DESAIN APLIKASI STEGANOGRAFI PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER

SKRIPSI

Disusun oleh:

LA SARMAN
NIM. 04.12.678

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011

LEMBAR PERSETUJUAN

DESAIN APLIKASI STEGANOGRAFI PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER

SKRIPSI

Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan
guna mencapai gelar Sarjana Teknik

Disusun Oleh :
La Sarman
NIM : 04.12.678

Mengetahui,
Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Diperiksa dan Disetujui,

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Aryuanto S, ST, MT	I Komang Somawirata, ST, MT
NIP.Y. 1030800417	NIP.Y. 1030100361

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Jl (PERSERO) MALANG
NK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA : LA SARMAN
NIM : 04.12.678
JURUSAN : Teknik Komputer dan Informatika S-1
JUDUL : **DESAIN APLIKASI STEGANOGRAFI PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Selasa
Tanggal : 9 Agustus 2011
Dengan Nilai : 82,55 (A)

PANITIA UJIAN SKRIPSI

KETUA,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

SEKRETARIS,

Dr. Arynanto S, ST. MT
NIP.P.1030800417

ANGGOTA PENGUJI

PENGUJI I,

Sotyohadi, ST,
NIP.Y. 1039700309

PENGUJI II,

Sonny Prasetio, ST, MT
NIP.P. 1031000433

ABSTRAK

DESAIN APLIKASI STEGANOGRAFI PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER

La Sarman, NIM 04.12.678
Dosen Pembimbing : Dr. Aryuanto S, ST, MT dan
I Komang Somawirata, ST, MT

Berbagai macam teknik digunakan untuk melindungi informasi yang dirahasiakan dari orang yang tidak berhak, salah satunya adalah teknik steganografi. Steganografi sebagai suatu seni penyembunyian pesan ke dalam pesan lainnya yang telah ada sejak sebelum Masehi dan kini seiring dengan kemajuan teknologi jaringan serta perkembangan dari teknologi digital, steganografi banyak dimanfaatkan untuk mengirim pesan melalui jaringan *internet* tanpa diketahui orang lain dengan menggunakan media digital berupa file citra.

Aplikasi ini dibangun dengan menggunakan bahasa pemrograman delphi. Metode pengembangan yang digunakan untuk menyelesaikan aplikasi ini yakni dengan melakukan pengembangan metode algoritma *caesar* dan konversi biner kemudian menerapkannya pada media penampungan citra.

Hasil keseluruhan dari skripsi ini berupa suatu aplikasi steganografi. Adapun kelebihan dari aplikasi ini yaitu terdapat pada metode yang diterapkan dalam proses penyisipan pesan, sehingga pihak lain tidak menduga algoritma apa yang sedang digunakan. Tetapi aplikasi ini tingkat keberhasilannya tidak 100% dikarenakan media penampung tidak tahan terhadap manipulasi palet warna.

Kata Kunci : Steganografi, Algoritma Caesar, Konversi Biner, Citra Digital

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan mengucap syukur kehadirat Allah SWT yang maha pengasih dan maha penyayang yang dengan segala Kasih dan Anugerah – Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul :

" DESAIN APLIKASI STEGANOGRAFI PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER "

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor ITN Malang.
2. Bapak Ir. H. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Dr. Aryuanto S, ST, MT. selaku Dosen Pembimbing I.
5. Bapak I Komang Somawirata, ST, MT selaku Dosen Pembimbing II.
6. Ayah dan Ibuku yang selalu memberikan semangat, dorongan dan do'a serta seluruh keluarga yang telah banyak memberikan bantuan baik moril maupun materi.

7. Rekan - rekan dan semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, 2011

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi pada keamanan data adalah hal yang sangat penting, apalagi data yang dikirimkan berupa pesan yang sangat rahasia. Berbagai usaha dilakukan untuk menjaga agar pesan rahasia yang dikirimkan tersebut tidak bisa diakses oleh pihak lain. Dewasa ini penyembunyian pesan rahasia tidak hanya dapat dilakukan dengan menyamarkan pesan tersebut, melainkan dapat pula menyisipkan pesan tersebut ke dalam media lain. Dengan demikian orang lain tidak akan curiga terhadap pesan rahasia yang dikirimkan, karena pesan tersebut tidak terlihat, yang terlihat hanyalah media penampung pesan tersebut. Hal ini akan lebih aman dibandingkan dengan mengirimkan pesan dalam bentuk berkas ter-enkripsi yang akan membuat orang lain curiga dan berusaha untuk mengetahui isi pesan yang dikirim (Hery Purwanto, 2009).

Dalam dunia keamanan data, istilah steganografi sangat dikenal. Steganografi adalah teknik menyembunyikan atau menyamarkan keberadaan pesan rahasia dalam suatu media penampungnya. Steganografi pada media digital digunakan untuk mengeksploitasi keterbatasan kekuatan sistem indera penglihatan dan pendengaran manusia, sehingga dengan keterbatasan tersebut sulit menemukan perubahan pada berkas yang telah disisipi pesan rahasia (Deni Prasetyo, 2010).

Secara teori, semua berkas yang ada di dalam komputer dapat digunakan sebagai media penampung pesan, seperti berkas citra berformat jpg, gif, bmp, berkas audio berformat mp3, wav, bahkan di dalam sebuah video dengan format avi, atau dalam format lainnya seperti txt, html, atau pdf. Semua berkas dapat dijadikan tempat bersembunyi, asalkan berkas tersebut memiliki bit - bit data *redundan* yang dapat dimodifikasi. Setelah dimodifikasi berkas media tersebut tidak akan banyak terganggu fungsinya dan kualitasnya tidak akan jauh berbeda dengan aslinya. Berkas citra merupakan media yang sering digunakan dalam dunia

internet maupun dunia citra digital, ukuran berkasnya relatif kecil apabila dibandingkan dengan berkas audio atau video. Salah satu format berkas citra yang paling sering ditemui di dunia *internet* maupun dunia digital adalah GIF. Berkas dengan format gif berukuran lebih kecil jika dibandingkan dengan format lain untuk citra yang sama, hal ini terjadi karena gif menggunakan tipe kompresi *lossless* (Agung Bahtiar Subardo 2009).

Salah satu algoritma yang digunakan adalah *Gifshuffle*. Sesuai dengan namanya *Gifshuffle* akan melakukan "*shuffle*" terhadap palet warna dari sebuah berkas gif, *shuffle* jika diterjemahkan ke dalam bahasa Indonesia berarti mengacak. Sehingga dapat diartikan bahwa *GifShuffle* adalah algoritma yang memanfaatkan penukaran posisi ke 256 palet warna dalam berkas citra berformat gif. Hal tersebut aman dilakukan karena dua buah berkas gif dengan palet warna yang berbeda akan ditampilkan secara sama persis. Penilaian sebuah algoritma steganografi yang baik dapat dinilai dari beberapa faktor yaitu *imperceptibility* atau keberadaan pesan dalam media penampung tidak dapat dideteksi, *fidelity* yaitu mutu media penampung setelah ditambahkan pesan rahasia tidak jauh berbeda dengan mutu media penampung sebelum ditambahkan pesan, *recovery* yaitu pesan rahasia yang telah disisipkan dalam media penampung harus dapat diungkap kembali dan *robustness* yaitu pesan yang disembunyikan harus tahan terhadap berbagai operasi manipulasi yang dilakukan pada media penampung.

Masalah yang timbul adalah apakah teknik – teknik tersebut baik atau layak untuk digunakan dalam steganografi. Oleh karena itu perlu dilakukan pengembangan terhadap metode lain yang lebih bervariatif. Pengembangan tersebut meliputi penggabungan dua metode berbeda yang dipadukan menjadi suatu teknik baru, dengan asumsi bahwa perpaduan dua metode ini mampu untuk melindungi data atau pesan rahasia yang akan dikirimkan.

1.2. Rumusan Masalah

Berdasarkan hal diatas permasalahan yang timbul adalah bagaimana membuat suatu aplikasi steganografi yang lebih baik, dengan cara menerapkan metode algoritma *caesar* dan pengkonversian biner pada media citra.

1.3. Tujuan

Tujuan yang ingin dicapai dari skripsi ini adalah untuk membuat dan mengembangkan suatu aplikasi steganografi dengan menggunakan metode algoritma *caesar* dan pengkonversian biner pada media citra.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi oleh hal – hal sebagai berikut :

a) Aplikasi steganografi yang akan dibangun menggunakan input–an berupa karakter.
b) Karakter yang diinputkan berupa kode ASCII, yaitu sebanyak 95 karakter huruf, angka, dan simbol.
c) Di dalam skripsi ini hanya membahas tentang steganografi dengan menggunakan metode algoritma *caesar* dan konversi biner serta tidak membahas *Steganalisist*.
d) Pada skripsi ini lebih ditekankan pada aplikasi steganografi–nya sendiri dari pada yang lain.

1.5. Metodologi

1.5.1. Metode Pengumpulan Data

Data merupakan sumber atau bahan mentah yang sangat penting untuk proses menghasilkan informasi. Oleh karena itu dalam pengambilan data perlu dilakukan secara cermat dan hati – hati, sehingga data yang diperoleh dapat bermanfaat dan berkualitas.

Dalam pengumpulan data penyusun menggunakan metode sebagai berikut :

1. Studi Lapangan

   Dengan metode ini data – data diperoleh langsung dari sumber yang bersangkutan, dimana peneliti berhadapan langsung dengan objek yang diteliti, yang dilakukan dengan cara :

   a. *Survey*

      Teknik pengumpulan data dengan cara terjun langsung dan mencatat secara sistematis terhadap objek masalah.

   b. Wawancara / *Interview*

      Teknik pengumpulan data dengan jalan mengadakan komunikasi atau tanya jawab secara langsung.

2. Studi Pustaka / *Literatur*

   Pengumpulan data ini dilakukan dengan cara mencari bahan – bahan kepustakaan sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian. Sumber – sumber pustaka bisa berupa buku atau artikel ilmiah.

1.5.2. Metode Pengembangan Sistem

Metode pengembangan sistem perangkat lunak menggunakan metodologi *waterfall* yaitu sebagai berikut :

1. Rekayasa Sistem

   Tahap ini ditekankan pada pengumpulan kebutuhan pengguna tingkatan sistem dengan mendefinisikan konsep sistem beserta *interface* yang menghubungkannya dengan lingkungan. Hasil dari tahap ini adalah spesifikasi sistem.

2. Analisis Sistem

   Melakukan analisis terhadap kebutuhan perangkat lunak, sehingga diperoleh gambaran umum tentang perangkat lunak yang akan dibangun.

3. Desain Sistem

   Dilakukan dengan cara membuat desain prototipe perangkat lunak yang dapat mengimplementasikan hasil analisis penyelesaian masalah di atas.

4. Implementasi Sistem

   Dilakukan berdasarkan hasil perancangan desain. Penelitian dilakukan dengan bentuk simulasi program dengan menggunakan *software* yang memungkinkan peneliti memanipulasi variabel – variabel input dan meneliti akibatnya terhadap performansi aplikasi steganografi.

5. Pengujian Sistem

   Pengujian ini dilakukan terhadap perangkat lunak yang telah diimplementasikan. Pengujian ini dilakukan untuk menguji kebenaran aplikasi yang telah dibangun.

1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, tujuan, permasalahan, batasan masalah, dan sistematika pembahasan dari skripsi ini.

BAB II: DASAR TEORI

Dasar teori, berupa penjelasan tentang pengertian atau definisi steganografi, metode algoritma *caesar* dan pengkonversian biner.

BAB III : ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi tentang perencanaan objek uji, prosedur percobaan dan menjelaskan metode penelitian yang dilakukan.

BAB IV : IMPLEMENTASI

Bab ini berisi pembahasan hasil pengujian serta pembahasan dari hasil analisa mengenai cara kerja dari sistem.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan skripsi ini.

BAB II
DASAR TEORI

2.1 Steganografi

Steganografi mempunyai peranan penting dalam dunia komputer, khususnya yang berhubungan dengan pengamanan informasi. Banyaknya informasi – informasi rahasia yang dikirimkan melalui media komputer membuat perlu diterapkannya ilmu steganografi agar bisa dikembangkan setiap saat untuk dapat selalu menjaga kerahasiaan informasi tersebut. Informasi informasi ini biasanya berisikan informasi atau data penting dari seseorang, perusahaan ataupun instansi yang tidak ingin dibaca oleh orang yang tidak berhak atas informasi tersebut.

2.1.1 Definisi Steganografi

Steganography adalah suatu teknik atau seni untuk menyembunyikan informasi yang bersifat pribadi ke dalam media lain, agar hasilnya akan tampak seperti informasi normal lainnya. Media yang digunakan umumnya merupakan suatu media yang berbeda dengan media pembawa informasi rahasia, disinilah fungsi dari teknik steganografi yaitu sebagai teknik penyamaran menggunakan media lain yang berbeda sehingga informasi rahasia dalam media awal tidak terlihat secara jelas.

Steganografi biasanya sering disalahkaprahkan dengan kriptografi karena keduanya sama – sama bertujuan untuk melindungi informasi yang berharga. Perbedaan yang mendasar antara keduanya yaitu steganografi berhubungan dengan informasi tersembunyi sehingga tampak seperti tidak ada informasi tersembunyi sama sekali. Jika seseorang mengamati objek yang menyimpan informasi tersembunyi tersebut, ia tidak akan menyangka bahwa terdapat pesan rahasia dalam objek tersebut, dan karenanya ia tidak akan berusaha memecahkan informasi *(dekripsi)* dari objek tersebut.

Kata *Steganography* berasal dari bahasa Yunani, yaitu dari kata *Steganos* (tersembunyi) dan *Graptos* (tulisan). Steganografi di dunia moderen biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam sebuh arsip citra digital, audio, atau video. Teknik Steganografi ini telah banyak

digunakan dalam strategi peperangan dan pengiriman sandi rahasia sejak jaman dahulu kala. Dalam perang Dunia II, teknik steganografi umum digunakan oleh tentara Jerman dalam mengirimkan pesan rahasia dari atau menuju Jerman.

Semakin pentingnya nilai dari sebuah informasi, maka semakin berkembang pula metode – metode yang dapat digunakan untuk melakukan penyisipan informasi yang didukung pula dengan semakin berkembangnya media elektronik. Berbagai macam media elektronik kini telah dapat digunakan untuk melakukan berbagai fungsi steganografi dengan berbagai macam tujuan dan fungsi yang diharapkan oleh penggunanya. Sebagai fungsi yang umum, steganografi digunakan untuk memberikan cap khusus dalam sebuah karya yang dibuat dalam format media elektronik sebagai identifikasi.

Satu hal esensial yang menjadi kelebihan steganografi adalah kemampuannya untuk menipu persepsi manusia. Manusia tidak memiliki insting untuk mencurigai adanya arsip – arsip yang memiliki informasi yang tersembunyi didalamnya, terutama bila arsip tersebut tampak seperti arsip normal lainnya. Namun begitu terbentuk pula suatu teknik yang dikenal dengan steganalisis (*steganalyst*), yaitu suatu teknik yang digunakan untuk mendeteksi penggunaan steganografi pada suatu arsip. Seorang *steganalyst* tidak berusaha untuk melakukan *dekripsi* terhadap informasi yang tersembunyi dalam suatu arsip, yang dilakukan adalah berusaha untuk menemukannya.

Terdapat beberapa cara yang dapat digunakan untuk mendeteksi steganografi seperti melakukan pengamatan terhadap suatu arsip dan membandingkannya dengan salinan arsip yang dianggap belum direkayasa, atau berusaha mendengarkan dan membandingkan perbedaannya dengan arsip lain bila arsip tersebut adalah dalam bentuk audio. Dengan semakin berkembangnya metode penyisipan pesan, maka seorang *steganalisis* akan merubah teknik yang sama untuk memecahkan data tersebut.

Adapun gambaran umum dari perbedaan steganografi dan kriptografi dapat dilihat dalam Gambar 2.1

Gambar 2.1 Perbedaan Steganografi Dan Kriptografi

2.1.2 Sejarah Steganografi

Seperti kriptografi, penggunaan steganografi sebetulnya telah digunakan berabad – abad yang lalu bahkan sebelum istilah steganografi itu sendiri muncul. Berikut adalah contoh penggunaan steganografi di masa lalu :

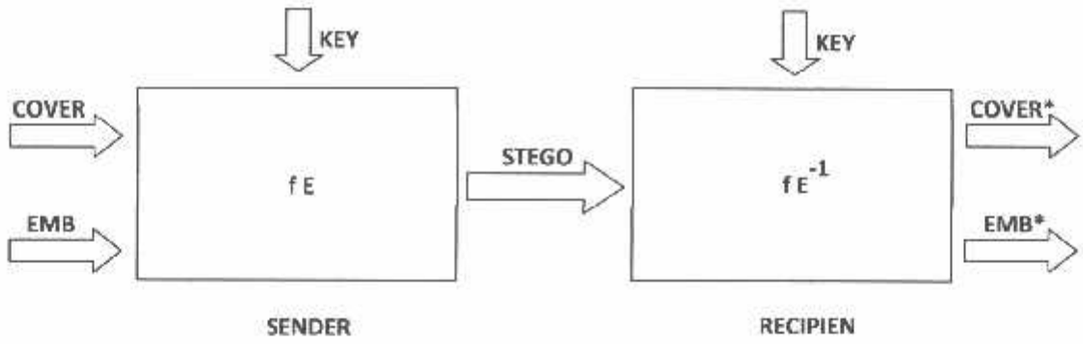
1. Selama terjadinya Perang Dunia ke – 2, tinta yang tidak tampak *(invisible ink)* telah digunakan untuk menulis informasi pada lembaran kertas sehingga saat kertas tersebut jatuh ditangan pihak lain hanya akan tampak seperti lembaran kertas kosong biasa. Cairan seperti air kencing *(urine)*, susu, *vinegar*, dan jus buah digunakan sebagai media penulisan sebab bila salah satu elemen tersebut dipanaskan, tulisan akan menggelap dan tampak melalui mata manusia.
2. Pada sejarah Yunani kuno, masyarakatnya biasa menggunakan seorang pembawa pesan sebagai perantara pengiriman pesan. Pengirim pesan tersebut akan dicukur rambutnya, untuk kemudian dituliskan suatu pesan pada kepalanya yang sudah botak. Setelah pesan dituliskan, pembawa pesan harus menunggu hingga rambutnya tumbuh kembali sebelum dapat mengirimkan pesan kepada pihak penerima.
3. Metode lain yang digunakan oleh masyarakat Yunani kuno adalah dengan menggunakan lilin sebagai media penyembunyi pesan mereka. Pesan dituliskan pada suatu lembaran, dan lembaran tersebut akan ditutup dengan lilin untuk menyembunyikan pesan yang telah tertulis. Pihak penerima kemudian akan menghilangkan lilin dari lembaran tersebut untuk melihat pesan yang disampaikan oleh pihak pengirim.

2.1.3 Kegunaan Steganografi

Seperti perangkat keamanan lainnya, steganografi dapat digunakan untuk berbagai macam alasan, beberapa diantaranya untuk alasan yang baik, namun dapat juga untuk alasan yang tidak baik. Untuk tujuan legitimasi dapat digunakan pengamanan seperti citra dengan *watermarking* dengan alasan untuk perlindungan *copyright*. *Digital watermark* (yang juga dikenal dengan *finger printing*, yang dikhususkan untuk hal – hal menyangkut *copyright*) sangat mirip dengan steganografi karena menggunakan metode penyembunyian dalam arsip, yang muncul sebagai bagian asli dari arsip tersebut dan tidak mudah dideteksi oleh kebanyakan orang.

Steganografi juga dapat digunakan sebagai cara untuk membuat pengganti suatu nilai *hash* satu arah (yaitu pengguna mengambil suatu masukan panjang variabel dan membuat sebuah keluaran panjang statis dengan *tipe string* untuk melakukan verifikasi bahwa tidak ada perubahan yang dibuat pada variabel masukan yang asli). Selain itu juga, steganografi dapat digunakan sebagai *tag - notes* untuk citra *online*. Terakhir, steganografi juga dapat digunakan untuk melakukan perawatan atas kerahasiaan informasi yang berharga, untuk menjaga data tersebut dari kemungkinan sabotasi, pencurian, atau dari pihak yang tidak berwenang. Sayangnya steganografi juga dapat digunakan untuk alasan yang ilegal. Sebagai contoh, jika seseorang telah mencuri data, mereka dapat menyembunyikan arsip curian tersebut ke dalam arsip lain dan mengirimkannya keluar tanpa menimbulkan kecurigaan siapapun karena tampak seperti *email* atau arsip normal. Selain itu, seseorang dengan hobi menyimpan *pornografi*, atau lebih parah lagi, menyimpannya dalam *hard disk*, mereka dapat menyembunyikan hobi buruk mereka tersebut melalui steganografi. Begitu pula dengan masalah terorisme, steganografi dapat digunakan oleh para teroris untuk menyamarkan komunikasi mereka dari pihak luar.

Adapun gambaran umum dari struktur sistem steganografi dapat dilihat dalam Gambar 2.2



Gambar 2.2 Struktur Sistem Steganografi

Keterangan :

- ✓ *Hiddentext* atau *embedded message* : pesan yang disembunyikan.
- ✓ *Covertext* atau *cover–object* : pesan yang digunakan untuk menyembunyikan *embedded message*.
- ✓ *Stegotext* atau *stego–object* : pesan yang sudah berisi *embedded message*.

2.1.3.1 Kriteria Steganografi

Penilaian sebuah algoritma steganografi yang baik dapat dinilai dari beberapa faktor yaitu :

1. **Imperectibility**

   Keberadaan pesan rahasia dalam media penampung tidak dapat dideteksi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan *covertext*–nya. Jika *covertext* berupa *audio* (misalnya berkas file *mp3*, *wav*, *midi* dan sebagainya), maka indera telinga tidak dapat mendeteksi perubahan pada file *stegotext*–nya.

2. **Fidelity**

   Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan itu tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan dapat membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*–nya. Jika *covertext* berupa *audio* (misalnya berkas file *mp3*, *wav*, *midi* dan sebagainya), maka *audio stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan pada file *stegotext*–nya.

3. **Recovery**

   Pesan yang disembunyikan harus dapat diungkapkan kembali (*reveal*). Karena tujuan steganografi adalah *data hiding*, maka sewaktu – waktu pesan rahasia didalam *stegotext* harus dapat diambil kembali untuk digunakan lebih lanjut.

2.1.4 Tipe Media Steganografi

Dalam steganografi, ada beberapa tipe media yang dapat digunakan untuk menyisipkan pesan rahasia. Tipe – tipe media ini dapat berfungsi sebagai media pembawa pesan rahasia, yang disebut dengan *host message*.

Terdapat beberapa media dalam data digital yang dapat digunakan sebagai media steganografi, diantarannya adalah File Sistem Komputer, Transmisi Protokol, Dokumen Teks dan File Audio. Secara detail empat tipe media steganografi tersebut dapat dijelaskan dalam sub bab dibawah ini.

a) File Sistem Komputer

Sebagaimana penyimpanan data secara normal, sebuah file sistem komputer juga dapat digunakan untuk menyembunyikan informasi diantara file yang tidak terlihat penting. Sebagai contoh sebuah *hard drive* ketika menampakkan partisi dalam komputer pemakai dapat berisi partisi tersembunyi yang dapat membawa informasi tersembunyi didalamnya.

Sebagai contoh *sfspatch* adalah sebuah potongan *kernel*, yang dapat berfungsi untuk memasukkan modul pendukung file steganografi dalam sistem *Linux*.

Sfspatch menggunakan *enkripsi* secara bersamaan dengan tehnik steganografi untuk menyembunyikan informasi rahasia di dalam *disk* sehingga tidak akan terlihat oleh pemakai awam.

b) Transmisi Protokol

Transmission Control Protocol (TCP) dan *Intenet Protocol (IP)* adalah sebagian dari protokol yang dapat digunakan untuk menyembunyikan informasi didalam bagian *header* tertentu. Beberapa bagian dari *TCP/IP* akan diubah atau dipotong melalui mekanisme *paket filter* atau melalui *fragment – fragment* yang dikumpulkan kembali.

Bagaimanapun, terdapat beberapa bagian yang tidak dapat diubah. Bagian – bagian tersebut meliputi : *Identification field, Sequence Number field,* dan *Acknowledge Sequence Number field.*

c) Penyembunyian Informasi Dalam Dokumen Teks

Menurut *Bender et al* **[Bender, 1996]** *softcopy text* merupakan salah satu tempat yang paling menarik untuk melakukan penyembunyian data. Karena kurangnya informasi *redundan* di dalam data teks. *Bender et al* mendiskusikan tiga cara berbeda untuk melakukan penyembunyian di dalam data teks.

Metode – metode tersebut adalah : Metode Spasi Terbuka, *Syntactic*, dan Metode *Semantic*.

d) Penyembunyian Informasi Dalam File Audio

Bender et al mengungkapkan penyembunyian informasi dalam sinyal audio sebagai tantangan khusus. Hal ini disebabkan karena fakta bahwa *Human Auditory System (HAS)* berjalan melebihi jangkauan dinamis yang luas. Tetapi sebagaimana yang didiskusikan dalam *paper* masih terdapat beberapa kemungkinan untuk mengungkap sebagian "*holes*" yang tersedia. Terlebih dahulu untuk menyembunyikan informasi dalam data audio tidak hanya cukup untuk mengingat dari sensitivitas *HAS*, namun juga fakta bahwa sinyal audio berjalan diantara *encoding* dan *decoding*. Sinyal audio tersebut juga dapat berjalan melalui sebuah media penyimpan atau ditransmisikan melalui suatu media. Ketika data audio direpresentasikan secara digital, metode *kuantisasi sample* dan penilaian *sampling temporal* akan menjadi faktor penting. Beberapa teknik dipresentasikan oleh *Bender et al* untuk menyembunyikan informasi melalui data audio meliputi *low – bit encoding* dan *Phase coding*. Dalam *low – bit encoding* informasi disimpan dengan mengubah *Least Siginificant Bit (LSB)* dari masing – masing *sampling point* menggunakan sebuah kode *string biner*.

Hasil ini dalam jumlah informasi yang besar dapat di–*encode* dalam sebuah *single data audio*. Sebagai contoh apabila *channel* tanpa suara idealnya berkapasitas I Kbps maka nilai bit yang diberikan sebesar 8 Kbps untuk 8Khz *sequence sample*. Sementara sebagai cara termudah untuk menyembunyikan informasi dalam data audio, skema *low – bit encoding* dapat dihancurkan dengan *niose channel* dan *re – sampling*. *Phase coding* telah terbukti sebagai teknik *coding* yang paling efektif dalam kasus sinyal ke *rasio noise*. Dalam metode ini *phase* dari sinyal audio asli akan diubah dengan referensi *phase* dari informasi yang akan disembunyikan. *Bender et al* menemukan bahwa sebuah kapasitas channel sekitar 8 bps dapat dicapai dengan mengalokasikan 128 slot frekuensi per bit dengan *background noise* yang minimum. *Bender et al* mendiskusikan metode untuk memperbaiki sinyal audio *encoding* dibawah *channel* komunikasi yang berbeda.

2.1.5 Metode Steganografi

Terdapat banyak metode yang digunakan dalam melakukan penyembunyian data kedalam data lainnya. Berikut adalah penjelasan mengenai beberapa metode yang banyak digunakan dalam steganografi.

1. **Metode Steganografi Pada *Text***

   - **Metode Spasi Terbuka**

     Terdapat beberapa cara untuk memanfaatkan spasi terbuka dalam data teks guna menyembunyikan informasi. Metode ini dapat berhasil karena buku bacaan pada umumnya menambahkan satu spasi tambahan pada akhir baris atau diantara dua kata sehingga tidak terbaca aneh. Bagaimanapun, metode spasi terbuka hanya dapat digunakan dengan memakai *ASCII (American Standard Character Interchange)* format. *Bender et al* memberikan tiga metode untuk mengungkap *white space* dalam proses penyembunyian. Spasi terbuka antar kalimat akan menghasilkan nilai "**0**" apabila hanya terdapat sebuah spasi yang ditambahkan diantara kalimat tersebut. Dengan menambahkan dua spasi akan menghasilkan nilai "**1**". Metode ini dapat berhasil, tetapi membutuhkan data dalam jumlah besar untuk menyembunyikan sebuah informasi kecil. Dan juga terdapat banyak *software word – processing*

yang akan secara otomatis membetulkan spasi antara kalimat, sehingga metode ini seringkali gagal. Metode spasi *end–of–line (EOL)* mengutarakan *white space* pada akhir dari masing – masing baris. Data disembunyikan menggunakan jumlah spasi yang telah ditentukan sebelumnya dari akhir untuk masing – masing kalimat. Sebagai contoh dua spasi akan menyembunyikan satu bit, empat spasi akan menyembunyikan dua bit dan delapan spasi akan menghasilkan tiga bit dan seterusnya. Teknik ini lebih baik dibandingkan metode spasi terbuka antar kalimat, karena dengan meningkatkan jumlah spasi akan dapat menyembunyikan lebih banyak data. Salah satu kekurangan dari tehnik ini adalah dapat hilangnya informasi tersembunyi jika *hard copy* data yang diberikan. Pada akhirnya, pemerataan kanan dari teks dapat digunakan pula untuk menyembunyikan informasi rahasia pada data teks. Penghitungan dan pengontrolan spasi diantara kata dapat menyembunyikan informasi dalam data teks yang terlihat tidak penting. Sebuah spasi antara kata akan menghasilkan nilai "**0**" dan dua buah spasi akan menghasilkan nilai "**1**". Bagaimanapun, pendekatan ini akan mempersulit untuk mengeluarkan informasi penting dari media data teks tersebut karena akan semakin tidak mungkin untuk membedakan sebuah spasi biasa dengan spasi yang berfungsi untuk penyembunyian data. Untuk mewujudkan hal ini, *Bender et al* menggunakan *Manchester coding* untuk mengelompokkan bit – bit. Sehingga "**01**" di–interpretasikan sebagai "**1**" dan "**10**" di–interpretasikan sebagai "**0**". Dimana "**00**" dan "**11**" akan dianggap sebagai *null bit string*.

- **Metode Syntactic**

  Metode *Syntactic* sebagaimana yang telah di sarankan oleh *Bender et al*, mengutarakan penggunaan *punktuasi* dan struktur teks untuk menyembunyikan informasi tanpa secara signifikan mengubah arti dari pesan pembawa. Sebagai contoh terdapat dua frase "*bread, butter, and milk*" dan "*bread, butter and milk*" secara gramatikal benar tetapi berbeda dalam penggunaan koma.

  Salah satu dapat digunakan secara alternatif dalam pesan teks guna mengintepretasikan nilai "**1**" apabila salah satu metode dipakai dan nilai "**0**" untuk metode lain yang dipakai.

- **Metode Semantic**

  Metode *Semantic* menggunakan dua sinonim sebagai nilai primer atau sekunder. Nilai tersebut akan diterjemahkan kedalam biner "**1**" atau "**0**". *Bender et al* menggunakan sebuah contoh dimana kata "*big*" berfungsi sebagai primer dan "*large*" berfungsi sebagai sekunder. Oleh karena itu, dalam menguraikan isi sebuah pesan akan menterjemahkan atas penggunaan primer sebagai "**1**" dan sekunder sebagai "**0**".

  Bender et al menyebutkan masalah yang dapat muncul dengan penggunaan metode ini adalah ketika sinonim tidak dapat digantikan karena dapat mengubah arti dari struktur kalimat. Sebagai contoh dalam memanggil seseorang dalam bahasa Inggris dengan "*cool*" mempunyai arti berbeda dibandingkan dengan memanggilnya "*chilly*".

2. Metode Steganografi Pada Gambar

Sudah banyak metode yang digunakan untuk menyembunyikan pesan di dalam sebuah *image* tanpa mengubah tampilan *image*, sehingga pesan yang disembunyikan tidak akan terlihat.

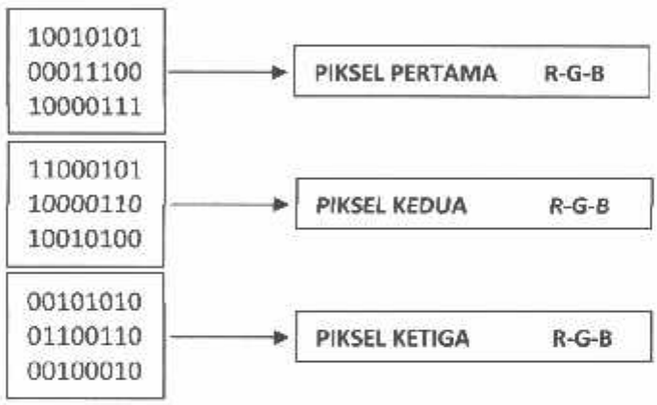
Berikut akan dibahas beberapa metode umum yang biasa digunakan pada *image* steganografi.

- **Metode Penyisipan Least Signicant Bit (LSB)**

  Cara paling umum untuk menyembunyikan pesan adalah dengan memanfaatkan *Least–Significant Bit (LSB)*. Walaupun banyak kekurangan pada metode ini, tetapi kemudahan implementasinya

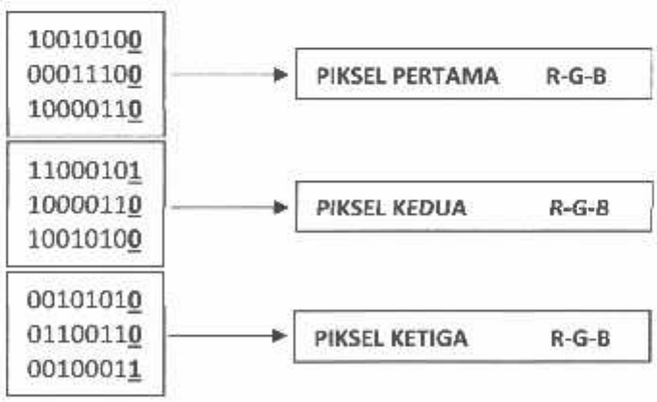
membuat metode ini tetap digunakan sampai sekarang. Metode ini membutuhkan syarat, yaitu jika dilakukan kompresi pada stego, harus digunakan format *lossless compression*, karena metode ini menggunakan bit – bit pada setiap piksel pada image. Jika digunakan format *lossy compression*, pesan rahasia yang disembunyikan dapat hilang. Jika digunakan *image 24 bit color* sebagai *cover*, sebuah bit dari masing – masing komponen *Red, Green*, dan *Blue*, dapat digunakan sehingga 3 bit dapat disimpan pada setiap piksel. Sebuah *image* 800 x 600 piksel dapat digunakan untuk menyembunyikan 1.440.000 bit (180.000 bytes) data rahasia.

Misalnya, di bawah ini terdapat 3 piksel dari *image 24 bit color* dan akan disispkan huruf **A** kedalam data, perhatikan Gambar 2.2.1.



Gambar 2.2.1 Bit Color RGB Sebelum Penyisipan

Huruf **A** mempunyai representasi biner **01000001**, schingga bila disisipkan data diatas akan berubah menjadi data, perhatikan Gambar 2.2.2.

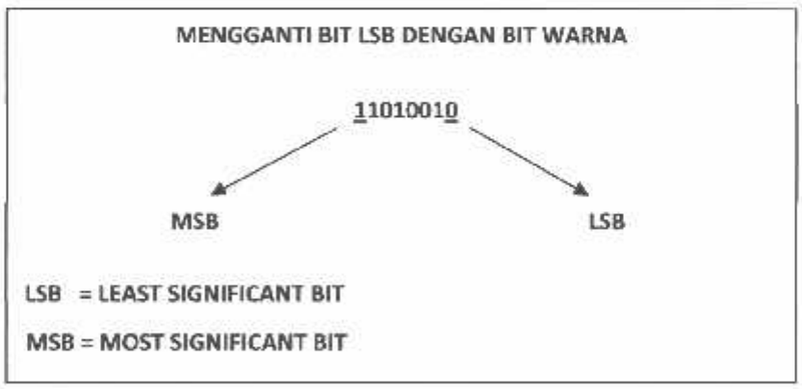


Gambar 2.2.2 Bit Color RGB Sesudah Penyisipan

Dapat dilihat bahwa hanya 3 bit saja yang perlu diubah untuk menyembunyikan karakter **A** ini. Perubahan pada *LSB* ini akan terlalu kecil untuk terdeteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Jika digunakan *image* 8 bit *color* sebagai *cover*, hanya 1 bit saja dari setiap piksel warna yang dapat dimodifikasi sehingga pemilihan *image* harus dilakukan dengan sangat hati – hati, karena perubahan *LSB* dapat menyebabkan terjadinya perubahan warna yang ditampilkan pada citra. Akan lebih baik jika *image* berupa *image grayscale* karena perubahan warnanya akan lebih sulit dideteksi oleh mata manusia. Proses ekstraksi pesan dapat dengan mudah dilakukan dengan meng–ekstrak *LSB* dari masing – masing piksel pada stego secara berurutan dan menuliskan–nya ke *output file* yang akan berisi pesan tersebut.

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

Adapun perbedaan dari MSB dan LSB dapat dilihat dalam Gambar 2.3



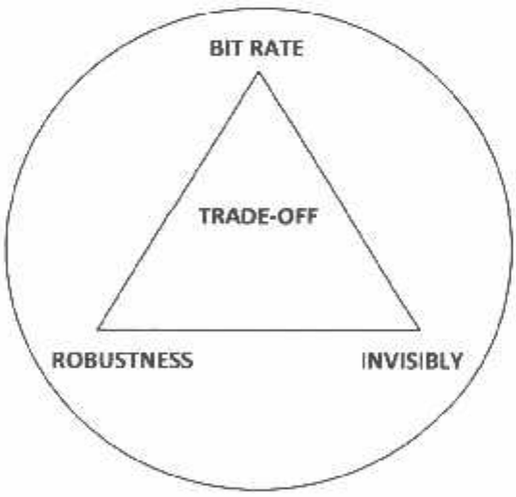
Gambar 2.3 Perbedaan LSB Dan MSB

- **Metode Masking Dan Filtering**

  Teknik *Masking* dan *Filtering* ini biasanya dibatasi pada image 24 bit *color* atau *image grayscale*. Metode ini mirip dengan *watermark*, dimana suatu *image* diberi tanda *(marking)* untuk menyembunyikan pesan rahasia. Hal ini dapat dilakukan, misalnya dengan memodifikasi *luminance* beberapa bagian dari *image*.

  Walaupun metode ini akan mengubah tampilan dari *image*, dimungkinkan untuk melakukannya dengan cara tertentu sehingga mata manusia tidak melihat perbedaannya. Karena metode ini menggunakan aspek *image* yang memang terlihat langsung, metode ini akan lebih "*robust*" terhadap kompresi (terutama *lossy compression), cropping*, dan beberapa *image processing* lain, bila dibandingkan dengan metode modifikasi *LSB*.

  Adapun gambaran umum dari metode *Masking* dan *Filtering* dapat dilihat dalam Gambar 2.4



Gambar 2.4 *Trade–Off* Dalam *Watermarking*

3. Metode Steganografi Pada Suara

Cara untuk mengaplikasikan steganografi pada *file audio* terdiri dari beberapa cara yang lazim digunakan dan prinsip kerja atau algoritma yang digunakan sama seperti pada metode steganografi pada media gambar.

Berikut adalah beberapa teknik yang sering pada suara (*audio*) digunakan :

1. **Low Bit Coding**

   Cara ini lazim digunakan dalam teknik digital steganografi yaitu mengganti *LSB* input setiap sampling–nya dengan data yang dikodekan. Dengan metode ini keuntungan yang didapatkan adalah ukuran pesan yang disispkan relative besar, namun berdampak pada hasil audio yang berkualitas kurang dengan banyaknya *noise*.

2. **Phase Coding**

   Metode kedua yang digunakan ini adalah merekayasa *fasa* dari sinyal masukan. Teori yang digunakan adalah dengan mensubstitusikan awal *fasa* dari tiap awal *segment* dengan *fasa* yang telah dibuat sedemikian rupa dan merepresentasikan pesan yang disembunyikan. *Fasa* dari tiap awal *segment* ini dibuat sedemikian rupa sehingga setiap segmen masih memiliki hubungan yang berujung pada kualitas suara yang tetap terjaga (tidak rusak). Teknik ini menghasilkan keluaran yang jauh lebih baik dari pada metode pertama namun dikompensasikan dengan kerumitan dalam realisasinya.

3. **Spread Spectrum**

   Metode yang ketiga adalah penyebaran spektrum. Dengan metode ini pesan dikodekan dan disebar ke setiap *spectrum* frekuensi yang memungkinkan. Maka dari itu akan sangat sulit bagi yang akan mencoba memecahkannya kecuali ia memiliki akses terhadap data tersebut atau dapat merekonstruksi sinyal *random* yang digunakan untuk menyebarkan pesan pada *range* frekuensi.

4. **Echo Hiding**

   Metode terakhir yang sering digunakan adalah menyembunyikan pesan melalui teknik *echo*. Teknik menyamarkan pesan ke dalam sinyal yang membentuk *echo*. Kemudian pesan disembunyikan dengan menvariasikan tiga parameter dalam *echo* yaitu besar *amplitude* awal,

tingkat penurunan *atenuasi*, dan *offset*. Dengan adanya *offset* dari *echo* dan sinyal asli maka *echo* akan tercampur dengan sinyal aslinya, karena sistem pendengaran manusia yang tidak memisahkan antara *echo* dan sinyal asli. Keempat metode di atas memiliki kesamaan yaitu menggunakan kelemahan dari sistem pendengaran manusia. Maka dari itu teknik steganografi pada citra juga akan menggunakan kelemahan ini untuk menyembunyikan pesan.

2.1.6 Terminologi

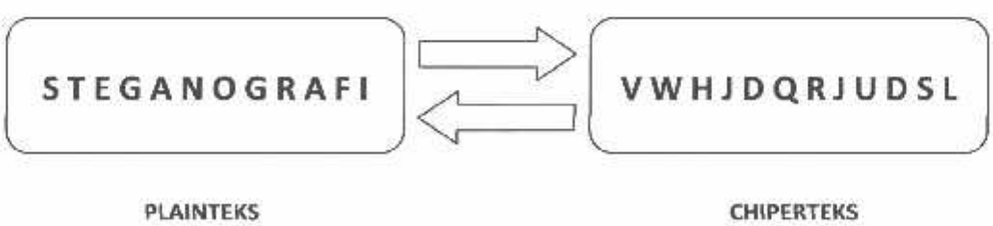
Di dalam steganografi akan sering menemukan berbagai istilah atau terminologi. Beberapa istilah penting untuk diketahui.

1. Plainteks Dan Chiperteks

Pesan ialah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain dari pesan adalah Plainteks. Plainteks dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman. Plainteks yang tersimpan tidak hanya berupa teks, tetapi juga berbentuk citra *(image)*, suara atau bunyi *(audio)*, dan video , atau berkas biner lainnya.

Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan (plainteks) yang tersandi disebut cipherteks *(ciphertext)*. Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca.

Adapun ilustrasi dari proses plainteks dan chiperteks dapat dilihat dalam Gambar 2.5



Gambar 2.5 Perbandingan Plainteks Dan Chiperteks

2. Enkripsi Dan Dekripsi

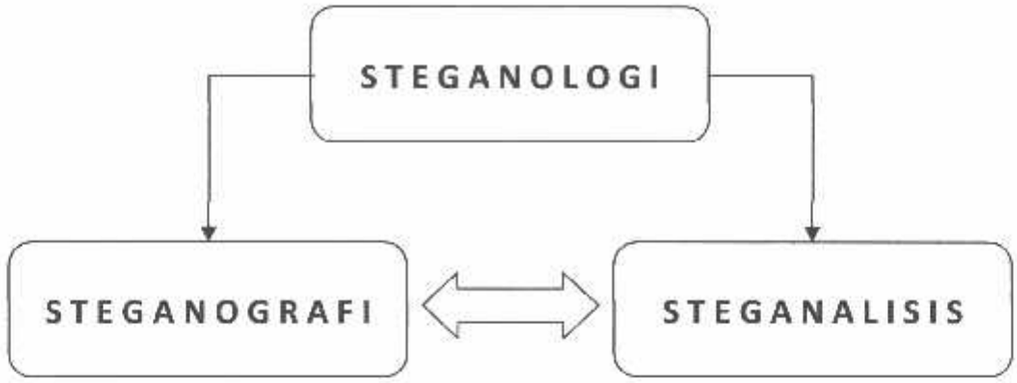
Proses menyandikan plainteks menjadi cipherteks disebut enkripsi *(encryption)* sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi *(decryption)*. Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pesan yang diterima. Istilah *encryption of data motion* mengacu pada enkripsi pesan yang ditransmisikan melalui saluran komunikasi, contohnya adalah pengiriman nomor *PIN* dari mesin *ATM* ke komputer server di kantor bank pusat. Sedangkan istilah *encryption of data at-rest* mengacu pada enkripsi dokumen yang disimpan di dalam *storage*, contohnya adalah *enkripsi file* basis data di dalam *hard disk*.

3. Steganalisis Dan Stegosistem

Seperti Kriptografi dan Kriptanalisis, Steganalisis didefinisikan sebagai suatu seni dan ilmu dalam mendeteksi informasi tersembunyi. Sebagai tujuan dari steganografi adalah untuk merahasiakan keberadaan dari sebuah pesan rahasia, satu keberhasilan penyerangan pada sebuah sistem steganografi terdiri dari pendeteksian bahwa sebuah file yang diyakini berisikan data terselubung. Seperti dalam kriptanalisis diasumsikan bahwa sistem steganografi telah diketahui oleh si–penyerang dan maka dari itu keamanan dari sistem steganografi bergantung hanya pada fakta bahwa kunci rahasia tidak diketahui oleh si penyerang.

Stegosystem disini berisi tentang penyerangan – penyerangan yang dilakukan terhadap suatu sistem steganografi, sebuah perbedaan penting harus dibuat diantara penyerangan – penyerangan pasif dimana penyerang hanya dapat memotong data dan penyerangan – penyerangan aktif dimana penyerang juga dapat memanipulasi data. Lingkaran – lingkaran menunjukkan tempat – tempat penyerang yang berpotensi memiliki jalan masuk ke satu atau lebih dari tempat – tempat tersebut akibat penyerangan – penyerangan yang berbeda jenis, dan juga berfungsi untuk melakukan sebuah penyerangan aktif. Jika lingkaran tidak terisi, penyerang hanya dapat melakukan penyerangan pasif yaitu menghalangi dan memotong data.

Adapun hubungan antar steganologi, steganografi dan steganalisis dapat dilihat dalam Gambar 2.6



Gambar 2.6 Hubungan Antara Steganologi, Steganografi Dan Steganalisis

Penyerangan – penyerangan berikut memungkinkan dalam model dari stegosistem ini :

- ***Stego–Only–Attack*** (Penyerangan hanya Stego).
  Penyerang telah menghalangi stego data dan dapat menganalisisnya.
- ***Stego–Attack*** (Penyerangan Stego).
  Pengirim telah menggunakan cover yang sama berulangkali untuk data terselubung. Penyerang memiliki file stego yang berasal dari cover file yang sama. Dalam setiap file – file stego tersebut, sebuah pesan berbeda disembunyikan.
- ***Cover–Stego–Attack*** (Penyerangan selubung Stego).
  Penyerang telah menghalangi file stego dan mengetahui cover file mana yang digunakan untuk menghasilkan file stego ini. Ini menyediakan sebuah keuntungan melalui penyerangan stego–only untuk si penyerang.
- ***Manipulating The Stego Data*** (Memanipulasi data stego).
  Penyerang memiliki kemampuan untuk memanipulasi data stego. Jika penyerang hanya ingin menentukan sebuah pesan disembunyikan dalam file–stego ini, biasanya ini tidak memberikan sebuah keuntungan tapi memiliki kemampuan dalam memanipulasi data stego.
- ***Manipulating The Cover Data*** (Memanipulasi data terselubung).
  Penyerang dapat memanipulasi data terselubung dan menghalangi hasil data stego. Ini dapat membuat tugas dalam menentukan apakah data stego berisikan sebuah pesan rahasia lebih mudah bagi si penyerang.

2.2 Algoritma *Caesar* Cipher

Salah satu metode penyandian yang pernah digunakan pada masa Yunani Kuno adalah Sandi Caesar *(Caesar Cipher)*. Sandi ini merupakan algoritma subtitusi tertua, dan proses enkripsi–nya mengganti (subtitusi) setiap huruf pada *plaintext* menjadi huruf ke–3 setelahnya. Dengan kata lain, setiap huruf digeser maju sebanyak tiga huruf. Dan untuk mendekripsi *ciphertext* cukup dengan menggeser mundur sebanyak tiga huruf. Sehingga *caesar cipher* ini digolongkan atau dikenal sebagai *monoalphabetic substitution chipper* karena satu huruf tertentu pasti akan berubah menjadi huruf tertentu yang lain.

2.2.1 Dasar Teknik Enkripsi Dan Dekripsi Algoritma *Caesar*

Dasar teknik *caesar cipher* adalah mengganti (mensubtitusi) setiap karakter dengan karakter lain dalam susunan abjad (alfabet). Misalnya, tiap huruf disubtitusi dengan huruf ketiga berikutnya dari susunan abjad. Dalam hal ini kuncinya adalah jumlah pergeseran huruf (yaitu k=3), seperti tampak dalam Tabel 2.1

Tabel 2.1 Subtitusi *Caesar Cipher*

Pi	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Ci	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

2.2.2 Teknik Kriptanalisis Algoritma Caesar

Caesar cipher mudah dipecahkan dengan metode *exhaustive key search* karena jumlah kuncinya sangat sedikit (hanya ada 26 kunci). Misalkan kriptanalis menemukan potongan *ciphertext* **XMZVII**. Diandaikan kriptanalis mengetahui bahwa *plaintext* disusun dalam Bahasa Inggris dan algoritma kriptografi yang digunakan adalah *caesar chipper*. Untuk memperoleh *plaintext*, harus melakukan dekripsi dari kunci yang terbesar (25) sampai kunci yang terkecil (1), kemudian memeriksa apakah dekripsi tersebut menghasilkan pesan yang mempunyai makna. Adapun metode *Exhaustive Key Search*, seperti dalam Tabel 2.2

Tabel 2.2 Metode Menggunakan *Exhaustive Key Search*

Kunci (k) Chipering	'Pesan' hasil Deskripsi	Kunci (k) Chipering	'Pesan' hasil Deskripsi	Kunci (k) Chipering	'Pesan' hasil Deskripsi
0	XMZVH	17	GVIEQ	8	PERNZ
25	YNAWI	16	HWJFR	7	QFSOA
24	ZOBXJ	15	IXKGS	6	RGTPB
23	ZPCYK	14	JYLHT	5	SHUQC
22	BQDZL	13	KZMIU	4	TIVRD
21	CREAM	12	LANJV	3	UJWSE
20	DSFBN	11	MBOKW	2	VKXTF
19	ETGCO	10	NCPLX	1	WLYUG
18	FUHDP	9	ODQMY		

Dari tabel diatas , kata dalam Bahasa Inggris yang potensial menjadi *plaintext* adalah **CREAM** dengan menggunakan $k = 21$. Kunci ini digunakan untuk mendekripsikan *Ciphertext* lainnya. Kadang – kadang satu kunci menghasilkan pesan yang bermakna tidak satu buah, untuk itu dibutuhkan informasi yang lain dengan mendekripsikan potongan Ciphertext lain untuk memperoleh kunci yang benar.

Cara lain yang digunakan untuk memecahkan *Ciphertext* dengan statistik, yaitu dengan menggunakan tabel kemunculan karaker, yang membantu mengidentifikasi karakter *Plaintext* yang berkoresponden dengan karakter di dalam *Ciphertext*. Dalam hal ini, kriptanalis menggunakan tabel frekuensi

kemunculan huruf – huruf dalam teks bahasa Inggris. Pada Tabel 2.4 memperlihatkan frekuensi kemunculan huruf – huruf abjad yang diambil dari sampel yang mencapai 300.000 karakter di dalam sejumlah novel dan surat kabar.

Tabel 2.3 Frekuensi Kemunculan (Relatif) Huruf – Huruf Dalam Teks Bahasa Inggris

HURUF	%	HURUF	%
A	8.2	N	6.7
B	1.5	O	7.5
C	2.8	P	1.9
D	4.2	Q	0.1
E	12.7	R	6.0
F	2.2	S	6.3
G	2.0	T	9.0
H	6.1	U	2.8
I	7.0	V	1.0
J	0.1	W	2.4
K	0.8	X	2.0
L	4.0	Y	0.1
M	2.4	Z	0.1

- Tabel 2.4 di atas pada mulanya dipublikasikan di dalam *Cipher–Systems: The Protection of Communications* dan dikompilasi oleh *H. J. Beker* dan *F.C. Piper*.
- Terdapat sejumlah tabel frekuensi sejenis yang dipublikasikan oleh pengarang lain, namun secara umum persentase kemunculan tersebut konsisten pada sejumlah tabel.
- Bila *Cipher* abjad – tunggal digunakan untuk meng–enkripsi pesan, maka kemunculan huruf – huruf di dalam plainteks tercermin pada tabel 2.4 diatas. Misalnya bila di dalam *Cipher* abjad tunggal huruf **R** menggantikan huruf **E**, maka frekuensi **R** di dalam Cipherteks sama dengan frekuensi E di dalam plainteks–nya.

2.3 Pengertian Diagram Alir

Flowchart adalah tabel keputusan dengan jalur yang terpisah melalui *flowchart* yang mampu menghasilkan aturan keputusan. Menurut Alton. R. Kindred (1985), *flowchart* adalah :

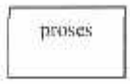
"Menjelaskan karakter dan data yang digunakan oleh programmer komputer untuk menjelaskan bagan alur dari program."

Menurut Jogiyanto H.M (1999), flowchart atau juga disebut dengan bagan alir program (*program flowchart*), adalah :

"Merupakan bagan alir yang menjelaskan secara rinci langkah – langkah dari proses program."

Berikut simbol – simbol yang digunakan untuk pembuatan *flowchart* menurut Jogiyanto H.M (1999) dalam Tabel 2.4 sebagai berikut :

Tabel 2.4 Simbol – Simbol *Flowchart*

SIMBOL	URAIAN
	Gambar kotak persegi menunjukkan proses yang digunakan untuk mewakili suatu proses.
 Garis Alir	Gambar anak panah menunjukkan aliran data atau arus dari proses.
 Penghubung	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang sama atau lainnya
	Menunjukkan input atau output yang digunakan untuk mewakili data input atau output.
	Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
	Digunakan untuk suatu penyelesaian kondisi di dalam program.
	Digunakan untuk memberikan nilai awal suatu besaran
	Atau disebut juga simbol titik terminal yang digunakan untuk menunjukkan awal dan akhir dari suatu proses.

Sehingga dapat disimpulkan bahwa flowchart adalah penggambaran aliran data dan karakter dari bagan alir program yang menjelaskan secara rinci langkah – langkah dari proses program.

2.4 Bahasa Pemrograman Borland Delphi

Borland Delphi merupakan salah satu IDE untuk membangun aplikasi *desktop* menggunakan bahasa pemrograman *Pascal*. *Pascal* adalah bahasa pemrograman yang cukup populer karena nyaris memiliki struktur seperti bahasa manusia.

Delphi7 adalah aplikasi utama mengembangkan lingkungan secara cepat dengan membangun aplikasi berkinerja tinggi. *Delphi7 Architect* dirancang untuk para pengembang dan tim membangun data–intensif klien / *server GUI* dan aplikasi *web* dengan sistem *database* yang besar atau kompleks. *Delphi Architect* menggabungkan pengembangan aplikasi berbasis data yang cepat dan *heterogen* kecepatan akses data yang tinggi dengan pemodelan data *visual* yang kaya untuk membantu para pengembang memahami, desain, dan mendapatkan nilai yang banyak dari *database* perusahaan yang ada dan struktur data.

Borland Delphi atau biasa disebut *Delphi* merupakan perangkat lunak pengembangan aplikasi yang sangat populer di lingkungan *Windows*. Perangkat lunak ini dapat digunakan untuk membuat aplikasi apa saja, dari permainan hingga ke aplikasi basis data.

Semenjak versi 6, *Delphi* telah dilengkapi dengan sejumlah komponen yang tergolong sebagai *dbExpress*, yang memungkinkan koneksi ke *MySQL* ataupun *Oracle* dilakukan dengan mudah, sehingga *Delphi* dapat digunakan sebagai aplikasi *front–end* yang berhubungan dengan *database server*. Sedangkan pada versi 7, komponen yang tergolong sebagai *dbExpress* sedikit berubah.

Perubahan umum pada *IDE (Integrated Development Environment)* adalah sebagai berikut :

- *Menu View | Additional Message Info* menampilkan jendela *message hint*, yaitu informasi tambahan tentang *compiler message* yang dapat di *download* dari *website Borland*.
- Perubahan pada *component pallete*.
- Jika membuat aplikasi baru untuk *CLX*, pada *page system* akan ditampilkan beberapa komponen yang berhubungan dengan file dan direktori.

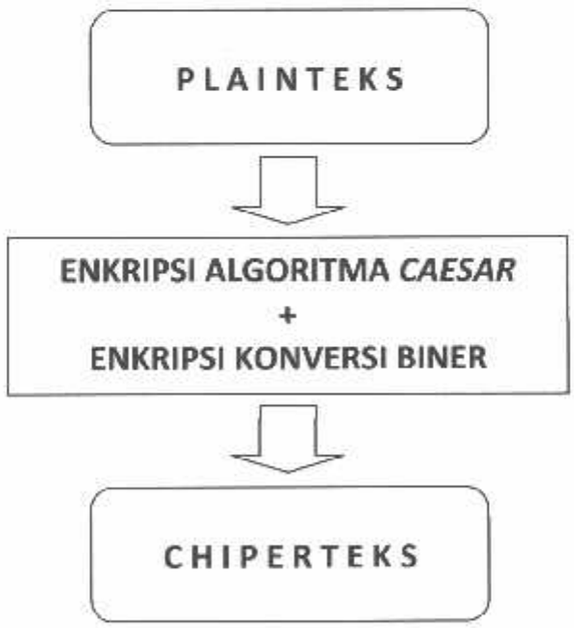
- Penambahan *page Indy Intercepts* dan *Indy I/O Handler Open Source* untuk komponen *Internet Protocol* pada edisi *Professional* dan *Enterprise.*
- Penambahan *page IW Standart, IW Data, IW Client Side, IW Control page* yang menyediakan komponen – komponen *IntraWeb* untuk pengembangan aplikasi berbasis *web.*
- Penambahan *page Rave* yang menyediakan komponen – komponen untuk keperluan pembuatan *report.*
- *Code Completion* yang lebih cepat. *Customize code completion manager* menggunakan *OpenTools API.*

BAB III
ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Masalah

Dalam merancang suatu sistem diperlukan analisis terhadap sistem yang akan dirancang tersebut terlebih dahulu. Tujuan dari analisis ini sendiri adalah agar sistem yang dirancang menjadi tepat guna dan ketahanan sistem tersebut akan lebih terjaga. Disamping itu dengan dilakukannya analisis akan mempermudah pekerjaan dalam membuat sistem, dan jika suatu saat nanti ada perbaikan atau penambahan dalam sistem tersebut, maka akan mudah diselesaikan.

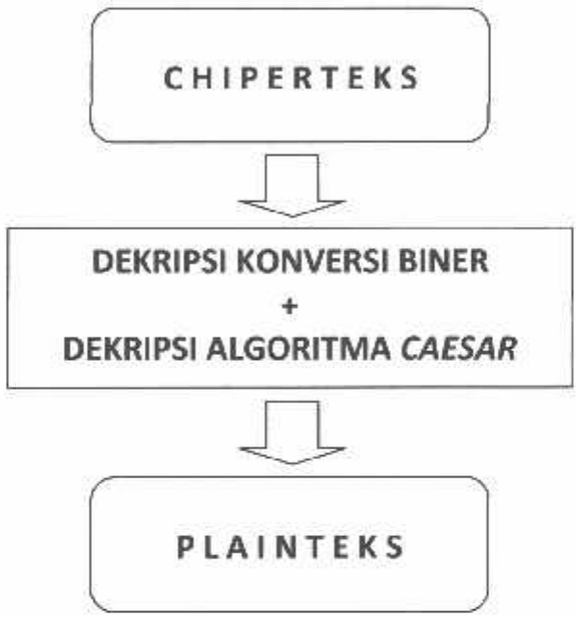
Sistem yang dirancang terdiri dari dua proses secara garis besar, yaitu proses enkripsi dan dekripsi. Dimana untuk proses enkripsi atau dekripsi menggunakan algoritma *Caesar* dan konversi biner. Adapun tujuan dari algoritma ini sendiri yaitu agar penggunaan karakter yang digunakan menjadi lebih luas lagi, yang semula hanya mencakup 26 karakter menjadi 95 karakter yang meliputi karakter huruf besar (A..Z), huruf kecil (a..z), angka(0..9) dan simbol – simbol (-! ... +\). Skema global untuk proses enkripsi dapat dilihat dalam Gambar 3.1



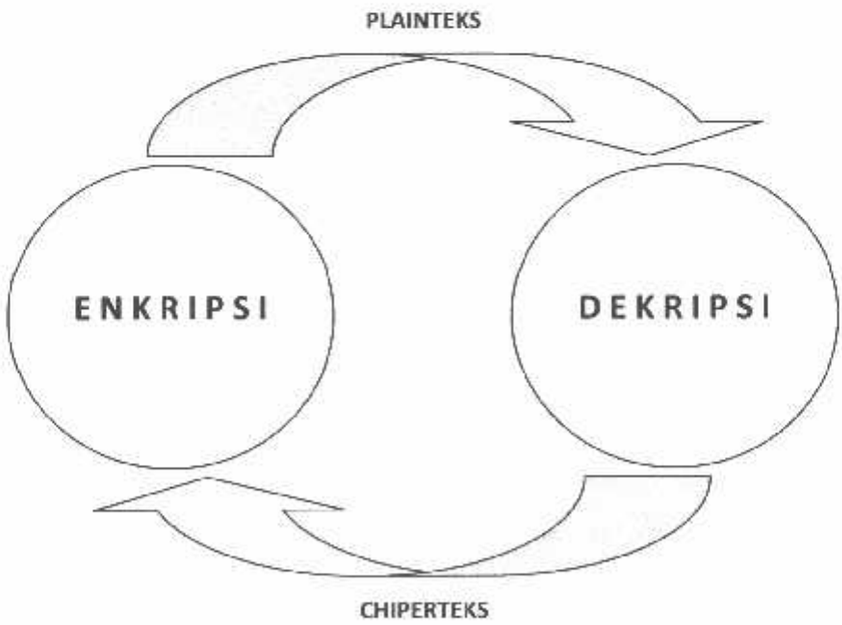
Gambar 3.1 Skema Enkripsi Algoritma Kriptografi Klasik

Dari gambar diatas dapat dilihat bahwa plainteks akan dienkripsi menggunakan algoritma kriptografi klasik, dimana nantinya di dalam algoritma ini ada dua proses enkripsi yaitu enkripsi Caesar dan enkripsi Biner. Sehingga nantinya akan menghasilkan suatu chiperteks yang akan disisipkan ke dalam media penampungan berupa citra (images).

Skema global untuk proses dekripsi seperti tampak dalam Gambar 3.2



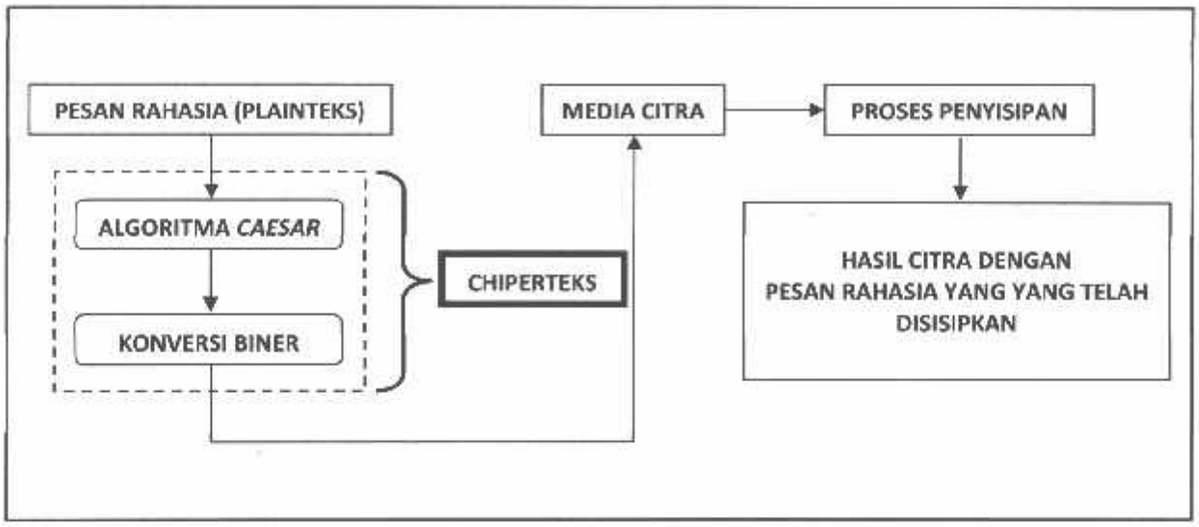
Gambar 3.2 Skema Dekripsi Algoritma Kriptografi Klasik



Gambar 3.3 Siklus Hubungan Antara Enkripsi Dan Dekripsi

3.2 Desain Sistem

Adapun desain sistem untuk aplikasi steganografi ini dapat di lihat dalam Gambar 3.4



Gambar 3.4 Desain Sistem Aplikasi Steganografi

Dari gambar diatas dapat dilihat bahwa terdapat dua penambahan metode atau cara untuk mengamankan pesan yang akan disisipkan, yaitu metode algoritma *Caesar* dan konversi biner.

Dasar teknik *Caesar Cipher* adalah mengganti (mensubtitusi) setiap karakter dengan karakter lain dalam susunan abjad (alfabet). Misalnya, tiap huruf disubtitusi dengan huruf ketiga berikutnya dari susunan abjad. Tujuannya adalah mengubah pesan asli menjadi pesan acak agar sulit untuk di pecahkan.

- **Proses Enkripsi :**

  C = E(P)=(P+k=3) mod 26

- **Proses Dekripsi :**

  P = D(C)=(C-k=3) mod 26

Dikarenakan dalam hal ini menggunakan 95 karakter maka akan berubah menjadi persamaan gabungan sebagai berikut :

- **Proses Enkripsi :**

  C = E((P-32)+(k=3)) mod 95+32

- **Proses Dekripsi :**

  P = D((C-32)-(k=3)) mod 95+32

Keterangan :

C = Chiperteks (pesan sandi)

P = Plainteks (pesan asli)

E = Enkripsi

D = Dekripsi

K = Key(3)

Mod/Modulo = sisa pembagian

32 = Nilai entitas awal pada kode ASCII

95 = Nilai entitas akhir pada kode ASCII

Contoh :

Plainteks : **MESIN**

Enkripsi :

C = E((P-32)+(k=3)) mod 95+32

Enkripsi **M** :

C = E(P-32)+(k=3) mod 95+32

C = E(**M**-32)+3) mod 95+32

C = E(77-32)+3) mod 95−32

C = E48 mod 95+32

C = E80

C = 80 atau **P**

Enkripsi **E** :

C = E(P-32)+(k=3) mod 95+32

C = E(**E**-32)+3) mod 95+32

C = E(69-32)+3) mod 95+32

C = E40 mod 95+32

C = E72

C = 72 atau **H**

Enkripsi **S** :

C = E(P-32)+(k=3) mod 95+32

C = E(**S**-32)+3) mod 95+32

C = E(83-32)+3) mod 95+32

C = E54 mod 95+32

C = E86

C = 86 atau **V**

Enkripsi **I** :

C = E(P-32)+(k=3) mod 95+32

C = E(**I**-32)+3) mod 95+32

C = E(73-32)+3) mod 95+32

C = E44 mod 95+32

C = E76

C = 76 atau **L**

Enkripsi **N** :

C = E(P-32)+(k=3) mod 95+32

C = E(**N**-32)+3) mod 95+32

C = E(78-32)+3) mod 95+32

C = E49 mod 95+32

C = E81

C = 81 atau **Q**

Jadi chiperteks yang didapat dari plainteks **MESIN** = **PIIVLQ**

Dekripsi :

P = D((C-32)-(k=3)) mod 95+32

Dekripsi **P** :

P = D(C-32)-(k=3) mod 95+32

P = D(**P**-32)-3) mod 95+32

P = D(80-32)-3) mod 95+32

P = D45 mod 95+32

P = D77

P = 77 atau **M**

Dekripsi **H** :

P = D(C-32)-(k=3) mod 95+32

P = D(**H**-32)-3) mod 95+32

P = D(72-32)-3) mod 95+32

P = D37 mod 95+32

P = D69

P = 69 atau **E**

Dekripsi **V** :

P = D(C-32)-(k=3) mod 95+32

P = D(**V**-32)-3) mod 95+32

P = D(86-32)-3) mod 95+32

P = D51 mod 95+32

P = D83

P = 83 atau **S**

Dekripsi **L** :

P = D(C-32)-(k=3) mod 95+32

P = D(**L**-32)-3) mod 95+32

P = D(76-32)-3) mod 95+32

P = D41 mod 95+32

P = D73

P = 73 atau **I**

Dekripsi **Q** :

P = D(C-32)-(k=3) mod 95+32

P = D(**Q**-32)-3) mod 95+32

P = D(81-32)-3) mod 95+32

P = D46 mod 95+32

P = D78

P = 78 atau **N**

Menjadi **PHVLQ = MESIN**

Sistem bilangan biner atau sistem bilangan basis dua adalah sebuah sistem penulisan angka dengan menggunakan dua simbol yaitu "0" dan "1". Sistem bilangan biner moderen ditemukan oleh Gottfried Wilhelm Leibniz pada abad ke–17. Sistem bilangan ini merupakan dasar dari semua sistem bilangan berbasis digital. Dalam istilah komputer, 1 Byte = 8 bit. Kode – kode rancang bangun komputer, seperti ASCII (*American Standard Code for Information Interchange*) menggunakan sistem pengkodean 1 Byte. Adapun tabel ASCII seperti tampak dalam Tabel 3.1

Tabel 3.1 Kode ASCII

(American Standard Code for Information Interchange)

DEC	OCT	HEX	BIN	Symbol	HTML Number	Description
32	040	20	**00100000**			Space
33	041	21	**00100001**	!	!	Exclamation mark
34	042	22	**00100010**	"	"	Double quotes
35	043	23	**00100011**	#	#	Number
36	044	24	**00100100**	$	$	Dollar
37	045	25	**00100101**	%	%	Procenttecken
38	046	26	**00100110**	&	&	Ampersand
39	047	27	**00100111**	'	'	Single quote
40	050	28	**00101000**	(	(	Open parenthesis
41	051	29	**00101001**	)	)	Close parenthesis
42	052	2A	**00101010**	*	*	Asterisk
43	053	2B	**00101011**	+	+	Plus
44	054	2C	**00101100**	,	,	Comma
45	055	2D	**00101101**	-	-	Hyphen
46	056	2E	**00101110**	.	.	Period
47	057	2F	**00101111**	/	/	Slash or divide
48	060	30	**00110000**	0	0	Zero
49	061	31	**00110001**	1	1	One
50	062	32	**00110010**	2	2	Two
51	063	33	**00110011**	3	3	Three
52	064	34	**00110100**	4	4	Four
53	065	35	**00110101**	5	5	Five
54	066	36	**00110110**	6	6	Six
55	067	37	**00110111**	7	7	Seven
56	070	38	**00111000**	8	8	Eight
57	071	39	**00111001**	9	9	Nine
58	072	3A	**00111010**	:	:	Colon
59	073	3B	**00111011**	;	;	Semicolon
60	074	3C	**00111100**	<	<	Less than
61	075	3D	**00111101**	=	=	Equals
62	076	3E	**00111110**	>	>	Greater than
63	077	3F	**00111111**	?	?	Question mark
64	100	40	**01000000**	@	@	At symbol
65	101	41	**01000001**	A	A	Uppercase A
66	102	42	**01000010**	B	B	Uppercase B
67	103	43	**01000011**	C	C	Uppercase C
68	104	44	**01000100**	D	D	Uppercase D
69	105	45	**01000101**	E	E	Uppercase E
70	106	46	**01000110**	F	F	Uppercase F
71	107	47	**01000111**	G	G	Uppercase G
72	110	48	**01001000**	H	H	Uppercase H
73	111	49	**01001001**	I	I	Uppercase I
74	112	4A	**01001010**	J	J	Uppercase J
75	113	4B	**01001011**	K	K	Uppercase K
76	114	4C	**01001100**	L	L	Uppercase L
77	115	4D	**01001101**	M	M	Uppercase M
78	116	4E	**01001110**	N	N	Uppercase N
79	117	4F	**01001111**	O	O	Uppercase O
80	120	50	**01010000**	P	P	Uppercase P
81	121	51	**01010001**	Q	Q	Uppercase Q
82	122	52	**01010010**	R	R	Uppercase R
83	123	53	**01010011**	S	S	Uppercase S
84	124	54	**01010100**	T	T	Uppercase T
85	125	55	**01010101**	U	U	Uppercase U

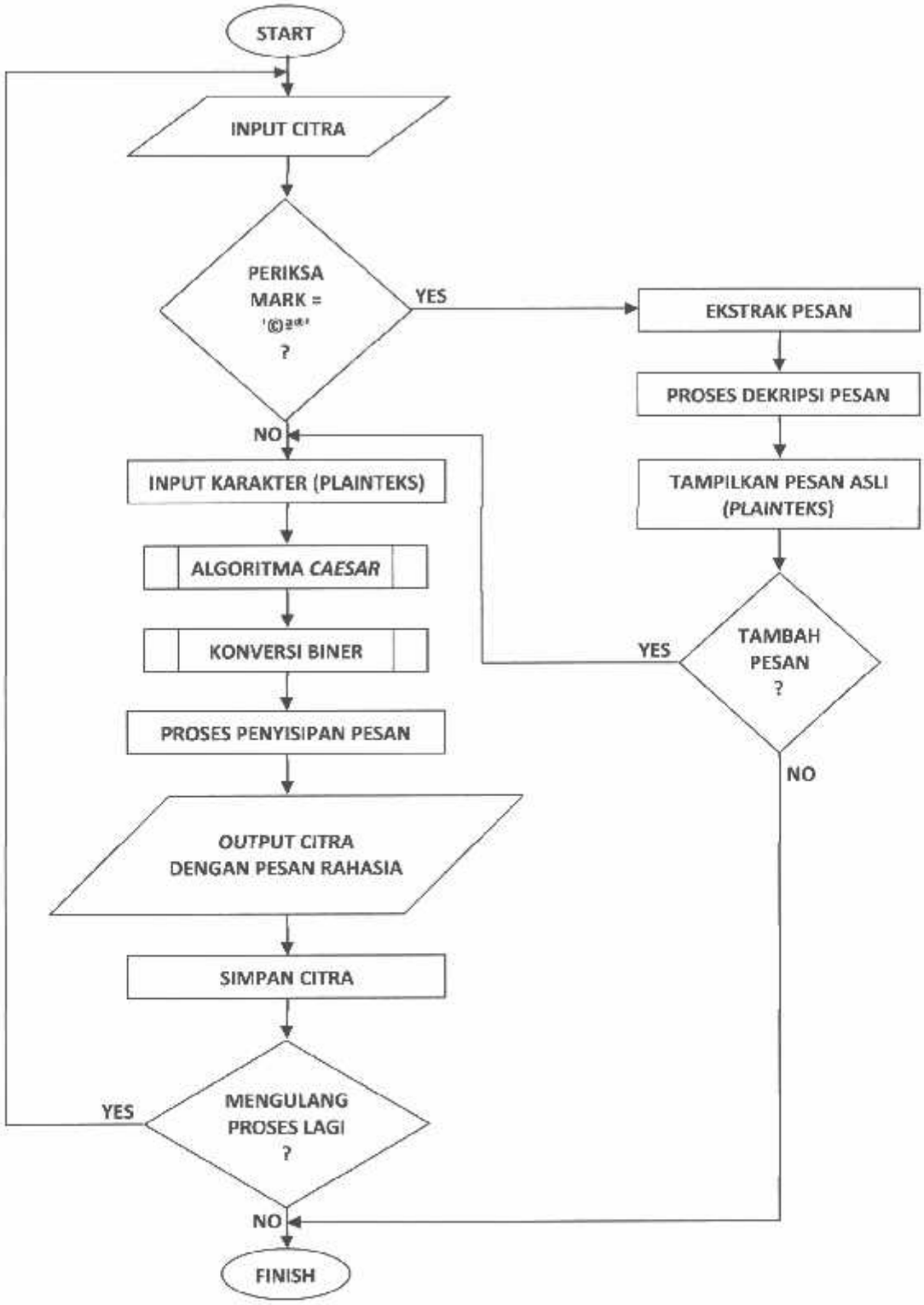
86	126	56	01010110	V	V	Uppercase V
87	127	57	01010111	W	W	Uppercase W
88	130	58	01011000	X	X	Uppercase X
89	131	59	01011001	Y	Y	Uppercase Y
90	132	5A	01011010	Z	Z	Uppercase Z
91	133	5B	01011011	[	[	Opening bracket
92	134	5C	01011100	\	\	Backslash
93	135	5D	01011101	]	]	Closing bracket
94	136	5E	01011110	^	^	Caret - circumflex
95	137	5F	01011111	_	_	Underscore
96	140	60	01100000	`	`	Grave accent
97	141	61	01100001	a	a	Lowercase a
98	142	62	01100010	b	b	Lowercase b
99	143	63	01100011	c	c	Lowercase c
100	144	64	01100100	d	d	Lowercase d
101	145	65	01100101	e	e	Lowercase e
102	146	66	01100110	f	f	Lowercase f
103	146	67	01100111	g	g	Lowercase g
104	150	68	01101000	h	h	Lowercase h
105	151	69	01101001	i	i	Lowercase i
106	152	6A	01101010	j	j	Lowercase j
107	153	6B	01101011	k	k	Lowercase k
108	154	6C	01101100	l	l	Lowercase l
109	155	6D	01101101	m	m	Lowercase m
110	156	6E	01101110	n	n	Lowercase n
111	157	6F	01101111	o	o	Lowercase o
112	160	70	01110000	p	p	Lowercase p
113	161	71	01110001	q	q	Lowercase q
114	162	72	01110010	r	r	Lowercase r
115	163	73	01110011	s	s	Lowercase s
116	164	74	01110100	t	t	Lowercase t
117	165	75	01110101	u	u	Lowercase u
118	166	76	01110110	v	v	Lowercase v
119	167	77	01110111	w	w	Lowercase w
120	170	78	01111000	x	x	Lowercase x
121	171	79	01111001	y	y	Lowercase y
122	172	7A	01111010	z	z	Lowercase z
123	173	7B	01111011	{	{	Opening brace
124	174	7C	01111100	|			Vertical bar
125	175	7D	01111101	}	}	Closing brace
126	176	7E	01111110	~	~	Equivalency sign

3.3 Analisa Flowchart Aplikasi Steganografi

Prinsip kerja dari algoritma steganografi ini adalah dengan menginputkan pesan rahasia (plainteks) berupa karakter huruf *alfabaeth* kecil (a...z) atau huruf alfabet besar (A...Z), angka (0..9) ataupun simbol – simbol (~! ... +\) yang disisipkan kedalam media penampungan.

Sebelum proses penyisipan pesan pada media citra, data asli di acak dengan menggunakan dua metode berbeda yaitu algoritma *Caesar* dan konversi biner.

Adapun skema global untuk proses aplikasi steganografi ini dapat dilihat dalam Gambar 3.5



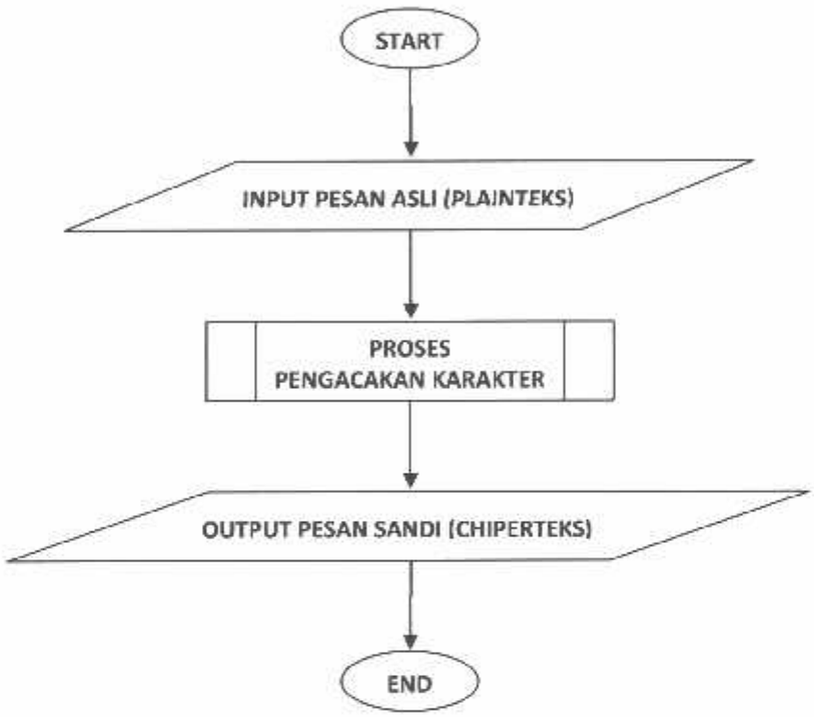
Gambar 3.5 Flowchart Desain Aplikasi Steganografi

Algoritma dari flowchart pada aplikasi steganografi diatas dapat dijabarkan sebagai berikut :

1. **Start** atau memulai program/aplikasi
2. **Input Citra** merupakan masukan untuk memilih citra mana yang akan digunakan dalam proses penyisipan pesan. Format citra yang biasa digunakan adalah citra dengan ekstensi sebagai berikut : Bitmap (BMP), GIF, PCX, JPEG, PNG dan lain – lain.
3. **Proses pemeriksaan citra** bertujuan untuk mengecek, apakah pada citra yang telah dipilih memiliki pesan rahasia atau tidak. Jika telah terdapat pesan rahasia pada citra yang dipilih, maka pesan akan ekstrak kemudian pesan ditampilkan atau ditambahkan dan apabila citra yang dipilih tidak terdapat pesan rahasia, maka dilanjutkan ke proses yang berikutnya.
4. **Input karakter** merupakan suatu masukan berupa karakter huruf *alfabaeth* kecil (a...z) atau huruf alfabet besar (A...Z), angka (0..9) ataupun simbol – simbol (~! ... +\). Atau dengan kata lain pesan rahasia yang akan disisipkan ke dalam media (penampung) citra.
5. **Algoritma *Caesar*** merupakan proses pengacakan data asli (plainteks) menjadi data acak (chiperteks).dengan melakukan pergeseran karakter sebanyak 3 kali perpindahan/pergeseran.
6. **Konversi biner** yaitu merubah chiperteks atau pesan acak menjadi bilangan biner sesuai dengan kode ASCII sebagai acuan standar internasional.
7. **Penyisipan pesan** merupakan proses memasukan kode biner ke dalam bit – bit citra dengan memanfaatkan bit – bit yang paling tidak berarti pada citra penampung LSB (*Least Significant Bit*). Pada proses ini pesan asli (plainteks) yang telah di enkripsi disebar ke dalam elemen citra yaitu RGB (*Red Green Blue*),piksel citra akan di rubah dari byte ke bit secara otomatis.
8. **Output citra** dengan pesan rahasia yang telah disisipi merupakan keluaran atau hasil akhir dari penyatuan dua metode ini.
9. **Proses** perulangan yang berisikan perintah apakah memulai proses lagi untuk mengulang dari awal atau tidak, kemudian di lanjutkan dengan mengakhiri aplikasi.
10. **Finish** atau selesai.

3.4 Algoritma Caesar

Prinsip kerja dari algoritma *Caesar* ini adalah mengganti (*mensubtitusi*) setiap karakter dengan karakter lain sebanyak pergeseran (k), dimana **k** pada *Caesar* Cipher ini bernilai tiga (k=3). Sehingga setiap karakter akan digeser sebanyak tiga kali perpindahan. Namun perkembangannya sekarang pemakaian nilai *key* sudah tidak terpaku lagi pada *key Caesar Cipher*, bisa juga bernilai k=1, k=2, k= 3…dst. Berikut flowchart enkripsi *Caesar* seperti dalam Gambar 3.6



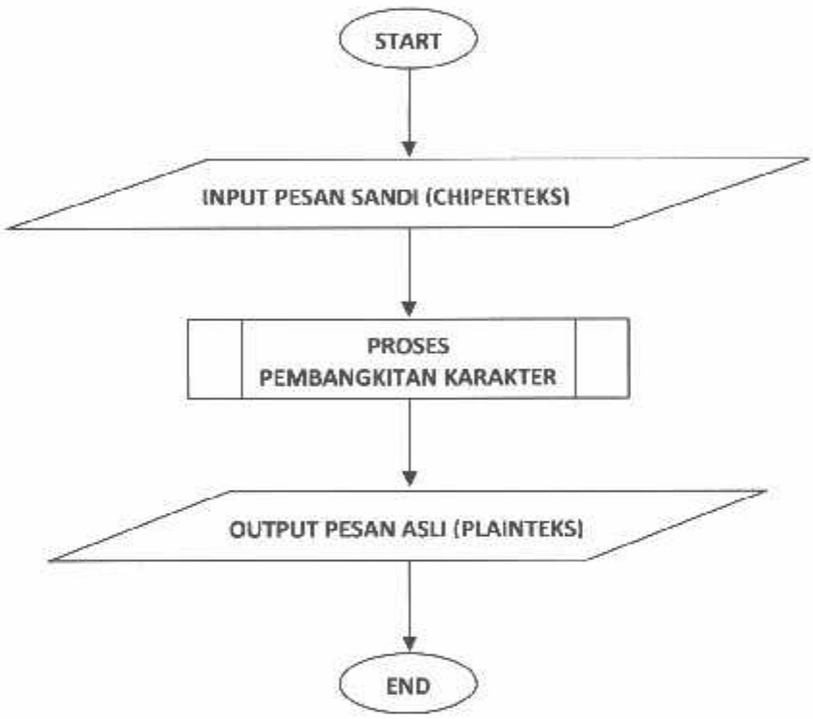
Gambar 3.6 Flowchart Enkripsi Algoritma *Caesar*

Dalam algoritma caesar terdapat variabel – variabel dan fungsi sebagai berikut :

1. **Start** atau memulai aplikasi.
2. **Input pesan asli** (plainteks) merupakan suatu masukan berupa karakter bebas yang ada pada tombol *keyboard*, dimulai dari huruf besar, huruf kecil, lambang dan juga simbol – symbol.
3. **Proses pengacakan karakter** yaitu proses di mana inputan atau masukan dari pesan asli (plainteks) yang di rubah menjadi pesan acak (chiperteks).

4. **Output** atau keluaran merupakan hasil dari pengacakan pesan asli (plainteks) menjadi pesan acak (chiperteks).
5. **End** atau selesai.

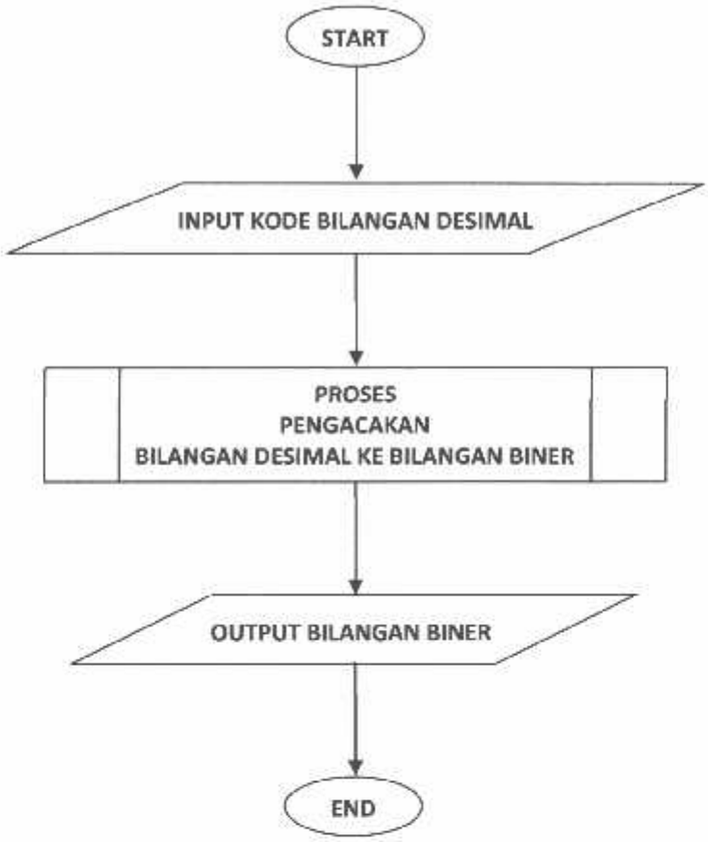
Proses pendekripsian, dilakukan dengan cara mengembalikan pesan asli ke bentuk semula. Jika pada waktu mengenkripsi dengan menggeser karakter ke arah kanan sebanyak tiga pergeseran, maka pada proses dekripsi menggeser karakter ke arah berlawanan sebanyak tiga pergeseran yang sama juga, begitu pula sebaliknya. Berikut *flowchart* dekripsi *caesar* seperti dalam Gambar 3.7



Gambar 3.7 Flowchart Dekripsi Algoritma *Caesar*

3.5 Konversi Bilangan Biner

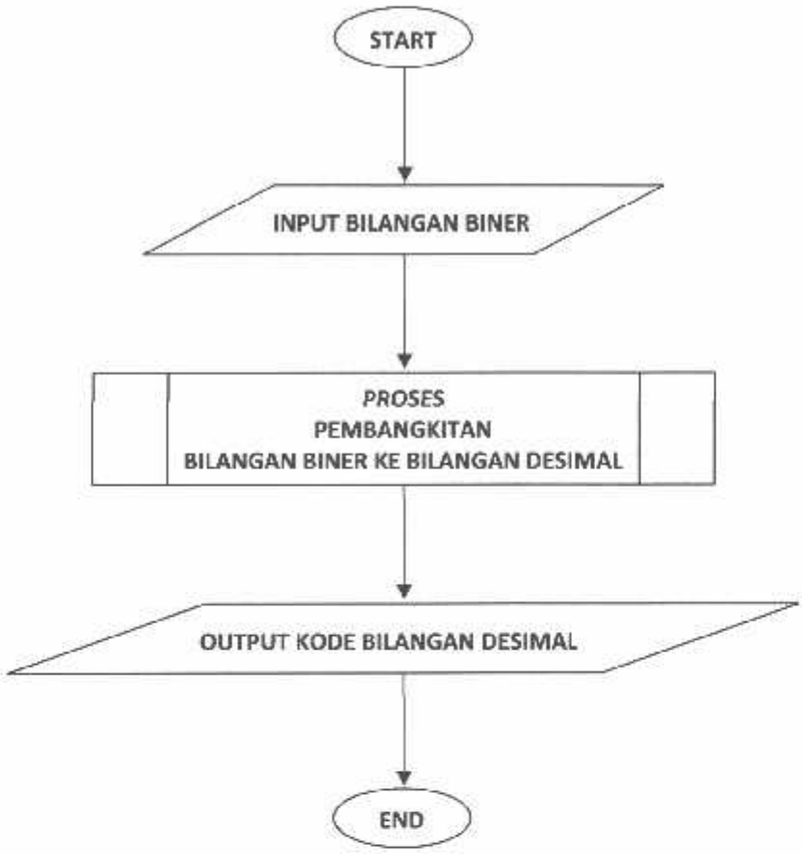
Prinsip kerja dari konversi bilangan biner ini adalah dengan mengganti (*mensubtitusi*) setiap bilangan desimal atau bilangan basis 10 menjadi bilangan biner atau bilangan berbasis dua. Berikut flowchart enkripsi bilangan desimal (basis 10) ke bilangan biner (basis 2) seperti dalam Gambar 3.8



Gambar 3.8 Flowchart Enkripsi Bilang Desimal Ke Bilangan Biner

Pada konversi bilangan desimal ke bilangan biner terdapat variable – variable dan fungsi sebagai berikut :

1. **Start** atau memulai program.
2. Input kode bilangan desimal dari hasil pengacakan algotitma *Caesar* yaitu 95 karakter ASCII.
3. Pada proses ini kode bilang desimal di rubah menjadi bilangan biner,yang mana nantinya bilangan biner ini akan disisipkan ke dalam bit – bit citra.
4. **Output** berupa keluaran bilangan biner.
5. **End** atau selesai.



Gambar 3.9 Flowchart Dekripsi Bilang Biner Ke Bilangan Desimal

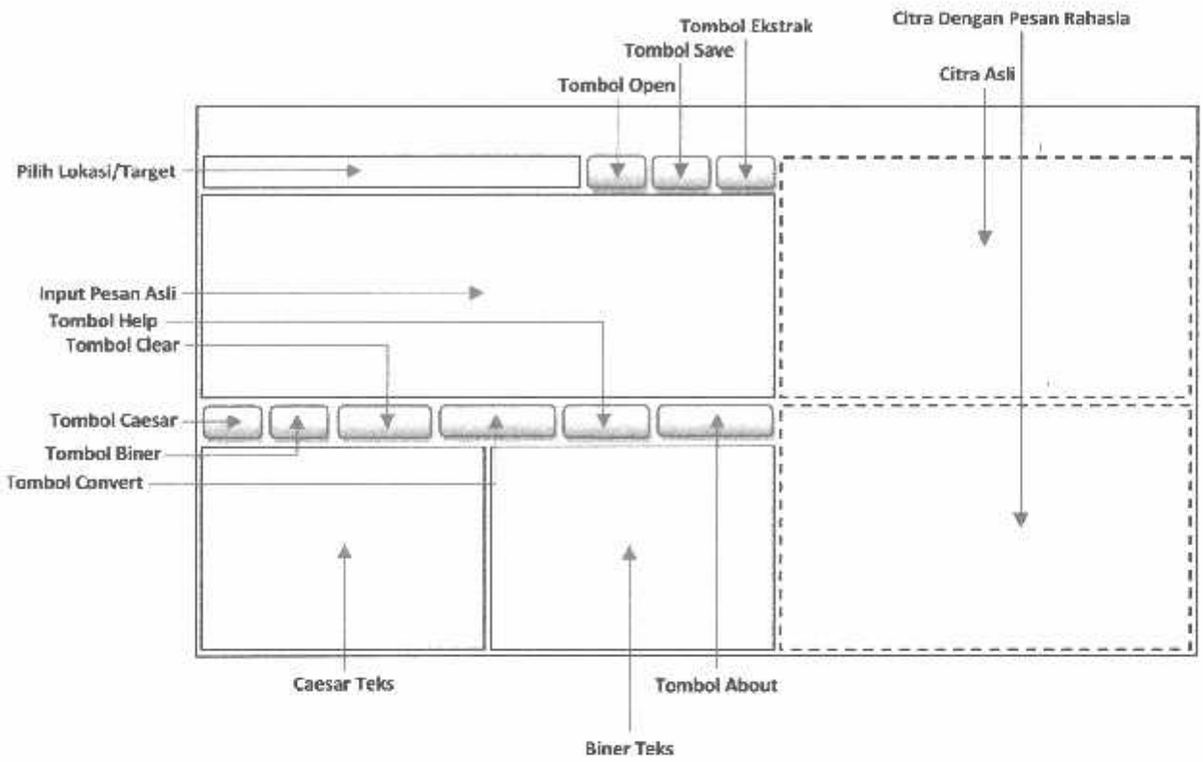
3.6 Deskripsi Sistem

Sistem algoritma steganografi ini merupakan salah satu solusi untuk mengamankan kerahasiaan suatu informasi dalam menyisipkan pesan rahasia dengan memanfaatkan kelemahan indra manusia. Sistem ini merupakan kombinasi serta pengembangan algoritma *Caesar* dan konversi biner. Pada aplikasi ini, sistem dirancang dengan menggunakan bahasa pemrograman *Borland Delphi 7*. Pesan yang berisi beberapa karakter ini bisa berupa huruf alfabet kecil (a...z) atau huruf alfabet besar (A…Z), angka (0..9) ataupun simbol – simbol (~! ... +\). Sebelum pesan disisipkan, pesan terlebih dahulu di enkripsi, setelah selesai menjadi chiperteks maka cipherteks tersebut akan disisipkan ke dalam media penampung citra.

Pada program ini tidak bisa menerima masukkan berupa suara (audio), video ataupun berupa file.

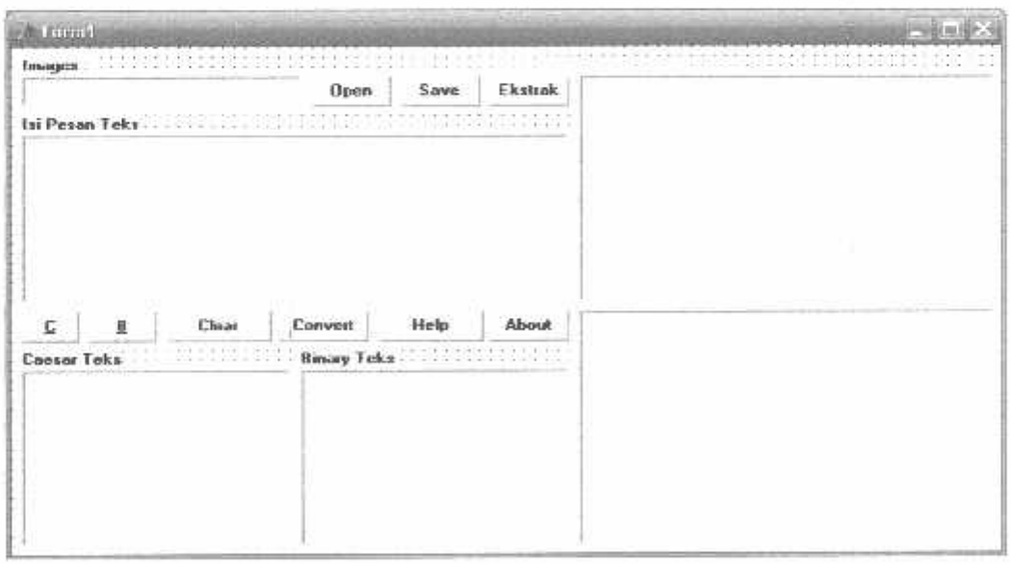
3.7 Perancangan Sistem

3.7.1 Desain Menu



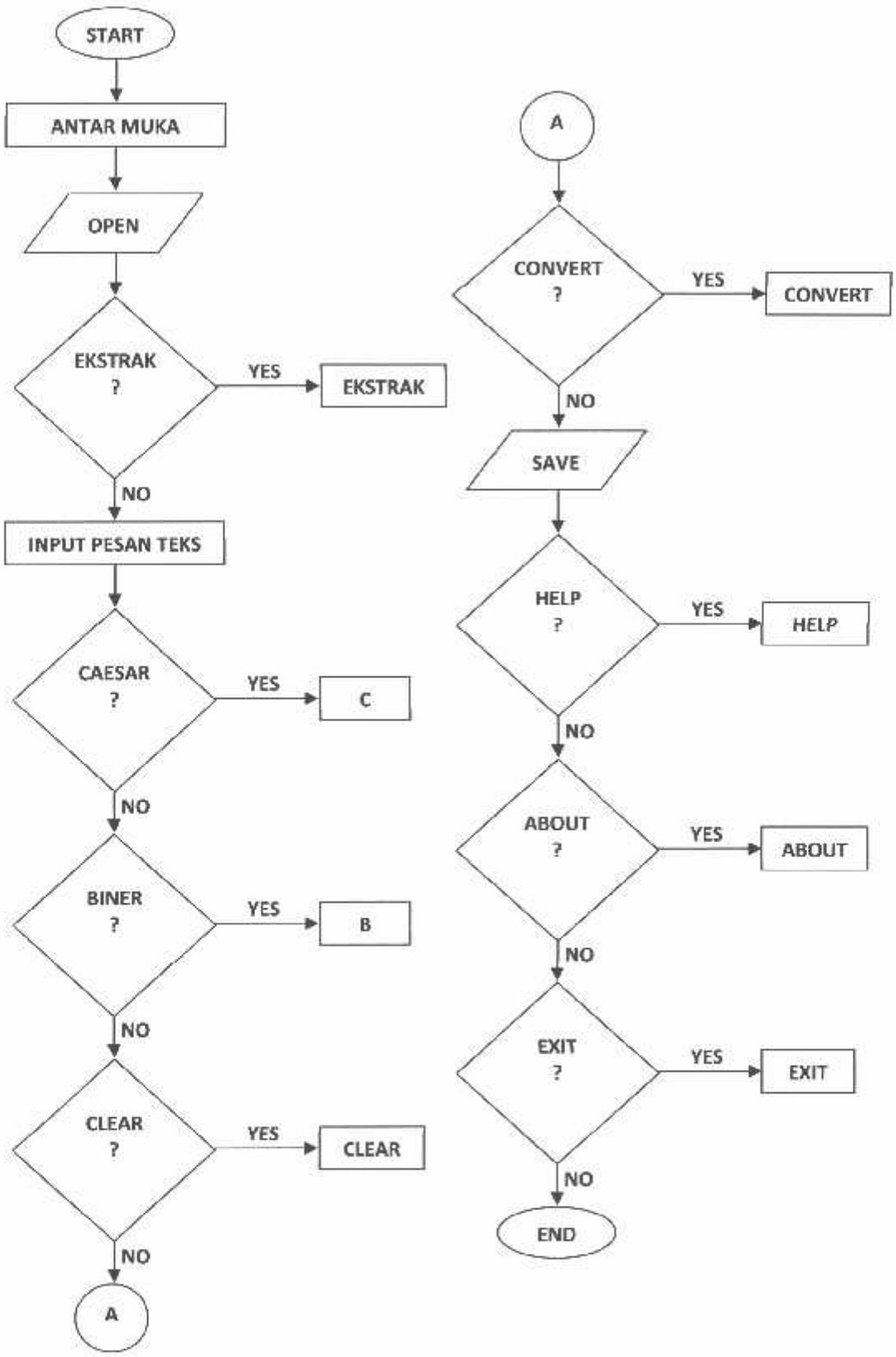
Gambar 3.10 Desain Menu Aplikasi Steganografi

3.7.2 Desain Form Aplikasi Steganografi



Gambar 3.11 Desain Form Aplikasi Steganografi

3.8 Flowchart



Gambar 3.12 Flowchart Aplikasi Steganografi

BAB IV
IMPLEMENTASI

4.1 Kebutuhan Hardware

Perangkat keras (*hardware*) adalah semua alat komputer yang objeknya *real* seperti monitor, CPU dll. Perangkart keras yang dibutuhkan harus di-*install* terlebih dahulu agar Sistem Operasi *Windows* bisa digunakan. Rincian *Software* dan *Hardware* pendukung seperti dalam Tabel 4.1

Tabel 4.1 Spesifikasi Implementasi Perlengkapan

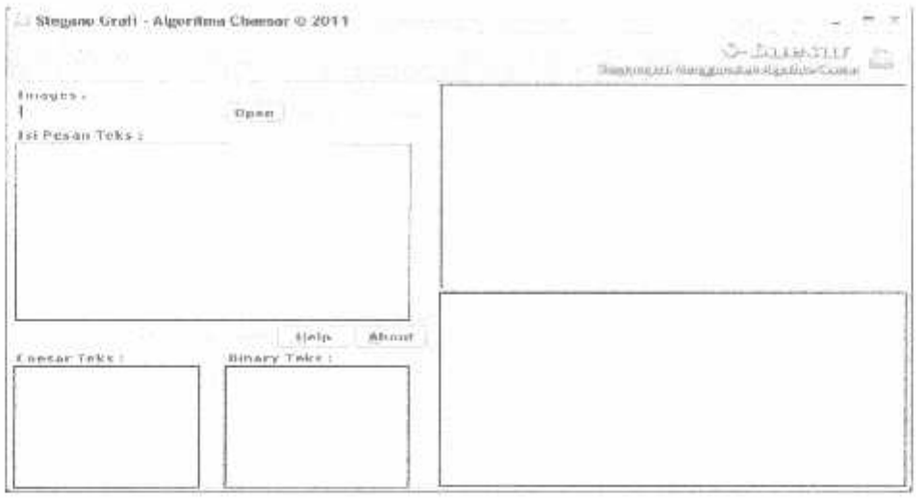
NO	Perlengkapan	Spesifikasi	Keterangan
1	Software	Sistem Operasi	Windows XP Profesional
		Bahasa Pemrograman	Borland Delphi 7 Architect
2	Personal Komputer	Processor	Pentium (R) Dual. 2.2 GHz
		Memori	2 Gb DDR2
		Hardisk	80 Gb
		Kartu Grafis (VGA Card)	Min. 32 Mbyte

4.2. Implementasi Sistem

Implementasi dilakukan dengan menerapkan hasil desain yang telah dbuat ke dalam bahasa pemrograman (*Coding*) *Borland Delphi 7 Architect*, sehingga prosedur – prosedur yang telah dibuat dapar dimengerti oleh mesin sehingga menghasilkan keluaran seperti yang diharapkan.

4.2.1 Form Aplikasi Steganografi

Form ini bentindak sebagai form utama, jadi ketika program pertama kali dijalankan akan muncul Form Aplikasi Steganografi. Gambar 4.1 menunjukkan tampilan *form* pada aplikasi steganografi.

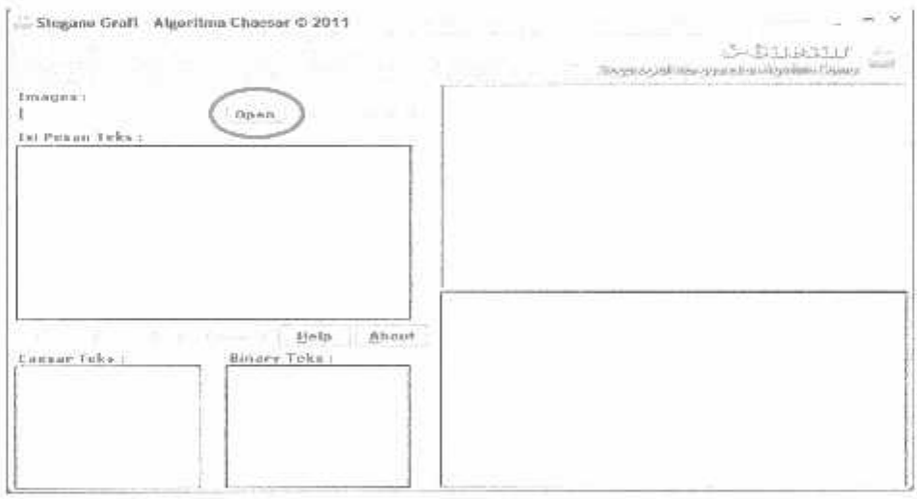


Gambar 4.1 Tampilan *Form* Aplikasi Steganografi

Adapun fungsi masing masing tombol (button) pada *Form* Aplikasi Steganografi sebagai berikut :

1. **Open** Open

   Tombol ini berguna untuk memilih gambar yang akan dijadikan objek untuk menyisipkan pesan rahasia (plainteks). Berikut tampilan tombol *Open* seperti dalam Gambar 4.2

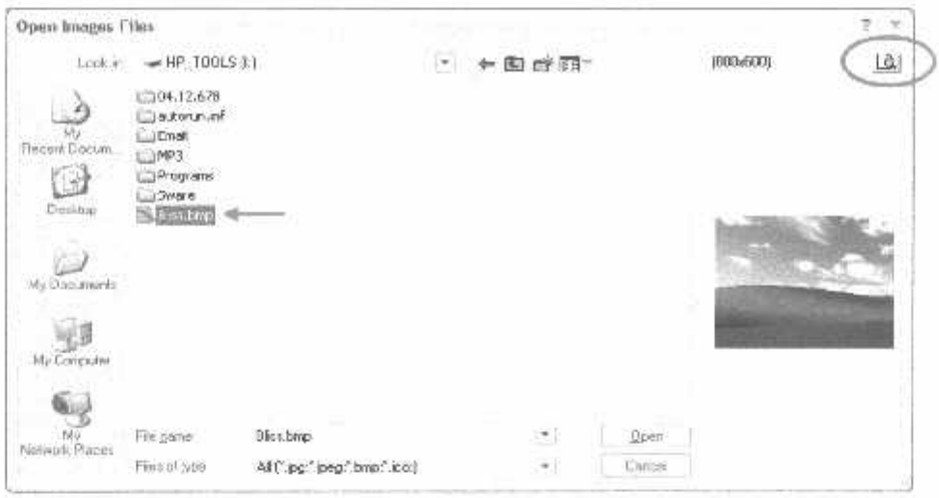


Gambar 4.2 Tampilan Letak Tombol *Open*

Gambar 4.3 Tampilan Jendela Untuk Mencari Lokasi Gambar
Yang Akan Disisipkan Pesan Rahasia (Plainteks)

Ketika gambar terpilih maka secara otomatis aplikasi akan menampilkan gambar serta kapasitas (ukuran) dari gambar tersebut dan juga tombol *Preview* yang terletak disebelah kanan atas, adapun fungsi dari tombol *preview* ini yaitu untuk melihat gambar secara utuh, seperti tampak dalam Gambar 4.4

Gambar 4.4 Tampilan Saat Gambar Dipilih

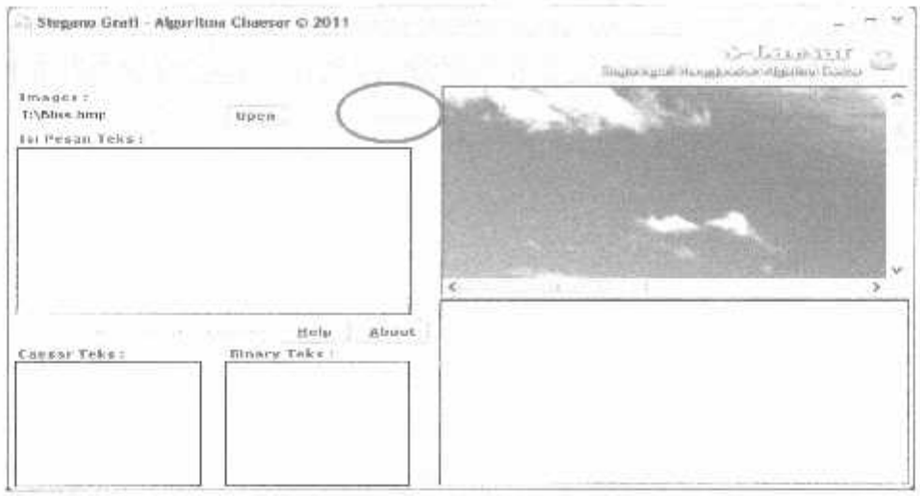


Gambar 4.5 Tampilan Letak Tombol *Preview*

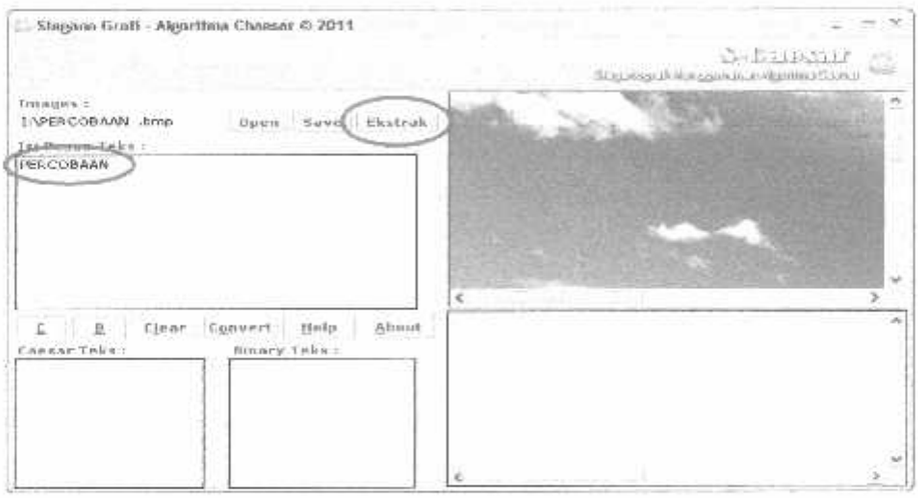
Gambar 4.6 Tampilan Saat Tombol *Preview* Ditekan

2. Ekstrak Ekstrak

Tombol ini digunakan untuk mengeluarkan pesan rahasia (plainteks) yang telah disisipkan pada gambar, tombol *Ekstrak* akan aktif hanya apabila pada gambar yang dipilih telah terdapat pesan rahasia. Berikut perbedaan tombol *Ekstrak* seperti pada gambar di bawah ini.

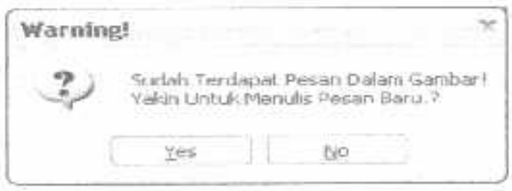


Gambar 4.7 Tampilan Gambar Yang Tidak Ada Pesannya



Gambar 4.8 Tampilan Saat Tombol *Ekstrak* Ditekan

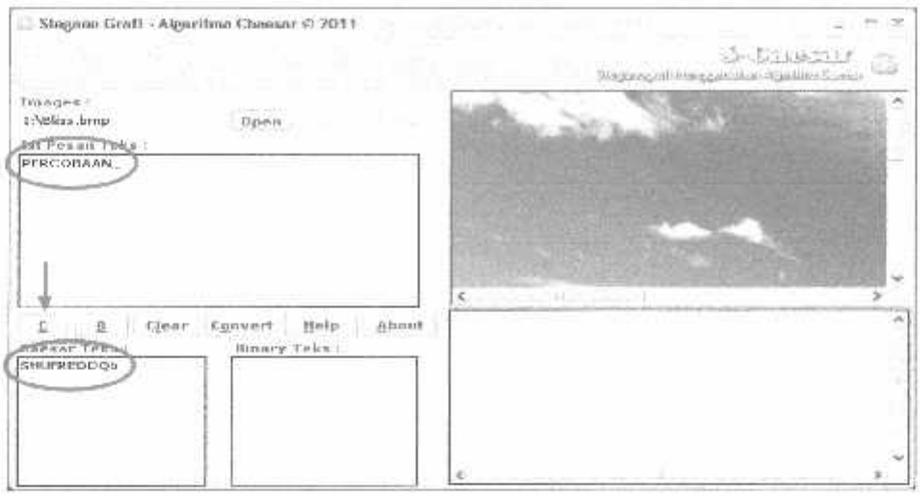
Setelah pesan rahasia (plainteks) dikeluarkan dari gambar, kemudian langkah berikutnya adalah menambahkan pesan lagi atau tidak, apabila akan menambahkan pesan lagi maka setelah di *Convert* akan muncul kotak dialog yang berisikan pemberitahuan agar melanjutkan atau tidak, seperti dalam Gambar 4.9



Gambar 4.9 Tampilan Kotak Dialog

3. **C (*Caesar*)** C

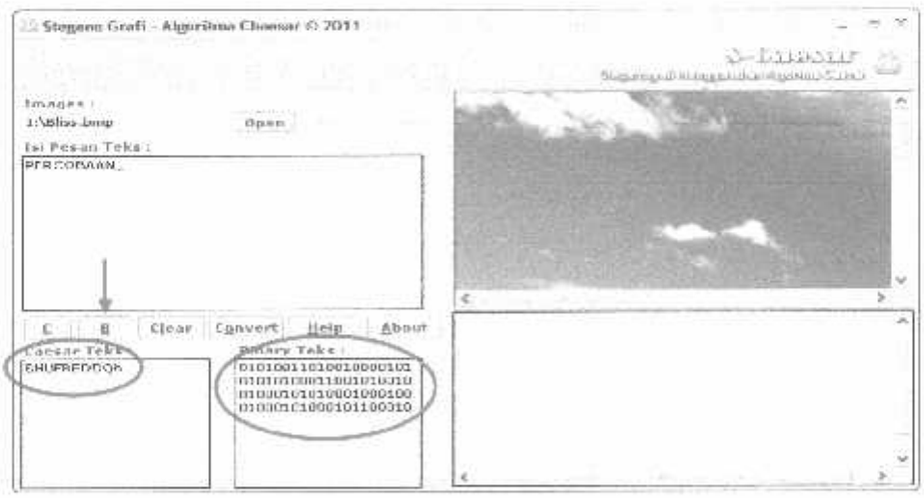
Tombol ***C*** digunakan untuk mengacak pesan asli (chiperteks) menjadi pesan sandi (chiperteks). Tombol ***C*** akan aktif apabila pada kolom pesan asli telah terisi seperti yang tampak dalam Gambar 4.10



Gambar 4.10 Tampilan Saat Tombol ***C*** Ditekan

4. **B (Biner)** B

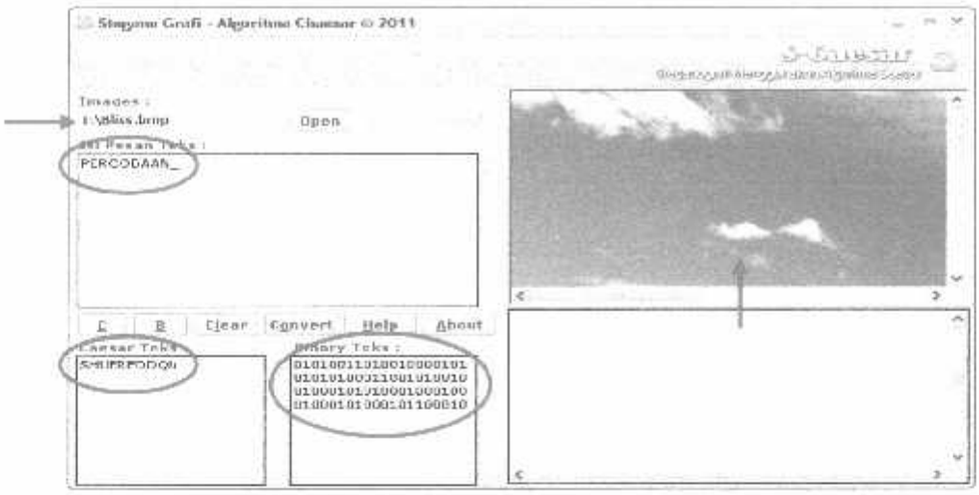
Tombol ***B*** digunakan untuk mengacak pesan sandi (plainteks) menjadi kode biner. Tombol ***B*** akan aktif apabila pada kolom pesan asli telah terisi seperti yang tampak dalam Gambar 4.11



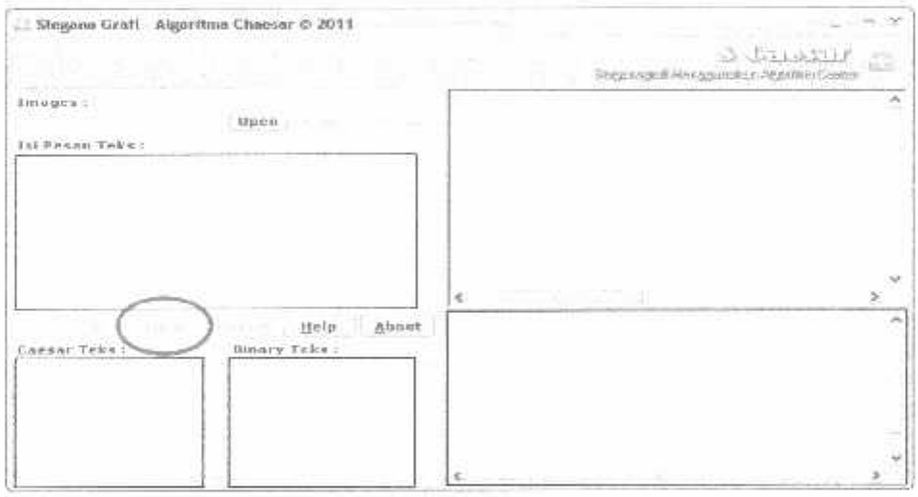
Gambar 4.11 Tampilan Saat Tombol ***B*** Ditekan

5. **Clear** Clear

Tombol ini digunakan untuk menghapus semua karakter yang telah di masukan seperti yang tampak dalam Gambar 4.12



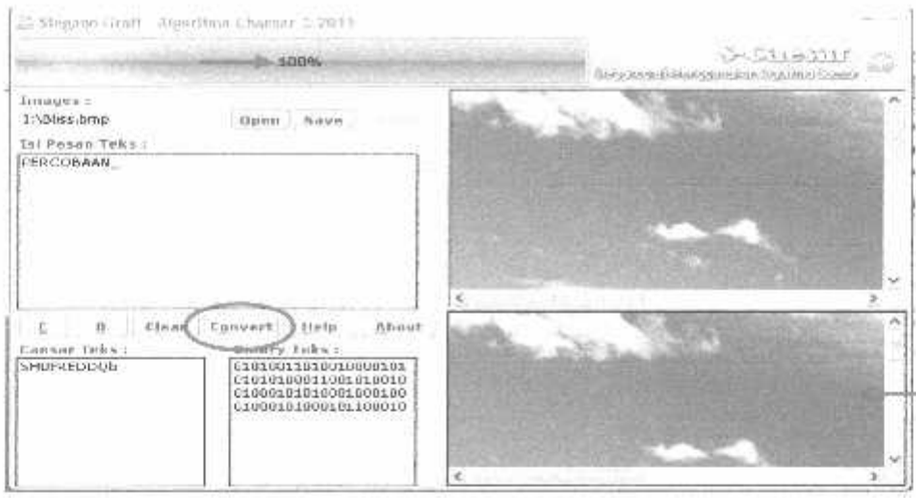
Gambar 4.12 Tampilan Sebelum Tombol *Clear* Ditekan



Gambar 4.13 Tampilan Setelah Tombol *Clear* Ditekan

6. **Convert** Convert

Tombol ini berfungsi untuk memasukan pesan asli (plainteks) yang telah di acak menjadi kode biner dan kemudian disisipkan kedalam gambar, seperti yang ditampilakan dalam Gambar 4.14



Gambar 4.14 Tampilan Setelah Tombol *Convert* Ditekan

Setelah pesan asli disisipkan ke dalam gambar, kemudian akan muncul kotak pemberitahuan yang menampilkan pesan telah berhasil di *Convert* kemudian gambar yang telah disisipkan pesan rahasia (chiperteks) akan muncul pada jendela disebelah kanan. Berikut tampilannya seperti dalam Gambar 4.15

Gambar 4.15 Tampilan Kotak Dialog

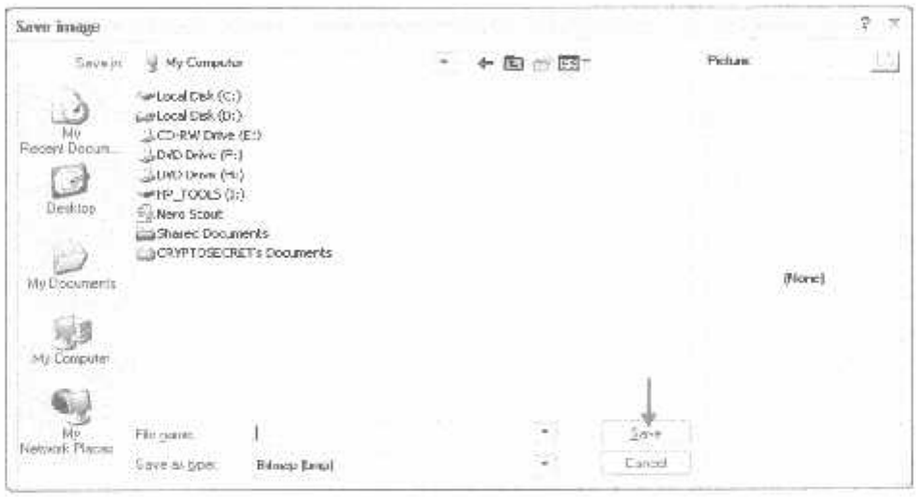
Gambar 4.16 Tampilan Kotak Dialog

Pada saat akan menyisipkan pesan asli (plainteks), pesan yang diinputkan tidak boleh melebihi kapasitas (ukuran) dari gambar yang akan dijadikan media penampung apabila pesan yang diinputkan melebihi kapasitas, maka pesan tidak dapat disisipkan. Seperti dalam Gambar 4.16

Save Save

Tombol ini berfungsi untuk menyimpan gambar hasil dari proses penyisipan pesan rahasia, gambar tidak dapat disimpan apabila belum disisipkan pesan didalamnya. Gambar 4.17 menampilkan jendela dialog apabila pesan belum disisipkan.

Gambar 4.17 Tampilan Kotak Dialog

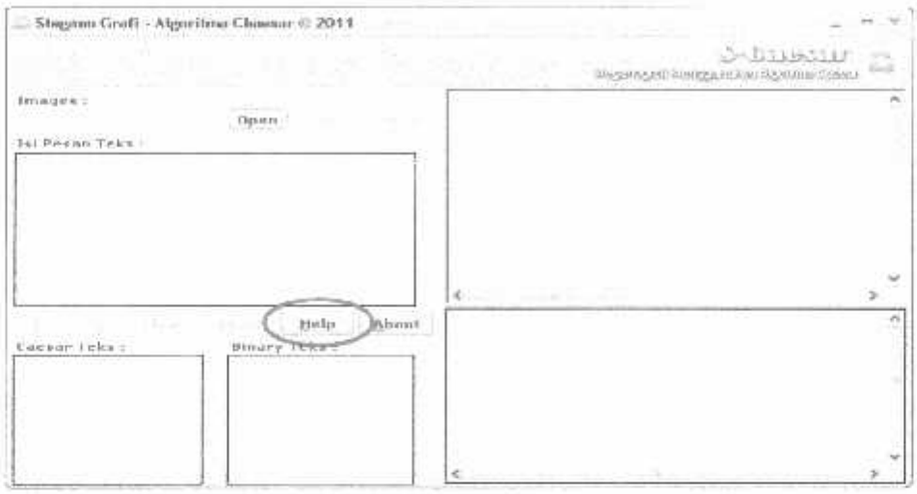


Gambar 4.18 Tampilan Jendela Untuk Mencari
Lokasi Penyimpanan Gambar Yang Telah Disisipi Pesan Rahasia

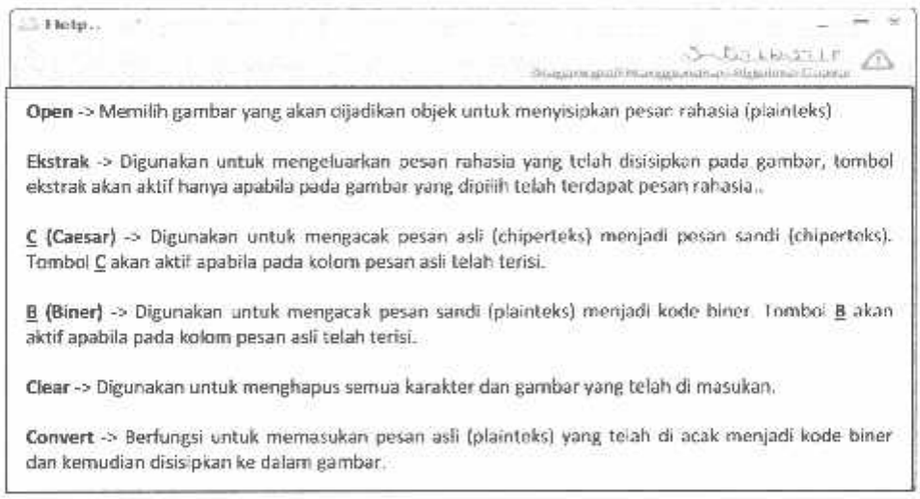
Gambar 4.19 Tampilan Kotak Dialog

7. **Help** Help

Tombol *Help* berfungsi untuk membuka *Help.rft* yang diletakan dalam suatu direktori dengan perangkat lunak ini. Tombol *Help* ini berisikan tentang petunjuk penggunaan dari aplikasi ini, berikut letak tombol *Help* seperti dalam Gambar 4.20



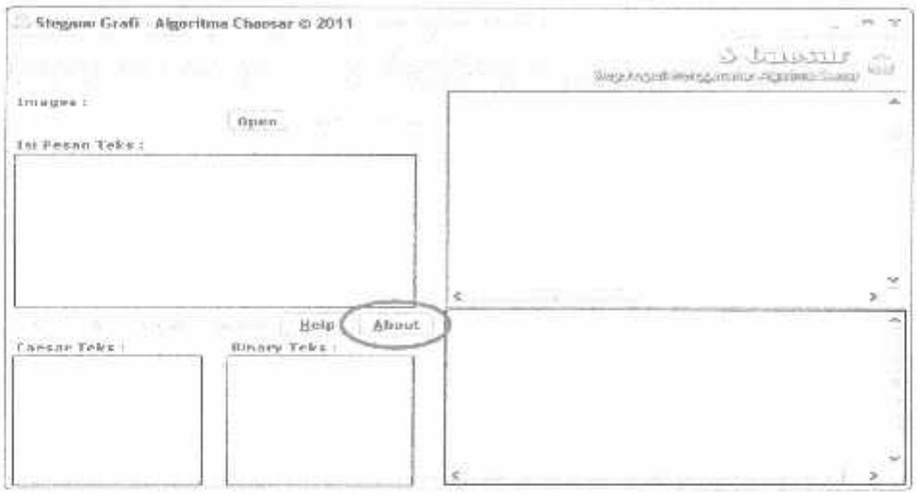
Gambar 4.20 Tampilan Letak Tombol *Help*



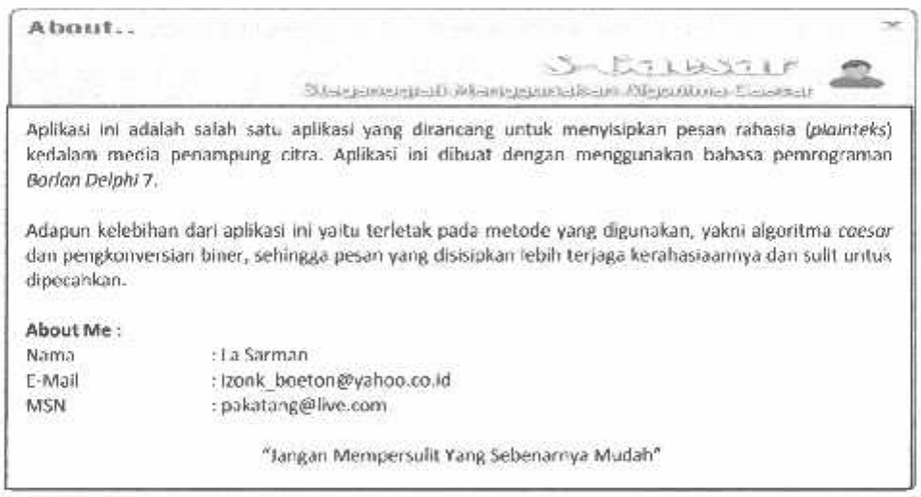
Gambar 4.21 Tampilan Setelah Tombol *Help* Ditekan

8. About About

Tombol ini berfungsi untu menampilkan penjelasan tentang aplikasi steganografi ini, dan informasi tentang pembuatnya serta informasi – informasi yang bersifat ringan seperti dalam Gambar 4.22



Gambar 4.22 Tampilan Letak Tombol *About*



Gambar 4.23 Tampilan Setelah Tombol *About* Ditekan

4.2.2 Coding

Secara garis besar penulisan program (*coding*) pada aplikasi steganografi ini terbagi menjadi dua bagian, yakni :

- Enkripsi / Dekripsi
- Dekripsi Biner

1. Coding Enkripsi Caesar

Digunakan untuk mengacak suatu pesan menjadi pesan yang sulit dimengerti atau biasa juga disebut pesan rahasia (cipherteks). Berikut penggalan *coding* untuk Enkripsi :

```
function CaesarEncrypt(s:String; keys:Integer): String;
var
   i : Integer;
begin
   for i:=1 to Length(s) do
    result:=result+Chr(Ord(s[i])+keys);
end;
```

2. Coding Dekripsi Caesar

Digunakan untuk mengembalikan pesan rahasia (cipherteks) menjadi pesan yang bisa dimengerti. Berikut penggalan *coding* untuk Enkripsi :

```
function CaesarDecrypt(s:String; n:Integer): String;
var
   i : Integer;
begin
   for i:=1 to Length(s) do
     result:=result+Chr(Ord(s[i])-n);
end;
```

3. Coding Konversi Bilangan

Digunakan untuk merubah bilangan desimal menjadi bilangan basis dua Berikut penggalan *coding* untuk konversi bilangan desimal menjadi bilangan basis dua :

```
function DesToBin8(angka:Integer): string;
var
    a, sisa : integer;
    car : string[1];
    hasil : string[8];
begin
    hasil:='00000000';
    a:=0;
    if angka <> 0 then
    begin
        while angka <> 1 do
        begin
            sisa:=angka mod 2;
            str(sisa, car);
            hasil[8-a]:=car[1];
            angka:=trunc(angka/2);
            inc(a);
        end;
        str(angka,car);
        hasil[8-a]:=car[1];
    end;
    Result :=hasil;
end;
```

4. Coding Konversi Bilangan Desimal Ke Bilangan Biner

Digunakan untuk merubah bilangan desimal menjadi bilangan biner Berikut penggalan *coding* untuk konversi bilangan desimal menjadi bilangan biner :

```
function BinToDes(Bit: String): Integer;
var
    i : Integer;
    Hasil: Integer;
begin
    hasil := 0;
    for i := 1 to Length(Bit) do
    begin
     if Bit[i] = '1' then
        hasil := hasil + Trunc(Exp((Length(Bit)-i)*ln(2)));
    end;
```

5. Coding Konversi Bilangan Biner Ke Bilangan Desimal

Digunakan untuk merubah bilangan desimal menjadi bilangan biner Berikut penggalan *coding* untuk konversi bilangan desimal menjadi bilangan biner :

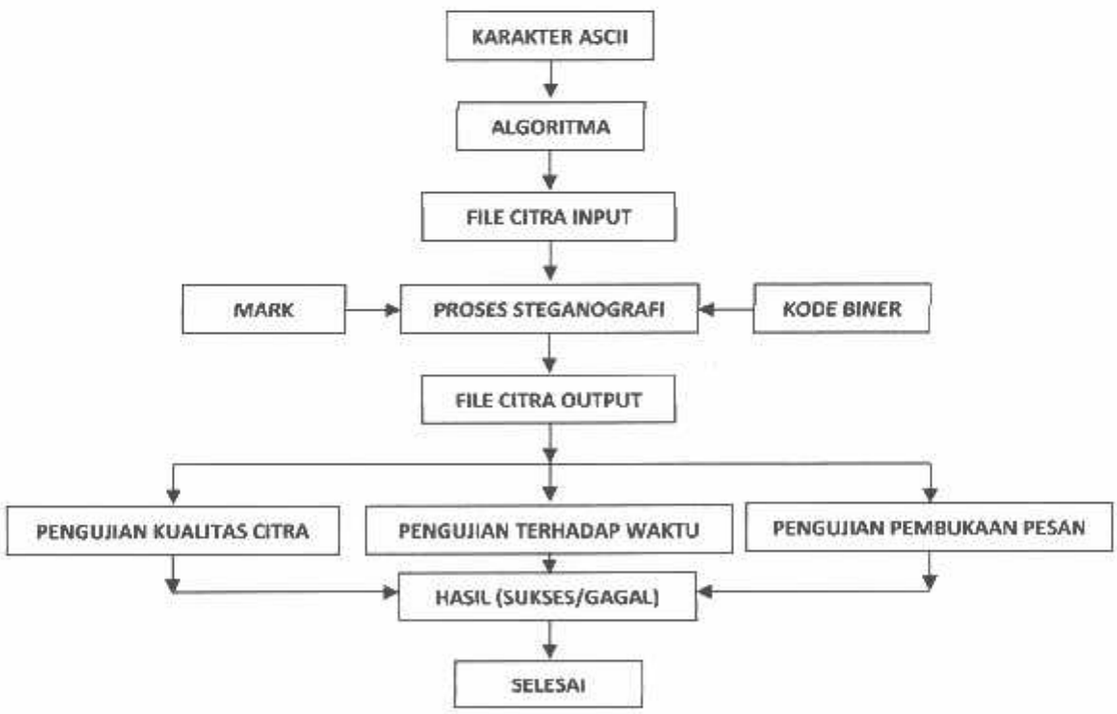
```
Function DesToNBin (Des :integer):String;
var
 Hasil,Temp : String;
begin
  if des = 0 then
  begin
   Result :='0';
   exit;
  end;
```

```
Hasil := '';
  Temp :='';
  repeat
      str(des mod 2, temp);
      Hasil:=temp+Hasil;
      des:=des div 2;
  until des=0;
  Result := Hasil;
end;
```

4.3 Pengujian Sistem

4.3.1 Pengujian Perangkat Lunak

Perangkat lunak ini didesain pada lingkungan *windows xp professional* dan akan diuji dengan menggunakan beberapa buah citra uji dengan resolusi yang berbeda – beda. Lihat diagram blok pengujian perangkat lunak dalam Gambar 4.24



Gambar 4.24 Diagram Blok Pengujian Perangkat Lunak

Pengujian pertama dilakukan dengan cara melakukan penyisipan dan kemudian melakukan pembukaan pesan rahasia (plainteks) dengan menggunakan perangkat lunak ini. Pada pengujian ini jika pesan yang telah disisipkan berhasil dibuka kembali sesuai aslinya maka hasil pengujian dinyatakan sukses.

Jika pengujian dan pembukaan pada pengujian pertama tadi dinyatakan sukses, kemudian akan dilakukan pengujian kedua yaitu pengujian kualitas citra. Pengujian kualitas citra dilakukan dengan cara melakukan survey pada 9 orang pengamat yang mempunyai penglihatan normal (pada usia 21-24 tahun) untuk melihat kualitas gambar sebelum dan sesudah disisipi pesan rahasia (plainteks). Jika kualitas gambar hasil mengalami perubahan dibandingkan dengan kualitas gambar aslinya, dan disahkan oleh lebih dari setengah jumlah pengamat, maka pengujian dinyatakan gagal.

Spesifikasi komputer yang digunakan untuk pengujian adalah :

- Komputer : Acer Aspire 4530
- Processor : AMD Turion (TM) X2 (2.20 GHz)
- Sitem Memory : 2Gb (DDR-2)
- Display Adapter : NVIDIA G-force 9100MG
- Disk Drive : 120Gb
- Images View : Windows Picture and Fax Viewer
- System Operasi : Windows XP Professional

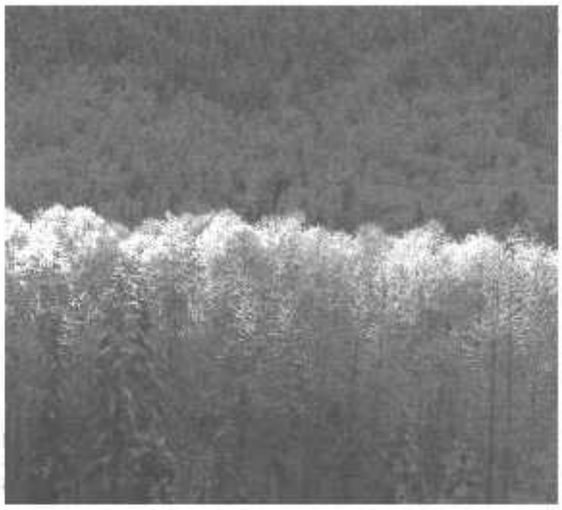
Berikut ini akan dilakukan pengujian terhadap duabelas buah citra uji yang nantinya keduabelas citra uji ini akan disisipkan sejumlah karakter dengan jumlah yang bervariasi mulai dari 1.000 karakter hingga 500.000 karakter, hal ini dimaksudkan untuk mengetahui seberapa besar perubahan yang terjadi pada citra uji yang akan diukur.

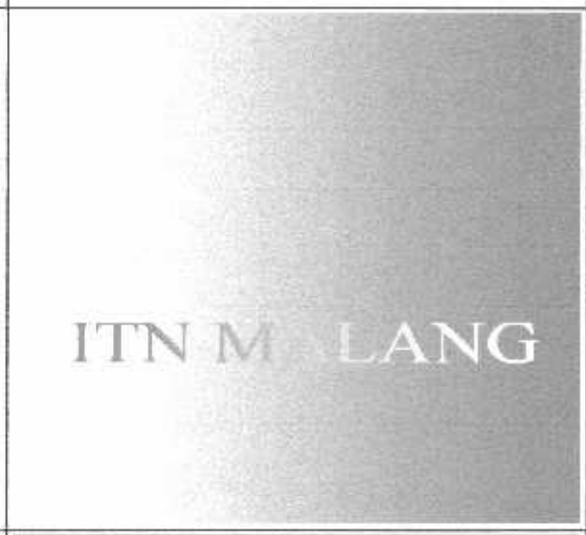
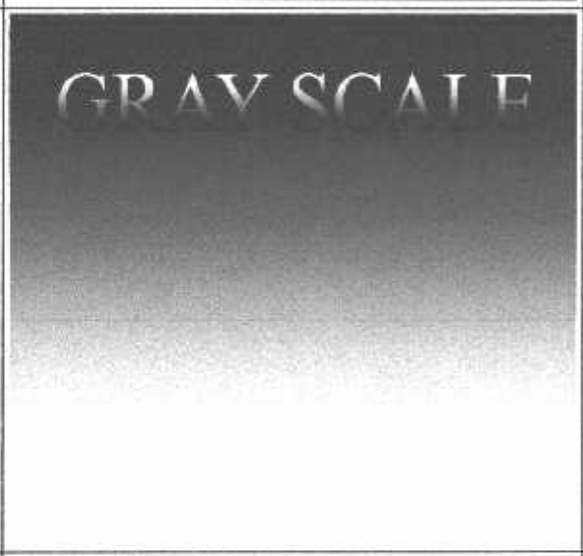
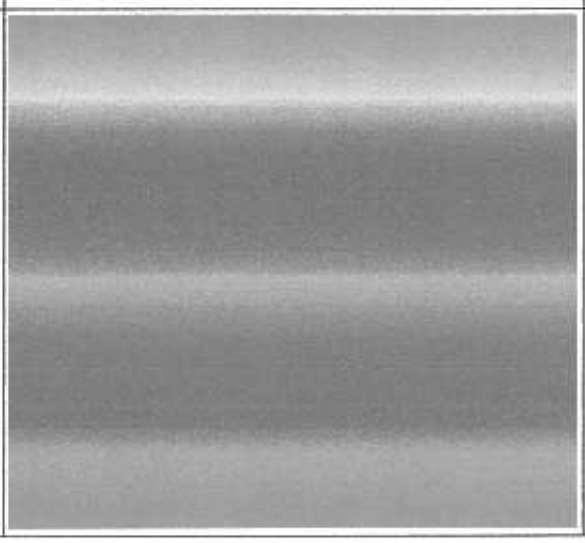
Citra pengujian memiliki ukuran yang bervariasi, yang diharapkan dapat menunjukkan kemampuan aplikasi steganografi yang dibuat terhadap berbagai macam ukuran citra uji.

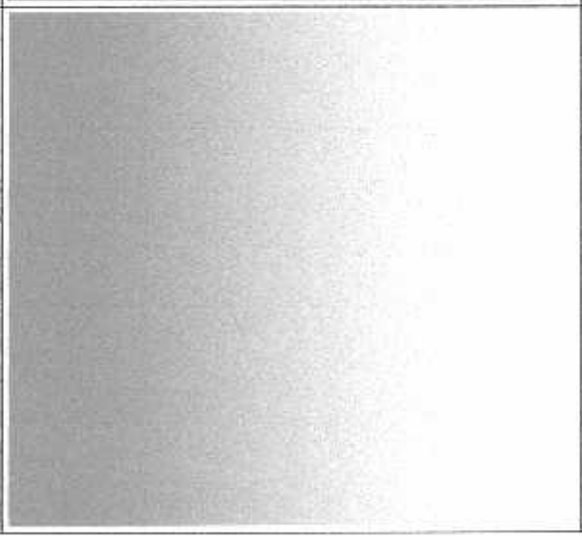
Tabel 4.2 menunjukkan citra uji yang digunakan beserta deskripsi terhadap citra tersebut yang diperlukan dalam proses pengujian.

Tabel 4.2 Deskripsi Citra Uji

NAMA CITRA UJI	GAMBAR CITRA	UKURAN CITRA PIXEL X PIXEL	UKURAN CITRA	DPI
BLISS.BMP		300 X 200	175 KB	71
NICE.JPG		305 X 204	223 KB	71
SUNSET.JPG		400 X 300	351 KB	71

NAMA CITRA UJI	GAMBAR CITRA	UKURAN CITRA PIXEL X PIXEL	UKURAN CITRA	DPI
WINTER .BMP		600 X 500	878 KB	71
WATER LILIES .JPG		250 X 170	124 KB	71
BLUE HILLS .JPG		440 X 320	412 KB	71

NAMA CITRA UJI	GAMBAR CITRA	UKURAN CITRA PIXEL X PIXEL	UKURAN CITRA	DPI
ITN.JPG		1024 X 768	180 KB	72
GRAY SCALE JPG		900 X 600	128 KB	72
RAINBOW JPG		900 X 600	143 KB	72

NAMA CITRA UJI	GAMBAR CITRA	UKURAN CITRA PIXEL X PIXEL	UKURAN CITRA	DPI
CITRA.JPG		800 X 600	142 KB	72
STEGO.JPG		600 X 300	74 KB	72
BLUE.JPG		700 X 640	95 KB	72

Hasil pengujian yang dilakukan ditampilkan dalam Tabel 4.3

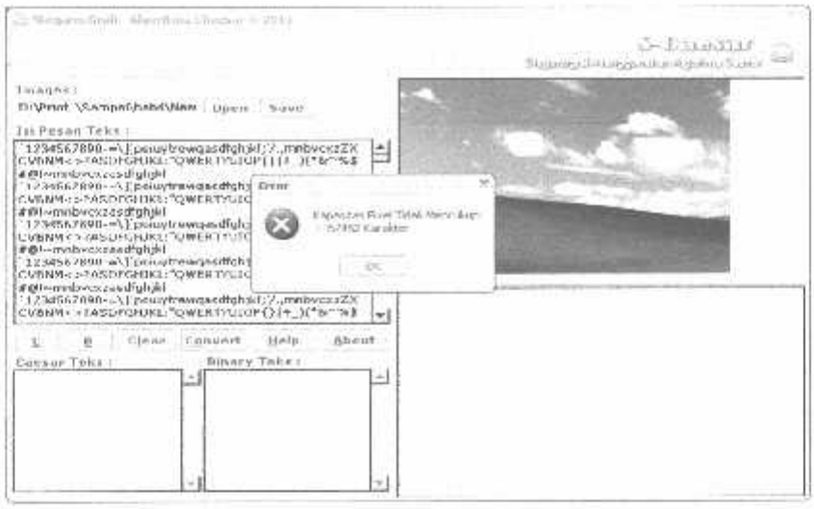
Tabel 4.3 Hasil Percobaan

NO	NAMA CITRA UJI	RESOLUSI (PIXEL X PIXEL)	DPI	KAPASITAS (KILO BYTE)	BANYAK KARAKTER	HASIL (SUKSES)	HASIL (GAGAL)
1	BLISS.BMP	300 X 200	71	175 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	-	GAGAL
2	NICE.JPG	304 X 204	71	223 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	-	GAGAL
3	SUNSET.JPG	400 X 300	71	351 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	-	GAGAL
4	WINTER.BMP	600 X 500	71	878 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	SUKSES	-
5	WATER LILIES.JPG	250 X 170	71	124 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	-	GAGAL
6	BLUE HILLS.JPG	440 X 320	71	412 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	-	GAGAL
7	ITN.JPG	1024 X 768	72	180 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	SUKSES	-

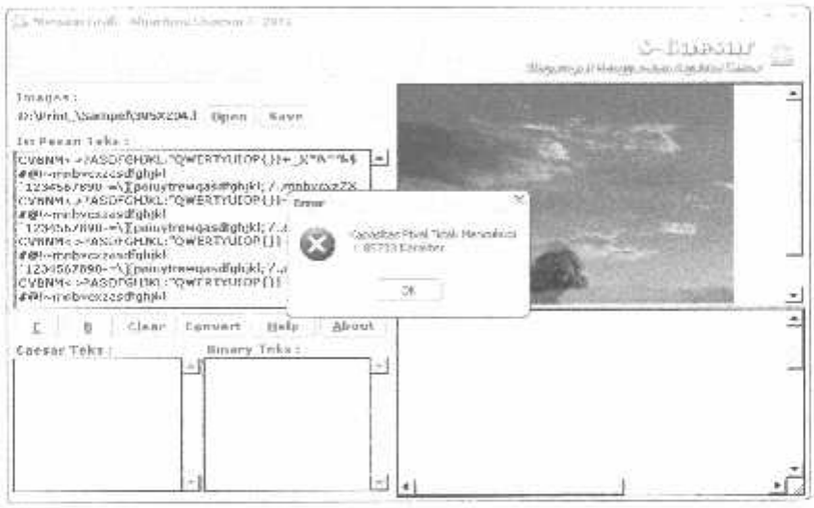
8	GRAY SCALE.JPG	900 X 600	72	128 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	SUKSES	-
9	RAINBOW.JPG	900 X 600	72	143 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	SUKSES	-
10	CITRA.JPG	800 X 600	72	142 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	SUKSES	-
11	STEGO.JPG	600 X 300	72	74 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	-	GAGAL
12	BLUE.JPG	700 X 640	72	95 KB	1.000	SUKSES	-
					5.000	SUKSES	-
					10.000	SUKSES	-
					50.000	SUKSES	-
					100.000	SUKSES	-
					500.000	SUKSES	-

Dari hasil percobaan terhadap keduabelas buah file citra uji yang kemudian disisipkan karakter dengan jumlah yang bervariasi serta jumlah karakter yang terus ditambahkan pada setiap pengujiannya yakni dari 1.000, 5.000, 10.000, 50.000, 100.000 dan 500.000 karakter, diperoleh hasil yang berbeda dari masing - masing file citra uji untuk setiap jumlah karakter yang disisipkan.

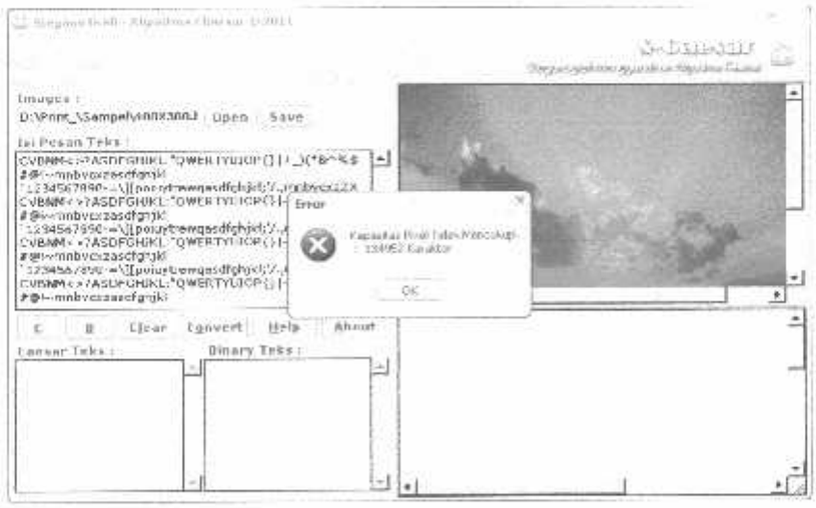
Pada tabel hasil percobaan diatas dapat dilihat bahwa tidak semua dari karakter dapat disisipkan, dikarenakan karakter yang diinputkan terlalu banyak sehingga melebihi kapasitas dari piksel citra uji.



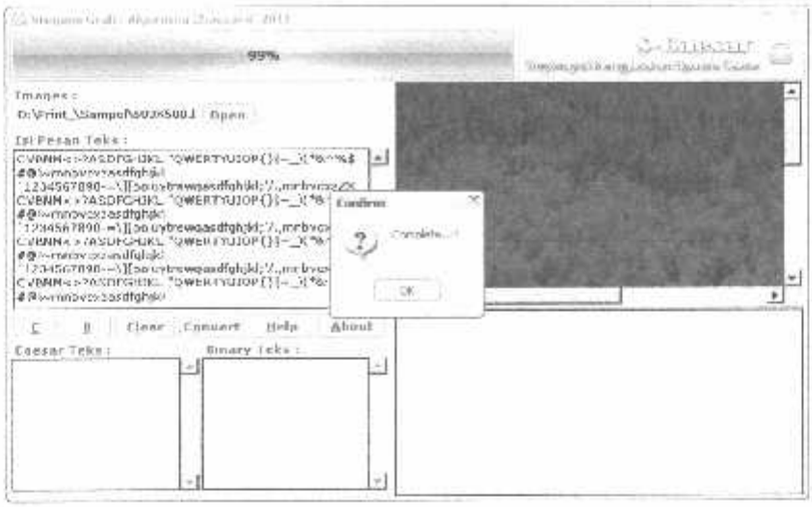
Gambar 4.25 Percobaan Gagal Pada Citra Bliss.BMP



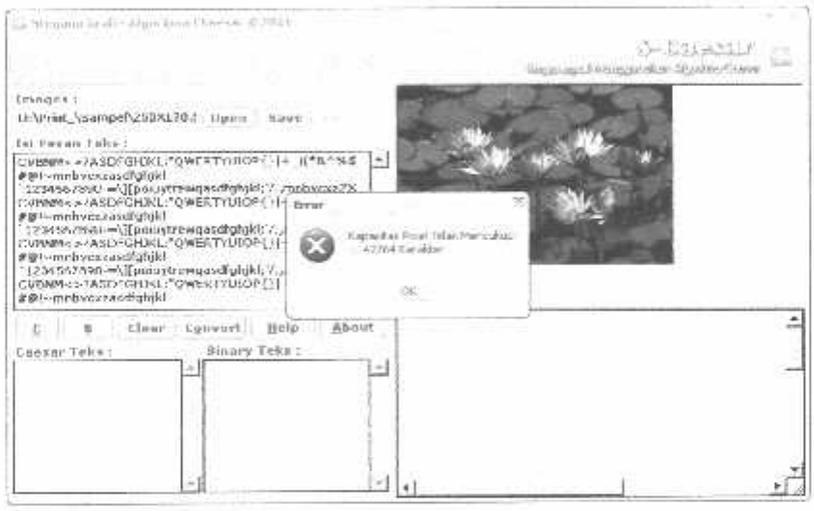
Gambar 4.26 Percobaan Gagal Pada Citra Nice.JPG



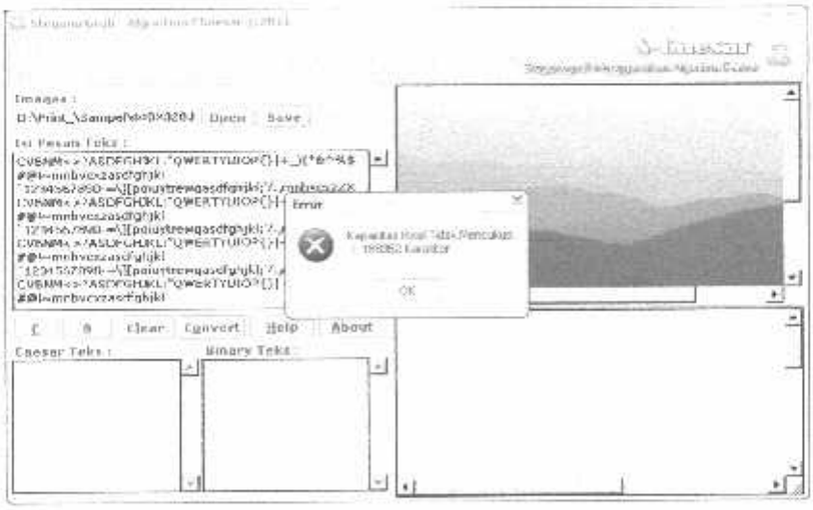
Gambar 4.27 Percobaan Gagal Pada Citra Sunset.JPG



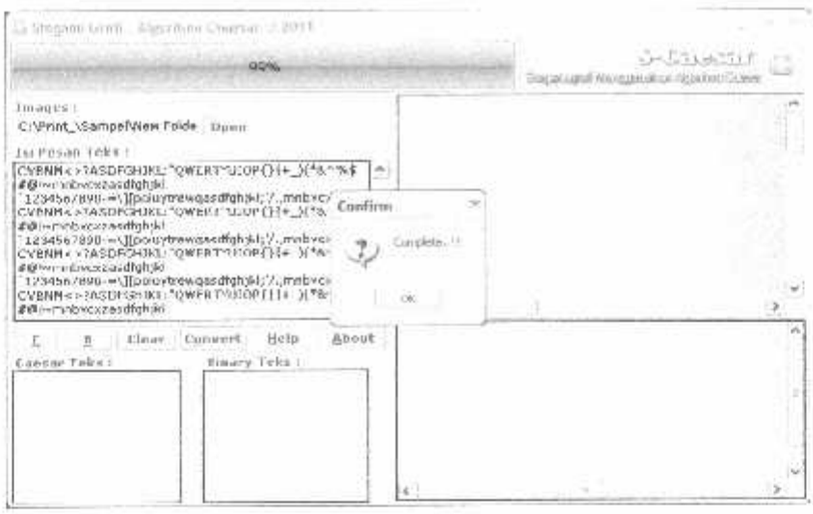
Gambar 4.28 Percobaan Sukses Pada Citra Winter.JPG



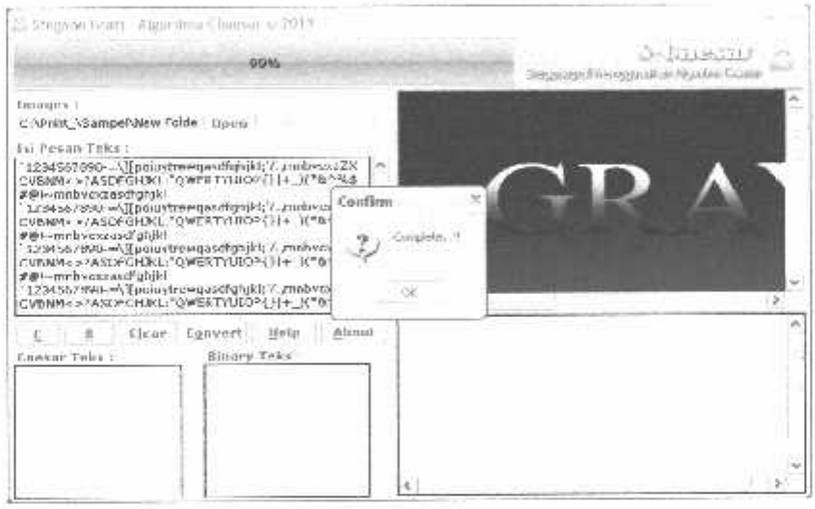
Gambar 4.29 Percobaan Gagal Pada Citra Water Lilies.JPG



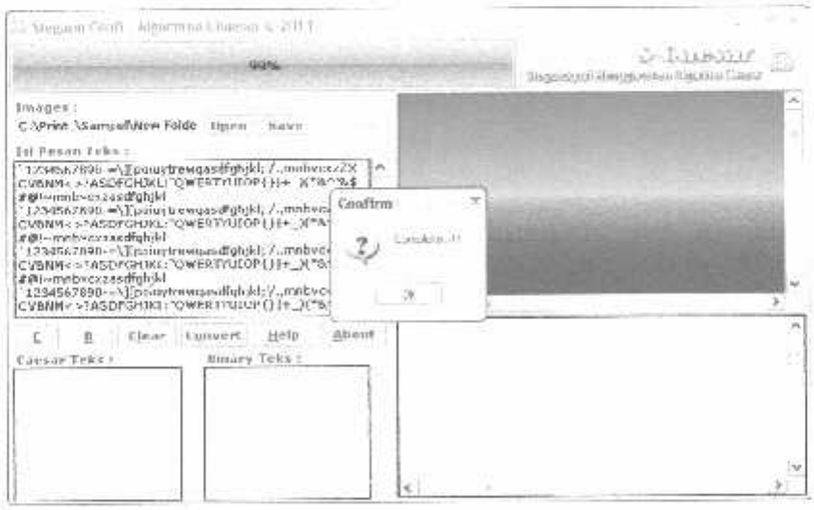
Gambar 4.30 Percobaan Gagal Pada Citra Blue Hills.JPG



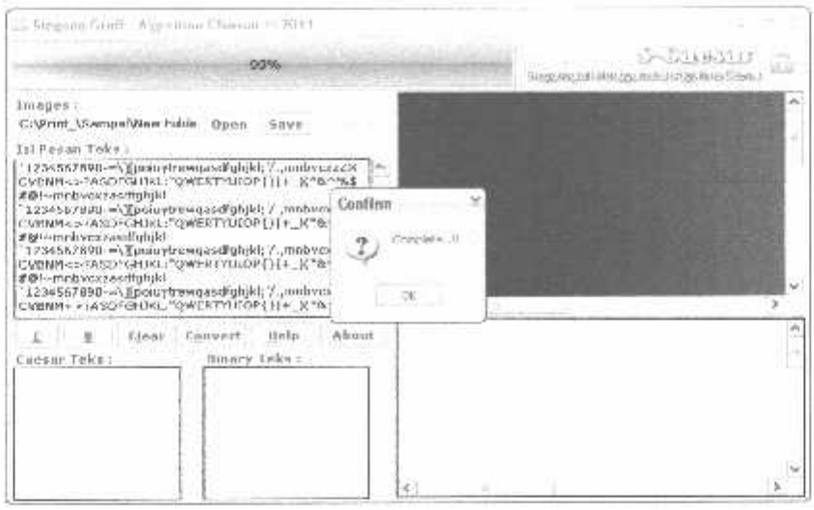
Gambar 4.31 Percobaan Sukses Pada Citra Itn.JPG



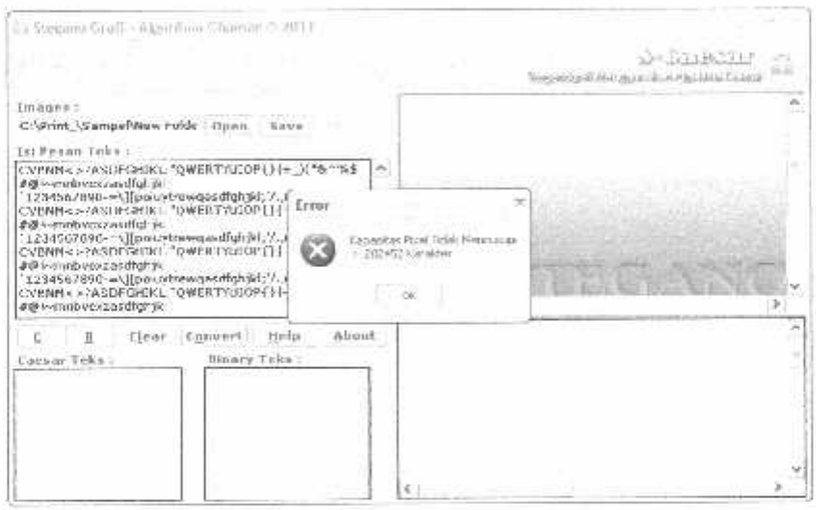
Gambar 4.32 Percobaan Sukses Pada Citra Gray Scale.JPG



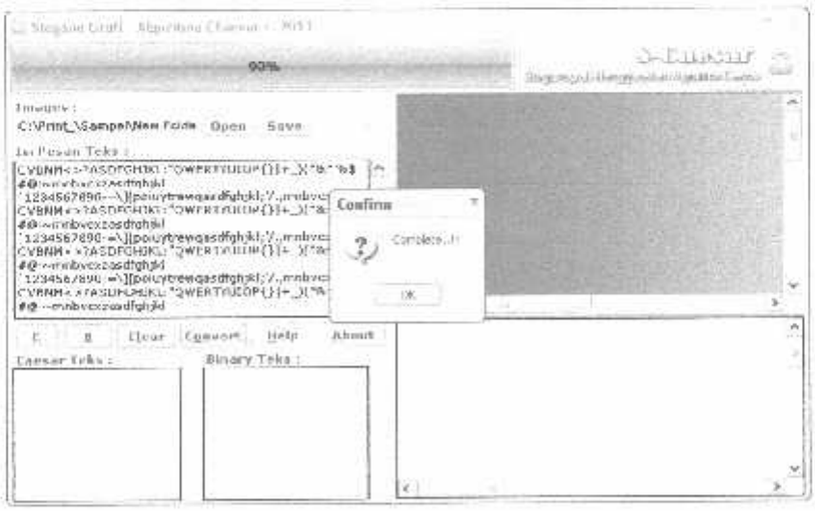
Gambar 4.33 Percobaan Sukses Pada Citra Rainbow.JPG



Gambar 4.34 Percobaan Sukses Pada Citra Citra.JPG



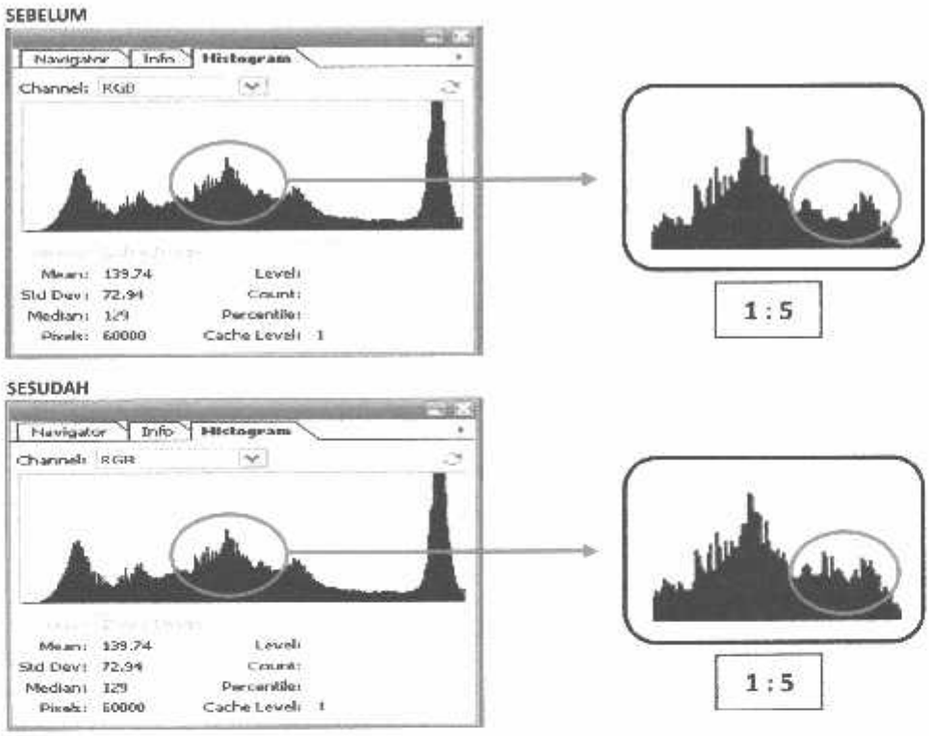
Gambar 4.35 Percobaan Gagal Pada Citra Stego.JPG



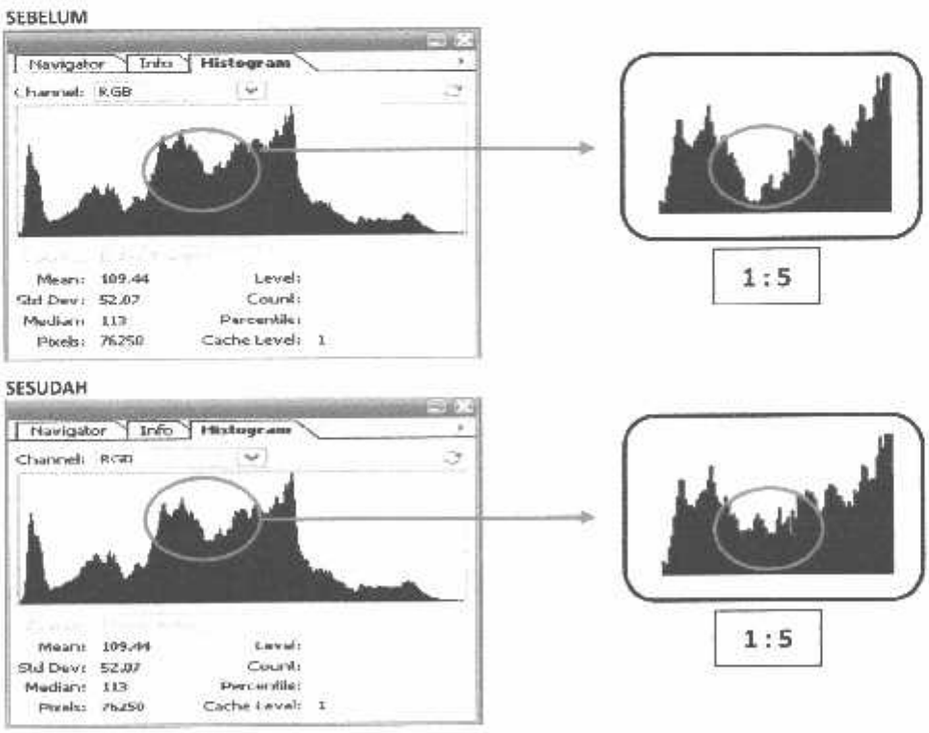
Gambar 4.36 Percobaan Sukses Pada Citra Blue.JPG

4.3.2 Pengujian Kualitas Citra

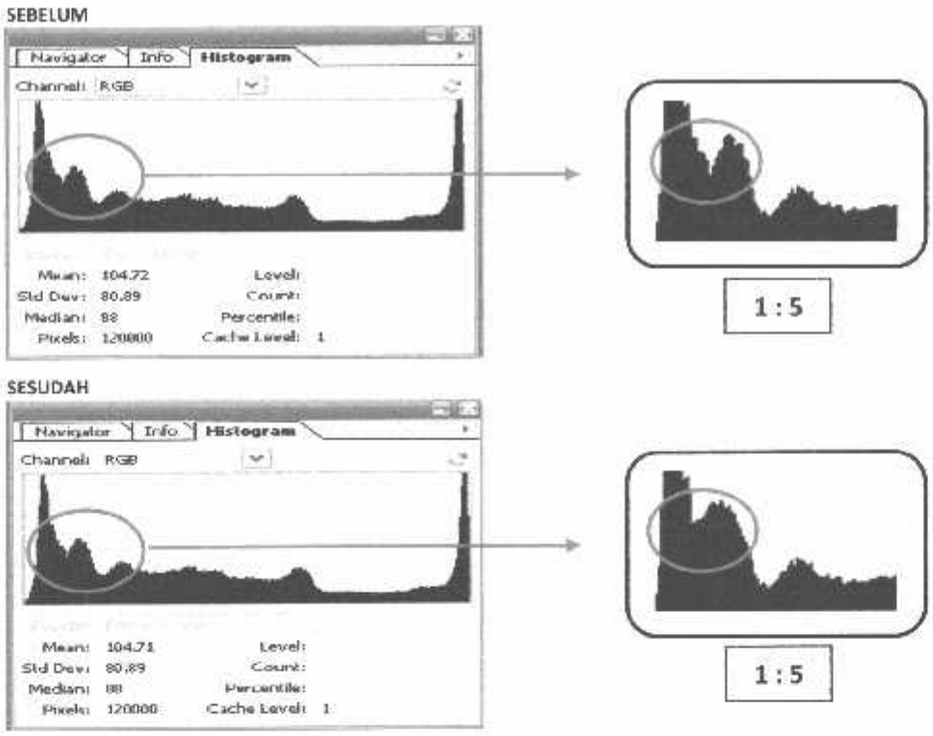
Pengujian ini pertujuan untuk membandingkan kualitas citra uji sebelum dan sesudah disisipi pesan rahasia (plainteks). Serta perubahan yang terjadi pada histogram citra uji maupun citra hasil.



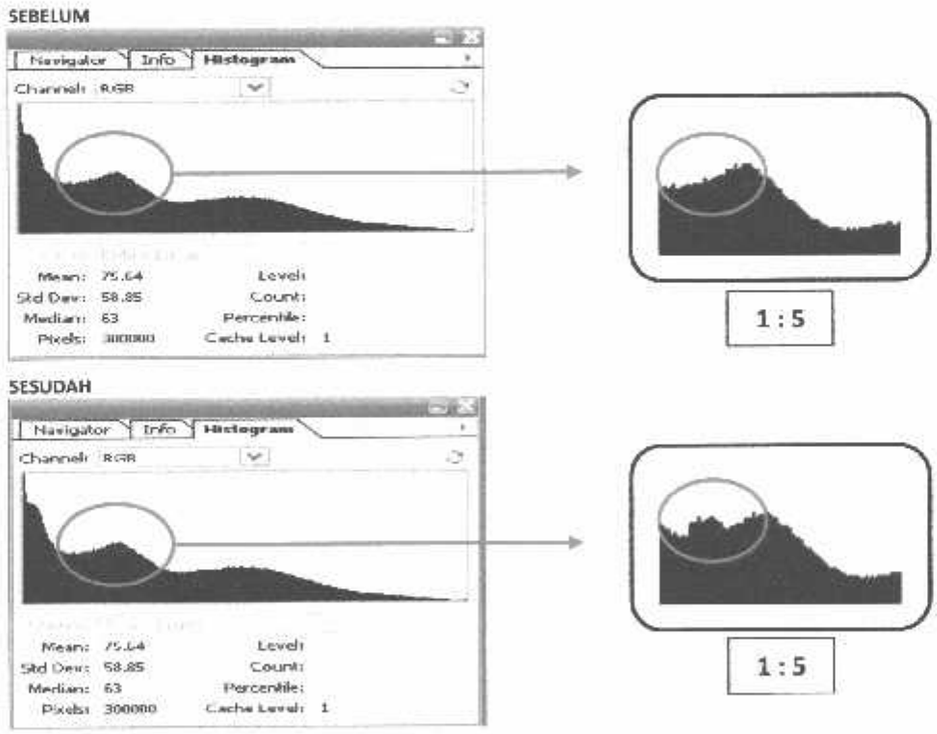
Gambar 4.37 Perbandingan Grafik Histogram Citra Bliss.BMP



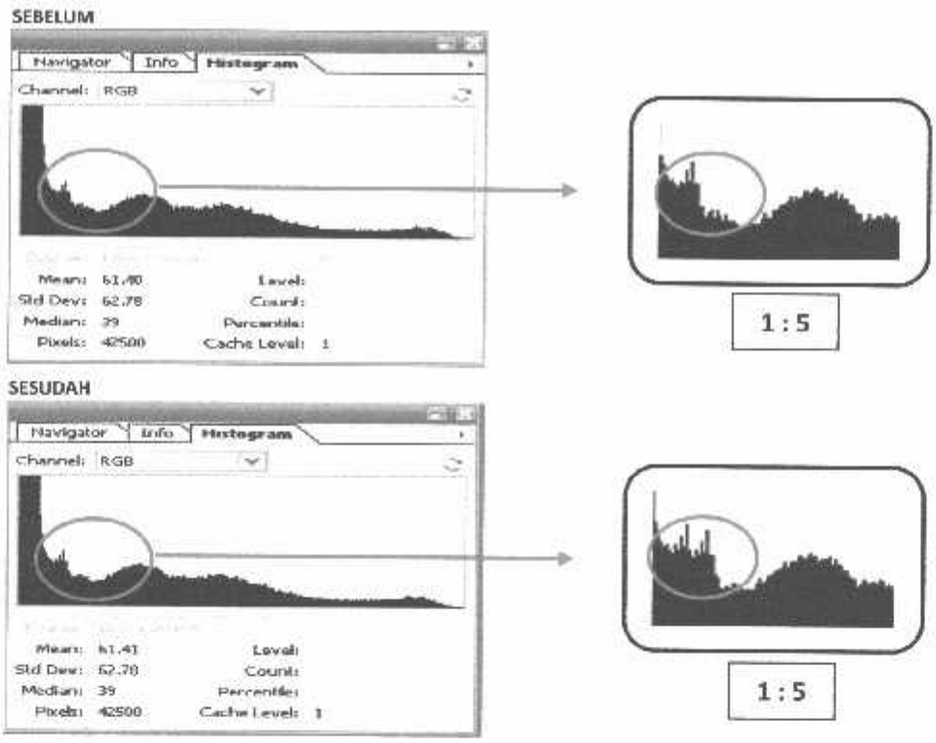
Gambar 4.38 Perbandingan Grafik Histogram Citra Nice.JPG



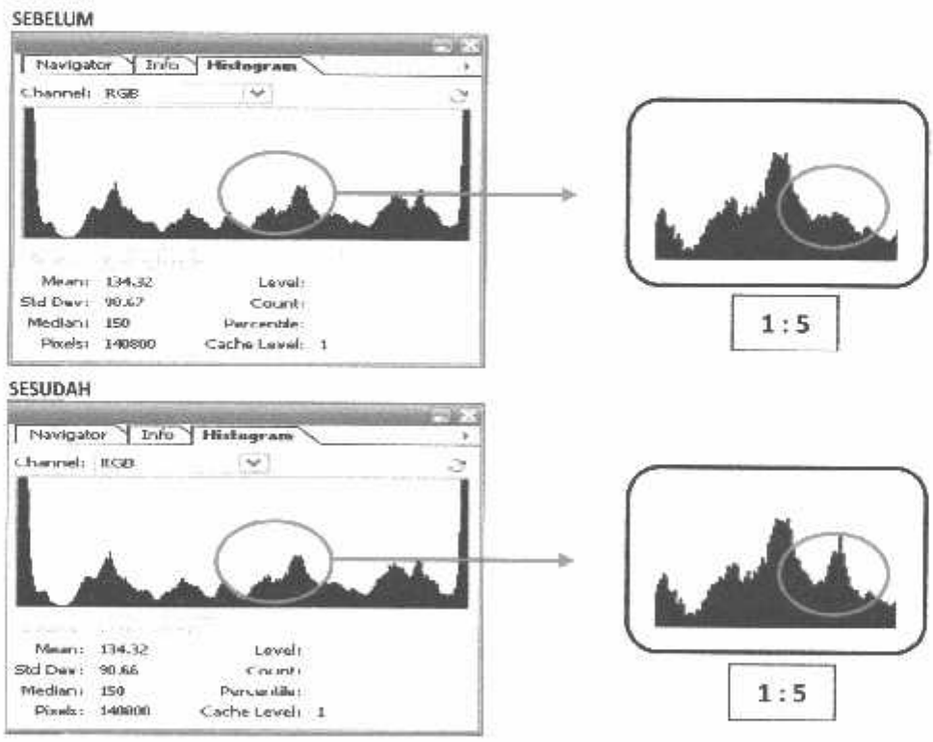
Gambar 4.39 Perbandingan Grafik Histogram Citra Sunset.JPG



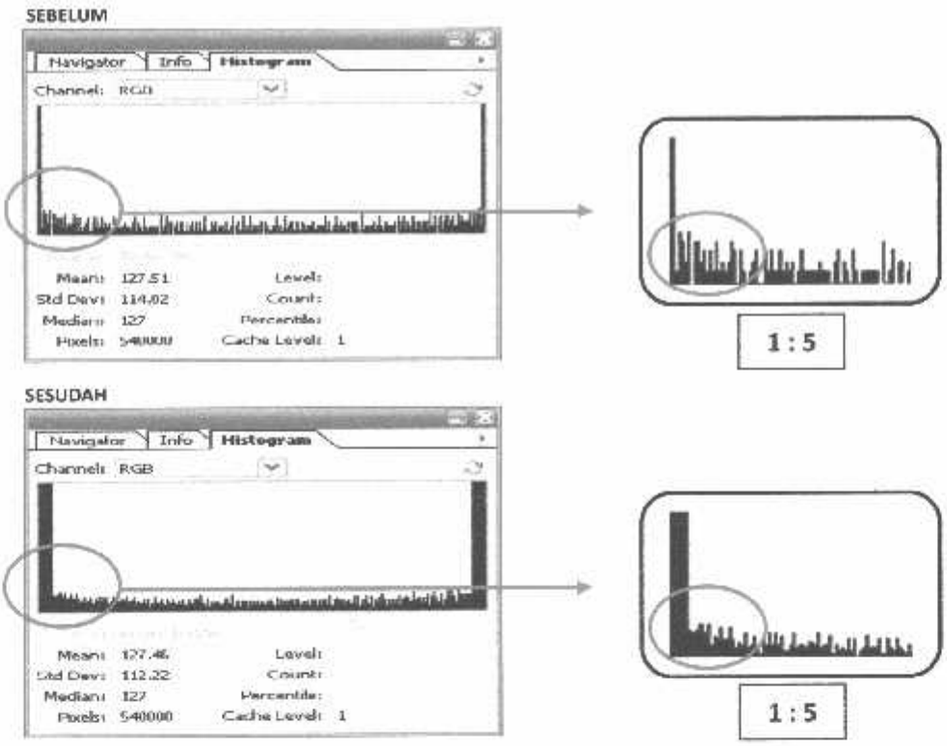
Gambar 4.40 Perbandingan Grafik Histogram Citra Winter.JPG



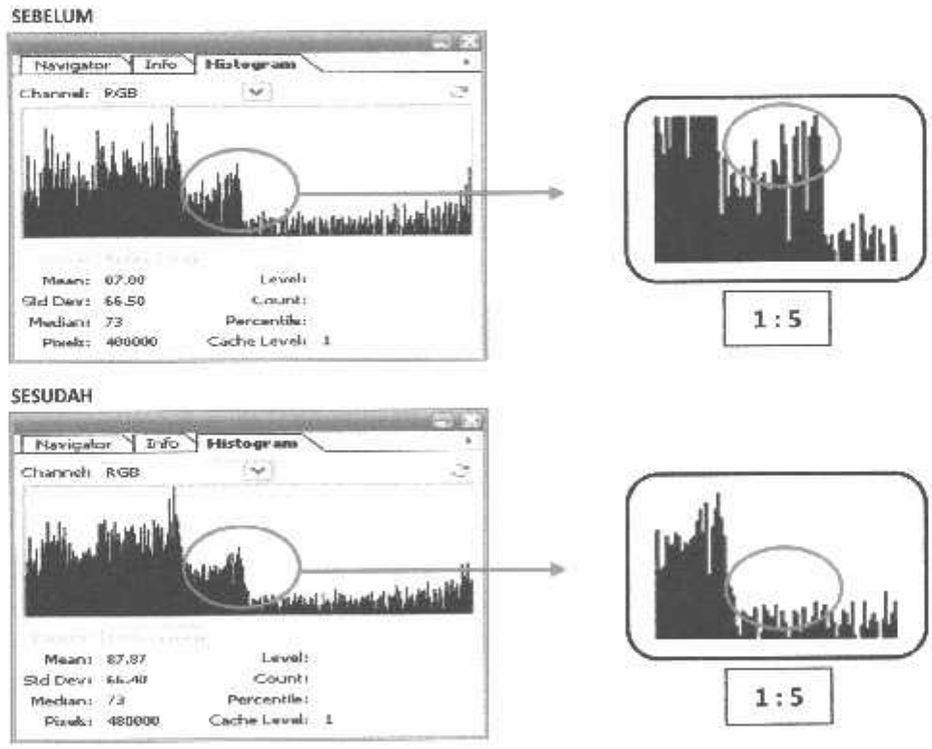
Gambar 4.41 Perbandingan Grafik Histogram Citra Water Lilies.JPG



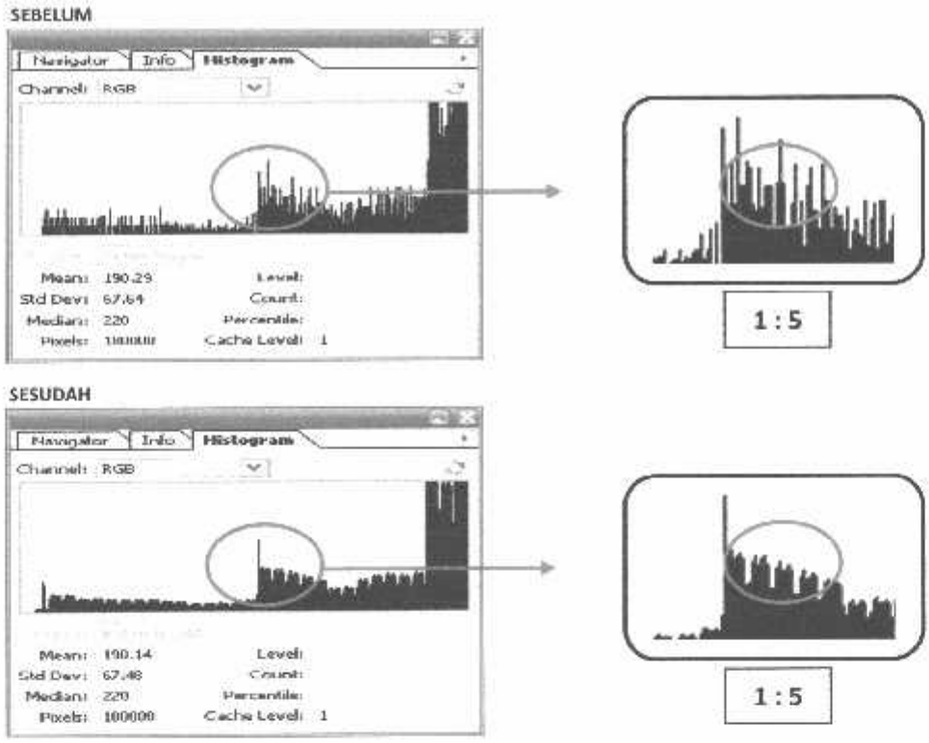
Gambar 4.42 Perbandingan Grafik Histogram Citra Blue Hills.JPG



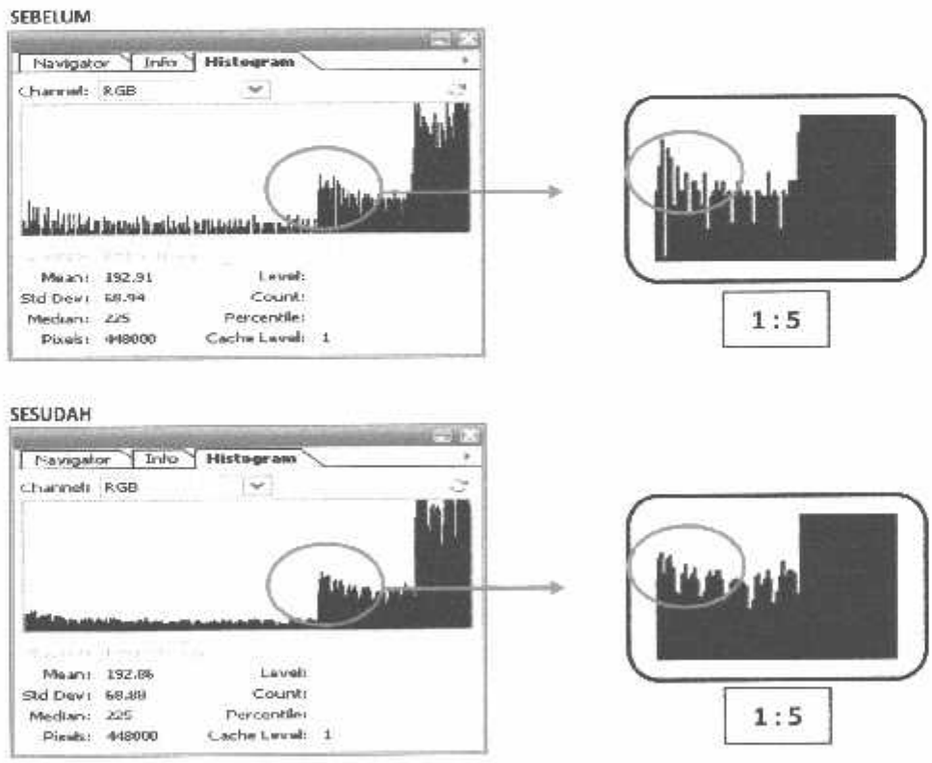
Gambar 4.45 Perbandingan Grafik Histogram Citra Rainbow.JPG



Gambar 4.46 Perbandingan Grafik Histogram Citra Citra.JPG



Gambar 4.47 Perbandingan Grafik Histogram Citra Stego.JPG



Gambar 4.48 Perbandingan Grafik Histogram Citra Blue.JPG

Gambar 4.49 Citra Bliss.BMP Sebelum Disisipi Karakter

Gambar 4.50 Citra Bliss.BMP Sesudah Disisipi Karakter

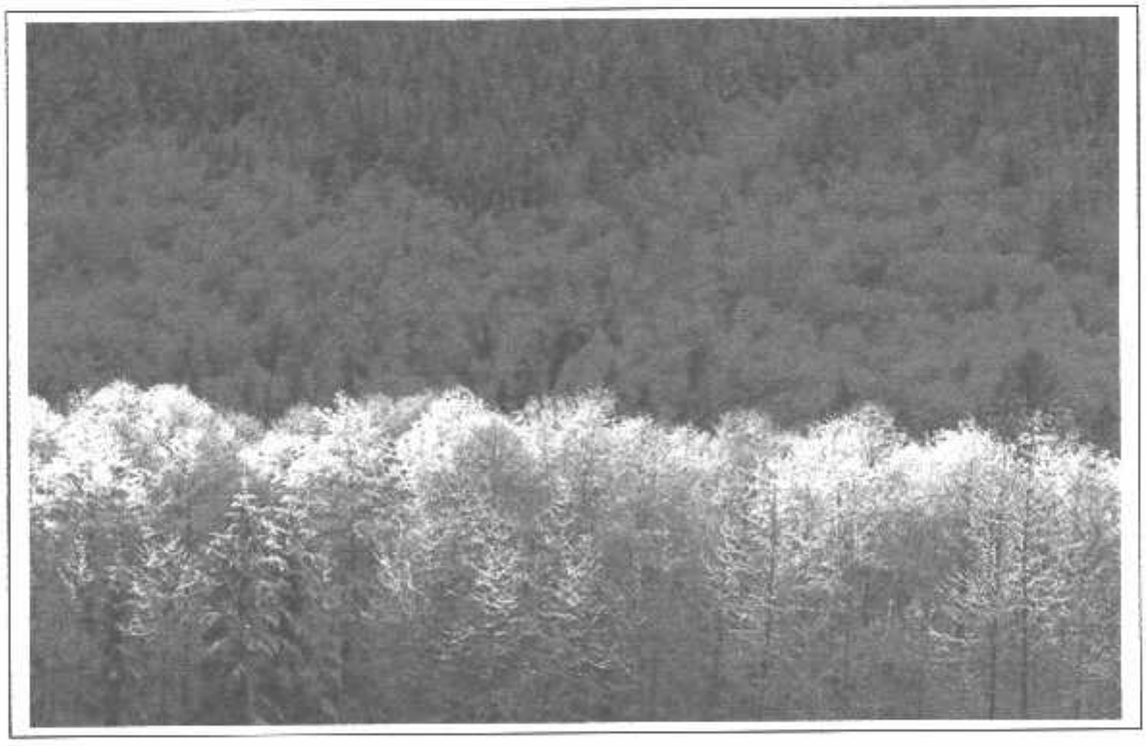
Gambar 4.51 Citra Nice.JPG Sebelum Disisipi Karakter

Gambar 4.52 Citra Nice.JPG Sesudah Disisipi Karakter

Gambar 4.53 Citra Sunset.JPG Sebelum Disisipi Karakter

Gambar 4.54 Citra Sunset.JPG Sesudah Disisipi Karakter

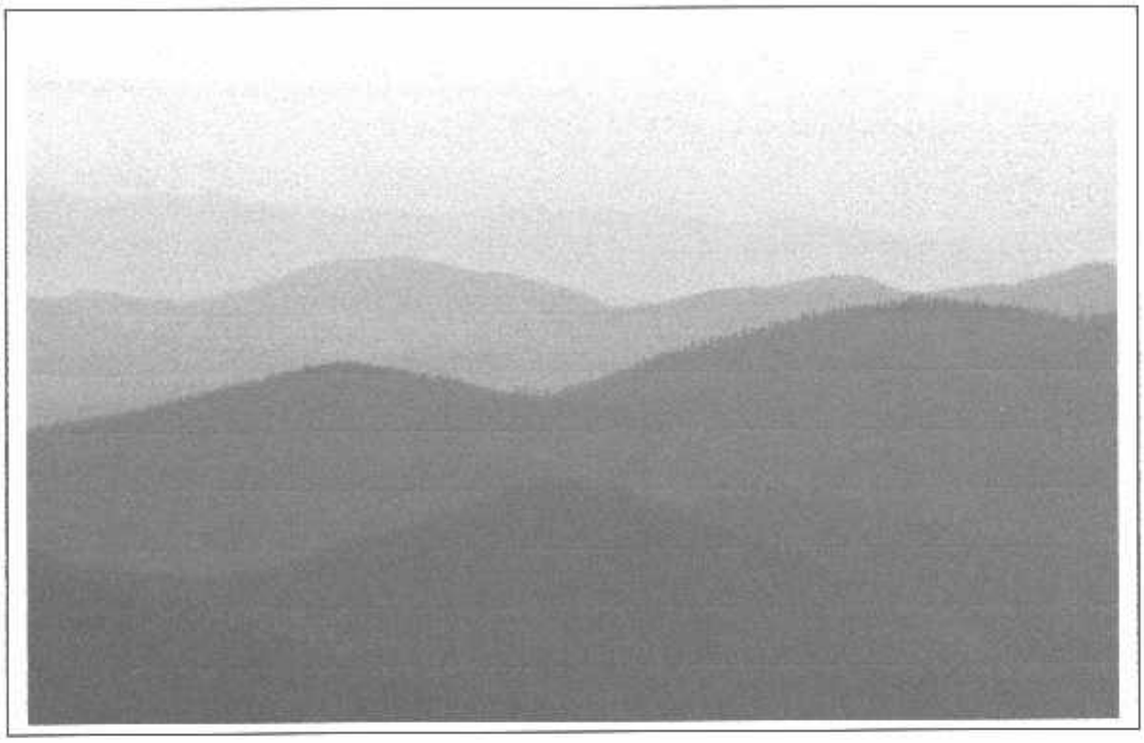
Gambar 4.55 Citra Winter.JPG Sebelum Disisipi Karakter



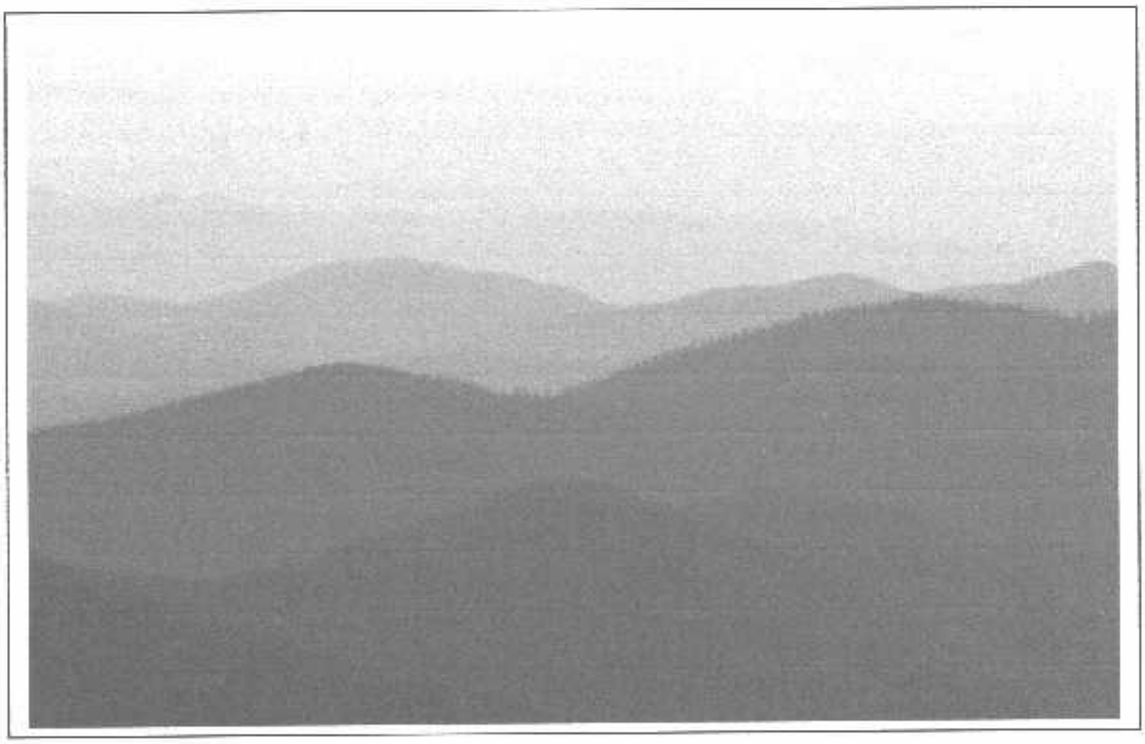
Gambar 4.56 Citra Winter.JPG Sesudah Disisipi Karakter

Gambar 4.57 Citra Water Lilies.JPG Sebelum Disisipi Karakter

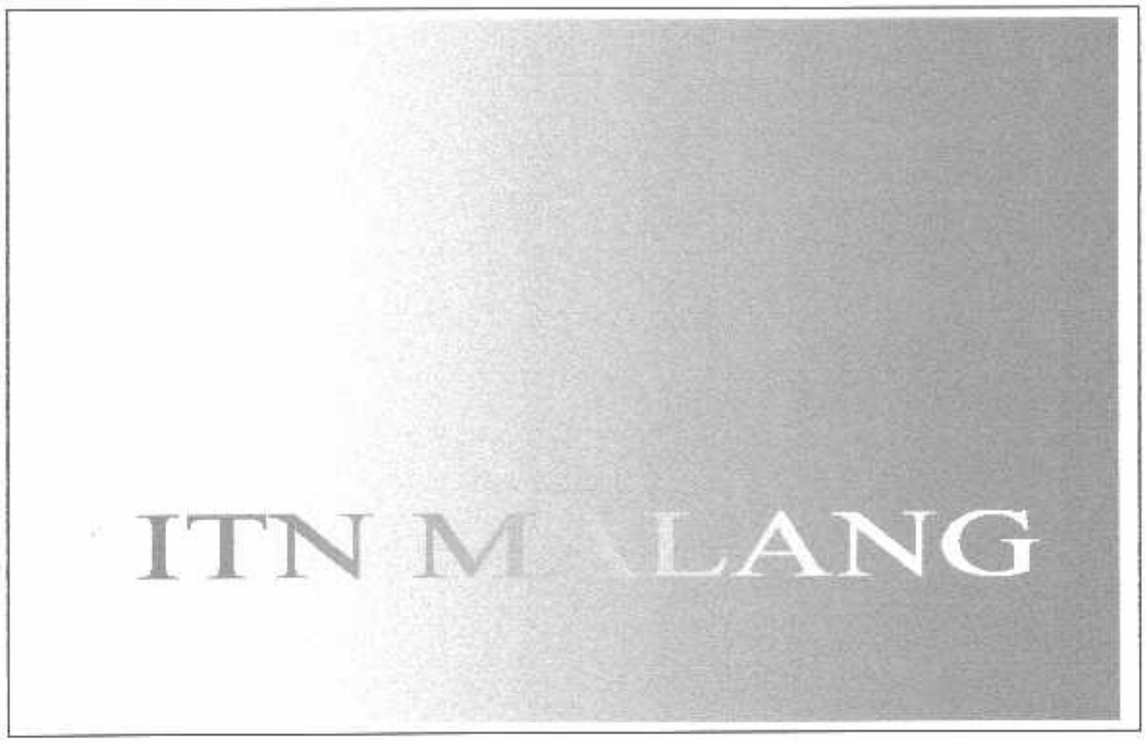
Gambar 4.58 Citra Water Lilies.JPG Sesudah Disisipi Karakter



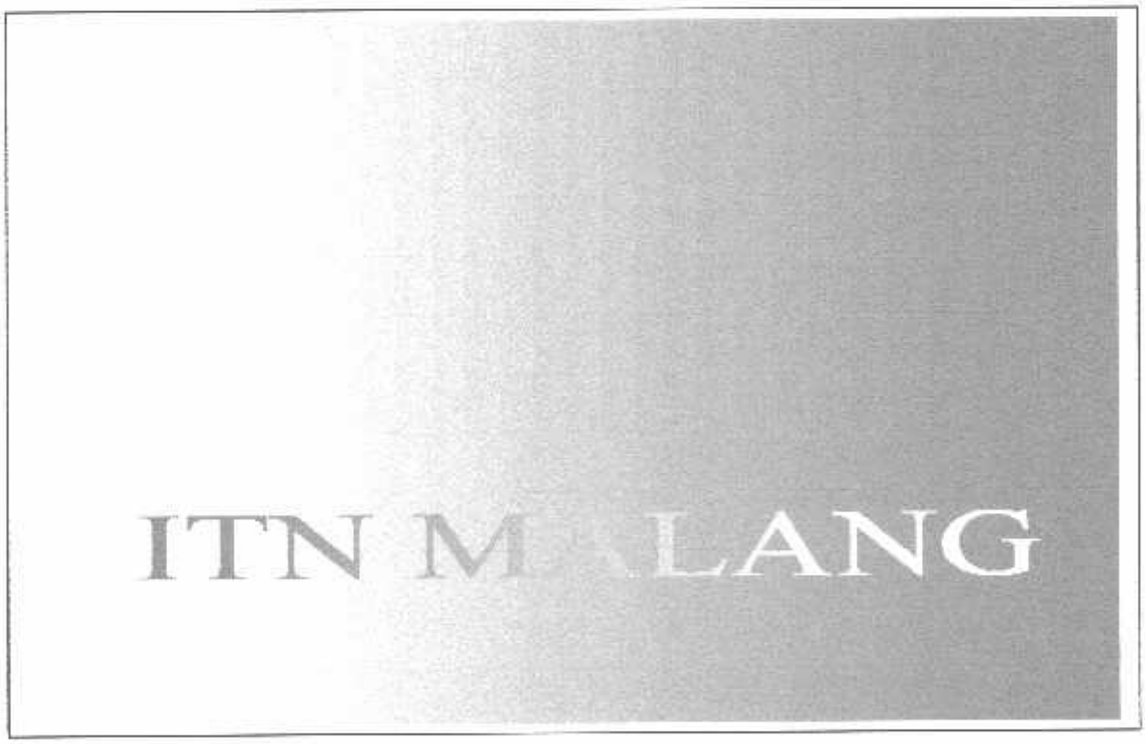
Gambar 4.59 Citra Blue Hills.JPG Sebelum Disisipi Karakter



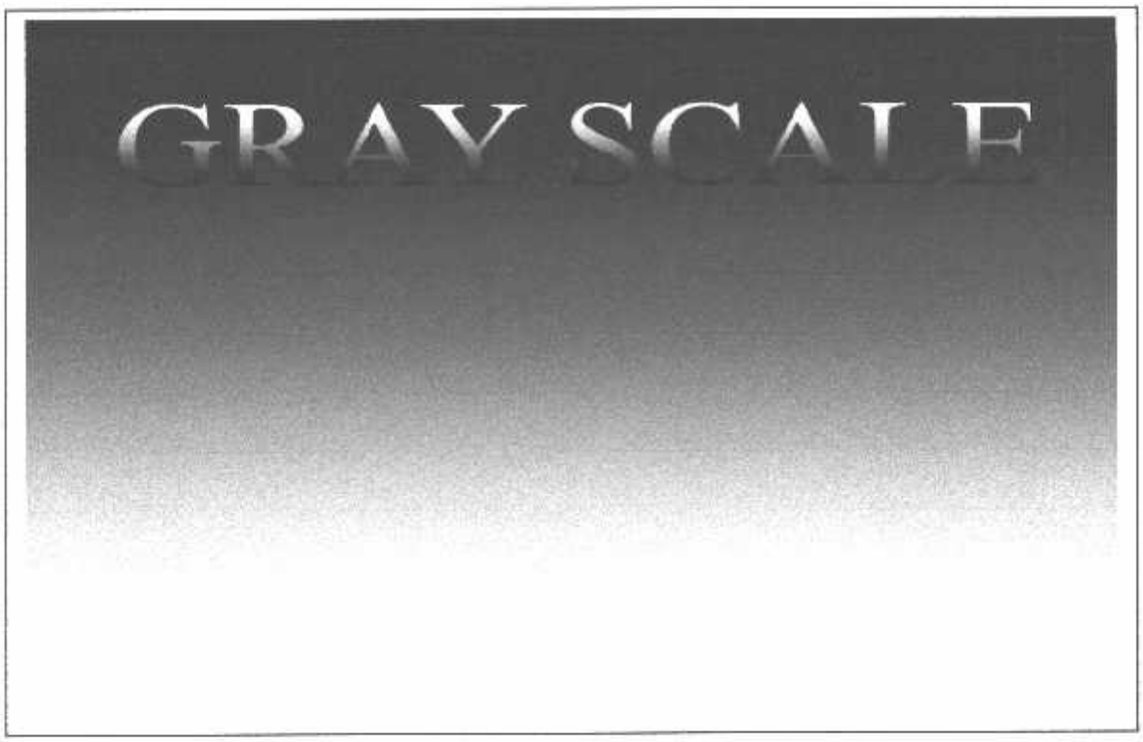
Gambar 4.60 Citra Blue Hills.JPG Sesudah Disisipi Karakter



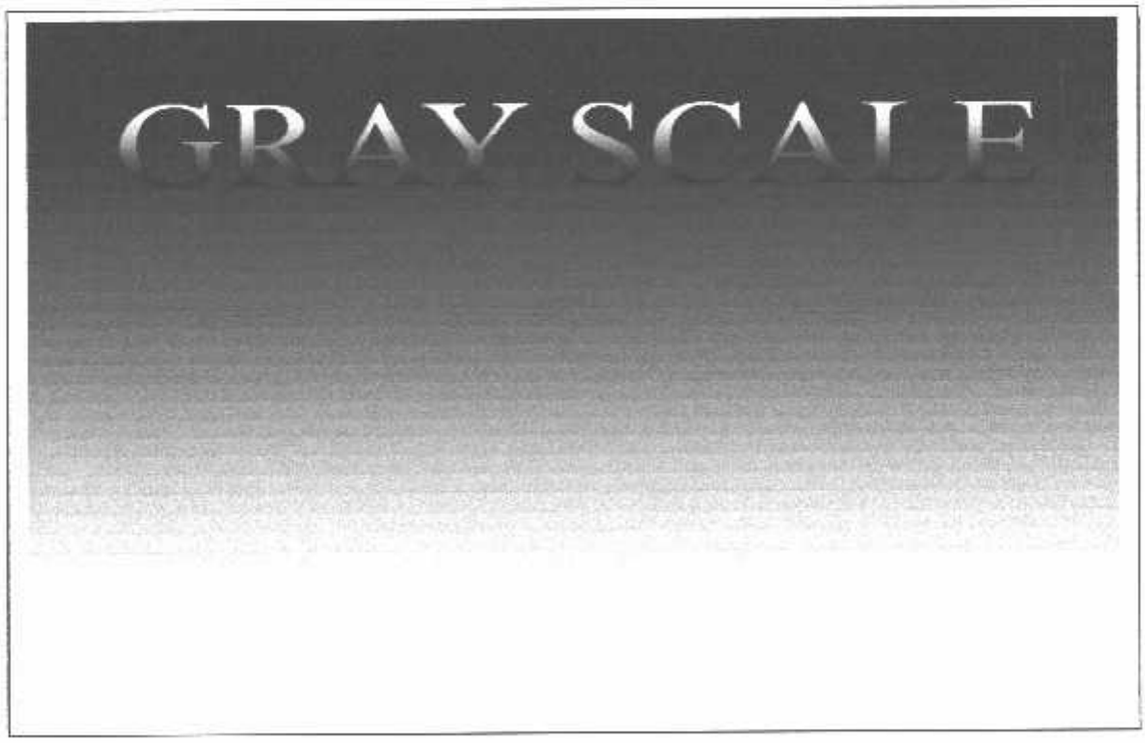
Gambar 4.61 Citra Itn.JPG Sebelum Disisipi Karakter



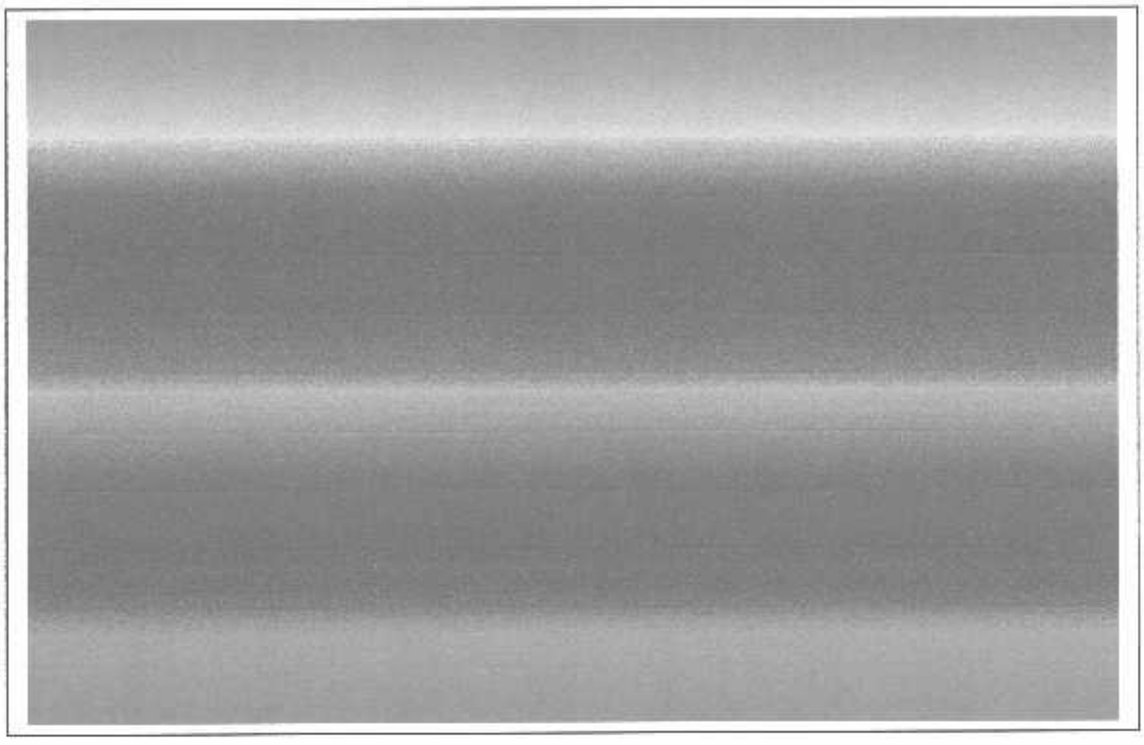
Gambar 4.62 Citra Itn.JPG Sesudah Disisipi Karakter



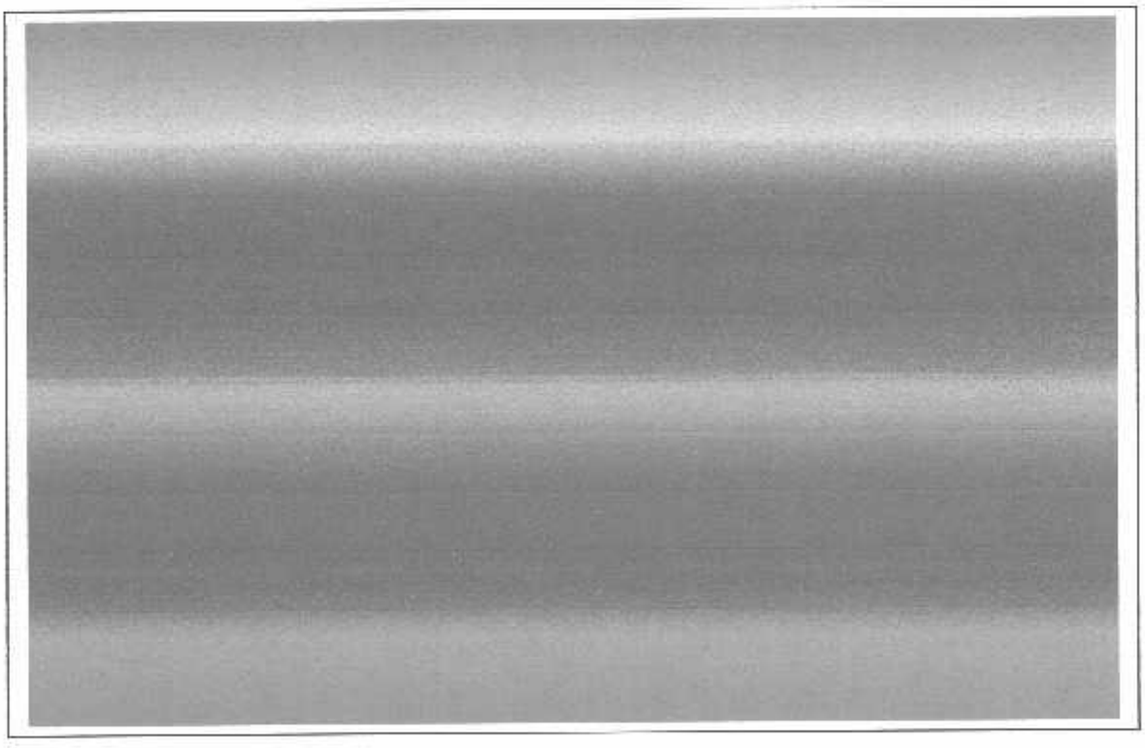
Gambar 4.63 Citra Gray Scale.JPG Sebelum Disisipi Karakter



Gambar 4.64 Citra Gray Scale.JPG Sesudah Disisipi Karakter



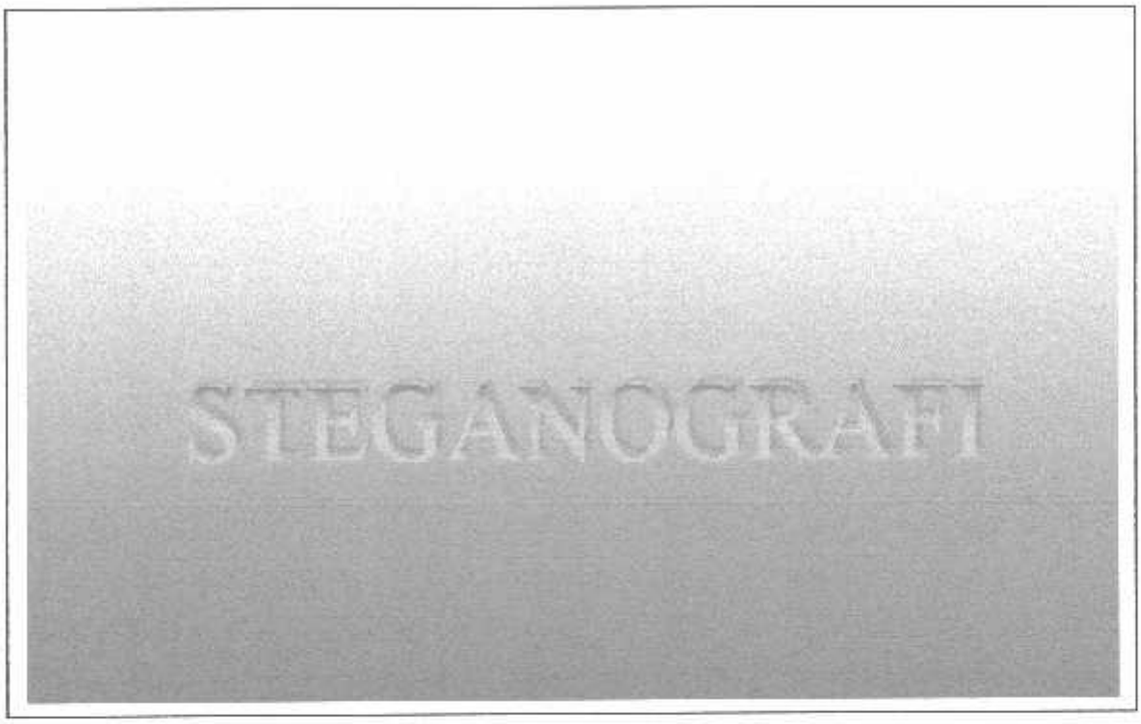
Gambar 4.65 Citra Rainbow.JPG Sebelum Disisipi Karakter



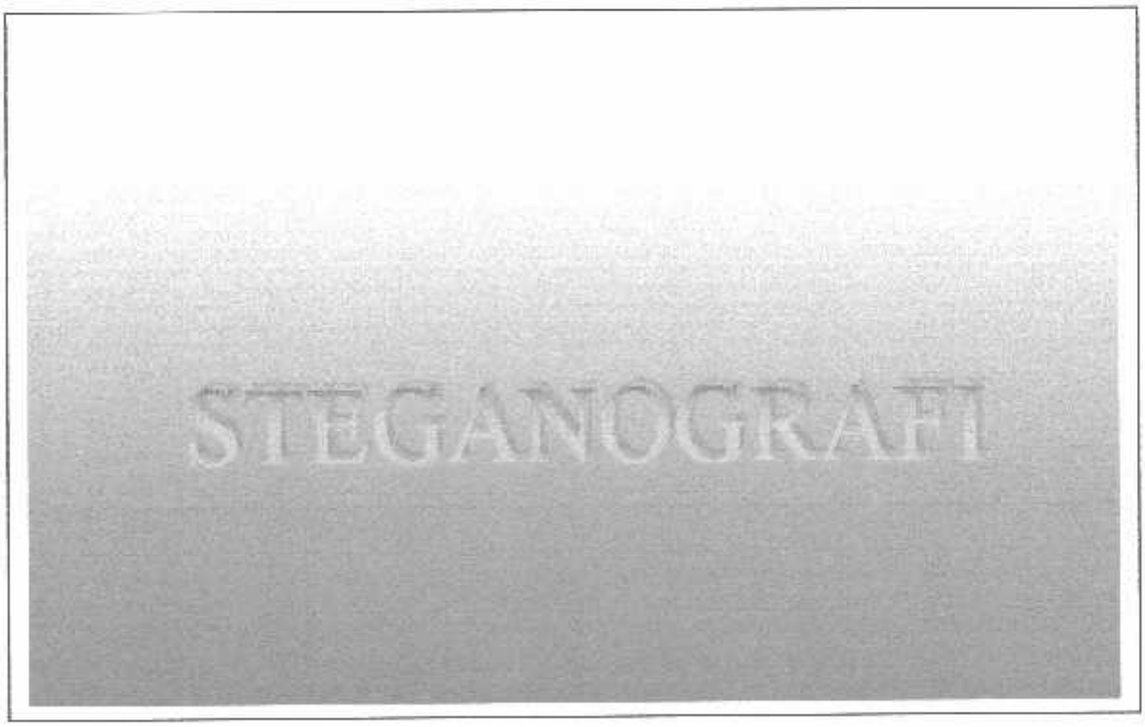
Gambar 4.66 Citra Rainbow.JPG Sesudah Disisipi Karakter

Gambar 4.67 Citra Citra.JPG Sebelum Disisipi Karakter

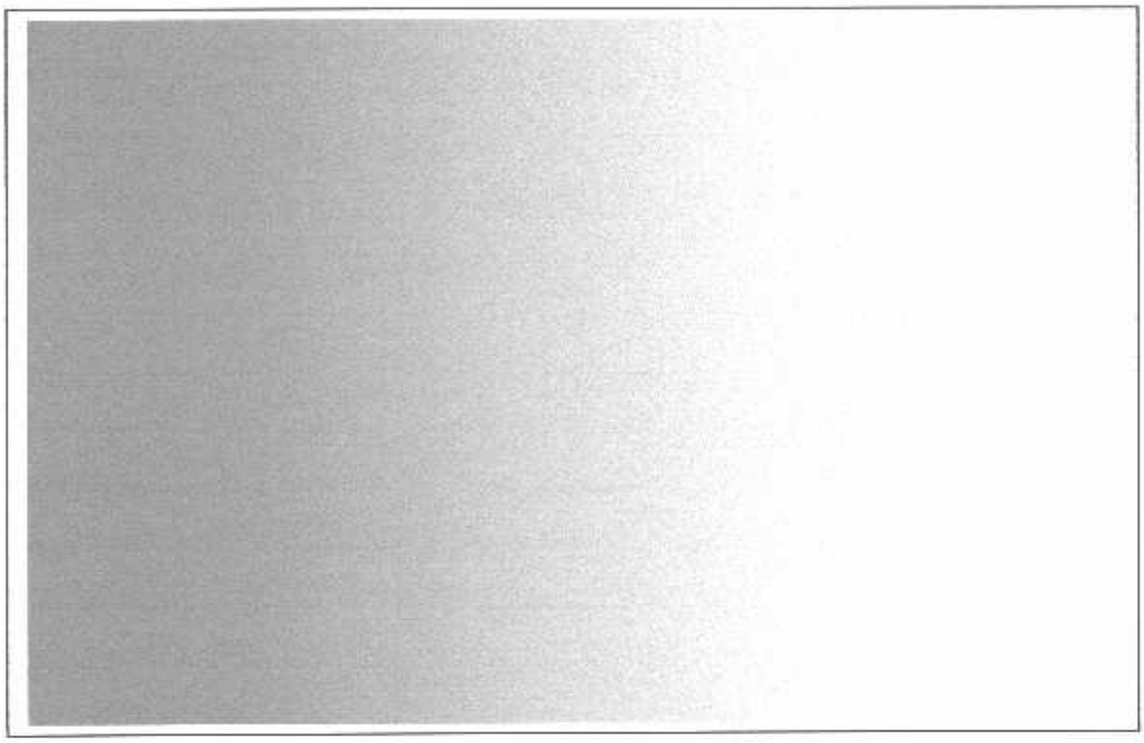
Gambar 4.68 Citra Citra.JPG Sesudah Disisipi Karakter



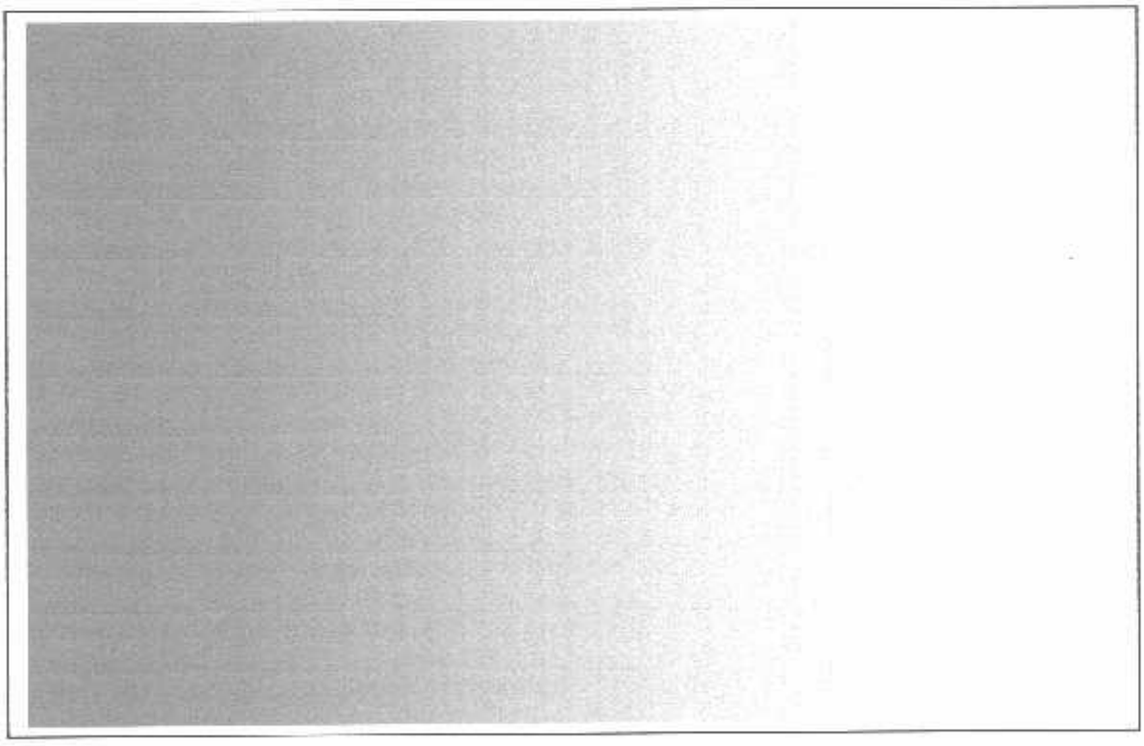
Gambar 4.69 Citra Stego.JPG Sebelum Disisipi Karakter



Gambar 4.70 Citra Stego.JPG Sesudah Disisipi Karakter



Gambar 4.71 Citra Bluc.JPG Sebelum Disisipi Karakter



Gambar 4.72 Citra Bluc.JPG Sesudah Disisipi Karakter

Hasil pengujian grafik histogram dari setiap file citra yang diuji dapat dilihat pada gambar diatas serta perubahan pada citra uji dengan jumlah karakter yang disisipkan. Pada grafik tersebut terlihat sangat jelas bahwa jumlah karakter yang disisipkan pada setiap file citra uji berpengaruh terhadap nilai atau grafik yang dihasilkan, dengan kata lain file citra uji yang digunakan mengalami perubahan sesuai dengan jumlah karakter yang disisipkan kedalam file citra sebelumnya. Semakin banyak karakter yang disisipkan maka semakin berkurang pula kualitas citra yang dihasilkan.

Kegagalan terjadi paling sering ketika karakter yang diinputkan terlalu banyak sehingga melebihi piksel dari citra uji.

Besarnya kapasitas dari citra uji juga berpengaruh pada proses penyisipan pesan rahasia (plainteks), karakter yang akan disisipkan tidak boleh melebihi kapasitas dari media penampung citra uji.

Pada pengujian diatas semua citra yang duiju pada dasarnya bisa disisipkan pesan rahasia, asalkan karakter yang diinputkan tidak melebihi piksel dari citra uji itu sendiri.

4.3.3 Pengujian Terhadap Waktu

Pengujian ini dilakukan untuk membandingkan waktu pada saat proses penyisipan dan pengambilan pesan rahasia, kemudian hasil dari percobaan tersebut di tampilkan dalam Tabel 4.4

Tabel 4.4 Hasil Percobaan Terhadap Waktu

NO	NAMA CITRA UJI	RESOLUSI (PIXEL X PIXEL)	DPI	KAPASITAS (KILO BYTE)	JUMLAH KARAKTER (INPUT)	JUMLAH KARAKTER (EKSTRAK)	WAKTU YANG DIBUTUHKAN
1	Bliss.BMP	300 x 200	71	175 KB	1.000	1.000	±1 Detik
					5.000	5.000	±1 Detik
					10.000	10.000	±1 Detik
					50.000	50.000	±1 Detik
					100.000	100.000	±2 Detik
					500.000	-	-
2	Nice.JPG	305 x 204	71	223 KB	1.000	1.000	±1 Detik
					5.000	5.000	±1 Detik
					10.000	10.000	±1 Detik
					50.000	50.000	±1 Detik
					100.000	100.000	±2 Detik

					500.000	-	-
3	Sunset.JPG	400 x 300	71	351 KB	1.000	1.000	±1 Detik
					5.000	5.000	±1 Detik
					10.000	10.000	±1 Detik
					50.000	50.000	±1 Detik
					100.000	100.000	±2 Detik
					500.000	-	-
4	Winter.JPG	600 x 500	71	878 KB	1.000	1.000	±1 Detik
					5.000	5.000	±1 Detik
					10.000	10.000	±1 Detik
					50.000	50.000	±1 Detik
					100.000	100.000	±2 Detik
					500.000	500.000	±2 Detik
5	Lilies.JPG	250 x 170	71	124 KB	1.000	1.000	±1 Detik
					5.000	5.000	±1 Detik
					10.000	10.000	±1 Detik
					50.000	50.000	±1 Detik
					100.000	100.000	±2 Detik
					500.000	-	-
6	Hills.JPG	440 x 230	71	412 KB	1.000	1.000	±1 Detik
					5.000	5.000	±1 Detik
					10.000	10.000	±1 Detik
					50.000	50.000	±1 Detik
					100.000	100.000	±2 Detik
					500.000	-	-
7	ITN.JPG	1024 X 768	72	180 KB	1.000	SUKSES	±1 Detik
					5.000	SUKSES	±1 Detik
					10.000	SUKSES	±1 Detik
					50.000	SUKSES	±1 Detik
					100.000	SUKSES	±2 Detik
					500.000	SUKSES	+2 Detik
8	GRAY SCALE.JPG	900 X 600	72	128 KB	1.000	SUKSES	±1 Detik
					5.000	SUKSES	±1 Detik
					10.000	SUKSES	±1 Detik
					50.000	SUKSES	±1 Detik
					100.000	SUKSES	±2 Detik
					500.000	SUKSES	±2 Detik
9	RAINBOW.JPG	900 X 600	72	143 KB	1.000	SUKSES	±1 Detik
					5.000	SUKSES	±1 Detik
					10.000	SUKSES	±1 Detik
					50.000	SUKSES	±1 Detik
					100.000	SUKSES	±2 Detik
					500.000	SUKSES	±2 Detik
10	CITRA.JPG	800 X 600	72	142 KB	1.000	SUKSES	±1 Detik
					5.000	SUKSES	±1 Detik
					10.000	SUKSES	±1 Detik
					50.000	SUKSES	±1 Detik

					100.000	SUKSES	±2 Detik
					500.000	SUKSES	±2 Detik
11	STEGO.JPG	600 X 300	72	74 KB	1.000	SUKSES	±1 Detik
					5.000	SUKSES	±1 Detik
					10.000	SUKSES	±1 Detik
					50.000	SUKSES	±1 Detik
					100.000	SUKSES	±2 Detik
					500.000	-	-
12	BLUE.JPG	700 X 640	72	95 KB	1.000	SUKSES	±1 Detik
					5.000	SUKSES	±1 Detik
					10.000	SUKSES	±1 Detik
					50.000	SUKSES	±1 Detik
					100.000	SUKSES	±2 Detik
					500.000	SUKSES	±2 Detik

Pada percobaan di atas dapat dilihat bahwa waktu yang dibutuhkan untuk proses pengambilan karakter relatif sama, yaitu antara 1 – 2 detik saja. Kecepatan terhadap waktu ekstrak (mengeluarkan) sangat bergantung pada spesifikasi computer yang digunakan pada saat uji coba.

Pesan yang diekstrak hanya pesan yang berhasil disisipkan, sebab tidak semua karakter yang diinputkan dapat disisipkan pada citra uji, ini terjadi karena karakter yang diinputkan terlalu banyak sehingga kapasitas dari piksel citra tidak memungkinkan proses penyisipan. Semakin banyak karakter yang disisipkan maka waktu yang dibutuhkan akan semakin lama. Tergantung dari banyaknya karakter yang diinputkan.

Hasil yang didapat pada tabel diatas dapat diambil suatu kesimpulan bahwa berapapun banyaknya karakter yang disisipkan ke dalam file citra uji tidak akan terlalu mempengaruhi besar dari file citra uji tersebut. Hal ini berlaku selama banyaknya karakter yang disisipkan tidak melebihi ukuran maksimum piksel citra uji yang dapat ditampung dengan masing-masing sampel yang digunakan. Bila karakter yang disisipkan besarnya melebihi dari ukuran maksimumnya maka karakter yang akan disisipkan kedalam file citra uji akan gagal (tidak bisa disisipkan).

Hal ini membuktikan bahwa pada aplikasi Steganografi yang di buat ini menghasilkan hasil yang cukup baik untuk setiap penyembuyian pesan kedalam file citra uji bergantung dari pemilihan *cover-object* atau file citra yang akan digunakan dan banyaknya karakter yang disisipkan pada file citra, karena semakin besar ukuran file citra yang digunakan dan semakin sedikit karakter

yang disisipkan pada file citra maka semakin sedikit perubahan yang terjadi setelah proses penyisipan pada file citra atau kualitas sebelum penyisipan dan setelah penyisipan tidak berpengaruh banyak pada perubahan kualitas citra sebelumnya.

BAB V
PENUTUP

5.1. Kesimpulan

Dari hasil penelitian dan pengujian yang dilakukan dapat dibuat beberapa kesimpulan sebagai berikut :

1. Kombinasi algoritma ini memungkinkan pihak lain tidak bisa mengetahui algoritma apa yang sedang digunakan sehingga pesan yang disembunyikan lebih aman.
2. Pesan rahasia (plainteks) yang disisipkan pada media penampung tidak boleh melebihi kapasitas dari citra uji. Semakin banyak karakter yang diinputkan maka kualitas dari citra uji akan menurun sehingga dapat menimbulkan kecurigaan dari pihak – pihak yang tidak berkepentingan.
3. Perubahan pada gambar hanya dapat dilihat dengan mengamati ataupun membandingkan histogram dari citra uji dan citra hasil.
4. Waktu yang dibutuhkan untuk melakukan proses penyisipan dan pengekstrakan pesan pada program ini cukup cepat yaitu antara ±1 detik sampai ±2 detik saja, sehingga untuk membangun program ini tidak memerlukan spesifikasi peralatan komputer yang lengkap dengan biaya yang mahal.
5. Aplikasi Steganografi ini keberhasilannya 85%, dikarenakan mutu dari media pembawa tidak tahan terhadap manipulasi palet warna.

5.2. Saran

Berikut adalah saran – saran untuk pengembangan lebih lanjut terhadap penelitian ini:

1. Untuk program pengembangan selanjutnya perlu dibuat tidak hanya menggunakan algoritma caesar dan konversi biner saja tapi juga dipadukan dengan algoritma yang lebih bervariatif.
2. Pada aplikasi steganografi ini, file citra yang dihasilkan setelah proses penyisipan mengalami pengurangan kualitas yang cukup banyak

DAFTAR PUSTAKA

Skripsi, Purwanto Hery, 2009 " Penggunaan Steganography Untuk Menyembunyikan Pesan Teks Ke Dalam File Audio Berformat Mp3 ".

Skripsi, Prasetyo deni, 2010, " Perancangan Dan Implementasi Steganography Citra Digital Pada Telpon Seluler ".

Anonim.*Delphi 2010 Architect*.2008. URL:http://shop.embarcadero.com/.

Anonim. URL:http://id.wikipedia.org/wiki/Format_bilangan_komputer.pdf

Darma Putra 2009 " PENGOLAHAN CITRA " Andi Publisher Yogyakarta.

T Sutoyo dkk 2009 " TEORI PENGOLAHAN CITRA DIGITAL " Andi Publisher Yogyakarta.

Anonim. Temmy Maradilla, 2009 "Aplikasi Steganografi Untuk Penyisipan Data Teks Ke Dalam Citra Digital ".

Putri Alatas. 2009 "Implementation Technique With Steganography Lsb Method In Digital Images ".

LAMPIRAN

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM.2 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : LA SARMAN
NIM : 04.12.678
JURUSAN : Teknik Komputer dan Informatika S-1
JUDUL : **DESAIN APLIKASI STEGANOGRAFI PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	09 Agustus 2011	-	
2.	Penguji II	09 Agustus 2011	1. Abstrak 2. Keterangan Gambar dan Tabel 3. Kesimpulan	19/2011

Disetujui :

Dosen Penguji I

Sotyohadi, ST
NIP.Y. 1039700309

Dosen Penguji II

Sonny Prasetio, ST, MT
NIP.P. 1031000433

Mengetahui :

Dosen Pembimbing I

Dr. Aryuanto S, ST, MT
NIP.P. 1030800417

Dosen Pembimbing II

I Komang Somawirata, ST, MT
NIP.Y. 1030100361

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : I.A SARMAN

Nim : 04.12.678

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan ini Menyatakan (bersedia / ~~tidak bersedia~~ *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

"DESAIN APLIKASI STEGANOGRAPHY PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER"

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Februari 2011
Kami yang membuat pernyataan,

Dr. Eng. Arynanto S,ST,MT.
NIP.P. 1030800417

Catatan :
Setelah disetujui agar formulir ini
Diserahkah mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
***) coret yang tidak perlu**

Form S-3b

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : LA SARMAN

Nim : 04.12.678

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan ini Menyatakan (bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

"DESAIN APLIKASI STEGANOGRAPHY PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER"

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Februari 2011
Kami yang membuat pernyataan,

I Komang Somawirata,ST, MT.
NIP.Y. 1030100361

Catatan :
Setelah disetujui agar formulir ini
Diserahkah mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
***) coret yang tidak perlu**

Form S-3b

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. ~~Energi Listrik~~ / ~~T. Elektronika~~ / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : La Sarman
N I M : 09 12 678
Perbaikan meliputi :

- Abstraksi
- Penulisan pada keterangan gambar & tabel
- ~~Kesimpulan~~

Malang, 9 - 8 - 2014

(SONNY PRASETIO, ST, MT)

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : La Sarman
Nim : 04.12.678
Masa Bimbingan : 2 April 2011 s/d 2 September 2011
Judul Skripsi : Desain Aplikasi Steganografi pada media citra dengan melakukan pengembangan metode Algoritma Caesar dan Pengkonversian Biner

No	Tanggal	Uraian	Paraf Pembimbing
1	10-06-2011	Konsultasi Bab I, II	
2	09-06-2011	Bimbingan Bab I, II, III, IV, V	
3	10-06-2011	Konsultasi Bab IV	
4	11-06-2011	Konsultasi Bab IV Revisi	
5	14-06-2011	Bimbingan Bab IV - perbaikan gambar	
6	15-06-2011	Konsultasi Bab IV - Revisi	
7	21-06-2011	Konsultasi Bab IV - perbaikan kelas Uji	
8	26-06-2011	Bimbingan Bab IV - Revisi	
9	27-06-2011	Konsultasi Bab IV - histogram citra	
10	30-06-2011	Konsultasi Bab IV - tambahan gambar	

Malang,
Dosen Pembimbing I

Dr. Eng. Aryuanto, ST, MT
NIP.Y.1030800417

Form S-4b

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : La Sarman
Nim : 04.12.678
Masa Bimbingan : 2 April 2011 s/d 2 September 2011
Judul Skripsi : Desain Aplikasi Steganografi pada media citra dengan melakukan pengembangan metode Algoritma Caesar dan Pengkonversian Biner

No	Tanggal	Uraian	Paraf Pembimbing
1	18-05-2011	Bimbingan Bab I, II, III	
2	20-05-2011	Konsultasi Seminar hasil	
3	21-05-2011	Konsultasi Bab IV, V	
4	27-05-2011	Bimbingan Bab IV - histogram Citrahasil	
5	30-05-2011	Konsultasi Mahala seminar hasil	
6	06-06-2011	Bimbingan Mahala seminar hasil - revisi	
7	02-07-2011	Konsultasi Mahala seminar hasil	
8			
9			
10			

Malang,
Dosen Pembimbing II

I Komang Somawirata, ST, MT
NIP.Y.1030100361

Form S-4b

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

Lampiran : 1 (Satu) Berkas
Pembimbing Skripsi

Kepada : Yth. Dr. Eng. Aryuanto S,ST,MT.
Dosen Institut Teknologi Nasional
M A L A N G

Yang bertanda tangan di bawah ini:

Nama : La Sarman
Nim : 04.12.678
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing (~~Utama~~ / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

"DESAIN APLIKASI STEGANOGRAPHY PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER"

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Skripsi Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang, Februari 2011

Ketua
Jurusan Teknik Elektro S-1,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 10188001189

Hormat kami,

La Sarman

*) coret yang tidak perlu

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

Lampiran : 1 (Satu) Berkas
Pembimbing Skripsi

Kepada : Yth. I Komang Somawirata,ST, MT.
Dosen Institut Teknologi Nasional
M A L A N G

Yang bertanda tangan di bawah ini:

Nama : La Sarman
Nim : 04.12.678
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing (Utama / ~~Pendamping~~ *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

"DESAIN APLIKASI STEGANOGRAPHY PADA MEDIA CITRA DENGAN MELAKUKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER"

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Skripsi Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang, Februari 2011

Ketua
Jurusan Teknik Elektro S-1,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Hormat kami,

La Sarman

*) coret yang tidak perlu

Form S-3a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

I (PERSERO) MALANG
NK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 April 2011

Nomor : ITN-214/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. **DR. ENG. ARYUANTO SOETEDJO, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: LA SARMAN
Nim	: 0412678
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

02 April 2011s/d 02 September 2011

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

: BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 April 2011

Nomor : ITN-215/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. **I KOMANG SOMAWIRATA, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: LA SARMAN
Nim	: 0412678
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

02 April 2011s/d 02 September 2011

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik / Teknik Elektronika /Teknik Komputer & Informatika /Teknik Komputer / Teknik Telekomunikasi*)

1.	Nama Mahasiswa: LA SARMAN			Nim: 04.12.678
2.	Waktu Pengajuan	Tanggal:	Bulan:	Tahun:
3.	Spesifikasi Judul (berilah tanda silang)**) a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri e. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainnya			
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*) Dr. Anggunto, ST, MT	Ketua Jurusan Ir. Yusuf Ismail Nakhoda, MT NIP. Y. 1018800189		
5.	Judul yang diajukan mahasiswa:	DESAIN APLIKASI STEGANOGRAPHY PADA MEDIA CITRA DENGAN MENGGUNAKAN PENGEMBANGAN METODE ALGORITMA CAESAR DAN PENGKONVERSIAN BINER		
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu	.....		
7.	Catatan: .... Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui 201 Dosen		

Perhatian:
1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ...atau g sesuai bidang keahlian

Form S-2

Tabel ASCII

DEC	OCT	HEX	BIN	Symbol	HTML Number	HTML Name	Description
0	000	00	00000000	NUL	&#000;		Null char
1	001	01	00000001	SOH	&#001;		Start of Heading
2	002	02	00000010	STX	&#002;		Start of Text
3	003	03	00000011	ETX	&#003;		End of Text
4	004	04	00000100	EOT	&#004;		End of Transmission
5	005	05	00000101	ENQ	&#005;		Enquiry
6	006	06	00000110	ACK	&#006;		Acknowledgment
7	007	07	00000111	BEL	&#007;		Bell
8	010	08	00001000	BS	&#008;		Back Space
9	011	09	00001001	HT			Horizontal Tab
10	012	0A	00001010	LF
	Line Feed
11	013	0B	00001011	VT	&#011;		Vertical Tab
12	014	0C	00001100	FF			Form Feed
13	015	0D	00001101	CR
	Carriage Return
14	016	0E	00001110	SO	&#014;		Shift Out / X-On
15	017	0F	00001111	SI	&#015;		Shift In / X-Off
16	020	10	00010000	DLE	&#016;		Data Line Escape
17	021	11	00010001	DC1	&#017;		Device Control 1 (oft. XON)
18	022	12	00010010	DC2	&#018;		Device Control 2
19	023	13	00010011	DC3	&#019;		Device Control 3 (oft. XOFF)
20	024	14	00010100	DC4	&#020;		Device Control 4
21	025	15	00010101	NAK	&#021;		Negative Acknowledgement
22	026	16	00010110	SYN	&#022;		Synchronous Idle
23	027	17	00010111	ETB	&#023;		End of Transmit Block
24	030	18	00011000	CAN	&#024;		Cancel
25	031	19	00011001	EM	&#025;		End of Medium
26	032	1A	00011010	SUB	&#026;		Substitute
27	033	1B	00011011	ESC	&#027;		Escape
28	034	1C	00011100	FS	&#028;		File Separator
29	035	1D	00011101	GS	&#029;		Group Separator
30	036	1E	00011110	RS	&#030;		Record Separator
31	037	1F	00011111	US	&#031;		Unit Separator

32	040	20	00100000		&#32;		Space
33	041	21	00100001	!	&#33;		Exclamation mark
34	042	22	00100010	"	&#34;	&quot;	Double quotes (or speech marks)
35	043	23	00100011	#	&#35;		Number
36	044	24	00100100	$	&#36;		Dollar
37	045	25	00100101	%	&#37;		Procenttecken
38	046	26	00100110	&	&#38;	&amp;	Ampersand
39	047	27	00100111	'	&#39;		Single quote
40	050	28	00101000	(	&#40;		Open parenthesis (or open bracket)
41	051	29	00101001	)	&#41;		Close parenthesis (or close bracket)
42	052	2A	00101010	*	&#42;		Asterisk
43	053	2B	00101011	+	&#43;		Plus
44	054	2C	00101100	,	&#44;		Comma
45	055	2D	00101101	-	&#45;		Hyphen
46	056	2E	00101110	.	&#46;		Period, dot or full stop
47	057	2F	00101111	/	&#47;		Slash or divide
48	060	30	00110000	0	&#48;		Zero
49	061	31	00110001	1	&#49;		One
50	062	32	00110010	2	&#50;		Two
51	063	33	00110011	3	&#51;		Three
52	064	34	00110100	4	&#52;		Four
53	065	35	00110101	5	&#53;		Five
54	066	36	00110110	6	&#54;		Six
55	067	37	00110111	7	&#55;		Seven
56	070	38	00111000	8	&#56;		Eight
57	071	39	00111001	9	&#57;		Nine
58	072	3A	00111010	:	&#58;		Colon
59	073	3B	00111011	;	&#59;		Semicolon
60	074	3C	00111100	<	&#60;	&lt;	Less than (or open angled bracket)
61	075	3D	00111101	=	&#61;		Equals
62	076	3E	00111110	>	&#62;	&gt;	Greater than (or close angled bracket)

63	077	3F	00111111	?	&#63;	Question mark
64	100	40	01000000	@	&#64;	At symbol
65	101	41	01000001	A	&#65;	Uppercase A
66	102	42	01000010	B	&#66;	Uppercase B
67	103	43	01000011	C	&#67;	Uppercase C
68	104	44	01000100	D	&#68;	Uppercase D
69	105	45	01000101	E	&#69;	Uppercase E
70	106	46	01000110	F	&#70;	Uppercase F
71	107	47	01000111	G	&#71;	Uppercase G
72	110	48	01001000	H	&#72;	Uppercase H
73	111	49	01001001	I	&#73;	Uppercase I
74	112	4A	01001010	J	&#74;	Uppercase J
75	113	4B	01001011	K	&#75;	Uppercase K
76	114	4C	01001100	L	&#76;	Uppercase L
77	115	4D	01001101	M	&#77;	Uppercase M
78	116	4E	01001110	N	&#78;	Uppercase N
79	117	4F	01001111	O	&#79;	Uppercase O
80	120	50	01010000	P	&#80;	Uppercase P
81	121	51	01010001	Q	&#81;	Uppercase Q
82	122	52	01010010	R	&#82;	Uppercase R
83	123	53	01010011	S	&#83;	Uppercase S
84	124	54	01010100	T	&#84;	Uppercase T
85	125	55	01010101	U	&#85;	Uppercase U
86	126	56	01010110	V	&#86;	Uppercase V
87	127	57	01010111	W	&#87;	Uppercase W
88	130	58	01011000	X	&#88;	Uppercase X
89	131	59	01011001	Y	&#89;	Uppercase Y
90	132	5A	01011010	Z	&#90;	Uppercase Z
91	133	5B	01011011	[	&#91;	Opening bracket
92	134	5C	01011100	\	&#92;	Backslash
93	135	5D	01011101	]	&#93;	Closing bracket
94	136	5E	01011110	^	&#94;	Caret - circumflex

95	137	5F	01011111	_	&#95;	Underscore
96	140	60	01100000	`	&#96;	Grave accent
97	141	61	01100001	a	&#97;	Lowercase a
98	142	62	01100010	b	&#98;	Lowercase b
99	143	63	01100011	c	&#99;	Lowercase c
100	144	64	01100100	d	&#100;	Lowercase d
101	145	65	01100101	e	&#101;	Lowercase e
102	146	66	01100110	f	&#102;	Lowercase f
103	147	67	01100111	g	&#103;	Lowercase g
104	150	68	01101000	h	&#104;	Lowercase h
105	151	69	01101001	i	&#105;	Lowercase i
106	152	6A	01101010	j	&#106;	Lowercase j
107	153	6B	01101011	k	&#107;	Lowercase k
108	154	6C	01101100	l	&#108;	Lowercase l
109	155	6D	01101101	m	&#109;	Lowercase m
110	156	6E	01101110	n	&#110;	Lowercase n
111	157	6F	01101111	o	&#111;	Lowercase o
112	160	70	01110000	p	&#112;	Lowercase p
113	161	71	01110001	q	&#113;	Lowercase q
114	162	72	01110010	r	&#114;	Lowercase r
115	163	73	01110011	s	&#115;	Lowercase s
116	164	74	01110100	t	&#116;	Lowercase t
117	165	75	01110101	u	&#117;	Lowercase u
118	166	76	01110110	v	&#118;	Lowercase v
119	167	77	01110111	w	&#119;	Lowercase w
120	170	78	01111000	x	&#120;	Lowercase x
121	171	79	01111001	y	&#121;	Lowercase y
122	172	7A	01111010	z	&#122;	Lowercase z
123	173	7B	01111011	{	&#123;	Opening brace
124	174	7C	01111100	|	&#124;	Vertical bar
125	175	7D	01111101	}	&#125;	Closing brace
126	176	7E	01111110	~	&#126;	Equivalency sign - tilde
127	177	7F	01111111		&#127;	Delete

128	200	80	10000000	€	€	€	Euro sign
129	201	81	10000001				
130	202	82	10000010	‚	‚	‚	Single low-9 quotation mark
131	203	83	10000011	ƒ	ƒ	ƒ	Latin small letter f with hook
132	204	84	10000100	„	„	„	Double low-9 quotation mark
133	205	85	10000101	…	…	…	Horizontal ellipsis
134	206	86	10000110	†	†	†	Dagger
135	207	87	10000111	‡	‡	‡	Double dagger
136	210	88	10001000	ˆ	ˆ	ˆ	Modifier letter circumflex accent
137	211	89	10001001	‰	‰	‰	Per mille sign
138	212	8A	10001010	Š	Š	Š	Latin capital letter S with caron
139	213	8B	10001011	‹	‹	‹	Single left-pointing angle quotation
140	214	8C	10001100	Œ	Œ	Œ	Latin capital ligature OE
141	215	8D	10001101				
142	216	8E	10001110	Ž	Ž		Latin captial letter Z with caron
143	217	8F	10001111				
144	220	90	10010000				
145	221	91	10010001	‘	‘	‘	Left single quotation mark
146	222	92	10010010	’	’	’	Right single quotation mark
147	223	93	10010011	“	“	“	Left double quotation mark
148	224	94	10010100	”	”	”	Right double quotation mark
149	225	95	10010101	•	•	•	Bullet
150	226	96	10010110	–	–	–	En dash
151	227	97	10010111	—	—	—	Em dash
152	230	98	10011000	˜	˜	˜	Small tilde
153	231	99	10011001	™	™	™	Trade mark sign
154	232	9A	10011010	š	š	š	Latin small letter S with caron
155	233	9B	10011011	›	›	›	Single right-pointing angle quotation mark
156	234	9C	10011100	œ	œ	œ	Latin small ligature oe
157	235	9D	10011101				
158	236	9E	10011110	ž	ž		Latin small letter z with caron
159	237	9F	10011111	Ÿ	Ÿ	ÿ	Latin capital letter Y with diaeresis

160	240	A0	10100000				Non-breaking space
161	241	A1	10100001	¡	¡	¡	Inverted exclamation mark
162	242	A2	10100010	¢	¢	¢	Cent sign
163	243	A3	10100011	£	£	£	Pound sign
164	244	A4	10100100	¤	¤	¤	Currency sign
165	245	A5	10100101	¥	¥	¥	Yen sign
166	246	A6	10100110	¦	¦	¦	Pipe, Broken vertical bar
167	247	A7	10100111	§	§	§	Section sign
168	250	A8	10101000	¨	¨	¨	Spacing diaeresis - umlaut
169	251	A9	10101001	©	©	©	Copyright sign
170	252	AA	10101010	ª	ª	ª	Feminine ordinal indicator
171	253	AB	10101011	«	«	«	Left double angle quotes
172	254	AC	10101100	¬	¬	¬	Not sign
173	255	AD	10101101				Soft hyphen
174	256	AE	10101110	®	®	®	Registered trade mark sign
175	257	AF	10101111	¯	¯	¯	Spacing macron - overline
176	260	B0	10110000	°	°	°	Degree sign
177	261	B1	10110001	±	±	±	Plus-or-minus sign
178	262	B2	10110010	²	²	²	Superscript two - squared
179	263	B3	10110011	³	³	³	Superscript three - cubed
180	264	B4	10110100	´	´	´	Acute accent - spacing acute
181	265	B5	10110101	µ	µ	µ	Micro sign
182	266	B6	10110110	¶	¶	¶	Pilcrow sign - paragraph sign
183	267	B7	10110111	·	·	·	Middle dot - Georgian comma
184	270	B8	10111000	¸	¸	¸	Spacing cedilla
185	271	B9	10111001	¹	¹	¹	Superscript one
186	272	BA	10111010	º	º	º	Masculine ordinal indicator
187	273	BB	10111011	»	»	»	Right double angle quotes
188	274	BC	10111100	¼	¼	¼	Fraction one quarter
189	275	BD	10111101	½	½	½	Fraction one half
190	276	BE	10111110	¾	¾	¾	Fraction three quarters
191	277	BF	10111111	¿	¿	¿	Inverted question mark
192	300	C0	11000000	À	À	À	Latin capital letter A with grave

193	301	C1	11000001	Á	Á	Á	Latin capital letter A with acute
194	302	C2	11000010	Â	Â	Â	Latin capital letter A with circumflex
195	303	C3	11000011	Ã	Ã	Ã	Latin capital letter A with tilde
196	304	C4	11000100	Ä	Ä	Ä	Latin capital letter A with diaeresis
197	305	C5	11000101	Å	Å	Å	Latin capital letter A with ring above
198	306	C6	11000110	Æ	Æ	Æ	Latin capital letter AE
199	307	C7	11000111	Ç	Ç	Ç	Latin capital letter C with cedilla
200	310	C8	11001000	È	È	È	Latin capital letter E with grave
201	311	C9	11001001	É	É	É	Latin capital letter E with acute
202	312	CA	11001010	Ê	Ê	Ê	Latin capital letter E with circumflex
203	313	CB	11001011	Ë	Ë	Ë	Latin capital letter E with diaeresis
204	314	CC	11001100	Ì	Ì	Ì	Latin capital letter I with grave
205	315	CD	11001101	Í	Í	Í	Latin capital letter I with acute
206	316	CE	11001110	Î	Î	Î	Latin capital letter I with circumflex
207	317	CF	11001111	Ï	Ï	Ï	Latin capital letter I with diaeresis
208	320	D0	11010000	Ð	Ð	Ð	Latin capital letter ETH
209	321	D1	11010001	Ñ	Ñ	Ñ	Latin capital letter N with tilde
210	322	D2	11010010	Ò	Ò	Ò	Latin capital letter O with grave
211	323	D3	11010011	Ó	Ó	Ó	Latin capital letter O with acute
212	324	D4	11010100	Ô	Ô	Ô	Latin capital letter O with circumflex
213	325	D5	11010101	Õ	Õ	Õ	Latin capital letter O with tilde
214	326	D6	11010110	Ö	Ö	Ö	Latin capital letter O with diaeresis
215	327	D7	11010111	×	×	×	Multiplication sign
216	330	D8	11011000	Ø	Ø	Ø	Latin capital letter O with slash
217	331	D9	11011001	Ù	Ù	Ù	Latin capital letter U with grave
218	332	DA	11011010	Ú	Ú	Ú	Latin capital letter U with acute
219	333	DB	11011011	Û	Û	Û	Latin capital letter U with circumflex
220	334	DC	11011100	Ü	Ü	Ü	Latin capital letter U with diaeresis
221	335	DD	11011101	Ý	Ý	Ý	Latin capital letter Y with acute
222	336	DE	11011110	Þ	Þ	Þ	Latin capital letter THORN

223	337	DF	11011111	ß	ß	ß	Latin small letter sharp s - ess-zed
224	340	E0	11100000	à	à	à	Latin small letter a with grave
225	341	E1	11100001	á	á	á	Latin small letter a with acute
226	342	E2	11100010	â	â	â	Latin small letter a with circumflex
227	343	E3	11100011	ã	ã	ã	Latin small letter a with tilde
228	344	E4	11100100	ä	ä	ä	Latin small letter a with diaeresis
229	345	E5	11100101	å	å	å	Latin small letter a with ring above
230	346	E6	11100110	æ	æ	æ	Latin small letter ae
231	347	E7	11100111	ç	ç	ç	Latin small letter c with cedilla
232	350	E8	11101000	è	è	è	Latin small letter e with grave
233	351	E9	11101001	é	é	é	Latin small letter e with acute
234	352	EA	11101010	ê	ê	ê	Latin small letter e with circumflex
235	353	EB	11101011	ë	ë	ë	Latin small letter e with diaeresis
236	354	EC	11101100	ì	ì	ì	Latin small letter i with grave
237	355	ED	11101101	í	í	í	Latin small letter i with acute
238	356	EE	11101110	î	î	î	Latin small letter i with circumflex
239	357	EF	11101111	ï	ï	ï	Latin small letter i with diaeresis
240	360	F0	11110000	ð	ð	ð	Latin small letter eth
241	361	F1	11110001	ñ	ñ	ñ	Latin small letter n with tilde
242	362	F2	11110010	ò	ò	ò	Latin small letter o with grave
243	363	F3	11110011	ó	ó	ó	Latin small letter o with acute
244	364	F4	11110100	ô	ô	ô	Latin small letter o with circumflex
245	365	F5	11110101	õ	õ	õ	Latin small letter o with tilde
246	366	F6	11110110	ö	ö	ö	Latin small letter o with diaeresis
247	367	F7	11110111	÷	÷	÷	Division sign
248	370	F8	11111000	ø	ø	ø	Latin small letter o with slash
249	371	F9	11111001	ù	ù	ù	Latin small letter u with grave
250	372	FA	11111010	ú	ú	ú	Latin small letter u with acute
251	373	FB	11111011	û	û	û	Latin small letter u with circumflex
252	374	FC	11111100	ü	ü	ü	Latin small letter u with diaeresis
253	375	FD	11111101	ý	ý	ý	Latin small letter y with acute
254	376	FE	11111110	þ	þ	þ	Latin small letter thorn
255	377	FF	11111111	ÿ	ÿ	ÿ	Latin small letter y with diaeresis

SOURCE CODE PROGRAM

UMain.pas

```
unit UMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, sSkinProvider, sSkinManager, ExtCtrls, sPanel, StdCtrls, sLabel,
  Buttons, sBitBtn, sEdit, sGauge, ExtDlgs,jpeg,strUtils;

type
  TFMain = class(TForm)
    sSkinManager1: TsSkinManager;
    sSkinProvider1: TsSkinProvider;
    sPanel1: TsPanel;
    sLabelFX1: TsLabelFX;
    sLabelFX2: TsLabelFX;
    Image1: TImage;
    sPanel2: TsPanel;
    EdImages: TsEdit;
    sLabel1: TsLabel;
    sLabel2: TsLabel;
    MPesan: TMemo;
    BConver: TsBitBtn;
    BClear: TsBitBtn;
    BBiner: TsBitBtn;
    BCesar: TsBitBtn;
    BHelp: TsBitBtn;
    BAbout: TsBitBtn;
    MCesar: TMemo;
    sLabel3: TsLabel;
    MBiner: TMemo;
    sLabel4: TsLabel;
    sPanel3: TsPanel;
    BSave: TsBitBtn;
    BOpen: TsBitBtn;
    GProses: TsGauge;
    Splitter1: TSplitter;
    ODialog: TOpenPictureDialog;
```

```
BEkstrak: TsBitBtn;
    ScrollBox1: TScrollBox;
    IHasil: TImage;
    ScrollBox2: TScrollBox;
    IAsli: TImage;
    SavePictureDialog1: TSavePictureDialog;
    procedure BAboutClick(Sender: TObject);
    procedure BHelpClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BOpenClick(Sender: TObject);
    procedure MPesanChange(Sender: TObject);
    procedure BCesarClick(Sender: TObject);
    procedure BBinerClick(Sender: TObject);
    procedure BClearClick(Sender: TObject);
    procedure BConverClick(Sender: TObject);
    procedure BSaveClick(Sender: TObject);
    procedure BEkstrakClick(Sender: TObject);
  private
    { Private declarations }
  public
    IOri : TBitmap;
  end;

const
  KEY = 3;
  Mark = '©ª¶';
var
  FMain: TFMain;

implementation

uses UAbout, UHelp, Math;

{$R *.dfm}

procedure TFMain.BAboutClick(Sender: TObject);
begin
  fAbout.ShowModal;
end;

procedure TFMain.BHelpClick(Sender: TObject);
begin
  fhelp.Show;
end;
```

```
procedure TFMain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
 Application.Terminate;
end;


function ConvertToBitmap(NamaFile : string):TBitmap;
var
 bmp   : TBitmap;
 jpg   : TJPEGImage;
 icon  : TIcon;
 wmf   : TMetafile;
 TipeFile : string;
 ConvertTo : TGraphic;
begin
 TipeFile := LowerCase(  ExtractFileExt(NamaFile));
 bmp := TBitmap.Create;
 bmp.PixelFormat := pf24bit;
 ConvertTo := bmp;
 if TipeFile = '.ico' then
  begin
   icon := TIcon.Create;
   icon.LoadFromFile(NamaFile);
   ConvertTo := icon;
  end
 else if TipeFile = '.jpg' then
  begin
   jpg   := TJPEGImage.Create;
   jpg.LoadFromFile(NamaFile);
   ConvertTo := jpg;
  end
 else if TipeFile = '.wmf' then
  begin
   wmf := TMetafile.Create;
   wmf.LoadFromFile(NamaFile);
   ConvertTo := wmf;
  end
 else begin
   bmp := TBitmap.Create;
   bmp.LoadFromFile(NamaFile);
   ConvertTo := bmp;
 end;
```

```
bmp.Width := ConvertTo.Width;
  bmp.Height := ConvertTo.Height;
  bmp.Canvas.Draw(0,0,ConvertTo);
  Result := bmp;
end;

function CaesarEncrypt(s:String; keys:Integer): String;
var
   i : Integer;
begin
   for i:=1 to Length(s) do
    result:=result+Chr(Ord(s[i])+keys);
end;
function CaesarDecrypt(s:String; n:Integer): String;
var
   i : Integer;
begin
   for i:=1 to Length(s) do
     result:=result+Chr(Ord(s[i])-n);
end;
function DesToBin8(angka:Integer): string;
var
     a, sisa : integer;
     car : string[1];
     hasil : string[8];
begin
     hasil:='00000000';
     a:=0;
     if angka <> 0 then
     begin
          while angka <> 1 do
          begin
               sisa:=angka mod 2;
               str(sisa, car);
               hasil[8-a]:=car[1];
               angka:=trunc(angka/2);
               inc(a);
          end;
          str(angka,car);
          hasil[8-a]:=car[1];
     end;
     Result :=hasil;
end;
```

```
function BinToDes(Bit: String): Integer;
var
    i : Integer;
    Hasil: Integer;
begin
    hasil := 0;
    for i := 1 to Length(Bit) do
    begin
     if Bit[i] = '1' then
        hasil := hasil + Trunc(Exp((Length(Bit)-i)*ln(2)));
    end;

    Result := hasil;
end;
Function DesToNBin (Des :integer):String;
var
 Hasil,Temp : String;
begin
  if des = 0 then
  begin
   Result := 0';
   exit;
  end;
  Hasil := '';
  Temp :='';
  repeat
       str(des mod 2, temp);
       Hasil:=temp+Hasil;
       des:=des div 2;
   until des=0;
   Result := Hasil;
end;
function GetBinerData(data :string):string;
var
    i :integer;
    hasil : string;
begin
    hasil := '';
    for i:=1 to length(data) do
    begin
      hasil := hasil + DesToBin8(ord(data[i]));
    end;
       Result := hasil;
end;
```

```
Function BinSizeData (dataSize : Integer ):String;//mendapatkan jml karakter di 24bit
var
  Hasil,Temp : String;
begin
   if dataSize  = 0 then
   begin
    Result :='000000000000000000000000'; // 24 bit = 3 pixel untuk image 3600x3600
    exit;
   end;
   Hasil := '';
   Temp :='';
   repeat
        str(dataSize  mod 2, temp);
        Hasil:=temp+Hasil;
        dataSize :=dataSize  div 2;
    until dataSize =0;
    if length(hasil) < 24 then
      Hasil := copy('000000000000000000000000',0,24-length(hasil))+hasil;
    Result := Hasil;
end;

function HasEmbbed(Img :TImage):Boolean ;
var
  RVal,GVal,BVal,X : integer;
  TempMark,DtBin : string;
begin
      DtBin  := '';
      for x:=0 to 2 do
      begin
           RVal :=GetRValue(Img.Picture.Bitmap.Canvas.Pixels [x,0]);
           GVal :=GetGValue(Img.Picture.Bitmap.Canvas.Pixels [x,0]);
           BVal :=GetBValue(Img.Picture.Bitmap.Canvas.Pixels [x,0]);
           DtBin  := DtBin  + RightStr(DesToBin8 (RVal),3) +
             RightStr (DesToBin8 (GVal),3) + RightStr (DesToBin8 (BVal),3);
      end;
      TempMark := chr(BinToDes (copy(DtBin ,1,8))) +
        chr(BinToDes (copy(DtBin ,9,8)))+
        chr(BinToDes (copy(DtBin ,17,8)));
      if TempMark = Mark then Result := true
      else Result := false;
end;
```

```
Function GetSizeDataEmbbed(Img :TImage):Integer ;
var
 RVal,GVal,BVal,X : integer;
 DtBin : string;
 TempSize : integer;
begin
      DtBin  := '';
      for x:= 2 to 5 do
      begin
          RVal :=GetRValue(Img.Picture.Bitmap.Canvas.Pixels [x,0]);
          GVal :=GetGValue(Img.Picture.Bitmap.Canvas.Pixels [x,0]);
          BVal :=GetBValue(Img.Picture.Bitmap.Canvas.Pixels [x,0]);
          DtBin  := DtBin  + RightStr(DesToBin8 (RVal),3) +
              RightStr (DesToBin8 (GVal),3) + RightStr (DesToBin8 (BVal),3);
      end;
      if IsNan(BinToDes(copy(dtbin,7,24))) then
       result := 0
      else
       begin
         TempSize := BinToDes(copy(dtbin,7,24)) * 8;
         Result := TempSize;
       end;
end;

Function GetCharDataEmbbed(dtBin :string;n : integer):string ;
var
 i : integer;
 Temp : string;
begin
      temp  := '';
      for i:=0 to n-1 do
      begin
          temp := temp  + chr(BinToDes(copy(dtBin,8*i+1,8)))
      end;
      Result := Temp;
end;
```

```
function JoinBiner(str,pixelData : string): string;
var temp : string;
 i :integer;
begin
 temp := '';
 for i := 1 to length(str) do
 begin
  if str[i]='1' then
   temp := temp + intToStr ( StrToInt(str[i]) or StrToInt (pixelData [i]) )
  else
   temp := temp + IntToStr ( StrToInt(str[i]) and StrToInt (pixelData [i]) )
 end;
 Result := temp;
end;


function ExtractData(img : TImage) : string;
label keluar,hasil;
var
 RVal,GVal,BVal,x,y,i,n : integer;
 TempMark,DtBin,strR,strG,StrB,temp : string;
begin
   DtBin  := '';
   i:=0;
   n := GetSizeDataEmbbed(Img) + 48;
   temp := FMain.Caption ;
   for y:=0 to Img.Picture.Bitmap.Height - 1 do
   begin
    for x:=0 to Img.Picture.Bitmap.Width - 1  do
     begin

          RVal :=GetRValue(Img.Picture.Bitmap.Canvas.Pixels [x,y]);
          GVal :=GetGValue(Img.Picture.Bitmap.Canvas.Pixels [x,y]);
          BVal :=GetBValue(Img.Picture.Bitmap.Canvas.Pixels [x,y]);

          strR := '';
          strG := '';
          strB := '';
          if n-i >=3 then  strR := RightStr(DesToBin8 (RVal),3)
          else
           begin
            strR := RightStr(DesToBin8 (RVal),n-i);
            goto hasil;
           end;
```

```
 inc(i,3);
            if n-i >=3 then  strG := RightStr(DesToBin8 (GVal),3)
            else
              begin
                strG := RightStr(DesToBin8 (GVal),n-i);
                goto hasil;
              end;
            inc(i,3);
            if n-i >=3 then  strB := RightStr(DesToBin8 (BVal),3)
            else
              begin
                strB := RightStr(DesToBin8 (BVal),n-i);
                goto hasil
              end;
        Hasil:
            DtBin  := DtBin  + strR + strG + strB;
            FMain.Caption := 'Complete = ' + IntToStr(round((i/n)*100)) +' %';
            inc(i,3);
            if (i>n)or (strR='') or (strB='') or (strB='') then goto keluar;
      end;
    end;
    keluar:
    TempMark :=
GetCharDataEmbbed(copy(DtBin,49,GetSizeDataEmbbed(Img)),GetSizeDataEmbbed(img)div
8);
    FMain.Caption := temp;
    result := TempMark;
end;
procedure TFMain.BOpenClick(Sender: TObject);
begin
  if ODialog.Execute then
  begin
    try
        IOri := TBitmap.Create;
        IOri := ConvertToBitmap(ODialog.FileName);
         IAsli.Picture.Bitmap := IOri;
        EdImages.Text := ODialog.FileName;
        BEkstrak.Enabled := HasEmbbed(IAsli);
        MPesanChange(sender);
    except on EInvalidGraphic do
    begin
     MessageDlg('Invalid Picture Format',mtwarning,[mbok],0 );
      end;
    end;
```

```
procedure TFMain.MPesanChange(Sender: TObject);
begin
  BCesar.Enabled := (trim(MPesan.Text)<>'');
  BBiner.Enabled := BCesar.Enabled;
  BClear.Enabled := BCesar.Enabled;
  MCesar.Clear;
  MBiner.Clear;
  BConver.Enabled := (IAsli.Picture.Bitmap.Empty=false) and (BCesar.Enabled ) ;
end;

procedure TFMain.BCesarClick(Sender: TObject);
begin
  MCesar.Text := CaesarEncrypt(trim(MPesan.Text),KEY);
end;

procedure TFMain.BBinerClick(Sender: TObject);
begin
  MBiner.Text := GetBinerData(MCesar.Text);
end;

procedure TFMain.BClearClick(Sender: TObject);
begin
  MPesan.Clear;
  MPesanChange(sender);
  EdImages.Clear;
  IAsli.Picture.Bitmap :=nil;
  IHasil.Picture.Bitmap :=nil;
  BConver.Enabled:=false;
end;

procedure TFMain.BConverClick(Sender: TObject);
label keluar;
var
  x,y,n : integer;
  R,G,B,newR,newG,newB : byte;
  temp,data,str, binR,binG,binB : string;
begin
   if trim(MPesan.Text) ='' then exit;
   if IAsli.Picture.Bitmap.Empty then
   begin
    MessageDlg('Tidak Ada Gambar yang Dipilih..!!',mtWarning,[mbOK],0);
    exit;
   end;
```

```
if (IAsli.Picture.Bitmap.Height < 3) and (IAsli.Picture.Bitmap.Width <3)  then
  begin
    MessageBox(Handle,'Proses Gagal, Kapasitas Gambar Terlalu Kecil;..!!','Warning!',MB_OK
or MB_ICONERROR);
    Exit;
  end;
  if length(DesToNBin((IAsli.Picture.Bitmap.Height * IAsli.Picture.Bitmap.Width * 9) div 8))>24
then
  begin
    MessageBox(Handle,'Gambar Terlalu Besar..!!','Warning!',MB_OK or MB_ICONERROR);
    Exit;
  end;
  if HasEmbbed(IAsli ) Then
  Begin
    if MessageBox(Handle,'Sudah Terdapat Pesan Dalam Gambar!'#13#10'Yakin Untuk
Menulis Pesan Baru.?','Warning!',MB_ICONQUESTION or MB_YESNO)<>ID_YES Then
      Exit;
  End;
  if MPesan.Lines.Count = 0 then
  begin
    MessageDlg('Tidak Ada Pesan Untuk disisipkan..'+ #13+#10 +'Isikan
Pesan..',mtError,[mbok],0);
    exit;
  end;
  if  length(MPesan.Text) * 8 >= ((IAsli.Picture.Bitmap.Height * IAsli.Picture.Bitmap.Width)*9)-
48 then
  begin
    MessageDlg('Kapasitas Pixel Tidak Mencukupi ' ,mtError,[mbok],0);
    exit;
  end;
  temp := Caption ;
  IAsli.Picture.Bitmap.PixelFormat := pf24bit;

  data := GetBinerData(Mark)+ BinSizeData(length(CaesarEncrypt( MPesan.Text,KEY))) +
GetBinerData(CaesarEncrypt( MPesan.Text,Key)) ;
  GProses.Visible := true;
  GProses.MaxValue := 100;
  n := 1;
  GProses.Progress := n;
  for y := 0 to IAsli.Picture.Height-1 do
  begin
    for x := 0 to IAsli.Picture.Width-1 do
    begin
```

```
R := GetRValue(IAsli.Picture.Bitmap.Canvas.Pixels[x,y]);
        G := GetGValue(IAsli.Picture.Bitmap.Canvas.Pixels[x,y]);
        B := GetBValue(IAsli.Picture.Bitmap.Canvas.Pixels[x,y]);
        binR := DesToBin8(R);
        binG := DesToBin8(G);
        binB := DesToBin8(B);
        str := trim(copy(data,n,3));
        if (str='') then  goto keluar;

        if  (JoinBiner(str,RightStr(binR,length(str)))<> '') then
          newR := BinToDes(LeftStr(binR,8-length(str)) + JoinBiner(str,RightStr(binR,length(str))))
        else  newR := R;
        str := trim(copy(data,n+3,3));

        if JoinBiner(str,RightStr(binG,length(str))) <> '' then
          newG := BinToDes(LeftStr(binG,8-length(str)) +
JoinBiner(str,RightStr(binG,length(str))))
        else newG := G;
        str := trim(copy(data,n+6,3));
        if JoinBiner(str,RightStr(binB,length(str))) <> '' then
          newB := BinToDes(LeftStr(binB,8-length(str)) +
JoinBiner(str,RightStr(binB,length(str))))
        else newB := B;
        SetPixel(IAsli.Canvas.Handle,x,y,RGB(newR,newG,newB)); //penyisipan
        Inc(n,9);
        GProses.Progress := round(((n)/length(data)*100))-1;
        Caption := 'Complete = ' + IntToStr(round(((n)/length(data)*100))-1) +' %'  ;
    end;
  end;
  keluar :
      Caption := temp;
      MessageDlg('Complete..!!' ,mtConfirmation,[mbOK ],0);
      GProses.Hide;
      IHasil.Picture.Bitmap := IAsli.Picture.Bitmap;
      IAsli.Picture.Bitmap := IOri;
      BSave.Enabled := true;


end;
```

```
procedure TFMain.BSaveClick(Sender: TObject);
begin
  if HasEmbbed(IHasil) Then
   Begin
     SavePictureDialog1.Filter := 'Bitmap (bmp) | *.bmp';
     SavePictureDialog1.Title := 'Save image';
     if SavePictureDialog1.Execute Then
       Begin
         IHasil.Picture.SaveToFile(SavePictureDialog1.FileName+'.bmp');
         MessageDlg('Proses Penyimpan Berhasil..',mtInformation,[mbOK],0);
         BClearClick(sender);
       End;
   End
   else
     MessageDlg('Tidak Pesan Dalam gambar..,'#13#10'Silahkan Sisipkan
Pesan..',mtInformation,[mbOK],0);

end;

procedure TFMain.BEkstrakClick(Sender: TObject);
begin
   MPesan.Text := CaesarDecrypt( ExtractData(IAsli),KEY);
end;

end.
```

UAbout.pas

```
unit UAbout;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, sLabel, sPanel, ImgList, ComCtrls;

type
  TFAbout = class(TForm)
    sPanel1: TsPanel;
    sLabelFX1: TsLabelFX;
    sLabelFX2: TsLabelFX;
    Image1: TImage;
    Image2: TImage;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FAbout: TFAbout;

implementation

{$R *.dfm}

end.
```