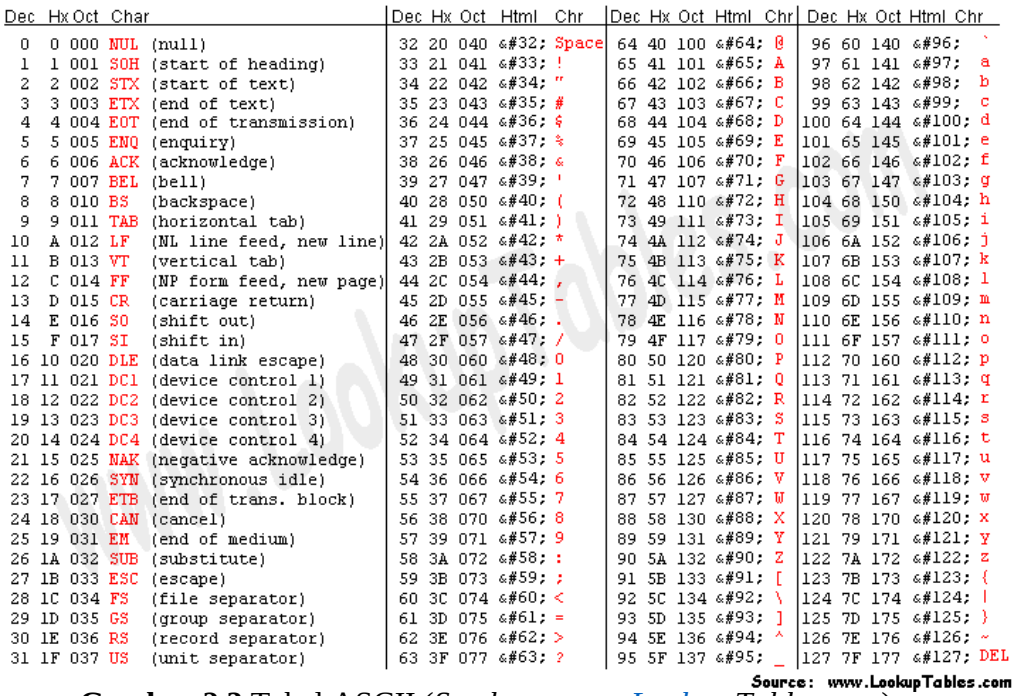
   

Hasil akhir nilai RGB piksel dari citra setelah dilakukan penyisipan data adalah : [0110001**1** 0110001**0** 0110101**1**].

Ukuran data yang akan disisipkan bergantung pada ukuran citra penampung. Pada citra 24-bit yang berukuran 256x256 piksel terdapat 65536 piksel, setiap piksel berukuran 3 *byte* (komponen *RGB*), berarti seluruhnya ada 65536 x 3 = 196608 *byte*. Karena setiap *byte* hanya dapat menyembunyikan satu bit di *LSB*, maka ukuran data yang akan disembunyikan dalam *media cover* maksimum adalah : $\frac{196608}{8}$ = 2457*byte*.

2.1.3 ASCII

ASCII (*American Standard Code for Information Interchange*) adalah salah satu standar internasional utamanya untuk pertukaran informasi. ASCII memiliki komposisi bilangan biner 8 bit dengan nilai terkecil adalah 0000 0000 sampai dengan 1111 1111. Contoh tabel ASCII ditunjukkan gambar 2.3:



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Gambar 2.3 Tabel ASCII (*Sumber : www.Lookup Tables.com*)

2.1.4 Citra Digital

Citra digital dapat mewakili sebuah matiks yang tersusun dari M kolom dan N baris, perpotongan antara kolom dan baris tersebut disebut piksel yang merupakan elemen terkecil dari sebuah citra. Piksel memiliki dua parameter lain yaitu koordinat piksel dan warna piksel. Warna dari piksel sering ditulis f(x,y) yang merupakan nilai dari piksel. Sebuah citra digital dapat ditulis dalam bentuk matriks seperti berikut.

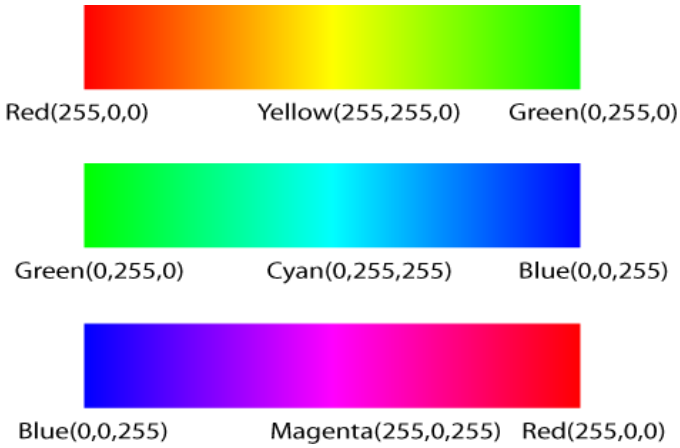
$$f=(x,y)\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & \cdots & \cdots & f(1,M-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Gambar 2.4 Rumus Matriks

Dari gambar rumus gambar 2.4 secara sistematis citra digital dapat dituliskan sebagai fungsi intensitas f (x,y), harga x (baris matriks) dan y (kolom matriks) merupakan koordinat posisi dan f(x,y) adalah nilai fungsi pada setiap titik (x,y) yang menyatakan intensitas citra atau warna dari piksel di titik tersebut. Setiap warna merupakan kombinasi dari tiga warna dasar yaitu merah, hijau, biru (*Red, Green, Blue –RGB*).

RGB merupakan suatu model pewarnaan yang terdiri dari tiga warna dasar, merah, hijau, dan biru. Setiap warna dasar memiliki rentang nilai. Berdasarkan pada 8 digit bilangan biner pada perangkat komputer maka dipilih skala 256 seperti ditunjukkan gambar 2.5, sehingga rentang nilai terkecil adalah 0 dan paling besar adalah 255 pada setiap warna dasar penyusunnya. Dengan nilai ini, maka akan diperoleh susunan warna sejumlah 256 x 256 x 256 = 1677726

jenis warna. Sebuah jenis warna mewakili sebuah vektor ruang 3 dimensi dengan koordinat yang menyatakan tiga nilai penyusunnya, yaitu komponen-*x*, komponen-*y*, dan komponen-*z*. sebagai ilustrasi, f(x,y,z) merupakan vektor warna setiap komponennya diwakili Red, Green, Blue. Sehingga sebuah warna dapat dituliskan RGB(45,60,235). Untuk nilai terkecil RGB(0,0,0) mewakili warna hitam, sedangkan nilai terbesar RGB(255,255,255) mewakili warna putih.



Gambar 2.5 Pewarnaan dalam RGB

2.1.5 Metode Penyisipan

Langkah-langkah metode penyisipan steganografi LSB :

1. Mengubah setiap piksel *media cover* menjadi raster data matriks desimal.

Untuk membaca piksel dari citra digital, Matlab menyediakan fungsi khusus, yaitu **imread();** untuk memanggil fungsi tersebut dapat menggunakan :

```
I=imread([direktori, namafile]);
```

Perintah untuk membaca file *media cover* adalah sebagai berikut :

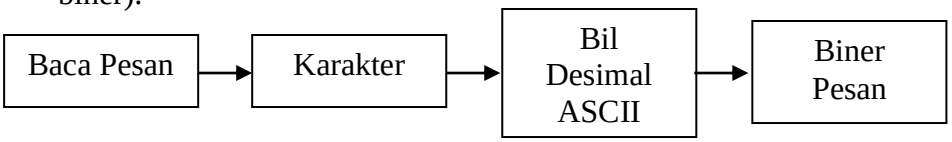
```
[namafie,direktori]=uigetfile({'*.bmp},'Buka
Gambar');

I = imread(namafile);
```

Setelah didapat nilai RGB dari setiap piksel, maka nilai tersebut diubah menjadi biner 8 bit menggunakan fungsi **dec2bin(***parameter*,8). Untuk memanggil fungsi tersebut dapat menggunakan :

```
dec2bin(parameter, 8);
```

2. Mengubah pesan (bertipe karakter) menjadi bit-bit pesan (bilangan biner).



Untuk membaca pesan yang akan disisipkan ke dalam *media cover*, menggunakan fungsi **fopen** yang disediakan oleh Matlab. Secara detail, **fopen** digunakan untuk membuka suatu file yang nantinya dapat digunakan untuk membaca, menulis, atau menyisipkan teks. Untuk membaca pesan, ditambahkan **' r '** pada fungsi pemanggilnya.

```
fopen(filename, 'r');
```

Perintah yang digunakan untuk membaca file pesan adalah sebagai berikut :

```
[namafile, direktori]=uigetfile('*.txt','Buka
Teks');
teks = fopen(namafile, 'r');
charteks = fread(teks, 'uint8=>char');
fclose(teks);
pesan = fprintf(charteks);
```

Variabel ‘teks’ merupakan keluaran fungsi ‘fopen’ yang berupa data biner. Agar dapat dimengerti oleh pengguna, data biner diubah ke dalam data karakter. Caranya adalah dengan menggunakan fungsi **fread(teks, ‘uint=>char’).** Variabel pesan adalah keluaran fungsi ‘fread’ berupa matriks berisi karakter. Sedangkan untuk mengubah data karakter menjadi format string menggunakan fungsi **sprintf.**

Setelah dapat membaca karakter dalam file pesan, selanjutnya adalah mengubah pesan manjadi raster data 8 bit biner dengan menggunakan perintah berikut ini :

```
function binteks=str2bin (teks)
%fungsi untuk mengubah string menjadi biner
ASCII

desteks=uint8(teks);
binteks=dec2bin(desteks,8);
```

Str2bin bukan merupakan fungsi bawaan dari Matlab, melainkan fungsi yang dibuat manual oleh pengguna. **Uint8( )** merupakan fungsi bawaan Matlab yang berfungsi untuk mengubah string menjadi bilangan integer (desimal ASCII). Sedangkan **dec2bin (desteks,8)** digunakan untuk mengubah bilangan integer ASCII yang didapat menjadi bilangan biner 8 bit.

3. Proses utama adalah mengganti setiap bit rendah (LSB) dengan bit pesan.

   Beberapa langkah yang dilakukan dalam mengganti LSB citra digital dengan bit pesan adalah sebagai berikut :

Mengidentifikasi banyaknya baris teksbin. Teksbin disini merupakan biner dari pesan yang akan disisipkan.

```
tbar=size(teksbin,1);
```

Setelah mengetahui banyaknya baris biner pesan, selanjutnya mentranspose biner tersebut (a[tbar,8] → a[tbar*8]) menggunakan perintah berikut:

```
a=[ ];
for i=1:tbar;
    a=[a teksbin (I,:)];

end
```

Selanjutnya, mengidentifikasi baris citra digital sebagai *media cover* disesuaikan dengan jumlah pesan yang akan disisipkan.

```
Xgbr=size(gbr1,1);
```

Variabel ‘gbr1’ merupakan variabel yang digunakan untuk membaca citra digital.

Untuk menyisipkan biner pesan dalam LSB *media cover* digunakan aturan berikut ini:

Jika LSB = bit pesan, maka raster data tidak berubah.

Jika LSB < bit pesan, maka raster data ditambah 1.

```
If mod (gbr1(i,j,1),2) < str2num(a(n))
       gbr2 (i,j,1)=gbr1(i,j,1)+1;
```

*Variabel i dan j digunakan untuk menunjuk posisi matriks yang digunakan

Jika LSB > bit pesan, maka raster data dikurang 1.

```
If mod (gbr1(i,j,1),2) > str2num(a(n))
       gbr2 (i,j,1)=gbr1(i,j,1)-1;
```

Proses penggantian tersebut terus diulang ke baris matriks selanjutnya pada citra digital sampai seluruh biner pesan yang akan disisipkan habis. Perintah lengkapnya sebagai berikut :

```
function gbr2=stegolsb(gbr1,teksbin)
gbr2=gbr1;
tbar=size(teksbin,1);


a=[];
for i=1:tbar
    a=[a teksbin(i,:)];

end

xgbr=size(gbr1,1);
n=1; i=1; j=1;
while n <=length(a)
    if mod(gbr1(i,j,1),2)<str2num(a(n))
        gbr2(i,j,1)=gbr1(i,j,1)+1;
    elseif mod (gbr1(i,j,1),2)>str2num(a(n))
        gbr2(i,j,1)=gbr1(i,j,1)-1;
    end
    i=i+1;
    if mod(n,xgbr)==0
        j=j+1;i=1;
    end
    n=n+1;
end
```

4. Menulis piksel yang baru sesuai dengan raster data.

Contoh kasus tabel 2.1 misalnya kita menyisipkan teks huruf " h ". Dengan proses diatas, maka akan didapat hasil sebagai berikut :

Tabel 2.1 contoh hasil raster biner steganografi LSB

piksel	RGB Media cover	Raster Data	Bit Rendah (LSB)	Bit Pesan (huruf 'h')	Hasil	RGB Stego image
[1,1]	46	00101110	0	0	00101110	46
[2,1]	39	00100111	1	1	00100111	39
[3,1]	36	00100100	0	1	00100101	37
[4,1]	29	00011101	1	0	00011100	28
[5,1]	21	00010101	1	1	00010101	21
[6,1]	30	00011110	0	0	00011110	30
[7,1]	46	00101110	0	0	00101110	46
[8,1]	47	00101111	1	0	00101110	46

2.1.6 Metode Ekstraksi

Langkah-langkah yang digunakan untuk mengambil kembali pesan yang disisipkan dari *stego image.*

1. Setiap piksel *stego image* diubah menjadi raster data agar memperoleh bit rendah (LSB). Prosesnya sama dengan tahap penyisipan.
2. LSB yang didapat dari setiap piksel dikumpulkan hingga terbentuk bit stream. Arah baca dari peksel adalah dari atas ke bawah serta dari

kiri ke kanan. Setiap 8 bit stream merepresentasikan sebuah karakter. Setelah semua bit stream diubah menjadi karakter, kita akan memperoleh pesan tersembunyi diantara piksel *stego image.*

Perintah yang digunakan adalah sebagai berikut :

```
function teksbin=ekstraklsb(gbr)

red=gbr(:,:,1);
redbin=dec2bin(red);
teks=(redbin(:,8))';
%mengambil bit ke-8, lalu ditransposes

n=(length(teks))/8';
%n
a=[];
i=0; Q=0;

while i<n & Q<4
    j=(8*i)+1;
    k=teks(j:j+7);
    if k=='01010001' %ASCII huruf Q
        Q=Q+1;
    else
        Q=0;
    end
    a=[a; k];
    i=i+1;
end

temp=size(a,1);
a(temp-3:temp,:)=[]; %menghilangkan batas Q
'QQQQ'
teksbin=(bin2str(a))';
```

Dalam penyisipan yang dilakukan, dalam setiap pesan yang disisipkan diakhiri dengan menyisisipkan karakter ‘QQQQ’. Hal itu digunakan untuk memberi batas pesan yang disisipkan sehingga tidak terjadi *over stream* ketika dilakukan proses ekstraksi pesan. Dengan asumsi, ‘QQQQ’ bukan merupakan karakter yang wajar/kata yang wajar dalam sebuah kalimat/pesan.

Matlab menyediakan fungsi **bin2str( )** yang digunakan untuk mengubah bilangan biner ASCII menjadi string. Algoritmanya adalah mengubah bilangan biner menjadi bilangan desimal (menggunakan fungsi **bin2dec**). Kemudian, bilangan desimal diubah menjadi string (menggunakan fungsi **char**). Perintah memanggil fungsinya adalah sebagai berikut :

```
Desteks=char(bin2dec(binteks));
```

'Desteks' adalah variabel yang digunakan untuk merepresentasikan nilai desimal dari karakter. Sedangkan 'binteks' merupakan variabel nilai biner dari pesan yang disisipkan.

Jika diterapkan berdasarkan contoh kasus sebelumnya, maka hasil ekstraksi *stego image* ditunjukkan oleh tabel 2.2 berikut :

Tabel 2.2 Tabel hasil ekstrak biner

Piksel	Gambar Stego RGB	Raster data	LSB	Bit Stream	Hasil
[1,1]	46	00101110	0	01101000	h
[2,1]	39	00100111	1		
[3,1]	37	00100101	1		
[4,1]	28	00011100	0		
[5,1]	21	00010101	1		
[6,1]	30	00011110	0		
[7,1]	46	00101110	0		

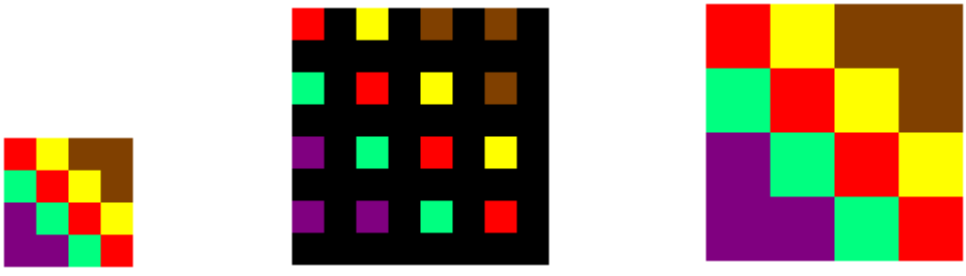
[8,1]	46	00101110	0		

2.1.7 Metode *Image Scaling*

Image scaling dalam ilmu pengolahan citra digital dikenal dengan nama *resampling. Resampling* merupakan suatu teknik matematis yang digunakan untuk menghasilkan citra baru dari citra sebelumnya dengan ukuran piksel yang berbeda atau sering disebut interpolasi. Menambah ukuran piksel dari citra sebelumnya disebut *upsampling*, sedangkan mengurangi ukuran pikselnya disebut *downsampling (*Sachs,Jonathan 2001:1*).*

Ketika ukuran suatu citra bertambah, baik itu lebar maupun tingginya maka jumlah piksel penyusunnya juga akan bertambah dengan tetap mengacu pada citra aslinya. Meskipun demikian, citra setelah mengalami *resampling* akan berkurang ketajamannya dari citra awal karena jumlah informasi per piksel juga berkurang.

Pada penelitian ini metode *upsampling* yang digunakan adalah *Nearest-neighbor*. karena berfokus dalam meningkatkan kapasitas penyimpanan steganografi yang mengacu pada bertambahnya piksel dari suatu citra melalui proses tersebut. Metode *Nearest-neighbor* adalah metode *upsampling* dengan cara menduplikasi piksel sesuai dengan perbesaran yang dilakukan. Sebagai ilustrasi dari perbesaran yang dilakukan dapat di jelaskan melalui gambar 2.6 berikut :



Gambar 2.6 *Nearest-neighbor*

Gambar A menunjukkan citra digital dengan piksel awal 4x4.

Gambar B menunjukkan piksel hasil perbesaran 2 kali dari ukuran piksel awal. Ukurannya menjadi 8x8 piksel. Piksel yang ditunjukkan dengan bidang warna hitam adalah piksel tambahan yang akan diisi informasi warna yang bersesuaian dengan piksel sebelum pembesaran.

Gambar C menunjukkan hasil akhir citra digital setelah pengisian informasi warna dari piksel yang bersesuaian.

Matlab menyediakan fungsi untuk melakukan proses tersebut. Melalui fungsi **Imresize(***variabel***,***parameter skala***,'nearest')** kitadapat melakukan perbesaran sebuah citra digital sesua skala yang kita inginkan. Dalam hal ini, langkah yang dilakukan program dalam memperbesar citra digital adalah :

1. Program membaca citra masukan sebagai citra awal sebelum dilakukan perbesaran.
2. Mendeteksi ukuran citra awal.
3. Mengambil nilai skala yang diinginkan.
4. Membaca nilai RGB disetiap piksel citra awal.
5. Membuat piksel baru sesuai dengan nilai perbesaran di setiap piksel citra awal.
6. Mengindeks nilai RGB pada piksel baru sesuai dengan nilai RGB dari piksel citra awal yang bersesuaian (model *Nearest-neighbor*).
7. Merepresentasikan nilai RGB baru menjadi citra digital baru yang sudah mengalami perbesaran dari citra digital awal.

Citra baru dari hasil pembesaran tidak akan mengalami perubahan intensitas derajat warna karena piksel baru memiliki nilai RGB yang sama dengan piksel terdekat yang bersuaian. Meskipun demikian, perbesaran tetap akan terdapat *jaggies* pada citra baru (Daryanto, 2016).

2.2 Kajian Penelitian yang Relevan

Penelitian yang dilakukan oleh Alson, dkk (2012) yang berjudul "*Implementasi Steganografi untuk Penyembunyian Pesan pada Video dengan Metode LSB"* penelitian tersebut mengembangkan aplikasi yang digunakan untuk menyembunyikan pesan ke dalam video sebagai pembawa pesan tersebut. Ukuran pesan yang dapat disimpan 4,1% dari ukuran file video. Sebagai tambahan keamanan file pesan yang disimpan, sebelum pesan disisipkan ke dalam video pesan dienkripsi dengan menggunakan algoritma *Blowfish* dengan kunci yang ditentukan user. Sebagai keluaran dari aplikasi ini adalah video dengan pesan yang tersembunyi di dalamnya. Tidak terjadi perubahan kualitas yang signifikan dari video tersebut dan PSNR dari video menunjukkan nilai diatas 30 dB.

Tujuan dari penelitian tersebut adalah mengimplementasikan steganografi dengan teknik LSB untuk menyisipkan teks ke dalam video. Untuk memperkuat keamanan, digunakan algoritma tambahan untuk enkripsi teks yang akan di sisipkan dan selanjutnya teks tersebut dapat diekstrak kembali dan didekripsi untuk mendapatkan teks asli.

Dari penelitian di atas dapat diketahui bahwa penyisipan teks ke dalam media dapat dilakukan dengan teknik LSB dan dapat dikombinasikan dengan

algoritma lain untuk memperkuat keamanan data. Media yang digunakan tidak hanya citra digital, akan tetapi dapat pula menggunakan video.

Penelitian yang dilakukan oleh Andrie Gunawan dan Muhammad Efi Fahlefi (2010) yang berjudul *"Pengembangan Aplikasi Steganografi Untuk Penyisipan Berkas Teks Ke Dalam Berkas Suara"* penelitian tersebut mengembangkan aplikasi steganografi dengan memanfaatkan berkas suara untuk menyisipkan data teks yang bersifat rahasia. Metode yang digunakan pada penelitian ini adalah metode *Least Significant Bit (LSB)*. Sampel yang akan digunakan untuk uji coba penelitian ini berupa data teks dengan kapasitas yang berbeda dan berkas suara dengan format wav.

Hasil akhir yang dicapai pada penelitian ini adalah membuat suatu aplikasi yang dapat menyisipkan data teks ke dalam berkas suara tanpa merusak kualitas dari berkas suara tersebut karena tidak ada perubahan yang sangat signifikan pada berkas suara tersebut. Data teks yang telah disisipkan tersebut juga dapat diekstraksi kembali.

Penelitian oleh Erdiansyah Fajar Nugraha (2011) yang berjudul *"Meningkatkan Kapasistas Pesan yang disisipkan dengan Metode Redundant Pattern Encoding"* dalam penelitian tersebut steganografi dilakukan dengan menggunakan metode *Redundant Pattern Encoding* dan menggunakan citra digital sebagai wadah pembawa teks yang disembunyikan. Kelebihan dari metode ini adalah tahan terhadap *cropping* dan pemampatan pada saat *media cover* dilakukan pemrosesan. Penyisipan teks pada metode *redundant pattern encoding* dilakukan dengan menyisipkan teks pada *noise* maupun bagian yang kurang

diperhatikan atau yang tidak terlihat secara visual pada file, contohnya *header image*. Dikarenakan tempat penyimpanan yang terbatas, maka kapasitas pesan yang disimpan juga terbatas.

Dalam penelitian tersebut, bertujuan untuk meningkatkan kapasitas teks yang disisipkan. Terdapat tiga cara yang dibahas dalam meningkatkan kapasitas pesan, yaitu menggunakan banyak file yang saling berhubungan, misalnya album foto atau kumpulan foto yang memiliki kategori yang sama, menggunakan file yang memiliki banyak *noise*, dan mengkombinasikan dua cara sebelumnya.

Selanjutnya adalah penelitian oleh Agus Prihanto, dkk (2010) yang berjudul *"Peningkatan Kapasitas Informasi Tersembunyi pada Image Steganografi dengan Menggunakan Teknik Hybrid"* dalam penelitian tersebut steganografi dilakukan dengan mengkombinasikan beberapa metode untuk meningkatkan kapasitas informasi yang dapat ditampung. Metode yang digunakan adalah multi LSB dan pemampatan. Selain itu, digunakan pula teknik *scaling* pada *media cover* Untuk meningkatkan keamanan informasi yang disimpan, dikombinasikan dengan algoritma enkripsi dan kriptografi *Triple* DES.

Tujuan dari penelitian tersebut adalah meningkatkan daya tampung informasi steganografi pada citra digital yang disisipkan melalui metode multi LSB. Untuk meningkatkan piksel dari citra digital yang digunakan maka dilakukan *scaling* dengan perbesaran tertentu. Sedangkan untuk meningkatkan keamanan informasi yang akan disipkan, dilakukan enkripsi dengan algoritma DES sebanyak 3 kali sebelum informasi itu disisipkan ke dalam media steganografi.

Dari penelitian tersebut didapatkan bahwa teknik *hybrid* yang digunakan meningkatkan kapasitas informasi bahkan secara teori dapat tak terhingga, namun disarankan untuk pembesaran image tidak terlalu tinggi. Untuk mengurangi efek mozaik setelah citra digital mengalami pembesaran, dapat dilakukan proses *cubic kernel, lanczos kernel* dan *spline kernel.*

Penelitian oleh Muhammad Hakim (2012) dengan judul " *Studi dan Implementasi Steganografi Metode LSB dengan Preprocessing Kompresi data dan Ekspansi Wadah "* salah satu kelemahan dari metode LSB adalah hanya dapat menyimpan seperdelapan dari ukuran pixel citra digital sebagai *media cover.* Dalam penelitian ini dilakukan optimalisasi penyimpanan pada metode LSB. Metode yang dilakukan adalah dengan melakukan *preprocessing* pada berkas data sebelum disisipkan yaitu dengan jalan memampatkan (*compression).* Selain itu juga dilakukan dengan ekspansi wadah atau media *cover.*

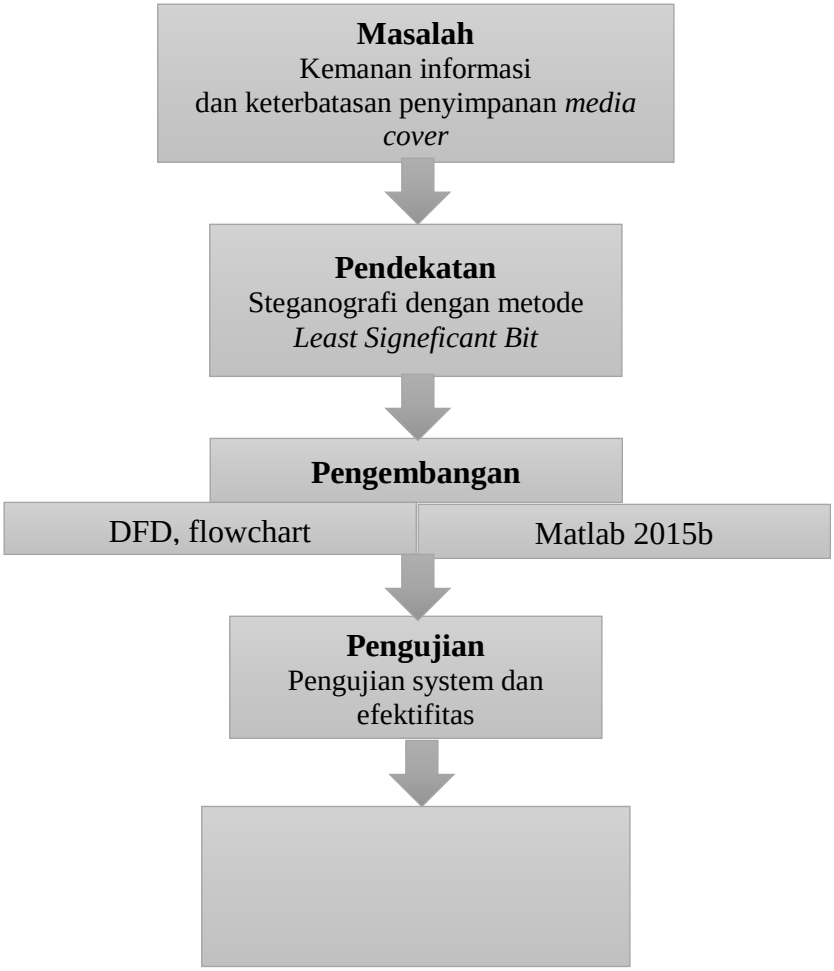
Dari penelitian tersebut didapati bahwa dengan melakukan *preprocessing* terlebih dahulu terhadap berkas data maupun berkas wadah yang akan digunakan, teknik steganografi ini mampu "menyimpan data yang lebih besar" dari kapasitas maksimum (*steganographic capacity*) wadah yang sebenarnya.

2.3 Kerangka Berpikir

Keamanan informasi menjadi isu penting bagi manusia zaman modern ini, steganografi dan kriptografi tidak dapat terlepas dari isu keamanan informasi tersebut. Steganografi berperan dalam menyembunyikan keberadaan dari informasi yang dirhasiakan. Langkah ini diambil agar tidak ada kecurigaan akan

adanya informasi rahasia tersebut. Langkah berpikir penulis digambarkan pada gambar 2.7.

Steganografi dengan metode LSB merupakan metode yang populer digunakan walaupun memiliki kelemahan yaitu hanya dapat menyimpan data informasi seperdelapan dari total ukuran piksel dari *media cover* yang digunakan. Dalam penelitian kali ini LSB akan dikombinasikan dengan *image processing* yaitu *image scaling* secara teori akan mengekspansi *media cover* sehingga ukuran piksel juga akan berubah. Berubahnya ukuran piksel akan menambah kapasitas penyimpanan informasi.



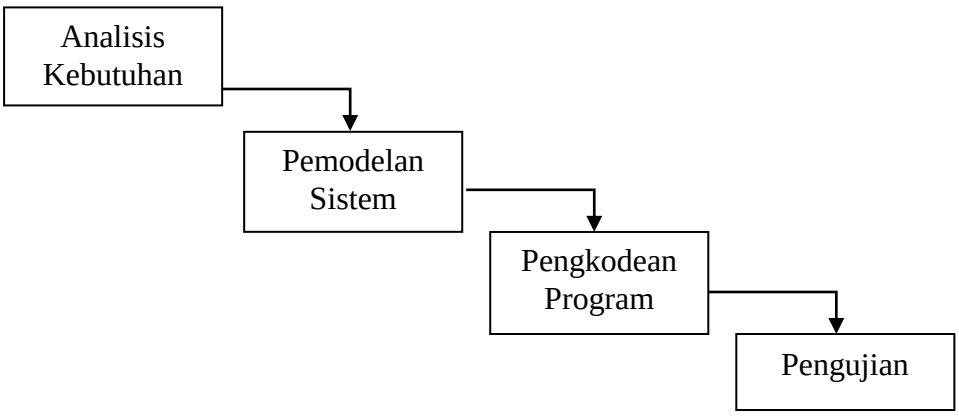
Gambar 2.7 Kerangka berpikir

BAB III

METODE PENELITIAN

3.1 Metode Pengembangan Sistem

Metode yang digunakan pada penelitian ini yaitu metode *waterfall* atau juga disebut *Linear Sequential Model*. Setiap tahapan dalam metode ini merupakan *input* bagi tahapan selanjutnya dengan pendekatan secara sekuensial atau berurutan dari analisi, desain, pengkodean dan pengujian (Pressman, 2010).



Gambar 3.1 Skema model *Waterfall*

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut :

1. Analisis

   Merupakan analisis terhadap kebutuhan sistem. Pengumpulan data dilakukan melalui penelitian, studi literatur dan wawancara untuk mendapatkan informasi sebanyak mungkin dari pengguna sehingga tercipta sistem yang

sesuai apa yang diinginkan pengguna. Tahap ini akan menghasilkan data yang berisi apa saja yang dibutuhkan pengguna dalam pembuatan sistem.

2. Desain atau pemodelan

   Tahap ini menerjemahkan syarat kebutuhan perancangan sebelum *coding* dilakukan. Pada tahap ini berfkous pada struktur data, arsitektur perangkat lunak, representasi *interface,* dan detail prosedural (algoritma). Tahapan ini menghasilkan dokument *software requirment* dan dokumen ini digunakan untuk melakukan aktivitas pembuatan sistem.

3. Pengkodean

   Tahap ini merupakan penerjemahan desain ke dalam bahasa yang dapat dikenali oleh komputer.

4. Pengujian

   Tahap ini dilakukan untuk menguji program apakah sudah berjalan sesuai dengan fungsinya atau belum.

3.2 Analisa Kebutuhan

Berdasarkan metode *waterfall* di atas, tahapan pertama yang harus dilakukan adalah persiapan awal. Tahapan ini terdiri dari :

1. Mengidentifikasi Masalah dan Kebutuhan Sistem

   Identifikasi masalah dan kebutuhan sistem merupakan tahapan awal dari suatu penelitian. Pada tahap ini peneliti mulai mengumpulkan data dan mengidentifikasi masalah serta kebutuhan sistem yang akan dibuat. Berikut ini adalah identifikasi masalah yang dirangkum :

a. Informasi rahasia dikemas dengan menggunakan enkripsi data. Hal ini hanya mengacak informasi, akan tetapi keberadaan informasi tersebut masih dapat diketahui dan menimbulkan kecurigaan bagi orang-orang yang tidak berkepentingan.
b. Steganografi LSB memiliki keterbatasan kapasitas penyimpanan informasi.

Berdasarkan identifikasi permasalahan di atas, maka kebutuhan sistem yang akan dibuat adalah :

a. Menggunakan Steganografi LSB (*Least Significant Bit*) untuk menyamarkan keberadaan suatu informasi rahasia agar tidak menimbulkan kecurigaan bagi orang lain sehingga keamanan informasi tersebut lebih terjamin.
b. Merancang pembuatan aplikasi Steganografi LSB (*Least Significant Bit*) berbasis Matlab yang dikombinasikan dengan metode *Image scaling* untuk meningkatkan kapasitas penyimpanan informasi.

2. Mempersiapkan alat dan bahan

Tahap ini peneliti mempersiapkan alat dan bahan yang digunakan dalam penelitian yang dilakukan sesuai dengan kebutuhan. Penelitian akan berjalan lancar jika alat dan bahan yang dibutuhkan dalam penelitian sesuai dengan yang dibutuhkan dan memenuhi kualifikasi minimal. Alat yang digunakan dalam pengembangan aplikasi ini dibagi menjadi 3 jenis yakni perangkat keras (*hardware*), perangkat lunak (*software*), dan *User* (*brainware*).

a. Perangkat Keras

Perangkat keras dapat diartikan sebagai komponen yang dibutuhkan berbentuk nyata dan fisiknya kasat mata. Pembuatan aplikasi steganografi ini memerlukan PC atau notebook dengan kebutuhan minimal untuk menjalankan program Matlab, berikut ini kebutuhan minimal yang didapat dari situs http://www.mathworks.com :

1) Prosesor Intel maupun AMD 32 atau 64 bit (disarankan 4 core)
2) Monitor SVGA
3) Memori RAM 2 GB
4) Hardisk 60 GB, 2 GB Hanya untuk Matlab, sedang 4-6 GB untuk instalasi penuh, sisanya untuk sistem operasi
5) Keyboard dan Mouse
6) Kartu grafis 1 GB untuk Matlab standar, serta 4 GB untuk simulink

b. Perangkat Lunak

Perangkat lunak diartikan sebagai komponen yang berbentuk program/aplikasi yang dapat dijalankan melalui PC atau notebook. Perangkat lunak yang digunakan dalam pengembangan aplikasi antara lain:

1) Sistem Operasi Komputer : Windows 7
2) Compiler : Matlab 2015b

c. Pengguna

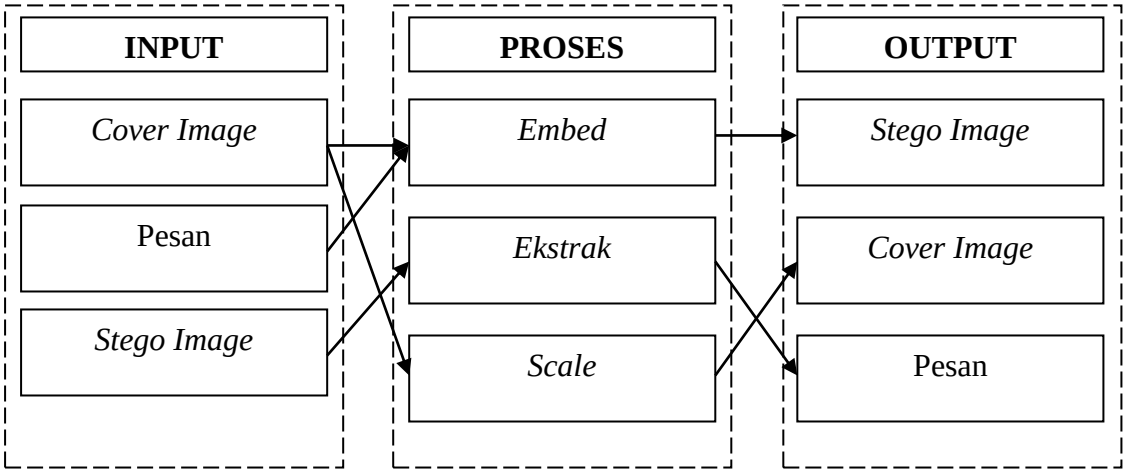
Aplikasi ini butuh user untuk mengoperasikannya. Dalam hal ini aplikasi hanya membutuhkan 1 pengguna untuk menggunakan aplikasi untuk steganografi.

3.3 Pemodelan Sistem

Pemodelan (*design*) aplikasi steganografi ini menggunakan model perancangan antarmuka (*interface*), dan diagram alir (*flowchart*).

3.3.1 Perancangan *Interface*

Perancangan antarmuka menfokuskan pada tiga area yaitu rancangan antarmuka antara modul-modul perangkat lunak, rancangan antarmuka antara perangkat lunak dengan entitas eksternal dan rancangan antarmuka antara perangkat lunak dengan pengguna perangkat lunak (manusia dengan komputer) (Prasetyo dkk, 2009 : 92). Perancangan antarmuka ini akan menghasilkan GUI (*Graphical User Interface*) yang menampilkan proses steganografi serta akan menghasilkan luaran steganografi yang berupa *stegi image* dan *plainteks* sebagai pesan yang disisipkan. Rancangan antara muka tampilan digambarkan pada gambar 3.2 :



Gambar 3.2 Rancangan Tampilan

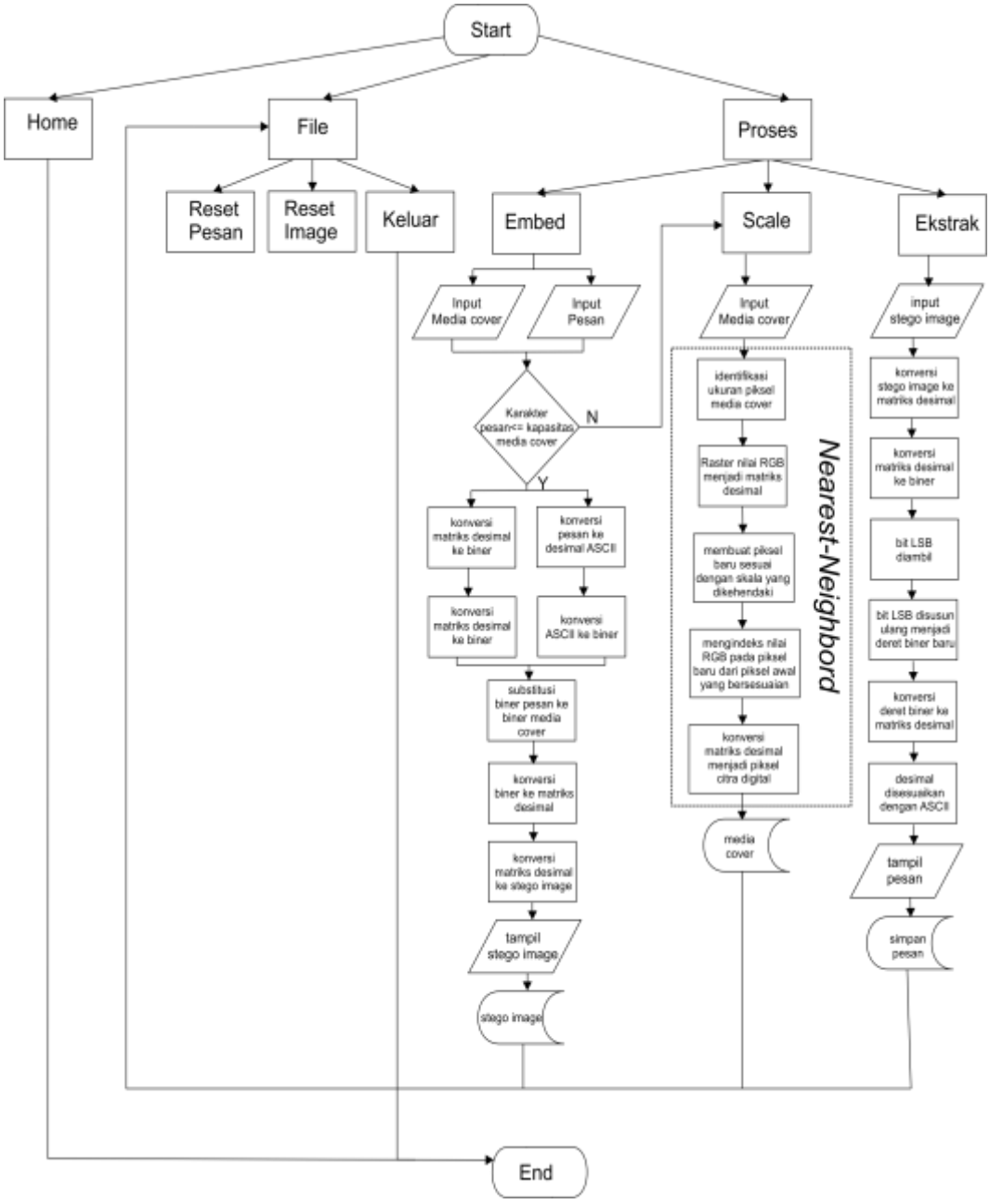
Penjelasan rancangan model :

1. Rancangan akan dibagi menjadi 3 bagian tampilan utama yaitu : *Input*, Proses, dan *Output*.
2. Pada Rancangan *Input* merupakan masukan yang diberikan oleh pengguna. Dalam hal ini akan dibagi menjadi 3 buah masukan yaitu : *cover image*, pesan, dan *stego image*.
3. Rancangan Proses merupakan proses utama yang akan dilakukan oleh program. Proses *Embed* merupakan proses penyisipan pesan ke dalam citra digital yang dalam hal ini akan mengambil masukan berupa *cover image* dan pesan. Proses *Scale* merupakan proses perbesaran citra digital sebagai pembawa pesan dengan mengambil masukan *cover image.* Proses *ekstrak* merupakan proses pengambilan kembali pesan yang disisipkan dari citra digital dengan mengambil masukan berupa *stego image.*
4. Rancangan *Output* merupakan tampilan yang akan menampilkan hasil dari proses yang dilakukan. Proses *Embed* akan menghasilkan *stego image* sebagai keluarannya. Proses *Scale* akan menghasilkan *cover image* baru dengan ukuran yang berbeda. Proses *ekstrak* akan menghasilkan pesan berupa *plainteks* sebagai pesan yang tersimpan.

3.3.2 Digram Alir (*Flowchart*)

Flowchart adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial yang menggambarkan langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* digunakan untuk

merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman. *Flowchart* menolong analis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian (Adelia dan Setiawan, 2011 : 116). Dalam penelitian ini *flowchart* steganografi lsb ditunjukkan pada gambar 3.3.



Gambar 3.3 Desain *Flowchart* aplikasi

3.4 Penulisan Kode (*scripting*)

Pembuatan antar muka, pemulisan kode proses, serta *compiler* menggunakan program Matlab R2015b karena Matlab telah menyediakan fungsi-fungsi yang memudahkan dalam pembuatan program steganografi.

3.5 Pengujian Sistem

Pengujian dilakukan untuk mengetahui apakah aplikasi ini sudah bisa berjalan dengan baik atau belum. Menurut Munir (2006) aplikasi steganografi memiliki tiga kriteria sebagai tolok ukur berhasil atau tidaknya aplikasi steganografi melakukan tugasnya, kriteria itu adalah *Imperectibility*, *Fidelity*, *Recoverable.*

3.5.1 Pengujian *Black Box*

Pengujian *black box* digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Pada teknik pengujian *black box* kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya (Astuti, 2012 : 90).

3.5.2 Pengujian *Imperectibility.*

Pengujian ini digunakan untuk mengetahui seberapa mudah sebuah *stego-image* dapat terdeteksi oleh inderawi manusia. Pengujian tingkat *imperectibility* dilakukan secara manual dengan melibatkan sejumlah responden. Responden

memberikan nilai secara subyektif terhadap apa yang dilihatnya. Langkah nya adalah :

1. Responden melihat *cover image* dan *stego image* citra digital yang sama yang disejajarkan pada layar monitor.
2. Responden memberikan nilai sesuai dengan rubrik yang disediakan pada tabel 3.1.
3. Penilaian dilakukan pada semua citra uji dengan susunan berbeda pada setiap responden.

Poin utama adalah untuk menganalisa tingkat distorsi akibat proses steganografi. Pengujian dilakukan dengan menggunakan HVS (*human visual system*) atau sistem indera penglihatan manusia. Koresponden akan diminta untuk memberikan penilaian berdasarkan tabel berikut :

Tabel 3.1 Keterangan nilai *imperectibility* (Piarsa, 2011)

NILAI	Keterangan Citra	Keterangan
5	Sama	Kesamaan citra mencapai 90-100%
4	Sedikit Sama	Kesamaan citra mencapai 70-90%
3	Sedikit Beda	Kesamaan citra mencapai 60-70%
2	Beda	Kesamaan citra mencapai 40-60%
1	Sangat Beda	Kesamaan citra mencapai <40%

3.5.3 Pengujian *Fidelity.*

Merupakan pengujian terhadap mutu (*fidelity*) citra hasil steganografi. Metode yang akan digunakan adalah dengan mengukur nilai MSE (*Mean Square*

Error) dan PNSR (*Peak Signal to Noise Ratio*) pada *stego image*. Keduanya merupakan sebuah nilai yang memiliki satuan dB (*desibels*). Semakin rendah nilai MSE maka kualitas citra semakin baik. Sementara itu, mutu *stego-image* dikatakan baik jika nilai PSNR lebih dari atau sama dengan 40 dB (solichin, 2015). Beberapa variabel yang akan diuji adalah :

a. Pengaruh jumlah karakter yang disisipkan terhadap nilai *error measurement.*
b. Pengaruh ukuran piksel data citra terhadap nilai *error measurement.*

Nilai PSNR dan MSE dapat dihitung dengan rumus sebagai berikut.

$$PNSR = 10\ log_{10}(\frac{C_{max}^2}{MSE})$$

$$MSE = \frac{1}{M.N}\sum_{x=1}^{M}\sum_{y=1}^{N}(S_{xy} - C_{xy})^2$$

Keterangan :

- C_{max} adalah nilai pixel terbesar pada keseluruhan citra.
- X dan Y adalah koordinat suatu titik pada citra.
- M dan N adalah simensi citra.
- S adalah citra tersisipi (*stego-image*)
- C adalah citra asli (*cover image*)

Berikut ini contoh perhitungan PSNR.

- Nilai pixel citra asli berukuran 3x3 pixel.

7	1	1
2	3	4
5	0	6

- Nilai pixel citra setelah disisipi berukuran 3x3 pixel.

6	1	1
4	3	4
5	3	6

Perhitungan dimulai dengan menghitung nilai MSE terlebih dahulu.

$$MSE = \frac{(6-7)^2 + (1-1)^2 + (1-1)^2 + (4-2)^2 + (3-3)^2 + (4-4)^2 + (5-5)^2 + (3-0)^2 + (6-6)^2}{3 \cdot 3}$$

$$MSE = \frac{1+0+0+4+0+0+0+9+0}{9}$$

$$MSE = 1{,}55$$

Setelah diketahui nilai MSE, maka dilanjutkan dengan menghitung PNSR.

$$PNSR = 10\ log_{10}(\frac{C_{max}^2}{MSE})$$

$$PNSR = 10\ log_{10}\left(\frac{7^2}{1{,}55}\right)$$

$$PNSR = 14{,}5$$

3.5.4 Pengujian *Recoverable.*

Pesan rahasia yang disisipkan pada sebuah citra harus dapat dipisahkan kembali dengan *stego-image* nya. Pengujian dapat dilakukan dengan melihat

keutuhan pesan yang diekstraksi dari sejumlah citra uji. Pengujian dilakukan dengan cara memproses *stego image* untuk dilakukan pengekstrakan pesan yang tersimpan. Selanjutnya pesan hasil ekstrak dibandingkan dengan pesan sebelum disisipkan.

BAB IV

HASIL PENELITIAN dan PEMBAHASAN

3.6 Hasil Penelitian

Pengujian aplikasi dilakukan untuk mengetahui apakah aplikasi yang dibuat berjalan dengan baik sesuai dengan fungsi dan tahap yang sudah dirancang sebelumnya. Hasil penelitian yang telah didapat meliputi tahap pengumpulan data, desain produk, uji hasil baik secara subyektif (berdasarkan pengguna) maupun obyektif (berdasarkan obyek yang diteliti).

4.1.1 Obyek Penelitian

Tahap pengumpulan data dalam penelitian ini dilakukan dengan mengumpulkan obyek citra yang digunakan sebagai obyek utama penelitian, serta pengumpulan data skor dari responden terhadap hasil penelitian. Data citra uji ditunjukkan oleh gambar 4.1, 4.2, 4.3 serta gambar 4.4 berikut :

Gambar 4.1 Lena RGB format Bitmap ukuran 512x512 (Sumber : http://sipi.usc.edu/database/database.php?volume=misc)

Gambar 4.2 Lena *grayscale* format Bitmap ukuran 512x512 (Sumber :

http://www.eecs.qmul.ac.uk/~phao/CIP/Images/)

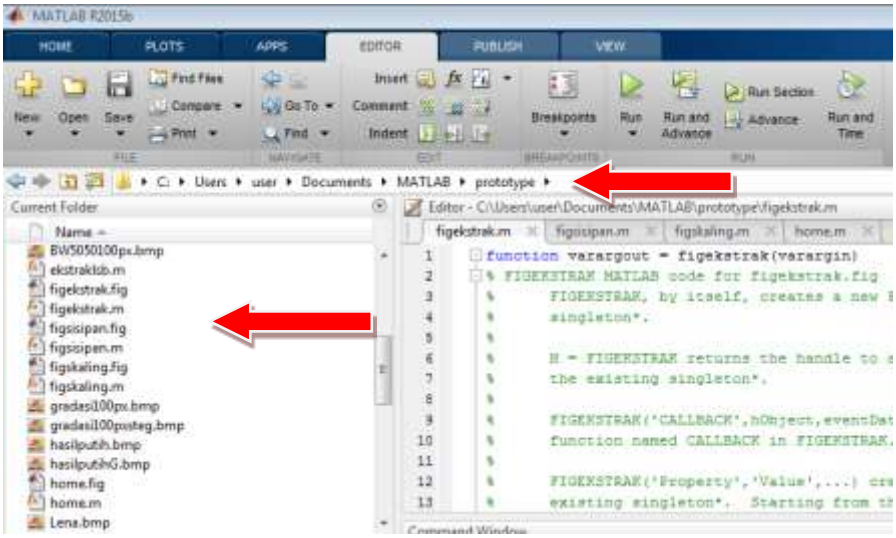
Gambar 4.3 Peppers RGB format Bitmap ukuran 512x512 (Sumber :

http://www.eecs.qmul.ac.uk/~phao/CIP/Images/)

Gambar 4.4 Baboon RGB format Bitmap ukuran 512x512 (Sumber : http://www.eecs.qmul.ac.uk/~phao/CIP/Images/)

4.1.2 Tampilan dan Cara Penggunaan Aplikasi

Hal pertama yang diperlukan dalam menjalankan aplikasi ini adalah memastikan komputer yang akan digunakan sudah terinstal aplikasi Matlab dan berjalan dengan baik dikarenakan aplikasi steganografi LSB ini dibuat dan dijalankan dengan aplikasi Matlab. Arahkan *Workspace* Matlab kedalam folder tempat dimana *source code* aplikasi ditempatkan seperti ditunjukkan gambar 4.5.



Gambar 4.5 Contoh *workspace* yang digunakan dalam Matlab

1. Tampilan Halaman Utama

Terdapat 4 buah .m file utama yang berperan dalam tampilan program steganografi ini yaitu : home.m, figskaling.m, figsisipan.m, dan figekstrak.m. Jalankan home.m untuk menampilkan halam utama aplikasi.

Gambar 4.6 Tampilan Halaman utama aplikasi.

Pada halaman utama (gambar 4.6), aplikasi menampilkan judul aplikasi serta memuat 3 menu utama di *menu bar* aplikasi. 3 menu utama serta turunannya adalah sebagai berikut :

Home

Digunakan untuk kembali ke halaman utama ketika di klik.

File

a. Reset Image

   Digunakan untuk membersihkan panel gambar.

b. Reset Pesan

Digunakan untuk membersihkan panel pesan.

c. Keluar

Digunakan untuk keluar aplikasi.

Proses

d. Embed

Digunakan untuk menyisipkan teks kedalam citra digital.

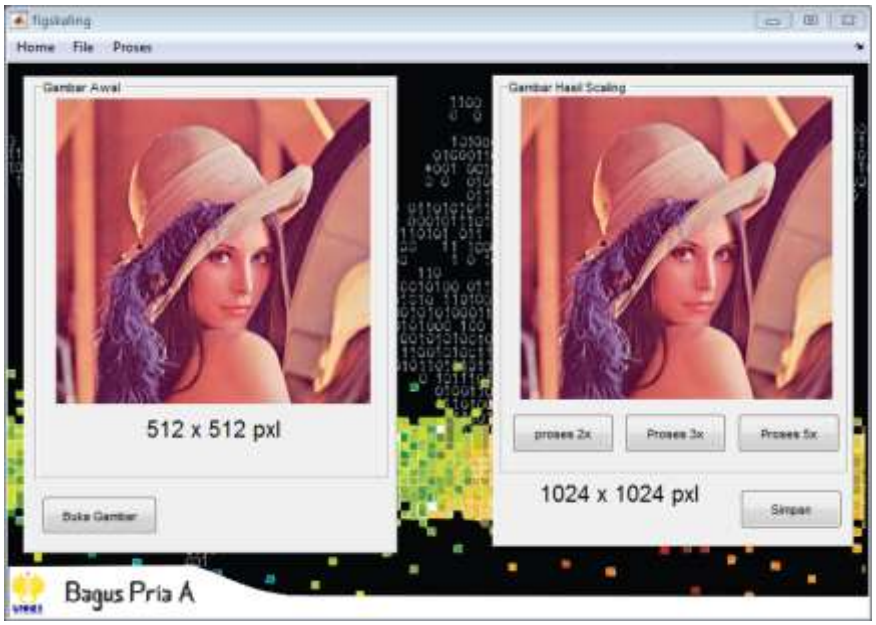
e. Ekstrak

Digunakan untuk mengekstraksi teks yang telah disisipkan.

f. Scale

Digunakan untuk mengatur perubahan skala citra digital.

2. Tampilan Menu Scale

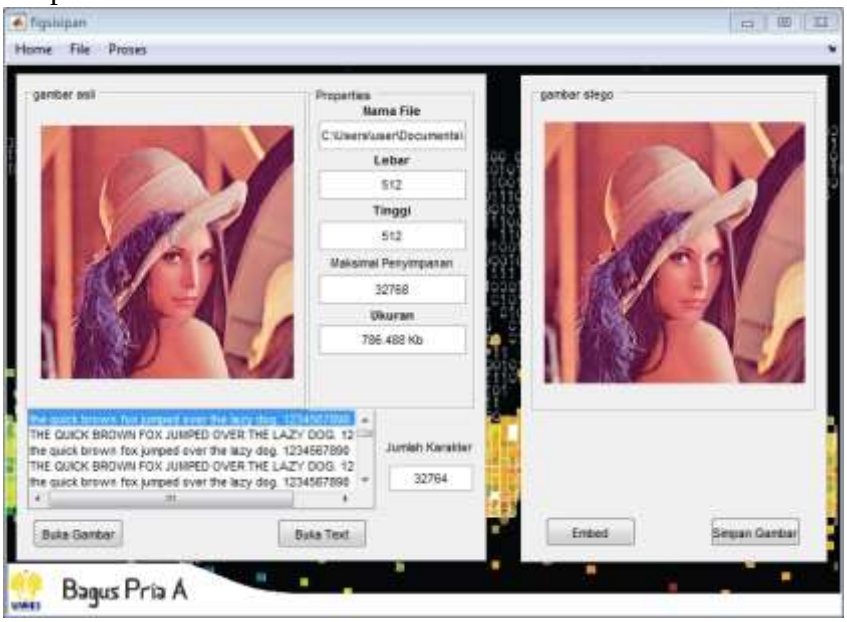


Gambar 4.7 Tampilan Menu Scale.

Menu ‘scale’ pada gambar 4.7 menampilkan proses perubahan skala citra digital, langkah pertama adalah dengan mengklik ‘Buka Gambar’ untuk memilih citra digital yang akan diproses. Selanjutnya memilih besaran skala yang dikehendaki (2x, 3x, 5x) lalu simpan

gambar hasil proses yang sudah dilakukan, maka jendela menu akan menampilkan ukuran citra digital setelah proses.

3. Tampilan Menu Embed



Gambar 4.8 Tampilan Menu Embed.

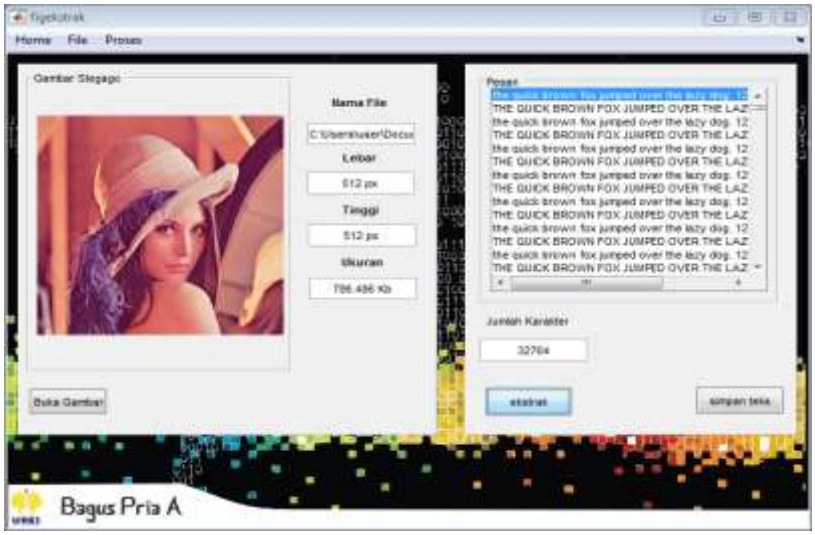
Menu ‘Embed’ seperti gambar 4.8 menampilkan proses menyisipkan pesan ke dalam gambar yang dikehendaki. Langkah pertama adalah memilih gambar yang dikehendaki, kemudian memilih pesan yang ingin disisipkan (format pesan .txt). Sebagai pesan yang digunakan sebagai pengujian kali ini adalah standar pangram yang menampilkan seluruh abjad dan angka yang ada pada tombol *keyboard "**the quick brown fox jumped over the lazy dog. 1234567890 THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG. 1234567890**"* Selanjutnya klik menu embed untuk memulai proses, setelah proses selesai maka jendela aplikasi akan menampilkan gambar yang telah disisipi pesan tersebut. Langkah terakhir adalah menyimpan gambar

hasil proses penyisipan pesan atau biasa disebut *stego image* (gambar 4.9).

(A) (B)

Gambar 4.9 (A) Citra Lena (RGB) sebelum dilakukan proses penyisipan pesan. (B) Citra Lena (RGB) setelah dilakukan proses penyisipan pesan.

4. Tampilan Menu Ekstrak



Gambar 4.10 Tampilan Menu Ekstrak

Menu ‘Ekstrak’ ditunjukkan gambar 4.10 menampilkan proses ekstraksi pesan yang disisipkan kedalam *stego image*, yang pertama

dilakukan adalah memilih *stego image* yang telah disimpan sebelumnya, selanjutnya klik menu ekstrak maka jendela aplikasi akan menampilkan hasil ekstrak pesan. Langkah terakhir adalah menyimpan pesan hasil ekstrak tersebut kedalam file dengan format .txt.

4.1.3 Uji *Black Box*

Pengujian dilakukan dengan menguji setiap proses dan kemungkinan kesalahan yang terjadi pada setiap proses. Pengujian ini dilakukan secara *black box* yaitu pengujian dilakukan dengan memperhatikan masukan ke sistem dan keluaran dari sistem apakah sesuai atau tidak yang diharapkan. Hasil pengujian ditunjukkan oleh tabel 4.1.

Tabel 4.1 Uji *Black Box*

No	Prosedur Pengujian	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
1	Klik pada menu Home	Masuk pada halaman utama	Ditampilkan halaman utama aplikasi	Sesuai
2	Klik pada menu scale	Masuk pada halaman scaling	Ditampilkan halaman scaling	Sesuai
3	Klik tombol buka gambar pada menu scale	Masuk jendela pemilihan direktori gambar	Ditampilkan direktori pemilihan gambar	Sesuai
4	Klik tombol proses pada halaman scale	Gambar diproses untuk perbesaran dan ditampilkan dihalaman	Ditampilkan gambar hasil proses scale	Sesuai
5	Klik simpan gambar pada halaman scale	Menampilkan dialog untuk menempatkan gambar hasil ke direktori yang dikehendaki	Ditampilkan pemilihan direktori penyimpanan gambar	Sesuai
6	Klik menu file >	Membersihkan	Gambar hasil	sesuai

	Reset image	gambar dihalaman	menjadi kosong seperti semua	
7	Klik menu proses > Embed	Masuk halaman penyisipan pesan (embedding)	Ditampilkan halaman penyisipan pesan	Sesuai
8	Klik tombol buka gambar pada halaman embed	Keluar jendela direktori tempat pemilihan gambar	Ditampilkan jendela direktori pemilihan gambar	Sesuai
9	Klik tombol buka pesan pada halaman embed	Keluar kendela direktori tempat pemilihan pesan (txt)	Ditampilkan jendela direktori	Sesuai
10	Klik tombol embed pada halaman embed	Gambar dan pesan yang dipilih diproses steganografi	Keluar gambar hasil proses steganografi	Sesuai
11	Klik simpan gambar pada halaman embed	Keluar jendela direktori untuk menempatkan gambar hasil proses embedding	Ditampilkan jendela penyimpanan direktori	Sesuai
12	Klik menu proses > ekstrak	Masuk halaman ekstrak pesan yang telah disisipkan	Ditampilkan halaman pengekstrakan pesan	Sesuai
13	Klik tombol buka gambar pada halaman ekstrak	Keluar jendela pemilihan gambar yang akan diekstrak pesan didalamnya	Ditampilkan jendela pemilihan gambar	Sesuai
14	Klik tombol ekstrak pada halaman ekstrak	Gambar yang dipilih diproses dan ditampilkan pesan yang telah diekstrak	Ditampilkan pesan yang diekstrak pada halaman ekstrak	Sesuai
15	Klik tombol simpan pesan	Keluar direktori pemilihan tempat dimana pesan hasil ekstrak akan disimpan dalam format txt	Keluar jendela pemilihan direktori penyimpanan hasil ekstrak pesan	Sesuai

4.1.4 Hasil Responden

Berikut ini ditampilkan responden yang mengamati gambar hasil dari proses penyisipan pesan dengan menggunakan aplikasi steganografi LSB yang telah dibuat. Hal ini dilakukan untuk membuktikan apakah terjadi perubahan yang signifikan terhadap gambar yang telah dilakukan proses *embedding* pesan. Untuk menentukan persentase hasil digunakan rumus sebagai berikut :

$$Pr = n/N \times 100\%$$

Keterangan : Pr : Persentase hasil respon pengguna

n : total skor yang didapat

N : skor maksimal

Tabel 4.2 Hasil Penilaian *Imperectibility Human Visual System* Responden.

No	NAMA	Lena RGB	Baboon RGB	Peppers RGB	Lena (grayscale)
1	Eko Budi Utomo	5	5	5	5
2	M. Nur Sobroni	5	5	5	5
3	Anton Wicaksana	5	5	4	4
4	Yoke Ana Marlina	5	4	5	4
5	Revita Anditya P	5	4	4	5
6	Yohana Ella K	5	4	4	4
7	Siti Nurkhayati	5	5	5	5
8	Ayu Novita F	5	5	5	5
9	Ruhaniah Atma Sari	4	5	4	4
10	Eka Yulianti F	5	4	5	4
11	Ifah Nur A	4	5	4	4
12	Risalatil Laeli	5	4	2	5
	Jumlah	58	55	52	54
	Rata-Rata	4.8	4.5	4.3	4.5

Berdasarkan hasil penilaian responden dengan pengamatan secara langsung dengan mata didapat bahwa penilaian terhadap *cover image* dan *stego image* sebagai berikut :

LenaRGB.bmp : (58/60) x 100% = 96.6%

Rata-rata = 4.8

BaboonRGB.bmp : (55/60) x 100% = 91.6%

Rata-rata = 4.5

PeppersRGB.bmp : (52/60) x 100% = 86.6%

Rata-rata = 4.3

Lena(grayscale).bmp : (54/60) x 100% = 90%

Rata-rata = 4.5

Berdasarkan tabel penilaian 4.2 dan rubrik yang ada dapat dikatakan bahwa setiap *stego image* yang dihasilkan dari steganografi ini memiliki kualitas yang bagus. Hasil yang didapat dapat digolongkan "sama" antara *cover image* dan *stego image* secara kasat mata.

4.1.5 Uji Pengaruh Dimensi Citra Digital Terhadap Kapasitas Penyimpanan

LSB Steganografi sangat erat hubungannya dengan manipulasi komponen piksel dari suatu citra digital. Kapasitas penyimpanan dalam citra digital sangat dipengaruhi oleh jumlah piksel dari citra digital. Semakin besar ukurannya, maka semakin banyak kapasitas penyimpanan citra tersebut dalam proses steganografi. Pengujian ini menggunakan citra digital uji dengan ukuran

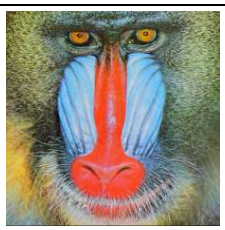
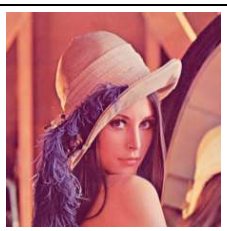
bervariasi: 512x512 px, 256x256 px, 128x128 px. Untuk menghitung kapasitas steganografi LSB digunakan rumus berikut ini :

$$Kapasitas = \frac{(ukuran\ citra)}{8}$$

Kapasitas : kapasitas steganografi LSB

Ukuran citra : dimensi ukuran citra (misal: 512 x 512 px)

Tabel 4.3 Tabel Pengaruh *scaling* Dimensi Citra Digital Terhadap Kapasitas Penyimpanan

NO	Citra Digital	Dimensi	File Size	Kapasitas Steganografi LSB
1	 PeppersRGB.bmp	512 X 512 px	769 Kb	32768 karakter
		256 X 256 px	193 Kb	8192 karakter
		128 X 128 px	49 Kb	2048 karakter
2	 BaboonRGB.bmp	512 X 512 px	769 Kb	32768 karakter
		256 X 256 px	193 Kb	8192 karakter
		128 X 128 px	49 Kb	2048 karakter
3	 LenaRGB.bmp	512 X 512 px	769 Kb	32768 karakter
		256 X 256 px	193 Kb	8192 karakter
		128 X 128 px	49 Kb	2048 karakter

4	 Lena.bmp	512 X 512 px	258 Kb	32768 karakter
		256 X 256 px	66 Kb	8192 karakter
		128 X 128 px	18 Kb	2048 karakter

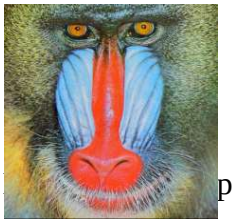
Berdasarkan data tabel 4.3 di atas, citra uji "peppersRGB.bmp" memiliki dimensi piksel 512x512, dengan *file size* 769 Kb memiliki kapasitas penyimpanan steganografi sebanyak 32768 karakter plain teks. Kapasitas LSB steganografi hanya dipengaruhi oleh jumlah total piksel dari suatu citra digital. Ukuran citra (*file size)* dan tipe citra digital (*true color* ataupun *grayscale*) tidak mempengaruhi kapasitas penyimpanan. Kapasitas penyimpanan suatu citra digital untuk proses LSB steganografi adalah seper delapan dari total piksel citra tersebut. Oleh karena itu citra digital dengan ukuran piksel 512x512 memiliki total kapasitas penyimpanan 32768 karakter plain teks, sedangkan citra dengan 256x256 piksel memiliki kapasitas penyimpanan total 8192 karakter plainteks sedangkan untuk ukuran 128x128 memiliki kapasitas penyimpanan 2018 karakter plain teks hal ini terlepas dari berapapun *file size* dari citra digital tersebut.

4.1.6 Uji Pengaruh Volume Penyisipan Karakter Terhadap *File Size* Citra Digital.

Proses steganografi pada intinya adalah melakukan substitusi nilai biner terhadap komponen piksel dari sebuah citra digital. Secara teori tidak akan mengubah *file size* dari citra digital tersebut. Hal itu dapat dibuktikan pada tabel 4.4 dibawah ini. Pengujian dilakukan dengan menggunakan citra uji dengan

ukuran 512x512 px, 256x256 px, dan 128x128 px. Sedangkan volume penyisipan yang digunakan adalah 100%,50%,25%,dan 15% dari kapasitas maksimal penyisipan pesan dengan LSB.

Tabel 4.4 Tabel Pengaruh Volume Penyisipan Karakter Terhadap *File Size*

NO	Citra Digital	Dimensi	File Size Sebelum Penyisipan	Volume Penyisipan (dihitung dari kapasitas maksimal)	File Size Sesudah Penyisipan
1		512 X 512 px	769 Kb	100 %	769 Kb
				50 %	
				25 %	
				15 %	
		256 X 256 px	193 Kb	100 %	193 Kb
				50 %	
				25 %	
				15 %	
		128 X 128 px	49 Kb	100 %	49 Kb
				50 %	
				25 %	
				15 %	
2		512 X 512 px	769 Kb	100 %	769 Kb
				50 %	
				25 %	
				15 %	
		256 X 256	193 Kb	100 %	193 Kb

		px		50 %	
				25 %	
				15 %	
		128 X 128 px	49 Kb	100 %	49 Kb
				50 %	
				25 %	
				15 %	
3	 LenaRGB.bmp	512 X 512 px	769 Kb	100 %	769 Kb
				50 %	
				25 %	
				15 %	
		256 X 256 px	193 Kb	100 %	193 Kb
				50 %	
				25 %	
				15 %	
		128 X 128 px	49 Kb	100 %	49 Kb
				50 %	
				25 %	
				15 %	
4	 Lena.bmp	512 X 512 px	258 Kb	100 %	258 Kb
				50 %	
				25 %	
				15 %	
		256 X 256 px	66 Kb	100 %	66 Kb
				50 %	
				25 %	
				15 %	
		128 X 128	18 Kb	100 %	18 Kb

		px		50 %	
				25 %	
				15 %	

Berdasarkan isi tabel 4.4 dari percobaan diatas, terlihat bahwa penyisipan pesan ke dalam citra digital tidak mempengaruhi *file size* citra digital. Percobaan dilakukan dengan melakukan penyisipan pesan sejumlah 100%, 50%, 25% dan 15% dari kapasitas maksimal penyisipan pesan citra digital tersebut. Citra PeppersRGB.bmp berdimensi 512x512 piksel dengan ukuran sebelum penyisipan 769 Kb dilakukan proses penyisipan 100% dari kapasitas maksimal penyisipan dihasilkan ukuran yang sama, yaitu 769 Kb.

Selanjutnya pada obyek citra digital yang sama dilakukan proses penyisipan 50% dari kapasitas maksimal penyisipan, didapatkan ukuran yang tetap yaitu 769 Kb. Dalam proses penyisipan dengan volume 25% dan 15% dari kapasitas penyimpanan, didapatkan ukuran yang tidak berubah 769 Kb. Proses serupa dilakukan pada citra digital PeppersRGB.bmp dengan dimensi 216x256 piksel dengan ukuran sebelum penyisipan 193 Kb dengan penyisipan 100%,50%,25%,dan 15% dihasilkan ukuran yang tidak berubah dari sebelum dan sesudah penyisipan yaitu 193 Kb. Setelah dilakukan dengan 4 obyek citra berbeda dengan dimensi 512x512 px, 256x256 px, serta 128x128 px dengan perlakuan yang sama yaitu penyisipan 100%,50%,25% dan 15% dari kapasitas maksimal penyisipan didapatkan bahwa tidak ada perubahan *file size* (ukuran) pada citra digital tersebut.

4.1.7 Uji Nilai MSE dan PSNR *Stego Image*

Dalam melakukan perbandingan antara 2 buah citra digital sebelum dan sesudah proses steganografi, penulis menggunakan nilai MSE (*Mean Square Error*) dan PSNR (*Peak Signal to Noise Ratio*) keduanya memiliki satuan dB (desibels). MSE dan PSNR merupakan nilai yang digunakan untuk mengukur kesamaan dan error yang ada pada citra digital setelah proses *image processing* dalam hal ini proses steganografi. Mutu *stego-image* dikatakan baik jika nilai PSNR lebih dari atau sama dengan 40 dB (solichin, 2015).

Tabel 4.5 Tabel nilai MSE dan PSNR citra digital setelah dilakukan penyisipan pesan

No.	**Citra Digital**	**Penyisipan**	**MSE**			**PSNR**		
			RED	**GREEN**	**BLUE**	**RED**	**GREEN**	**BLUE**
1.	PeppersRGB.bmp	**100%**	**0.4982**	**0**	**0**	**61.3476**	**0**	**0**
2.	BaboonRGB.bmp	**100%**	**0.4998**	**0**	**0**	**62.3472**	**0**	**0**
3.	LenaRGB.bmp	**100%**	**0.5021**	**0**	**0**	**62.3031**	**0**	**0**
4.	Lena.bmp (grayscale)	**100%**	**0.5000**			**61.9995**		

Berdasarkan tabel 4.5, pengukuran MSE dan PSNR dilakukan pada citra digital dengan melakukan penyisipan 100% dari kapasitas maksimal penyisipan proses steganografi. Citra yang digunakan terdiri dari 2 tipe citra, yaitu *True Color (RGB)* yaitu "PeppersRGB;BaboonRGB;LenaRGB" dan *Grayscale* yaitu "Lena". Untuk nilai MSE hanya didapat dari komponen RGB(*Red Green Blue*) yaitu

komponen R (*Red*) saja. Hal ini dikarenakan proses steganografi yang dilakukan oleh peneliti hanya memanipulasi komponen piksel RGB tepatnya komponen R (*Red*).

Sedang untuk komponen G (*Green*) dan B (*Blue*) tidak mengalami perubahan antara sebelum dan sesudah proses steganografi, sehingga nilai MSE pada komponen G dan B adalah 0. Nilai MSE berbanding terbalik dengan PSNR. Jika citra uji memiliki nilai MSE yang tinggi, maka akan memiliki nilai PSNR yang rendah. Hal ini mengindikasikan bahwa terdapat banyak error pada citra tersebut. Sebaliknya, jika sebuah citra memiliki nilai MSE yang rendah, maka akan memiliki nilai PSNR yang tinggi, hal ini menunjukkan bahwa terjadi sedikit error setelah proses steganografi.

Pada citra "PeppersRGB.bmp" MSE komponen R didapat nilai MSE=0,4982 dan dari nilai MSE tersebut didapatkan nilai PSNR=61,3476 dB. Berdasarkan nilai tersebut, "PeppersRGB.bmp" dapat dikatakan berkualitas bagus. Citra "BaboonRGB.bmp" didapat nilai MSE=0,4998 sedangkan PSNR=62,3472 dB, "BaboonRGB.bmp" berkualitas bagus. Citra "LenaRGB.bmp" memiliki nilai MSE=0,5021 dengan PSNR=62,3031 dB, "LenaRGB.bmp" berkualitas bagus. Untuk citra "Lena.bmp" tipe citra adalah *grayscale* karenanya tidak dibagi atas RGB, hanya ada satu komponen warna (*grayscale*). Nilai MSE yang didapat MSE=0,5 dengan PSNR=61,9995 berdasarkan nilai PSNR tersebut "Lena.bmp" dikatakan bagus.

Berdasarkan hasil dari keempat obyek citra yang diujikan, didapat nilai rata-rata nilai MSE=0,5000 dan rata-rata nilai PSNR=61,9993 dB. Jika merujuk

pada nilai minimal PSNR=40 dB sebuah citra dikatakan bagus, yang telah dijelaskan sebelumnya. Maka semua citra uji proses steganografi berkualitas bagus. Sebagai pembanding, berikut penulis berikan contoh citra "LenaRGB.bmp" yang dapat dikatakan buruk secara piksel dan kasat mata ditunjukkan oleh gambar 4.11 beserta data MSE dan PSNR yang didapat berdasarkan citra tersebut dibawah

Gambar 4.11 LenaRGB.bmp dengan *noise Salt and Pepper*

Dari gambar 4.11, jika diproses dengan citra asli (LenaRGB.bmp) maka didapatkan data MSE dan PSNR sebagai berikut :

MSE *Red*	: 1.8285e+04	PSNR *Red*	: **-42,7255 dB**
MSE *Green*	: 1.6951e+04	PSNR *Green*	: **-41,9682 dB**
MSE *Blue*	: 1.5210e+04	PSNR *Blue*	: **-40,8841 dB**

Berdasarkan contoh di atas, didapat nilai PSNR dari komponen RGB (-42,7255 dB;-41,9682 dB;-40,8841 dB) PSNR tersebut jauh dibawah batas minimal suatu citra digital dikatakan berkualitas baik yaitu lebih dari atau sama dengan 40 dB. Dengan nilai PSNR tersebut citra dikatakan berkualitas buruk.

4.1.8 Uji Histogram Citra Digital Sebelum dan Sesudah Penyisipan

Berikut ini akan disajikan histogram citra digital untuk mengetahui perbedaan histogram citra sebelum dan sesudah penyisipan. Citra hasil penyisipan (*stego image)* yang digunakan adalah citra yang telah disisipi pesan sebanyak 100% dari daya tampung citra tersebut. Teks yang disipkan adalah ***the quick brown fox jumped over the lazy dog. 1234567890 THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG. 1234567890.*** Sedangkan citra uji yang digunakan adalah "PeppersRGB.bmp, BaboonRGB.bmp, LenaRGB.bmp dan Lena.bmp". Keempat citra tersebut terdiri dari 3 citra dengan format *true color* (RGB) yaitu : PepperRGB.bmp, BaboonRGB.bmp, serta LenaRGB.bmp dan 1 citra *grayscale* yaitu : Lena.bmp. Menampilkan histogram citra uji, penulis menggunakan fungsi yang sudah ada pada Matlab 2015b.

Untuk menampilkan histogram citra *true color* (RGB) menggunakan fungsi

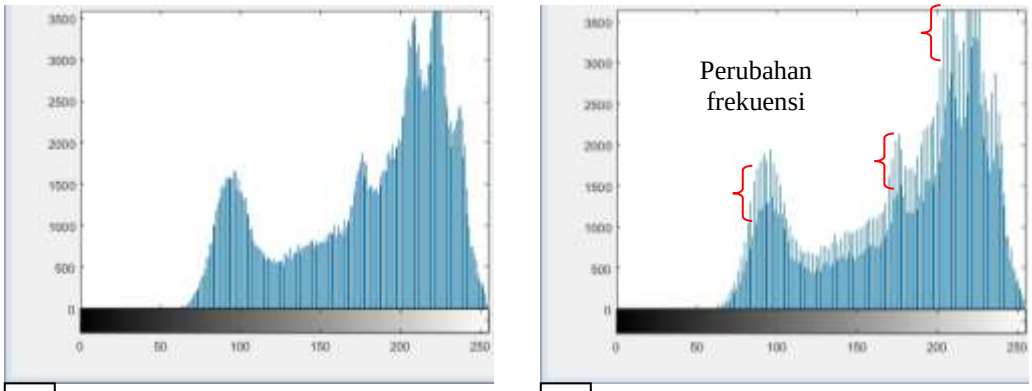
```
clc;
gbr=imread('Lenafull.bmp');
figure,imhist(gbr(:,:,1));
figure,imhist(gbr(:,:,2));
figure,imhist(gbr(:,:,3));
```

Sedangkan untuk menampilkan histogram citra *grayscale* menggunakan fungsi

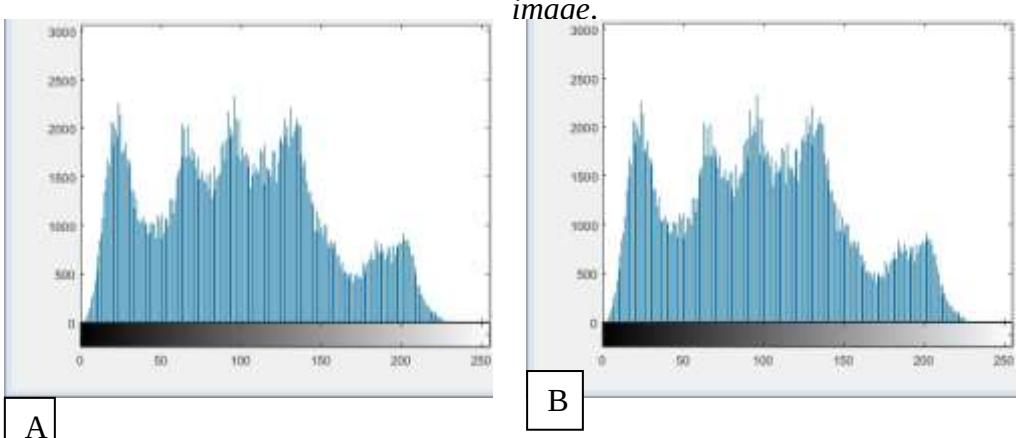
```
clc;
gbr=imread('Lenafull.bmp');
figure,imhist(gbr);
```

1. Uji menggunakan citra LenaRGB.bmp (*cover image*) dengan LenaRGBfull.bmp (*stego image*).

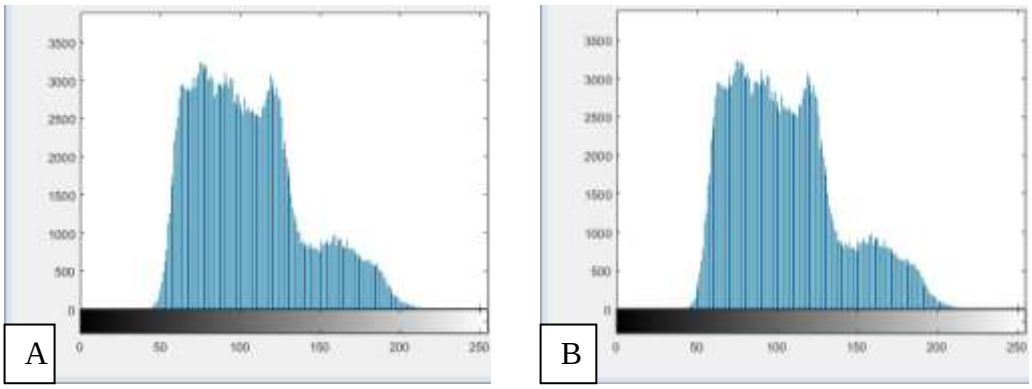
Gambar 4.12 Lena RGB format Bitmap ukuran 512x512 (Sumber : http://sipi.usc.edu/database/database.php?volume=misc)



Gambar 4.13 gambar A=Histogram komponen warna Red (R) citra *cover image* dan **gambar B**=Histogram komponen warna Red (R) citra *Stego imaae*.



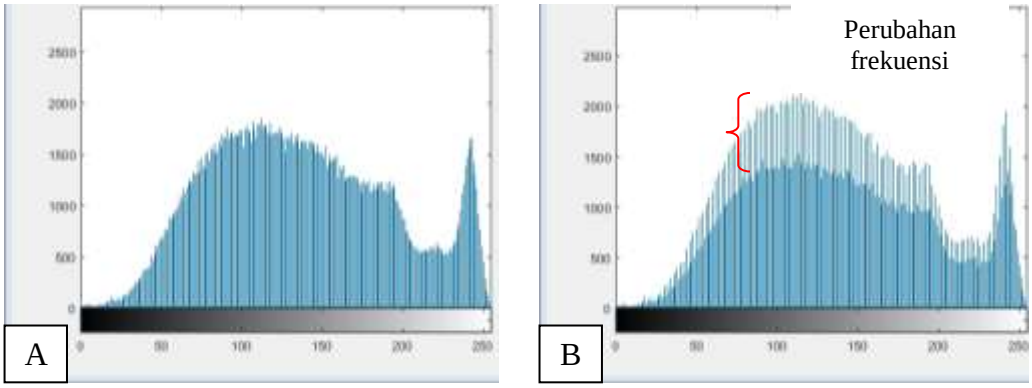
Gambar 4.14 gambar A=Histogram komponen warna Green (G) citra *cover image* dan **gambar B**=Histogram komponen warna Green (G) citra *Stego image*.



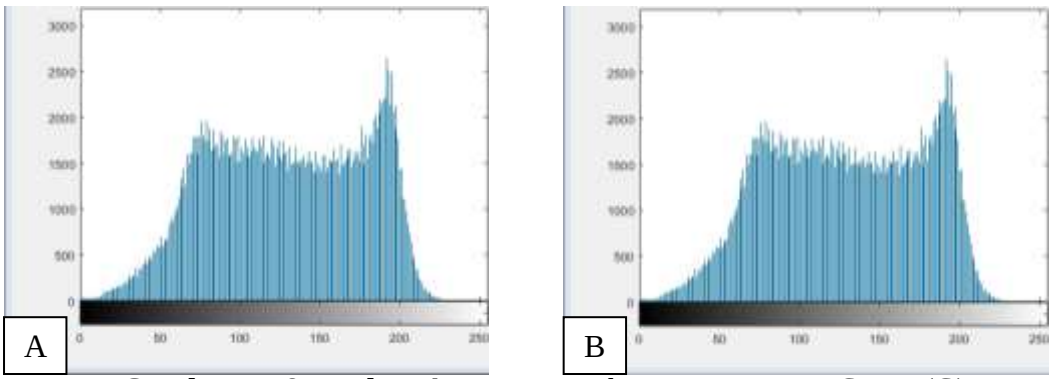
Gambar 4.15 gambar A=Histogram komponen warna Blue (B) citra *cover image* dan **gambar B=**Histogram komponen warna Blue (B) citra *Stego image*.

2. Uji menggunakan citra BaboonRGB.bmp (*cover image*) dengan BaboonRGBfull.bmp (*stego image*).

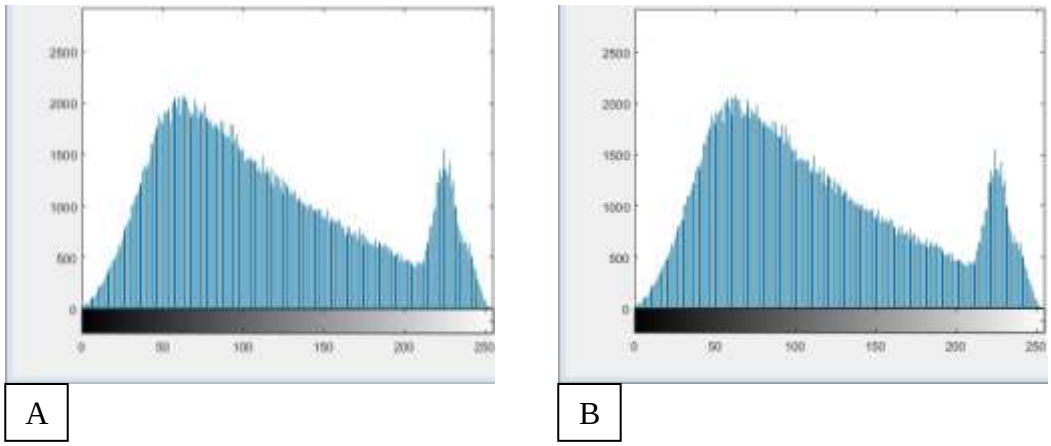
Gambar 4.16 Baboon RGB format Bitmap ukuran 512x512 (Sumber : http://sipi.usc.edu/database/database.php?volume=misc)



Gambar 4.17 gambar A=Histogram komponen warna Red (R) citra *Cover image* dan **gambar B=**Histogram komponen warna Red (R) citra *Stego image* .



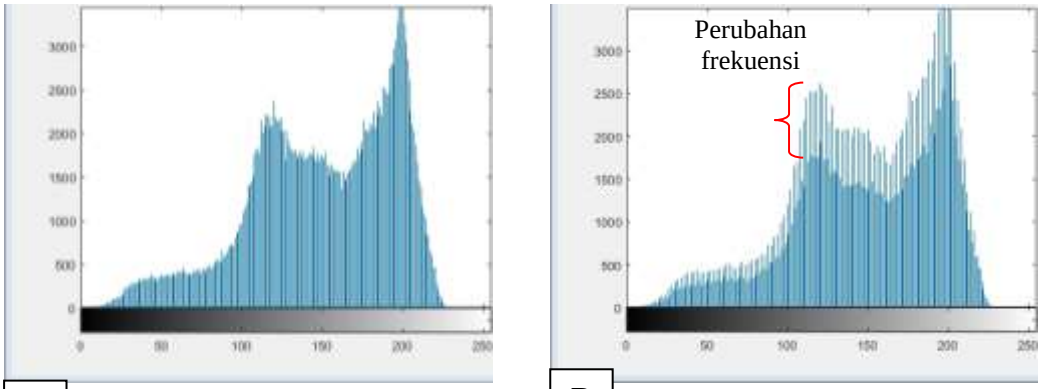
Gambar 4.18 gambar A=Histogram komponen warna Green (G) citra *cover image* dan **gambar B=**Histogram komponen warna Green (G) citra *Stego image*.



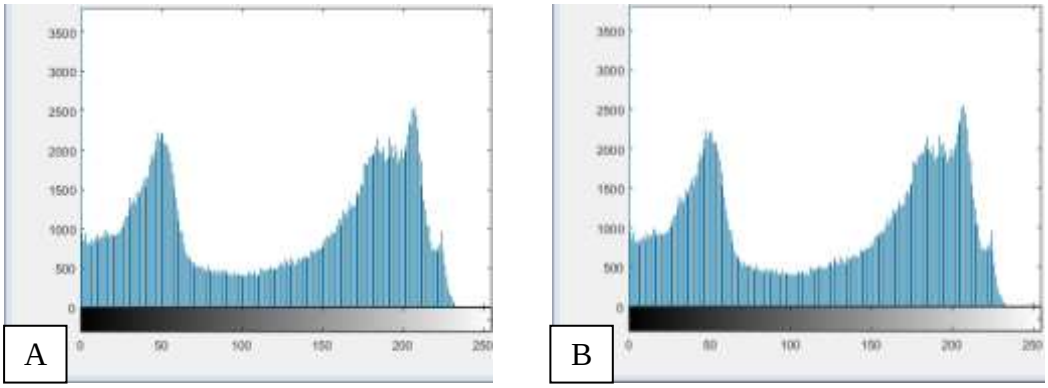
Gambar 4.19 gambar A= Histogram komponen warna Blue (B) citra *cover image* dan **gambar B=**Histogram komponen warna Blue (B) citra *Stego image*.

3. Uji menggunakan citra PeppersRGB.bmp (*cover image*) dengan PeppersRGBfull.bmp (*stego image*).

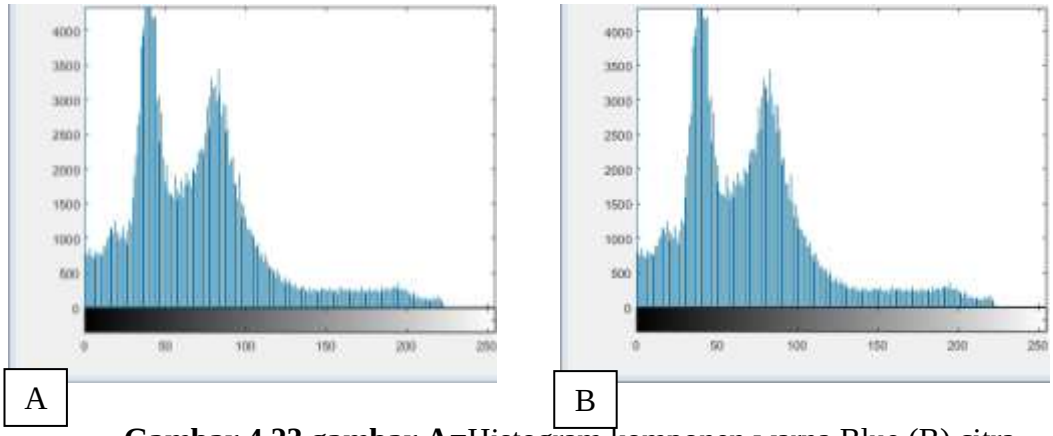
Gambar 4.20 Peppers RGB format Bitmap ukuran 512x512 (Sumber : http://sipi.usc.edu/database/database.php?volume=misc)



Gambar 4.21 gambar A=Histogram komponen warna Red (R) citra *Cover image* dan **gambar B=**Histogram komponen warna Red (R) citra *Stego image* .



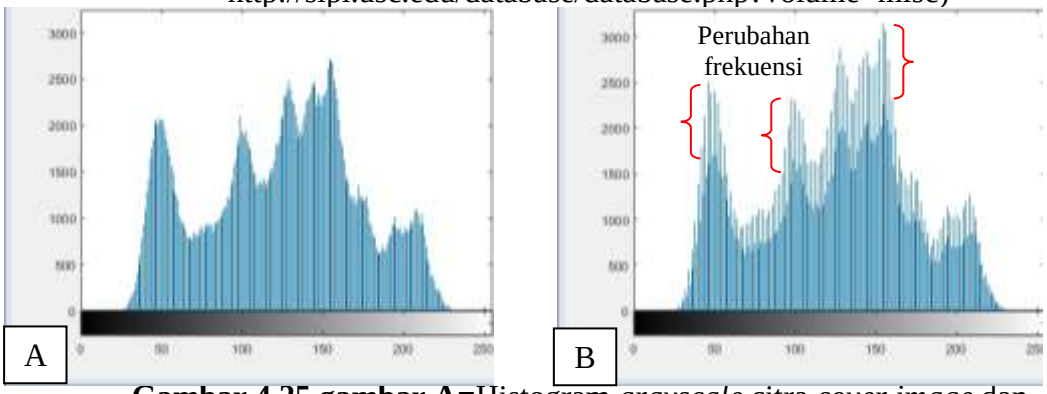
Gambar 4.22 gambar A=Histogram komponen warna Green (G) citra *cover image* dan **gambar B=**Histogram komponen warna Green (G) citra *Stego image*.



Gambar 4.23 gambar A=Histogram komponen warna Blue (B) citra *cover image* dan **gambar B=**Histogram komponen warna Blue (B) citra *Stego image*.

4. Uji menggunakan citra Lena.bmp (*cover image*) dengan Lenafull.bmp (*stego image*) format *grayscale*.

Gambar 4.24 Lena *grayscale* format Bitmap ukuran 512x512 (Sumber : http://sipi.usc.edu/database/database.php?volume=misc)



Gambar 4.25 gambar A=Histogram *grayscale* citra *cover image* dan **gambar B=**Histogram *grayscale* citra *Stego image*.

Berdasarkan hasil histogram citra yang digunakan untuk uji histogram tersebut diketahui bahwa untuk citra dengan jenis *true color* (RGB) terdapat perubahan frekuensi intensitas warna pada komponen warna *Red* (R) ditunjukkan dengan perubahan pada histogram sebelah kanan. Sedangkan untuk komponen yang lain *Green* (G) dan *Blue* (B) tidak mengalami perubahan frekuensi intensitas warna. Hal ini dikarenakan penyisipan karakter pesan hanya dilakukan pada komponen warna *Red* (R) sehingga mempengaruhi frekuensi intensitas warna. Meskipun demikian secara kasat mata citra tidak mengalami perubahan yang signifikan.

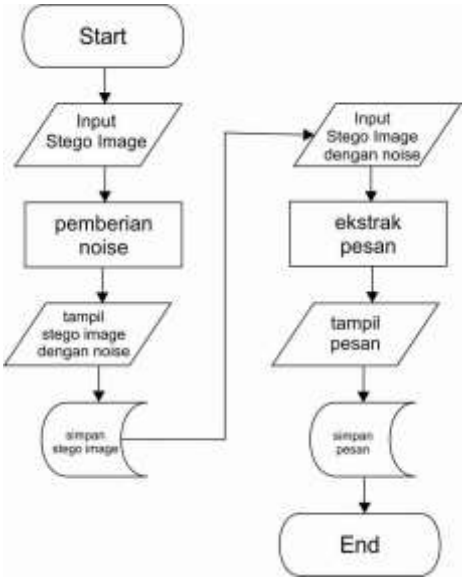
Citra dengan jenis *grayscale* dengan perlakuan yang sama menunjukkan perubahan frekuensi intensitas warna citra. Uji ini dilakukan untuk membuktikan bahwa proses steganografi LSB memang benar-benar memanipulasi citra digital sehingga terjadi perubahan pada tingkat komponen piksel citra uji.

4.1.9 Pemberian *Noise* Gaussian Pada *Stego Image*

Sebagai pengujian selanjutnya adalah menguji *stego image* dengan pemberian *noise* Gaussian. Pengujian ini dilakukan untuk mengetahui apakah pesan yang tersimpan dalam *stego image* masih dapat diekstrak sempurna setelah diberikan *noise.* Pengujian dilakukan pada 4 citra uji (*stego image*) yang telah disisipi sebanyak 100% dari jumlah maksimal kapasitas penyimpanan LSB. Pemberian *noise* dilakukan dengan program Matlab serta menggunakan fungsi yang sudah ada dalam program Matlab 2015b yaitu sebagai berikut :

```
proyek=guidata(gcbo);
I=get(proyek.imagesblm,'Userdata');
gbr2=imnoise(I,'gaussian',0.00001);
```

Alur yang dilalui dalam pengujian ini adalah sebagai berikut :



Gambar 4.26 *Flowchart* pengujian pemberian *noise*

Sebagai langkah awal, program akan memilih citra *stego image* yang akan diberi *noise*. Setelah itu program akan memperoses citra untuk *noise*. Selesai proses noise, citra akan diekstrak untuk mengetahui bagaimana pesan setelah diekstrak, pakah masih sama dengan pesan awal atau sudah rusak.

Berikut ini ditampilkan hasil pemberian *noise* yang dilakukan pada *stego image* .

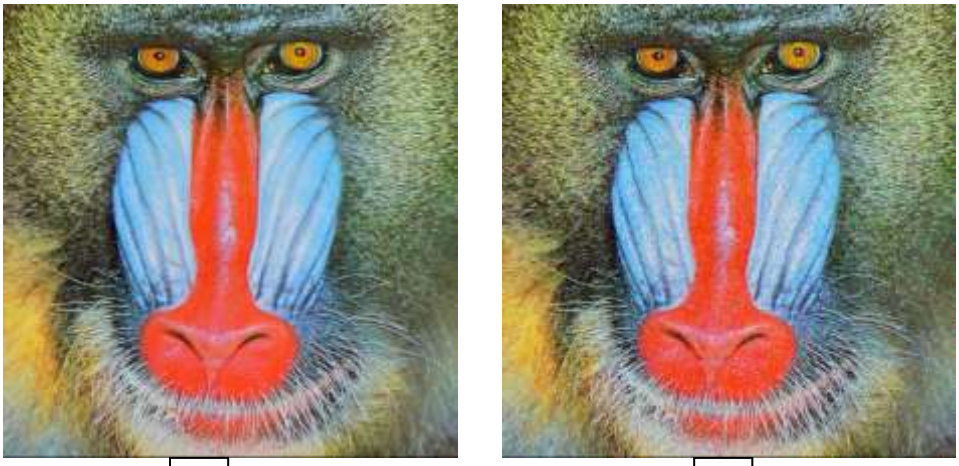
1. Percobaan dengan menggunakan *stego image* Lena RGB.

A B

Gambar 4.27 Gambar (A) menunjukkan *stego image* sebelum pemberian *noise*. Gambar (B) menunjukkan *stego image* setelah pemberian *noise*

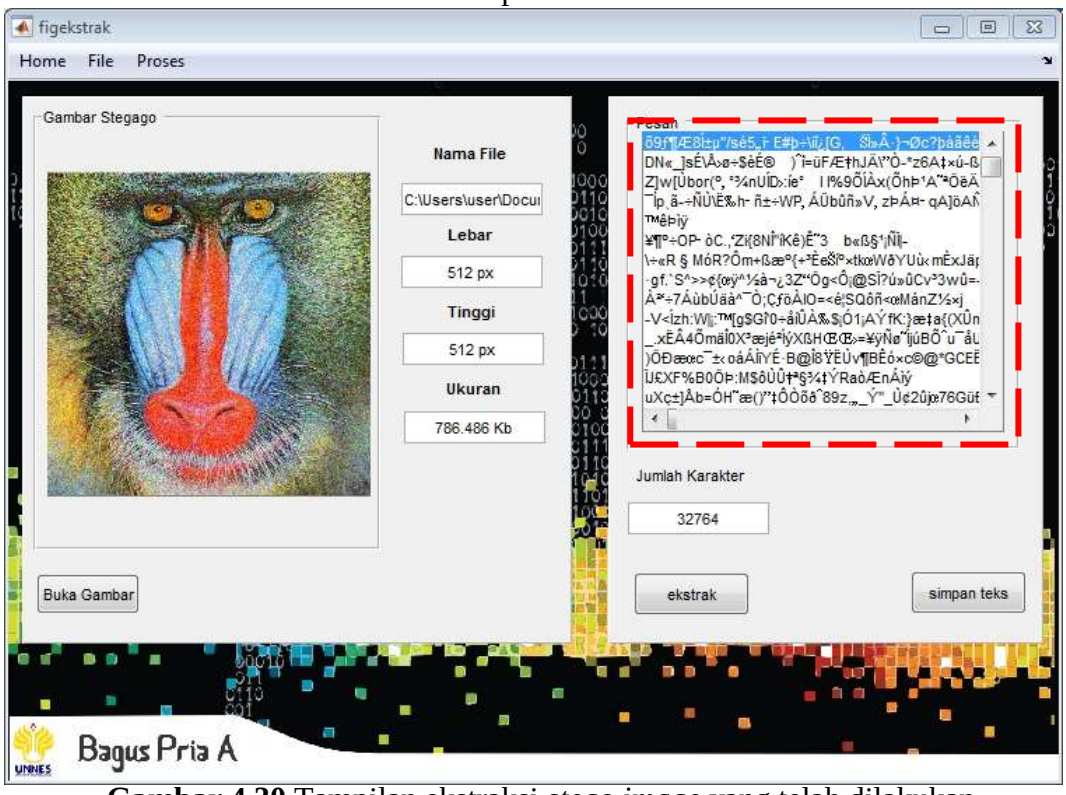
Gambar 4.28 Tampilan ekstraksi *stego image* yang telah dilakukan pemberian *noise*.

2. Percobaan dengan menggunakan *stego image* Baboon RGB.



A B

Gambar 4.29 Gambar (A) menunjukkan *stego image* sebelum pemberian *noise*. Gambar (B) menunjukkan *stego image* setelah pemberian *noise*

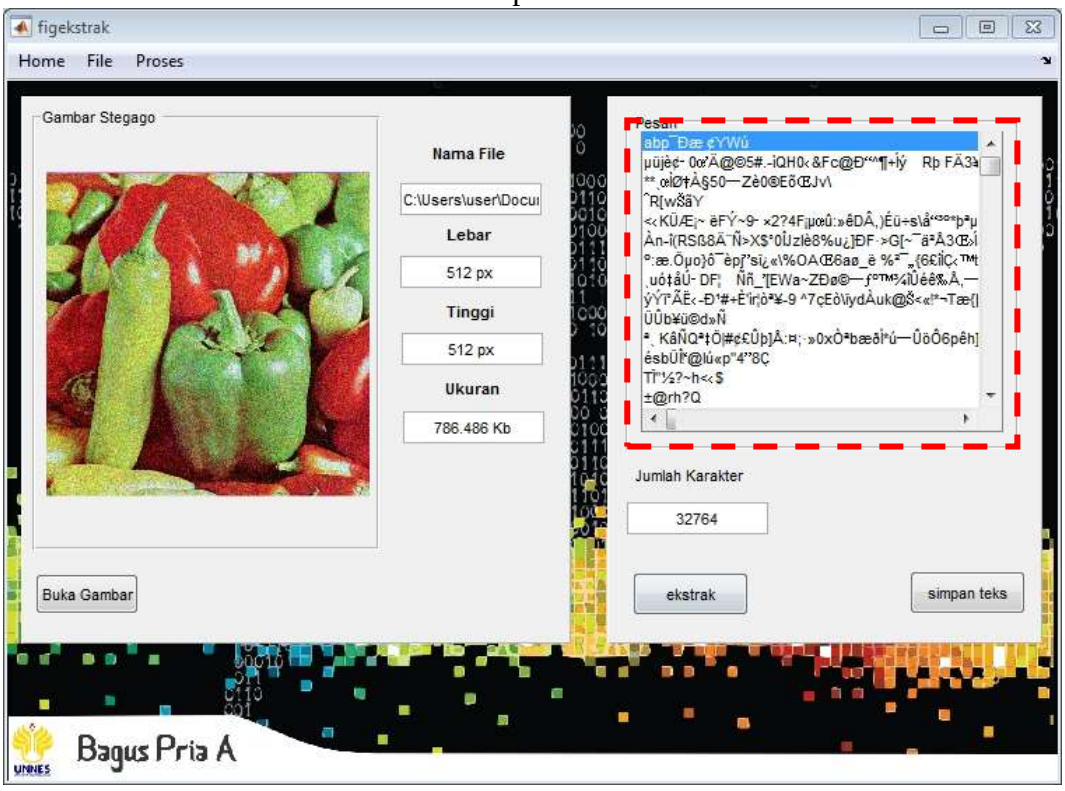


Gambar 4.30 Tampilan ekstraksi *stego image* yang telah dilakukan pemberian *noise*.

3. Percobaan dengan menggunakan *stego image* Peppers RGB.

A B

Gambar 4.31 Gambar (A) menunjukkan *stego image* sebelum pemberian *noise*. Gambar (B) menunjukkan *stego image* setelah pemberian *noise*

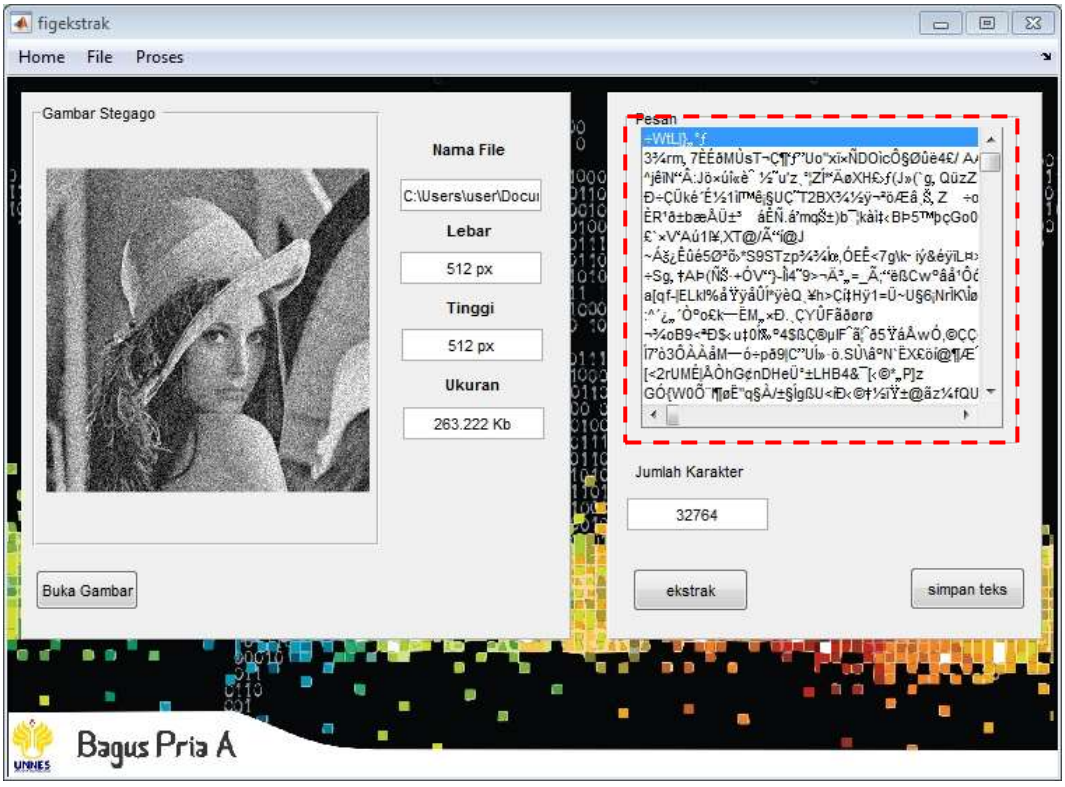


Gambar 4.32 Tampilan ekstraksi *stego image* yang telah dilakukan pemberian *noise*.

4. Percobaan dengan menggunakan *stego image* Lena Grayscale.

A B

Gambar 4.33 Gambar (A) menunjukkan *stego image* sebelum pemberian *noise*. Gambar (B) menunjukkan *stego image* setelah pemberian *noise*



Gambar 4.34 Tampilan ekstraksi *stego image* yang telah dilakukan pemberian *noise*.

Berdasarkan pengujian yang dilakukan diketahui bahwa *stego image* yang dilakukan pemberian *noise* tidak dapat menampilkan isi pesan yang telah disisipkan secara utuh. Hal itu dibuktikan dengan hasil ekstraksi (ditunjukkan dengan kotak merah gambar 4.28, 4.30, 4.32, dan 4.34) dengan keterangan pesan dapat diekstrak dengan jumlah karakter yang sama, namun isi pesan sudah rusak (tidak sama sebagaimana pesan awal yang disisipkan).

3.7 Pembahasan

Steganografi pada citra digital merupakan salah satu topik yang diteliti dalam bidang pengolahan citra digital. Pengolahan citra digital memiliki berbagai macam tujuan, diantaranya menyembunyikan informasi rahasia, hal itu lah yang menjadi tujuan utama steganografi. Selanjutnya, yang menjadi fokus utama penelitian adalah bagaimana menyembunyikan data secara maksimal dan keberadaannya tidak terdeteksi indera penglihatan serta data tersebut dapat diambil kembali. Selain itu, mengatasi keterbatasan kapasitas penyimpanan citra digital.

Penelitian penerapan metode *image scaling* pada steganografi LSB dilakukan dengan membuat dan mengembangkan aplikasi steganografi. Pembuatan aplikasi dilakukan dengan menggunakan program Matlab versi R2015b. Pengolahan citra pada tingkat komponen piksel merupakan landasan bagi peneliti dalam merancang algoritma sistem. Pembuatan GUI digunakan untuk dapat menampilkan hasil dari proses steganografi yang dilakukan.

Pengujian citra pada penelitian ini dilakukan dengan pengujian *black box* serta perlakuan khusus pada citra digital. Citra digital yang digunakan hanya citra

dengan format bitmap, terdiri dari 3 buah citra dengan jenis *true color* dan 1 buah citra dengan jenis *grayscale*. Penyisipan pesan kedalam citra digital menggunakan metode LSB (*Least Significant Bit*). Sedangkan untuk meningkatkan kapasitas penyimpanan karakter, digunakan *image scaling*. Hasil pengujian steganografi ini adalah citra digital yang didalamnya sudah disisipkan pesan/teks rahasia.

Citra hasil akan diproses kembali untuk mengekstraksi pesan yang telah disisipkan. Selain itu, citra hasil diujikan kepada responden, untuk mengetahui apakah citra hasil memiliki perbedaan yang signifikan dari citra awal secara kasat mata. Berdasarkan pengujian responden, didapatkan bahwa rata-rata nilai yang diperoleh adalah 4,5 dengan penjelasan bahwa citra akhir berkualitas baik, dengan kesamaan antara 70%-90% dari citra awal. Proses perhitungan MSE (*Mean Square Error*) dan PSNR (*Peak Signal to Noise Ratio*) dilakukan pada citra hasil steganografi untuk mengetahui kualitas citra. Nilai yang didapat dari perhitungan MSE dan PSNR pada keempat citra rata-rata adalah MSE=0,5 dan PSNR=61,9993 dB, dari nilai tersebut dapat digolongkan bahwa citra hasil berkualitas baik. Setelah pengujian kualitas pada citra dilakukan pengujian ketahanan pesan yang disisipkan dengan pemberian *noise* pada citra hasil. Dari hasil pemberian *noise* pada citra hasil, pesan yang disimpan dapat diekstrak kembali namun sudah tidak sama dengan pesan awal yang dimasukkan, meskipun jumlah karakter masih sama.

Pada penelitian steganografi lain yang dilakukan Champakamala, dkk (2010) yang berjudul "*Least Significant Bit Algorithm for Image Steganography*" dengan algoritma LSB berhasil digunakan untuk menyisipkan citra digital

kedalam citra digital lain. Selain itu, penelitian yang dilakukan oleh Champaka dkk berhasil mensimulasikan steganografi tersebut dengan *hardware* ARM7TDMI (*processor*) serta GSM 900 (*modem*). Jika dibandingkan dengan penelitian yang penulis lakukan, Champaka menggunakan citra digital sebagai obyek yang disisipkan kedalam *cover image.* Sedangkan penulis menggunakan teks sebagai obyek yang disisipkan. Dalam penelitian Champaka tidak menyinggung kapasitas penyimpanan *cover image* dalam steganografi. Sehingga penelitian tersebut hanya menggunakan citra digital dengan ukuran lebih kecil atau sama dengan ukuran *cover image.* Sedangkan penelitian yang dilakukan penulis membahas bagaimana memberikan solusi keterbatasan kapasitas penyimpanan suatu *cover image* dalam proses steganografi yaitu dengan menambahkan algoritma *image scaling.* Keduanya sama-sama menggunakan algoritma LSB dalam proses steganografi yang dilakukan.

Penelitian oleh Shilpa Gupta, dkk (2012) yang berjudul "*Enhanced Least Significant Bit Algorithm for Image Steganography*" menggunakan algoritma LSB. Penelitian tersebut fokus pada pembahasan sisi LSB dari steganografi. Secara *default* steganografi LSB memanipulasi komponen piksel (RGB) pada setiap piksel citra digital (*cover image*). Manipulasi dapat dilakukan pada komponen R, G atau B bahkan dapat ketiga komponen. Dalam memanipulasi ke tiga komponen piksel sekaligus, daya tampung citra digital akan semakin banyak, dapat mencapai 3 kali lipat jika dibandingkan dengan yang hanya memanipulasi pada satu komponen saja. Namun, sebagai konsekwensinya, akan menurunkan kualitas citra. Hal itu dikarenakan timbulnya lebih banyak *noise* pada citra digital.

Banyaknya *noise* pada citra digital akan mengakibatkan nilai MSE semakin tinggi sedangkan PSNR citra akan semakin rendah, hal itu akan berakibat berkurangnya kualitas citra. Pada penelitiannya, Gupta dkk memberikan solusi dengan memanipulasi 3 bit terakhir pada salah satu komponen piksel (R,G/B). Dibandingkan dengan penelitian yang dilakukan oleh peneliti, Gupta meningkatkan kapasitas penyimpanan pesan pada *cover image* dengan melakukan manipulasi pada 3 bit terakhir *cover image.* Sedangkan peneliti tetap menggunakan sistem memanipulasi 1 bit terakhir pada salah satu komponen piksel. Sedangkan untuk meningkatkan kapasitas digunakan algoritma *image scaling.*

Dalam perkembangannya steganografi dapat di kombinasikan dengan kriptografi untuk meningkatkan keamanan pesan yang disisipkan. Seperti penelitian yang dilakukan oleh Vikas Tyagi, dkk (2012) dengan judul "*Image Steganography Using Least Signoficant Bit with Cryptography*". Penelitian tersebut menggunakan algoritma LSB dalam proses steganografinya. LSB yang digunakan adalah memanipulasi 1 bit terakhir pada komponen piksel (RGB). Dari segi peningkatan, penelitian ini lebih fokus dalam peningkatan keamanan pesan yang disisipkan. Sehingga Vikas Tyagi dkk mengkombinasikan dengan kriptografi agar isi pesan lebih sulit untuk diketahui orang lain. Jika dibandingkan dengan penelitian penulis, keduanya sama-sama menggunakan algoritma LSB dan sama-sama meningkatkan, yang membedakan adalah Vikas Tyagi meningkatkan keamanan isi pesan yang disipkan dengan mengkombinasikan steganografi LSB

dengan kriptografi. Sedangkan penulis menigkatkan kapasitas penyimpanan karakter pesan dengan *image scaling.*

Selanjutnya, dalam penelitian "Metode *Least Significant Bit (LSB)* dan *End Of File (EOF)* Untuk Menyisipkan Teks ke dalam Citra *Grayscale*" oleh Krisnawati membahas steganografi dengan mengimplementasikan 2 metode, yaitu LSB dan EOF. Untuk metode LSB dilakukan dengan mensubstitusi bit terakhir pada komponen piksel. Jika metode LSB ini digunakan maka ukuran citra tidak akan berubah, namun kapasitas penyimpanannya terbatas dan bergantung ukuran citra digital. Sedangkan metode EOF jika digunakan maka tidak ada batasan banyaknya pesan yang dapat disisipkan kedalam citra digital. Pesan dimasukkan ke dalam citra digital dengan meletakkannya pada baris akhir citra digital. Karena bit dari pesan tidak disubstitusikan, melainkan ditambahkan maka metode EOF akan mempengaruhi ukuran citra digital. Semakin banyak pesan yang dimasukkan, maka akan semakin besar *file size* citra digital tersebut. Jika dibandingkan dengan penelitian penulis, penulis menggunakan *image scaling* untuk mengatasi keterbatasan penyisipan karakter pada LSB. Sedangkan Krisnawati untuk menyiasati keterbatasan LSB, digunakan metode EOF dengan segala kelebihan dan kekurangan metode tersebut.

Penelitian Lain yang dilakukan oleh Budi Prasetyo (2013) dengan judul "Kombinasi Steganografi *Bit Matching* dan Kriptografi DES Untuk Pengamanan Data" memberikan pandangan untuk menggunakan metode lain dalam steganografi. Penelitian tersebut menggunakan algoritma *bit matching* untuk proses steganografi yang dilakukan. Sebagai gambaran proses dari metode

tersebut adalah mencocokkan bit pesan dengan bit pada citra digital sebagai pembawa pesan tersebut. Berbeda dengan LSB yang mensubstitusikan bit pesan kedalam bit citra digital, *bit matching* tidak mengubah bit pada citra digital melainkan menyimpan indeks dimana bit pada citra digital cocok dengan bit pesan yang akan dimasukkan. Keuntungan dari metode ini adalah, citra digital tidak mengalami perubahan sedikitpun, sehingga nilai MSE bernilai = 0 sedangkan PSNR bernilai tinggi. Kapasitas penyimpanan pesan lebih banyak jika dibandingkan dengan LSB yang hanya seperdelapan dari total ukuran dimensi citra digital. Untuk peningkatan keamanan, penelitian Budi Prasetiyo mengkombinasikan dengan kriptografi DES sehingga, pesan yang akan dimasukkan ke dalam citra digital dienkripsi terlebih dahulu dengan algoritma DES sampai beberapa permutasi. Selanjutnya pesan diubah kedalam bentuk biner dan disesuaikan dengan biner pada citra digital untuk didapatkan indeks lokasi yang cocok untuk pembangkitan pesan kembali. Dibandingkan dengan penelitian yang penulis lakukan, penelitian milik Budi Prasetiyo tidak mempengaruhi citra digital sedikitpun bahkan dalam setiap bit citra digital. Sehingga kualitas citra digital sebelum dan sesudah proses steganografi adalah sama. Sedangkan penelitian yang dilakukan penulis mempengaruhi citra digital hanya pada bit terakhir setiap piksel citra digital, serta untuk meningkatan kapasitas penyisipan pesan digunakan *image scaling.* Penelitian milik Budi Prasetiyo tingkat keamanannya lebih terjamin jika dibandingkan dengan yang penulis lakukan, karena penulis tidak mengkombinasikan dengan kriptografi.

BAB V

SIMPULAN DAN SARAN

1.8 Simpulan

1. Penelitian ini menghasilkan sebuah aplikasi steganografi dengan menggunakan algoritma LSB (*Least Significant Bit*) yang dapat digunakan untuk menyisipkan pesan misalnya pesan rahasia ke dalam citra digital atau biasa disebut *cover image.* Aplikasi ini dibuat dengan menggunakan program Matlab R2015b dan hanya dapat dijalankan dengan menggunakan program Matlab. Berkenaan dengan terbatasnya kapasitas penyisipan pesan pada metode LSB, penulis mengkombinasikannya dengan metode *image scaling. Image scaling* memiliki peran yang strategis dalam steganografi ini, yaitu ekspansi wadah yang mana akan berpengaruh pada ukuran dimensi citra digital sehingga kapasitas penyisipan akan semakin meningkat.

   Berdasarkan pengujian yang telah dilakukan kepada responden, rata-rata melihat bahwa citra digital sebelum dan sesudah proses steganografi memilki kualitas bagus dengan tingkat kesamaannya antara 90-100% *stego image* dengan *cover image*. Hal itu diperkuat dengan pengujian parameter nilai MSE dan PSNR. Pengujian terhadap 4 citra uji menghasilkan MSE yang rendah, yaitu 0,5 sedangkan PSNR yang didapatkan adalah 61,99 dB. Angka tersebut mengindikasikan bahwa *stego image* sebagai pembawa pesan

berkualitas baik dan minim adanya *noise.* Sehingga keberadaan pesan dalam citra digital tidak mudah diketahui.

2. Menggunakan aplikasi steganografi ini, citra digital telah mampu menampung setiap karakter pesan yang diinputkan oleh pengguna. Jumlah pesan yang diinputkan pengguna harus disesuaikan dengan kapasitas maksimal penyisipan pada citra bitmap yang digunakan. Pesan yang diinputkan oleh pengguna dapat dibangkitakn kembali melalui proses ekstraksi secara utuh dan disimpan dalam format .txt.

3. Menggunakan algoritma LSB memiliki kekurangan yaitu daya tampung penyisipan karakter citra digital yang digunakan hanya sebesar seperdelapan dari total ukuran dimensi citra digital. Untuk mengatasi hal tersebut, penulis menggunakan *image scaling* untuk meningkatkan ukuran citra digital. Meningkatnya ukuran citra digital akan mempengaruhi jumlah piksel pada citra digital tersebut, sehingga daya tampung citra digital juga semakin meningkat.

1.9 Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa saran yang dapat digunakan untuk penelitian lanjutan. Diantaranya adalah :

1. Aplikasi perlu dikembangkan dalam hal kepraktisannya, sehingga aplikasi dapat bediri sendiri tanpa harus menggunakan aplikasi lain untuk menjalankannya.

4. Steganografi LSB perlu dikembangkan untuk menemukan modifikasi-modifikasi algoritma yang lain sehingga algoritma ini akan semakin efektif dalam steganografi. Misalnya LSB yang diberlakukan pada ketiga komponen piksel (RGB) yang saat ini peneliti hanya memberlakukan algotima LSB pada komponen piksel R saja.

5. Pengembangan steganografi dengan menggunakan metode yang lain dan mungkin dapat dikombinasikan. Sehingga mengatasi setiap kelemahan yang ada. Misalnya dikombinasikan dengan kriptografi untuk meningkatkan keamanan pesan yang disisipkan.

6. Pengembangan aplikasi dengan menggunakan *engine* yang lain, sehingga aplikasi dapat berjalan multi *platform.*

DAFTAR PUSTAKA

Afyenni, Rita. 2014. Perancangan Data Flow Diagram Untuk Sistem Informasi Sekolah (Studi Kasus pada SMA Pembangunan Laboraturium UNP). Jurnal TEKNOIF Vol. 2 (1). 35.

Ariyus, D. 2007. *Keamanan Multimedia.* Yogyakarta : Andi.

Astuti, Maria, dkk. 2012. Pembuatan Perangkat Lunak Untuk Workflow Pengelolaan Surat Dinas Bagian Surat Keluar di Pemerintah Kabupaten Buton Utara. Jurnal Teknik ITS Vol.1. 25-29.

Champakamala.B.S. dkk. *Least Significant Bit Algorithm for Image Steganography*. *International Journal of Advanced Computer Technology (IJACT)* Vol. 3 (4).

Chan, Chi-Kwong and Cheng, L.M. 2003. *Hiding Data in Image by Simple LSB Substitution*. *The Journal of Pattern Recognition Society* 37 : 469-474

Gupta,Shilpa, dkk. 2012. *Enhanced Least Significant Bit Algorithm for Image Steganography. International Kournal of Computational Engineering & Management* Vol. 15 (4).

Hakim, Muhammad. 2012. Studi dan Implementasi Steganografi Metode LSB dengan Preprocessing Kompresi data dan Ekspansi Wadah. Bandung: ITB.

Katzenbeisser, Stefan and Petitcolas, Fabien A.P. 2002. *Information Hiding Techniques for Stegaography and Digital Watermarking. London:Artech House*.

Krisnawati. 2008. Metode Least Significant Bit (LSB) dan End Of File (EOF) Untuk Menyisipkan Teks Ke Dalam Citra Grayscale. Makalah diseminarkan dalam Seminar Nasional Informatika. UPN "Veteran" Yogyakarta.

Munir, R. 2004. Pengantar Kriptografi. Bandung: ITB.

Nugraha, Erdiansyah Fajar. 2011. Meningkatkan Kapasitas Pesan yang Disisipkan dengan Metode Redundant Pattern Encoding. Bandung : ITB.

Paulus, Eric dan Natalia, Yessica. 2007. Cepat Mahir GUI Matlab. Yogyakarta: Andi.

Piarsa, I Nyoman. 2011. Steganografi Pada Citra JPEG dengan Metode Sequential dan Spreading. Lontar Komputer Vol. 2 (1).

Prasetiyo, Budi. 2013. Kombinasi Steganografi Bit Matching dan Kriptografi DES Untuk Pengamanan Data. Tesis. Universitas Diponegoro, Semarang

Pressman, R.S. 2012. Rekayasa Perangkat Lunak Pendekatan Praktisi. Yogyakarta : Andi.

Prihanto, Agus. Dkk. 2010. Peningkatan Kapasitas Informasi Tersembunyi Pada Image Steganography dengan Menggunakan Teknik Hybrid. Makalah seminar nasional Pascasarjana Teknik Informatika ITS.

Provos, N. and Honeyman, P. 2003.*Hide and Seek: An Introduction to Steganography. IEEE Computer Society: Security & Privacy*.

Rakhmat, Basuki dan Fairuzabadi, M. M.Kom. 2010. Steganografi Menggunakan Metode Least Significant Bit dengan Kombinasi Algoritma Kriptografi Vigenere dan RC4. Jurnal Dinamika Informatika Vol.5, 5-6.

Sachs, Jonathan. 2001. *Image Resampling. Digital Light & Color*. Cambridge.

Schneier, Bruce. 1996. *Applied Cryptography 2nd Edition*. Illinois.

Syakura, Shauma H. 2010. Studi Analisis Perbandingan Metode Steganalisis Terhadap LSB Image Steganography. Bandung: ITB.

Tyagi, Vikas,dkk. 2012. *Image Steganography Using Least Significant Bit with Cryptography. Journal of Global Research in Computer Science* Vol 3 (3).

Wijaya, Alston Evan, dkk. 2012. Implementasi Steganografi Untuk Penyembunyian Pesan pada Video dengan Metode LSB. Jurnal Teknik Informatika Vol 1.

LAMPIRAN

Lampiran 1 : Surat Penetapan Dosen Pembimbing Skripsi

KEPUTUSAN
DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
Nomor: 190/FT.H46.1/2015
Tentang
PENETAPAN DOSEN PEMBIMBING SKRIPSI/TUGAS AKHIR SEMESTER GASAL/GENAP
TAHUN AKADEMIK 2014/2015

Menimbang: Bahwa untuk mempelancar mahasiswa Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Fakultas Teknik membuat Skripsi/Tugas Akhir maka perlu menetapkan Dosen-dosen Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Fakultas Teknik UNNES untuk menjadi pembimbing.

Mengingat:
1. Undang-undang No 20 Tahun 2003 tentang Sistem Pendidikan Nasional (Tambahan Lembaran Negara RI No 4301, penjelasan atas Lembaran Negara RI Tahun 2003 Nomor 78)
2. Peraturan Rektor No 21 Tahun 2011 tentang Sistem Informasi Skripsi UNNES
3. SK Rektor UNNES No. 164/O/2004 tentang Pedoman penyusunan Skripsi/Tugas Akhir Mahasiswa Strata Satu (S1) UNNES.
4. SK Rektor UNNES No 162/O/2004 tentang penyelenggaraan Pendidikan UNNES

Menimbang: Usulan Ketua Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Tanggal 19 Maret 2015

MEMUTUSKAN

Menetapkan:

PERTAMA: Menunjuk dan menugaskan kepada:

Nama	Dr. Hari Wibawanto, M.T
NIP	196501071991021001
Pangkat/Golongan	IV/A
Jabatan Akademik	Lektor Kepala
Sebagai	Pembimbing

Untuk membimbing mahasiswa penyusun skripsi/Tugas Akhir:

Nama	BAGUS PRIA AMBADA
NIM	5302411180
Jurusan/Prodi	Teknik Elektro/Pend. Teknik Informatika dan Komputer
Topik	PENERAPAN METODE LSB DAN IMAGE STRETCHING DALAM STEGANOGRAFI

KEDUA: Keputusan ini mulai berlaku sejak tanggal ditetapkan.

DITETAPKAN DI SEMARANG
PADA TANGGAL 19 Maret 2015
DEKAN

Dr. Muhammad Harlanu, M.Pd
NIP 196602151991021001

Tembusan
1. Pembantu Dekan Bidang Akademik
2. Ketua Jurusan
3. Petinggal

5302411180
FM-03-AKD 24/Rev. 00

Lampiran 2 : Surat Tugas Panitia Ujian Sarjana

KEMENTERIAN RISTEK DAN PENDIDIKAN TINGGI
UNIVERSITAS NEGERI SEMARANG
FAKULTAS TEKNIK
Gedung E11 Lt 2, Kampus Sekaran, Gunungpati, Semarang 50229
Telepon: 024 8508104
Laman: www.ft.unnes.ac.id, surel:

UNNES

No : 8995/UN37.1.5/DT/2016
Lamp :
Hal : Surat Tugas Panitia Ujian Sarjana

Dengan ini kami tetapkan bahwa ujian Sarjana Fakultas Teknik UNNES untuk jurusan Teknik Elektro adalah sebagai berikut :

I. Susunan Panitia Ujian
a. Ketua : Dr. Ing. Dhidik Prastiyanto, S.T., M.T.
b. Sekretaris : Ir. Ulfah Mediaty Arief, M.T.
c. Pembimbing Utama : Dr. Hari Wibawanto, M.T.
d. Penguji : 1. Dr. Ing. Dhidik Prastiyanto, S.T., M.T.
2. ANGGRAINI MUKWINDA, S.T., M.Eng.

II. Calon yang diuji
Nama : BAGUS PRIA AMBADA
NIM/Jurusan/Program Studi : 5302411180/Teknik Elektro/Pendidikan Teknik Informatika dan Komputer, S1
Judul Skripsi : Implementasi Metode Image Scaling untuk meningkatkan Penyisipan Informasi Media Cover pada steganografi LSB

II. Waktu dan Tempat Ujian
Hari/Tanggal : Selasa / 22 November 2016
Jam : 13.00.00
Tempat : E11 R Sidang 2
Pukulan :

Semarang, 21 November 2016
Dekan,
Dr. Ing. Dhidik Prastiyanto, S.T., M.T.
NIP. 196911301994031001

Tembusan :
1. Ketua Jurusan TEKNIK ELEKTRO
2. Calon yang diuji

Lampiran 3 : Berkas Hasil Penilaian Responden

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
19

Nama : Risalatil Laeli

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp				✓	
3.bmp		✓			
4.bmp					✓

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, 2 Agustus 2016

Risalatil Laeli

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
18

Nama : IFAH NUR AZIZAH

Pekerjaan : MAHASISWA

Obyek	skor				
	1	2	3	4	5
1.bmp				✓	
2.bmp					✓
3.bmp				✓	
4.bmp				✓	

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

IFAH NUR A

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
17

Nama : Eka Yulianti Fajlin

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp				✓	
3.bmp					✓
4.bmp				✓	

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

Eka YF

Uji *Imperceptibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN: 16

Nama : Rahmaniah Anna Sari

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1 bmp				✓	
2 bmp					✓
3 bmp				✓	
4 bmp				✓	

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

Rahmaniah Anna S

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
15

Nama : Ayu Novita F

Pekerjaan : Pelajar

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp					✓
3.bmp					✓
4.bmp					✓

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

(Ayu N.F.)

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN: 14

Nama : Siti Nurkhayati

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp					✓
3.bmp					✓
4.bmp					✓

*Berilah tanda check (✓) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

Siti Nurkhayati

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
13

Nama : Yohana Ella Kristina

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp				✓	
3.bmp				✓	
4.bmp				✓	

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, 1 Agustus 2016

Yohana Ella K.

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
12

Nama : Revita Andhya Pratiwi

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp				✓	
3.bmp				✓	
4.bmp					✓

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

Revita A.P

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
11

Nama : Yoke Ana Marlina

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp				✓	
3.bmp					✓
4.bmp				✓	

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

Yoke A.M.

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN
4

Nama : Anton Wicaksana

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					√
2.bmp					√
3.bmp				√	
4.bmp				√	

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, Agustus 2016

Anton Wicaksana

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN: 3

Nama : M. Nur Shokroni

Pekerjaan : Mahasiswa

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp					✓
3.bmp					✓
4.bmp					✓

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, 1 Agustus 2016

Uji *Imperectibility* HVS (*Human Visual System*)
Metode *Mean Opinion Score*

NOMOR RESPONDEN: 2

Nama : Eko Budi Utomo

Pekerjaan : Mahasiswa PTO

Obyek	skor				
	1	2	3	4	5
1.bmp					✓
2.bmp					✓
3.bmp					✓
4.bmp					✓

*Berilah tanda check (√) pada kolom nilai sesuai rubrik dibawah ini

Skor	Kualitas Citra	Keterangan
5	Sangat Bagus	Kesamaan citra mencapai 90-100%
4	Bagus	Kesamaan citra mencapai 70-90%
3	Sedang	Kesamaan citra mencapai 60-70%
2	Buruk	Kesamaan citra mencapai 40-60%
1	Sangat Buruk	Kesamaan citra mencapai <40%

Semarang, 7 Agustus 2016

.....