IMPLEMENTASI PENYISIPAN PESAN FILE KE DALAM GAMBAR DENGAN ALGORITMA HUFFMAN

DIAN NAZELLIANA
AMBAR TRI HAPSARI
nazel.arka@gmail.com
Mahasiswa Pasca Sarjana
Universitas Budi Luhur

Abstrak. Kemajuan teknologi informasi mendorong manusia untuk lebih kreatif lagi membuat trobosan baru dalam berbagai bidang. Sama halnya dalam pengiriman pesan, yang awalnya hanya dapat mengirim pesan teks saat ini sudah dapat mengirim video terlebih dengan teknologi keamanan computer yang makin kompleks. Steganografi merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan sebuah informasi. Metode ini sudah di kenal sejak 2500 tahun yang lalu. Kata steganografi *(steganography)* berasal dari bahasa Yunani *steganos*, yang artinya 'tersembunyi/terselubung', dan *graphein*, 'menulis' sehingga kurang lebih artinya "menulis (tulisan) terselubung". Teknik ini meliputi banyak sekali metoda komunikasi untuk menyembunyikan pesan rahasia. Metode ini termasuk tinta yang tidak tampak, *microdots*, pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi spektrum lebar. Teknik Stenografi digunakan untuk menyisipkan file kedalam sebuah gambar sehingga orang ketiga tidak dapat membuka pesan ini. Sebelum file dikirim harus diubah terlebih dahulu kedalam format file terkompresi, dalam hal ini digunakan aplikasi Winrar, kemudian diselubung dengan file gambar sehingga yang tampak hanya file gambar saja dalam format jpg. Digunakan pemrograman Visual Basic 6 untuk melakukan pengkodean file tersebut. Algoritma Huffman adalah salah satu algoritma kompresi tertua yang disusun oleh David Huffman pada tahun 1952. Algoritma tersebut digunakan untuk membuat kompresi jenis lossy compression, yaitu pemampatan data dimana tidak satu byte pun hilang sehingga data tersebut utuh dan disimpan sesuai dengan aslinya. Algoritma Huffman menggunakan prinsip pengkodean yang mirip dengan kode Morse, yaitu tiap karakter (simbol) dikodekan hanya dengan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang.

Kata Kunci : Steganografi, Winrar, Algoritma Huffman

PENDAHULUAN

Kemajuan teknologi membuat pengiriman pesan semakin mudah, cepat dan murah. Hal ini membuat meningkatnya pengiriman pesan melalui dunia maya dimana kita dapat mengirim pesan berupa teks, audio dan video. Peningkatan pengiriman pesan yang signifikan membuat segelintir orang memcari cara untuk mengakses pesan yang bukan haknya. Salah satu cara untuk mengatasi hal tersebut yaitu dengan metode steganografi yang telah digunakan sejak 2500 tahun yang lalu untuk menyembunyikan pesan rahasia.

Steganografi adalah ilmu atau seni menyembunyikan pesan kedalam sebuah media sedemikian rupa sehingga keberadaan pesan tidak di ketahui atau disadari oleh orang selain yang mengirim pesan dan yang menerima pesan tersebut.[3]

Walaupun steganografi dapat dikatakan mempunyai hubungan yang erat dengan kriptografi, tapi metoda ini sangat berbeda dengan kriptografi. Kriptografi mengacak pesan sehingga tidak dimengerti, sedangkan steganografi menyembunyikan pesan

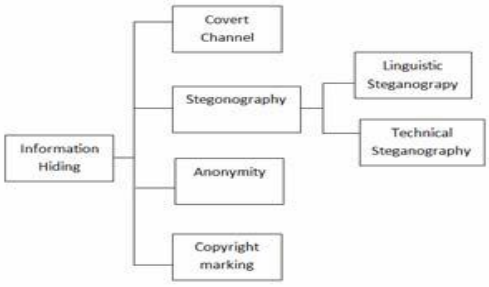
sehingga tidak terlihat. Pesan dalam cipherteks mungkin akan menimbulkan kecurigaan sedangkan pesan yang dibuat dengan steganografi tidak akan. Kedua teknik ini dapat digabungkan untuk mendapatkan metoda pengiriman rahasia yang sulit dilacak.

Pertama pesan dienkrip, kemudian cipherteks disembunyikan dengan cara steganografi pada media yang tampak tidak mencurigakan. Cara ini sangat berguna jika digunakan pada cara steganografi komputer karena banyak format file digital yang dapat dijadikan media untuk menyembunyikan pesan. Format yang biasa digunakan diantaranya:

- Format image: bitmap (bmp), gif, pcx, jpeg, dll.
- Format audio: wav, voc, mp3, dll.
- Format lain: teks file, html, pdf, dll.

Metoda yang digunakan untuk menyembunyikan pesan pada media digital tersebut berbeda-beda. Contohnya pada file image pesan dapat disembunyikan dengan menggunakan cara menyisipkanya pada bit rendah (lsb) pada data pixel yang menyusun file tersebut. Seperti kita ketahui untuk file bitmap 24 bit maka setiap pixel (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian pada setiap pixel file bitmap 24 bit kita dapat menyisipkan 3 bit data.

Melalui Penelitian ini di buat program yang dapat menyisipkan pesan berupa file kedalam sebuah gambar dengan program Visual Basic 6. Dalam program ini file harus diubah dahulu ke dalam bentuk WinRAR agar dapat di sembunyikan ke dalam gambar yang lebih besar ukuran filenya .



Gambar 1

TINJAUAN PUSTAKA

Definisi Visual Basic

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan Integrated Development Environment (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM).

Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.

Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic Program-program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi luar tambahan.

Dalam pemrograman untuk bisnis, Visual Basic memiliki pangsa pasar yang sangat luas. Sebuah survey yang dilakukan pada tahun 2005menunjukkan bahwa 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk Visual Basic, yang diikuti oleh C++, JavaScript

Definisi WinRAR

WinRAR adalah sebuah aplikasi yang sering di gunakan untuk memadatkan file. Memadatkan di sini artinya file di kompres sehingga menjadi lebih ringan. Selain itu winrar sering di gunakan untuk menyederhanakan banyak file sehingga dapat di satukan menjadi satu file. Winrar juga dapat di gunakan untuk memecah file yang berektensi ZIP,ACE, AJR dll.[1]

Dalam banyak kasus, format RAR menciptakan kompres/tekanan yang mantap/padat jika dibandingkan dengan format ZIP. Kelebihan RAR yang lain adalah mendukung multivolume. Jika Anda harus menciptakan suatu arsip multi-volume, maka menggunakan RAR adalah pilihahnnya.

Format RAR juga mempunyai beberapa hal penting lain yang tidak dijumpai dalam format ZIP. Misalnya, kemampuan RAR untuk merekontruksi arsip yang rusak, perlindungan/kunci arsip, dan melindungi arsip penting dari modifikasi yang tidak disengaja.

Format RAR bisa menangani file yang bisa dikatakan ukuran tak terbatas (sampai kepada 8.589.934.591 GB). Sedangkan ukuran maksimum dari satu file ZIP hanya 2 GB. Sistem file yang mendukung file berukuran lebih dari 4 GB adalah NTFS atau yang lebih terbaru dari itu.

Fitur-Fitur WinRAR:

1. Mendukung arsip *.RAR dan *.ZIP 2.0;
2. Sangat canggih dengan algoritma kompres asli;
3. Memiliki algoritma khusus yang dioptimalkan untuk text, audio, graphics, 32-bit dan 64-bit Intel kompres executables;
4. Memiliki antar muka Shell yang menyertai fasilitas drag-and-drop dan Wizard;
5. Memiliki antar muka Command Line;
6. Mampu menangani berkas: 7Z, ACE, ARJ, BZ2, CAB, GZ, ISO, JAR, LZH, TAR,
7. Mengkompres padat, dan dapat dinaikkan tingkat kompresi 10% s.d. 50% dari kompresi normal, terutama sekali ketika pengepakan sejumlah besar file kecil yang serupa;
8. Dapat membuat arsip-arsip multivolume;
9. Dapat membuat arsip yang mampu mengekstrak dirinya sendiri (self-extracting), dapat juga multivolume, menggunakan modul SFX bawaan atau modifikasi;
10. Dapat memulihkan (recovery) arsip yang rusak;
11. Dapat memulihkan (recovery) arsip multivolume dan merekontruksi bagian yang hilang dari arsip multivolume;
12. Mendukung nama file Unicode;
13. Dan memiliki banyak fasilitas lainnya, seperti: encryption, archive comments, error logging, dll.

Definisi Jpeg

Join Photografic Experts Group (JPEG) adalah format gambar yang banyak di gunakan untuk menyimpan gambar-gambar dengan ukuran lebih kecil. Beberapa karakteristik gambar JPEG mampu menayangkan warna dengan kedalaman 24-bit

true color. Mengkompresi gambar dengan sifat lossy. Umumnya untuk menyimpan gambar-gambar hasil foto.

Algoritma Huffman

Algoritma Huffman adalah salah satu algoritma kompresi tertua yang disusun oleh David Huffman pada tahun 1952. Algoritma tersebut digunakan untuk membuat kompresi jenis lossy compression, yaitu pemampatan data dimana tidak satu byte pun hilang sehingga data tersebut utuh dan disimpan sesuai dengan aslinya. Algoritma Huffman menggunakan prinsip pengkodean yang mirip dengan kode Morse, yaitu tiap karakter (simbol) dikodekan hanya dengan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang.[2]

Cara Kerja dalam menggunakan algoritma Huffman, yaitu:

Mengubah sebuah string atau masukan dari user dan menghitung kemunculan setiap huruf. Setelah itu, buat daftar dari huruf tersebut beserta peluang kemunculannya karena huruf tersebut akan menjadi daun dalam pohon Huffman. Kode ini biasanya identik dengan pohon biner yang diberi label 0 untuk cabang kiri dan 1 untuk cabang kanan.

Mengubah kembali daftar yang telah dibuat untuk kemudian membedakan daun (berupa huruf dengan pelung terkecil) dan penjumlahan 2 daun yang akan menjadi akar dari dua daun sebelumnya.

CONTOH --> ADADIAM

Tabel 1. Kode ASCII

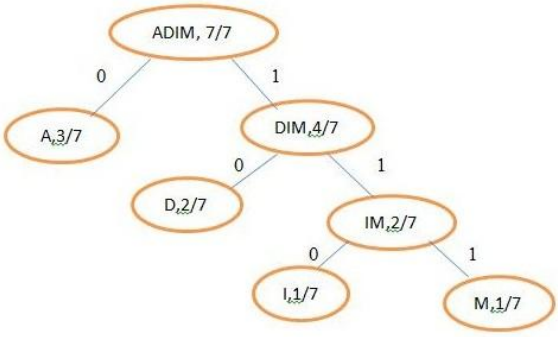
Simbol	Kode ASCII
A	01000001
D	01000100
A	01000001
D	01000100
I	01001001
A	01000001
M	01001101

Terdapat 7 karakter dalam string , maka memori yang dibutuhkan adalah 7 x 8 bit = 56 bit. Memori = n x 8 bit n = jumlah karakter dalam sebuah string. Selanjutnya panjang kode pada tiap karakter dipersingkat, terutama untuk karakter yang frekuensi kemunculan besar. A = 3, D = 2, I = 1, M = 1.

Tabel 2. Kode Huffman

Simbol	Kemunculan	Peluang	Kode Huffman
A	3	3/7	0
D	2	2/7	10
I	1	1/7	110
M	1	1/7	111

Buat kode tersebut ke bentuk pohon(Tree).



Gambar 2. Pohon Kode Huffman

Sehingga string 'ADADIAM' jika representasikan dalam bit menjadi --> 0100101100111

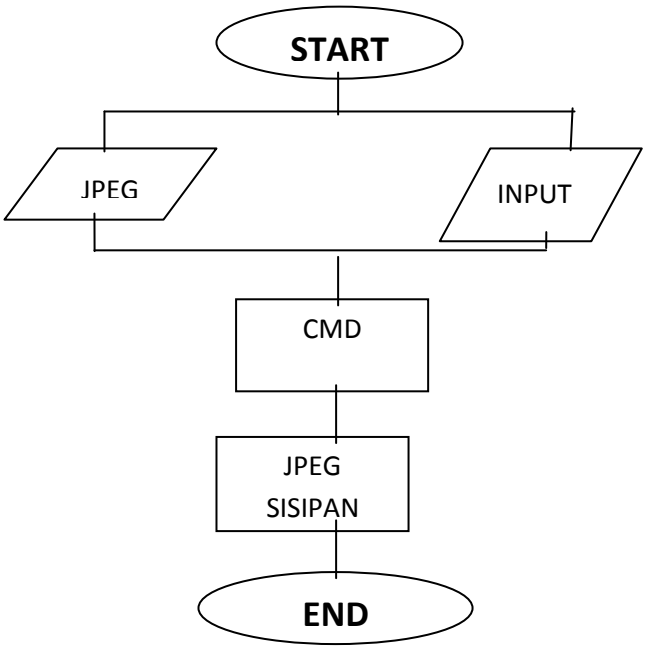
Maka, bit yang dibutuhkan hanya 13 bit dengan Algoritma Huffman. Lebih kecil bukan

ANALISIS KEBUTUHAN DAN PERANCANGAN

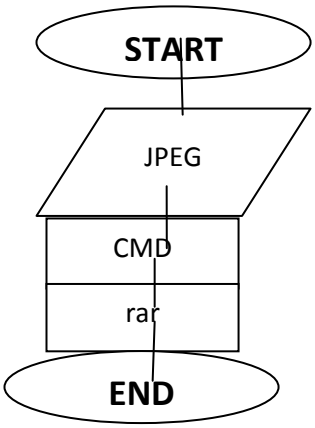
Sistem steganografi yang dibahas akan di fokuskan kepada bagaimana cara membangun suatu sistem steganografi pada citra digital file gambar yang efisien dan untuk mengeksploitasi keterbatasan sistem penglihatan manusia.

Sistem ini terdiri dari dua buah sub sistem yaitu : sistem penyisipan dan sistem pengekstrakkan. Sistem penyisipan berfungsi untuk melakukan proses penyembunyian pesan berupa file ke file citra digital gambar.

Sistem pengekstrakkan berfungsi untuk melakukan pengekstrakkan file untuk memeperoleh pesan yang telah disisipkan ke dalam file gambar tersebut. Komponen pada sistem pengekstrakkan ini terdapat komponen untuk membuka File pesan yang berada pada gambar.



Gambar 3. FlowChart Penyisipan Gambar



Gambar 4. FlowChart Pengekstrakan File

PERMASALAHAN

Steganografi mempunyai kelebihan dalam aspek penyembunyian pesan di mana pesan yang disembunyikan tidak terlihat kasat mata berupa kode tertentu seperti kriptografi, dikarenakan dalam steganografi pesan dititipkan pada suatu gambar. Permasalahannya bagaimana agar pesan tersebut dapat dititipkan pada gambar tanpa terlihat berkurangnya kualitas dari gambar tersebut, dan metode apa yang tepat agar pesan yang dititipkan tidak mengurangi kualitas gambar?

PEMBAHASAN

Steganografi yang dibahas di sini adalah penyembunyian data di dalam citra digital. Meskipun demikian, penyembunyian data dapat juga dilakukan pada wadah berupa suara digital, teks, ataupun video. Penyembunyian data rahasia ke dalam citra digital akan mengubah kualitas citra tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data adalah:

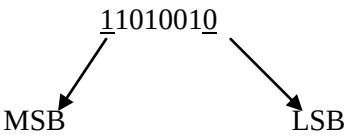
1. Mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2. Data yang disembunyikan harus mampu bertahan terhadap manipulasi yang dilakukan pada citra penampung. Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak.
3. Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*).

File yang akan dsisipkan berupa file winrar.

Teknik Penyembunyian Data

Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia. Salah satu metode penyembunyian data yang sederhana adalah *LSB Modification*.[4]

Perhatikan contoh sebuah susunan bit pada sebuah *byte*:



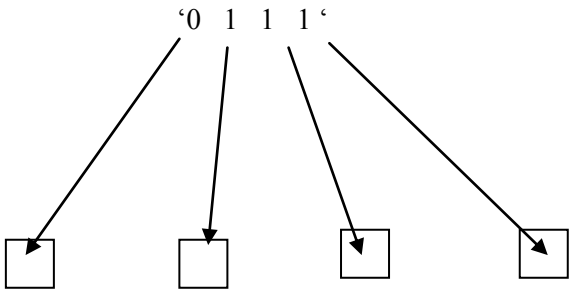
LSB = Least Significant Bit
MSB = Most Siginificant Bit

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

Misalkan segmen data citra sebelum perubahan :

0 0 1 1 0 0 1 1 1 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 1 1 1

Segmen data citra setelah di sembunyikan



0011 0 0 1 0 101000 1 1 111000 1 1 0 1 1 0 1 1 1

Untuk memperkuat teknik penyembunyian data, bit-bit data rahasia tidak digunakan mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka maka *byte* yang diganti bit *LSB*-nya dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49. Bilangan acak dibangkitkan dengan *pseudo-random-number-generator* (*PRNG*) kriptografi. *PRNG* kriptografi sebenarnya adalah algoritma kriptografi yang digunakan untuk enkripsi. *PRNG* dibangun dengan algoritma *DES* (*Data Encryption Standard*), algoritma *hash MD5*, dan mode kriptografi *CFB* (*Chiper-Feedback Mode*).

Tujuan dari enkripsi adalah menghasilkan sekumpulan bilangan acak yang sama untuk setiap kunci enkripsi yang sama. Bilangan acak dihasilkan dengan cara memilih bit-bit dari sebuah blok data hasil enkripsi. Teknik penyembunyian data untuk citra 8-bit berbeda dengan citra 24-bit. Seperti diketahui berkas citra *bitmap* terdiri atas bagian *header*, palet *RGB*, dan data *bitmap*.

Tetapi bagaimanapun dengan steganografi bukan berarti pengiriman pesan menjadi benar-benar aman karena telah dikembangkan juga metoda-metoda untuk mendeteksi keberadaan pesan yang dibuat dengan cara ini. Software untuk mendeteksi tersebut diantaranya Stegdetect, http://www.outguess.org/ yang juga merupakan pembuat Steghide yang telah dikembangkan sedemikian rupa sehingga tidak dapat dideteksi oleh Stegdetect.

Pada citra 8-bit, setiap elemen data *bitmap* menyatakan indeks dari peta warnanya di palet *RGB*.

Format citra 8-bit (256 warna)

<*header*>

<palet *RGB*>

	R	*G*	*B*
1	20	45	24
2	14	13	16
3	12	17	15
…			
256	46	78	25

<data *bitmap*>
2 2 1 1 1 3 5 …
.

Pada citra 24-bit, tidak terdapat palet *RGB*, karena nilai *RGB* langsung diuraikan dalam data *bitmap*. Setiap elemen data *bitmap* panjangnya 3 *byte*, masing-masing *byte* menyatakan komponen *R*, *G*, dan *B*.

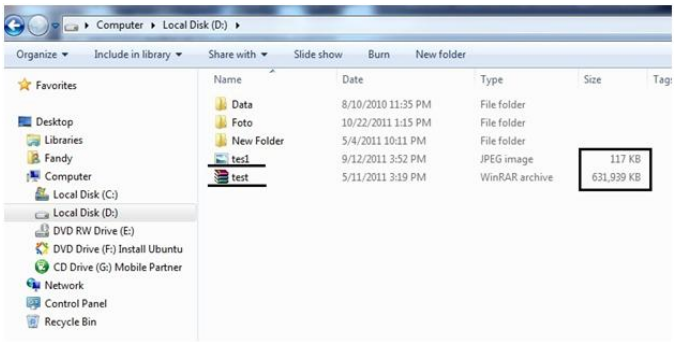
Format citra 24-bit (16 juta warna)

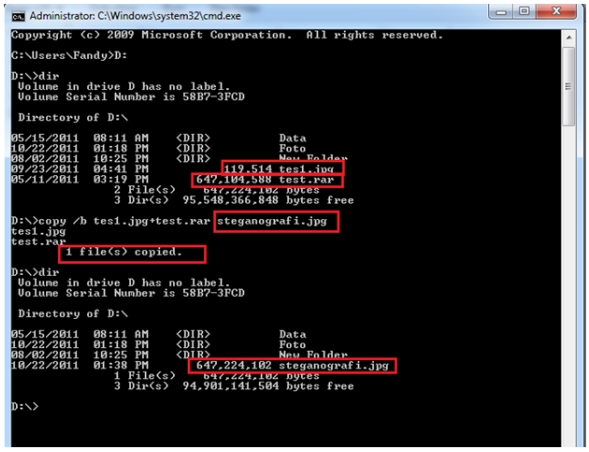
<*header*>
<data *bitmap*>
2 2 1 1 1 3 5 …

Pada contoh format citra 24-bit di atas, *pixel* pertama mempunyai $R = 2$, $G = 2$, $B = 1$.

Tutorial Melakukan Teknik Steganografi

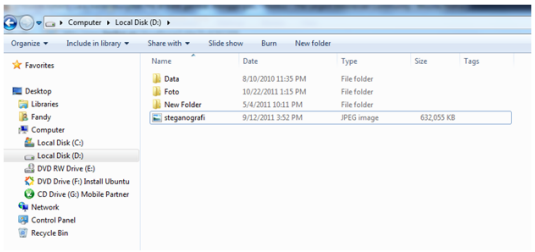
- Sediakan file dengan format rar atau zio. contoh: test.rar
- Sediakan gambarnya. contoh: tes1.jpg\ Masuk CMD, caranya tekan windows+r, ketik cmd
- Ketik copy /b tes1.jpg+test.rar steganografi.jpg
- Lihat jika ada gambar dengan nama steganografi.jpg berarti berhasil





Gambar 5.

Dalam data D: akan muncul gambar Steganografi.JPEG yang sebelum nya tidak ada dengan Size gambar yang lebih besar

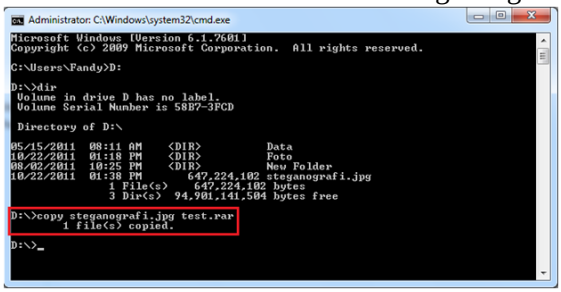


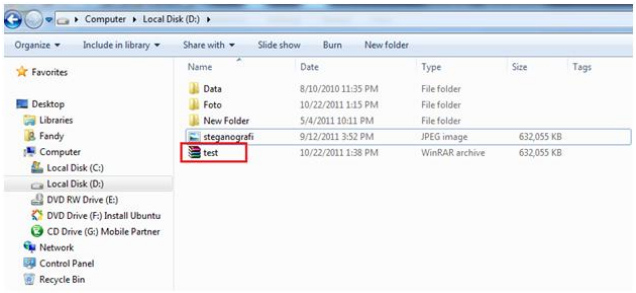
Gambar 6.

File berhasil di sembunyikan pada gambar *(steganografi)*

Cara membuka file yang ada dibalik gambarnya

- Buka CMD
- Ketik copy steganografi.jpg test.rar
- Maka akan muncul file test.rar dan anda tinggal mengekstraknya saja
- Atau Klik kanan di file hasil gabungannya, lalu gunakan open with "winrar"





Gambar 7.

Teknik penggantian bit pada citra bukan 24-bit.

Sebelum melakukan penggantian bit *LSB*, semua data citra yang bukan tipe 24-bit diubah menjadi format 24-bit. Jadi, setiap data *pixel* sudah mengandung komponen *RGB*. Setiap *byte* di dalam data *bitmap* diganti satu bit *LSB*-nya dengan bit data yang akan disembunyikan. Jika *byte* tersebut merupakan komponen hijau (G), maka penggantian 1 bit LSB-nya hanya mengubah sedikit kadar warna hijau, dan perubahan ini tidak terdeteksi oleh mata manusia.

Teknik penggantian bit pada citra 24-bit.

Karena data bitmap pada citra 24-bit sudah tersusun atas komponen *RGB*, maka tidak perlu dilakukan perubahan format.
Setiap *byte* di dalam data bitmap diganti satu bit *LSB*-nya dengan bit data yang akan disembunyikan.

Teknik Pengungkapan Data

Data yang disembunyikan di dalam citra dapat dibaca kembali dengan cara pengungkapan (*reveal* atau *extraction*). Posisi *byte* yang menyimpan bit data dapat diketahui dari bilangan acak yang dibangkitkan oleh *PRNG*. Karena algoritma kriptografi yang digunakan menggunakan kunci pada proses enkripsi, maka kunci yang sama digunakan untuk membangkitkan bilangan acak. Bilangan acak yang dihasilkan sama dengan bilangan acak yang dipakai pada waktu penyembunyian data. Dengan demikian, bit-bit data rahasia yang bertaburan di dalam citra dapat dikumpulkan kembali.

Algoritma ZIP Compression

Algoritma pemampatan data dengan format data ZIP termasuk dalam algoritma kompresi atau pemampatan yang bersifat *lossless*. Berbeda dengan algoritma pemampatan yang bersifat *lossy*yang menghilangkan sebagian informasi dari berkas yang di mampatkan untuk mendapatkan hasil yang optimum, algoritma kompresi yang bersifat lossless seperti ZIP tidak membuang sedikitpun informasi yang dimiliki oleh berkas asal. Algoritma kompresi yang bersifat *lossy* umumnya digunakan untuk memampatkan berkas–berkas gambar, video ataupun suara, hal ini menimbang perubahan (*penghilangan*) sedikit pada berkas asal tidak akan menimbulkan efek yang mampu ditangkap oleh indra manusia.

Sedangkan algoritma kompresi yang bersifat *lossless* umumnya digunakan untuk berkas teks atau binary (*executable*). Hal ini mengingat perubahan yang kecil pada berkas yang dikompresi akan memberi pengaruh besar pada berkas hasil kompresi saat di dekompresi ulang. Misalnya pada suatu berkas program computer (*source code*), perubahan yang terjadi walaupun sedikit akan berakibat pada kesalahan kode program tersebut saat di kompilasi setelah didekompresi.

Berkas termampatkan dengan format zip dibuat dengan menggunakan algoritma kompresi deflate. Sebagaimana format gzip berkas terkompresi dengan format zip dibuat dengan algoritma deflate yang pertama kali didisain oleh Philip Katz (1962-2000), algoritma deflate sendiri merupakan algoritma yang berbasiskan algoritma LZ77 dan kode Huffman (*Huffman Codes*). Spesifikasi format kompresi zip distandardisasi melalui RFC1952 yang ditulis oleh *Peter Deutsch*.

Meskipun *algoritma deflate* tidak dirancang untuk suatu tipe berkas secara spesifik, akan tetapi metode –metode pemampatan data yang dirancang khusus untuk tipe berkas tertentu, yang umumnya memiliki kerumitan yang lebih tinggi, umumnya memiliki performansi (*dalam segi ukuran berkas hasil kompresi*) yang lebih tinggi. Pada umumnya algoritma deflate (*termasuk zip*) memiliki nilai faktor kompresi (*compression factor*) antara 2.5 sampai 3 untuk pemampatan berkas tipe teks dan memiliki nilaiyang lebih kecil jika yang berkas dimampatkan adalah tipe binary (*executable*). Factor kompresi (*compression factor*) merupakan invers dari nilai rasio kompresi (*compression ratio*) yangmenunjukkan persentase ukuran berkas hasil pemampatan dibandingkan ukuran berkas sebelum dimampatkan.

Persamaan 1. Rasio Kompresi

$$RK = \frac{Ukuran\_file\_Output}{Ukuran\_File\_Input}$$

RK = rasio kompresi

Persamaan 2. FaktorKompresi

$$FK = \frac{Ukuran\_File\_Input}{Ukuran\_File\_Output}$$

FK = faktor kompresi

Dari persamaan tersebut terlihat, bahwa rasio kompresi akan selalu bernilai kurang dari 1, jadi jika suatu algoritma kompresi memiliki nilai rasio kompresi 0,5 maka algoritma ini mampu memampatkan berkas hingga menjadi separuh (50%) dari ukuran semula. Jadi semakin kecil nilai rasio kompresi dari suatu algoritma kompresi maka semakin bagus algoritma tersebut. Berkebalikan dengan nilai rasio kompresi adalah nilai faktor kompresi. Sehingga semakin besar nilai faktor kompresi dari suatu algoritma pemampatan data, maka algoritma tersebut berarti semakin baik. Pada umumnya nilai faktor kompresi lebih sering digunakan sebagai standard ukuran mengingat secara alamiah nilainya menunjukkan tingkat keandalan dari suatu algoritma (semakin besar nilai = semakin bagus kualitas).

DESAIN

Desain anatar muka yang di buat bertujuan untuk memudahkan user dalam melakukan proses penyisipan dan pengambilan file ke dan dari media gambar. Dalam desain antar muka ini, penggunaan sistem antar muka dibedakan menjadi 2 bagian utama yaitu bagian untuk menyembunyikan informasi yang nantinya akan disebut encode data dan bagian untuk mengambil informasi dari file stegoyang nantinya akan disebut decode data.

Tampilan awal dari program utama – aplikasi steganografi untuk menyisipkan pesan file ke dalam image :

Gambar 8.

Cara Menjalankan Programnya :

- **PROSES ENCODE**

Pertama – tama kita klick compressed File lalu di pilih file yang akan di sisipkan ke dalam gambar. File yang disisipkan harus sudah berekstensi rar. Karena ukurannya akan menjadi lebih kecil. Setelah itu kita masukan gambar yang akan menjadi cover imagenya. Ukuran cover image harus lebih lebih besar daripada file yang akan di sisipkan kedalamnya.
Setelah kita pilih gambar yang akan menjadi cover image kita pilih Output yang kita inginkan. Lalu kita klik sisipkan file. Maka pesan yang akan kita kirimkan akan berada pada tempat yang kita inginkan.

Setelah di pilih outputnya maka tampilan akan berada di tempat yang kita inginkan

Untuk Program Penyisipan gambarnya adalah :

Gambar 9.

Gambar 10.

- **PROSES DECODE**

Pada Proses ini adalah untuk membuka File yang telah dikirimkan. Tahap pertamanya adalah mencari gambar yang telah disipkan file dan output filenya. Setelah itu di klik ambil file. Maka akan terbuka file yang telah di sisipkan tadi.

Gambar 11.

Setelah di pilih outputnya maka tampilan akan berada di tempat yang kita inginkan

Gambar 12.

HASIL PENGUJIAN

Dalam melakukan pengujian kami memanfaatkan beberapa berkas cover dan beberapa file data yang akan disimpan dalam cover. Berkas–berkas yang akan di gunakan tesebut adalah sebagai berikut :

Tabel 3.

Nama File	Ukuran Asli	Ukuran Encoded	Ukuran Decoded
File.rar	2,84 MB	3,43 MB	3.16 MB
Tulip.jpg	606 KB		3,43 MB

PENUTUP

Kesimpulan dari penulisan ini adalah sebagai berikut:

1. Metode Steganografi menyembunyikan pesan file kedalam file gambar yang merupakan pembungkus pesan file.
2. File yang sudah terselubung dalam proses encode menjadi file gambar dengan ukuran file yang lebih besar namun tidak merubah komposisi fisik gambar.
3. Dalam penelitian ini digunakan pemrograman Visual Basic 6 dan format file yang disembunyikan dikompresi dengan aplikasi Winrar. Format gambar dalam ekstensi jpg. Hasil encode dalam format gambar berekstensi jpg.
4. Hasil pemisahan file ter-encode (decode) dalam bentuk file terkompresi berekstensi rar yang isinya adalah file gambar dan pesan file asli dengan ukuran tetap.

5. Efektivitas steganografi dalam pengiriman pesan file rahasia cukup kompeten dalam dunia keamanan computer, mengingat secara kasat mata file dalam bentuk file (gambar) biasa yang tidak terkira tersimpan pesan rahasia di dalamnya.

DAFTAR PUSTAKA

http://tutorialwinrar.frombanda.com/id/ rar

http://dekadwi.com/artikel2.html algoritma huffman

Iswahyudi, C. (2008). **Penyisipan Pesan Rahasia pada Citra Digital dengan Teknik Steganografi.**

Santana, W. (2005). **Tugas Akhir: Peningkatan Ketahanan Steganografi LSB Pada Media Audio Dengan Menggunakan Metode Penambahan Novel**. Bandung: IT Telkom.