MAKALAH SEMINAR TUGAS AKHIR

APLIKASI PENYISIPAN DATA KE DALAM CITRA TERMAMPAT BERFORMAT GIF DENGAN BAHASA PEMROGRAMAN DELPHI

Arif Nur Budi Setiyo*, Achmad Hidayatno**, R. Rizal Isnanto**

Abstrak **- Teknologi digital saat ini telah memberi kemudahan untuk melakukan akses serta mendistribusikan berbagai informasi dalam format digital.Steganografi adalah teknik menyembunyikan data rahasia di dalam media digital sehingga keberadaan data rahasia tersebut tidak dapat diketahui oleh orang lain. Pemampatan GIF merupakan bentuk pemampatan tak berugi sehingga tidak menghilangkan data yang terkandung di dalamnya. Delphi merupakan bahasa pemrograman visual yang memiliki fasilitas pengolah angka, basis data, grafik, dan animasi, sehingga memungkinkan untuk membuat program dengan komponen visual dalam bentuk objek.**

Piksel suatu citra berwarna tersusun oleh tiga elemen warna RGB (red, green, dan blue). Dengan mengganti satu bit LSB elemen RGB tersebut, dapat diperoleh ruang pada citra untuk menyimpan pesan rahasia. Langkah-langkah penelitian ini adalah: (1) Citra yang telah diambil di reduksi warna dari 24-bit menjadi 8-bit yang berfungsi untuk menyiapkan citra ke dalam format GIF, (2) Menduplikasi palet yang berfungsi untuk menyamakan palet warna pada citra asli, (3) Menyisipkan satu bit ke dalam bit terakhir citra, (4) Menyimpan citra hasil steganografi dalam format GIF, (5) Menguji keberhasilan program dengan melakukan proses desteganografi.

Hasil penelitian adalah bahwa ukuran citra menentukan kapasitas data yang dapat disisipkan. Semua data mampu disisipkan ke dalam citra jika kapasitas di bawah maksimal yang mampu ditampung citra uji. Namun, pengujian tidak berhasil dilakukan dengan data uji di atas kapasitas maksimal yang dapat ditampung oleh citra. Proses steganografi dapat dilakukan jika citra dari aplikasi ini maksimal berukuran 6800×5000 piksel karena kebutuhan memori yang besar untuk membuka citra tersebut. Untuk menyisipkan data dalam format teks hanya dibutuhkan waktu yang singkat karena ukuran berkas yang relatif kecil. Waktu yang dibutuhkan untuk menyisipkan data audio dan video sangat lama karena ukuran berkas yang relatif besar.

Kata- kunci : Citra Digital, GIF, Palet, Steganografi, LSB (Least Significant Bit), piksel.

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Kemajuan teknologi informasi yang pesat serta potensi pemanfaatannya yang luas, membuka peluang bagi pengaksesan, pengelolaan, dan pendayagunaan citra digital dalam volume yang besar secara tepat dan akurat.

Steganografi merupakan salah satu cara untuk menyembunyikan suatu pesan data rahasia di dalam data atau pesan lain yang tampak tidak mengandung apa-apa, kecuali bagi orang yang mengerti kuncinya. Dalam bidang keamanan komputer, steganografi digunakan untuk menyembunyikan data rahasia saat enkripsi yang dilakukan tidak bersamaan. Jadi, walaupun enkripsi berhasil dipecahkan pesan / data rahasia tetap tidak terlihat. Selain itu, pada kriptografi pesan disembunyikan dengan diacak sehingga pada kasus-kasus tertentu dapat dengan mudah mengundang kecurigaan, sedangkan pada steganografi pesan disamarkan dalam bentuk yang relatif aman sehingga tidak terjadi kecurigaan itu. Steganografi dapat digunakan pada berbagai macam bentuk data, yaitu citra, audio, dan video.

Pada tugas akhir ini akan dibahas penyisipan data ke dalam sebuah citra yang hasil akhirnya berupa citra berformat GIF dengan metode penyisipan LSB.

1.2 Tujuan Penelitian

Tujuan dari pembuatan Tugas Akhir ini adalah untuk menyisipkan data rahasia kedalam sebuah citra berformat GIF dengan metode penyisipan LSB.

1.3 Batasan Masalah

Pembatasan masalah pada penulisan tugas akhir ini sebagai berikut :

1. Proses Steganografi yang dilakukan adalah menyisipkan LSB ke dalam citra yang akan disisipi oleh data rahasia.
2. Data rahasia yang akan dsisipkan berupa data teks (***. txt, *.doc,*. Pdf**), data citra (***.jpeg, *.gif, *.PNG)** serta data audio (***.mp3**) dan video (***.rm, *.3gpp**).
3. Citra masukan berupa citra berwarna yang akan disisipi oleh data rahasia berupa citra belum dimampatkan berformat ***.bmp** maupun citra yang telah dimampatkan berformat ***.jpeg,*. jpg, *.ico, *.wmf**.
4. Hasil akhir dari data yang telah disisipi data rahasia disimpan ke dalam citra 8 bit dalam citra berformat GIF.

II. DASAR TEORI

2.1 Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang seperti yang diinginkan. Teknik pengolahan citra adalah mentransformasikan citra menjadi citra lain. Jadi masukannya berupa citra, keluarannya juga

* Mahasiswa Jurusan Teknik Elektro UNDIP
** Staf Pengajar Jurusan Teknik Elektro UNDIP

citra dengan kualitas yang lebih baik dari citra masukan. Beberapa contoh aplikasi pengolahan citra digital, namun pada dasarnya tujuan dari pengolahan citra digital adalah peningkatan kualitas citra (*Image enhancement*), pemulihan citra (*Image restoration*), pengukuran dan analisis citra (*Image measurement and analysis*), rekonstruksi citra (*Image reconstruction*), Pemampatan data citra (*Image data compression*).

2.2 Pemampatan Data Citra

Secara umum teknik pemampatan dapat dibagi menjadi dua bagian besar, tak berugi dan berugi. Algoritma pemampatan tak berugi menghilangkan hanya informasi berulang saja, sehingga pada saat depemampatan, citra yang telah dimampatkan dapat ditampilkan tepat seperti aslinya. Algoritma pemampatan berugi selain menghilangkan informasi yang berulang, juga menghilangkan informasi yang tidak relevan, dan dengan demikian hanya memungkinkan rekonstruksi yang mendekati citra aslinya, bukannya duplikat yang sama persis. Sebagaimana dapat kita perkirakan, algoritma pemampatan berugi memungkinkan rasio yang lebih tinggi. Contoh jenis pemampatan data citra yang berkarakteristik tak berugi adalah GIF *Encoding*.

2.3 Pemampatan GIF

Graphic Interchange Format (GIF, dibaca *jiff*, tetapi kebanyakan orang menyebutnya dengan *giff*) yang dibuat oleh Compuserve pada tahun 1987 untuk menyimpan berbagai citra dengan format bitmap menjadi sebuah berkas yang mudah untuk diubah pada jaringan komputer. GIF adalah berkas format citra yang paling tua pada web, dan begitu dekatnya berkas format ini dengan web pada saat itu sehingga menggunakan format ini. GIF mendukung sampai 8 bit piksel , itu berarti maksimum jumlah warnanya 256 warna (2^8= 256 warna), dan mengunakan varian dari algoritma pemampatan Lempel-Ziv Welch (LZW).[2]

2.4 Algoritma LZW

Algoritma LZW dikembangkan dari metode pemampatan yang dibuat oleh Ziv dan Lempel pada tahun 1977. Algoritma ini melakukan pemampatan dengan menggunakan kamus, dimana fragmen-fragmen teks diganti dengan indeks yang diperoleh dari sebuah kamus. Prinsip sejenis juga digunakan dalam sandi Braille, di mana sandi-sandi khusus digunakan untuk menunjukkan kata-kata yang ada. Pendekatan ini bersifat adaptif dan efektif karena banyak karakter dapat disandikan dengan mengacu pada string yang telah muncul sebelumnya dalam teks. Prinsip pemampatan tercapai jika referensi dalam bentuk pointer dapat disimpan dalam jumlah bit yang lebih sedikit dibandingkan string aslinya.

2.4.1 Algoritma pemampatan LZW

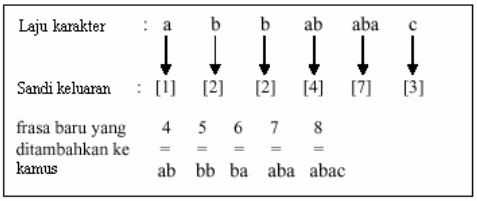
Algoritma pemampatan LZW sebagai berikut :

1. Kamus diinisialisasi dengan semua karakter dasar yang ada : {‘A’..’Z’,’a’..’z’,’0’..’9’}.
2. Jika *P* menunjukkan karakter pertama dalam laju karakter dan *C* merupakan karakter berikutnya dalam laju karakter. Carilah string (*P+C*) dalam kamus. Apakah string *P+C* terdapat dalam kamus? Jika ya, maka *P* merupakan *P+C* (gabungkan *P* dan *C* menjadi string baru).
   a. Keluaran sebuah sandi untuk menggantikan string *P*.
   b. Tambahkan string *(P + C)* ke dalam kamus dan berikan nomor/sandi berikutnya yang belum digunakan dalam kamus untuk string tersebut.
   c. String *P* menggantikan string *C*.
3. Apakah masih ada karakter berikutnya dalam laju karakter ? Jika ya, maka kembali ke langkah 2.
   Jika tidak, maka keluaran sandi yang menggantikan string *P*, lalu terminasi proses berhenti.

Sebagai contoh, string “ABBABABAC” akan dimampatkan dengan algoritma LZW. Isi kamus pada awal proses diset dengan tiga karakter dasar yang ada: “A”, “B”, dan “C”. Tahapan proses pemampatan ditunjukkan pada Tabel 2.1. Kolom posisi menyatakan posisi sekarang dari laju karakter dan kolom karakter menyatakan karakter yang terdapat pada posisi tersebut. Kolom kamus menyatakan string baru yang sudah ditambahkan ke dalam kamus dan nomor indeks untuk string tersebut ditulis dalam kurung siku. Kolom keluaran menyatakan sandi keluaran yang dihasilkan oleh langkah pemampatan. Hasil proses pemampatan ditunjukkan pada Gambar 2.1.

Tabel 2.1 Tahapan proses pemampatan LZW

Langkah	Posisi	Karakter	Kamus	Keluaran
1.	1	A	[4] A B	[1]
2.	2	B	[5] B B	[2]
3.	3	B	[6] B A	[2]
4.	4	A	[7] A B A	[4]
5.	6	A	[8] A B A C	[7]
6.	9	C	- - -	[3]



Gambar 2.1 Hasil pemampatan LZW

Proses pemampatan balik pada LZW dilakukan dengan prinsip yang sama seperti proses pemampatan.

Algoritma pemampatan balik dijelaskan sub bab 2.4.2. Pada awalnya, kamus diinisialisasi dengan semua karakter dasar yang ada. Lalu pada setiap langkah, sandi dibaca satu per satu dari laju sandi, dikeluarkan string dari kamus yang berkorespondensi dengan sandi tersebut, dan ditambahkan string baru ke dalam kamus. Tahapan proses pemampatan balik ini ditunjukkan pada Tabel 2.2. Metode LZW yang diterapkan dalam penelitian ini berbentuk dinamis, dimana hanya dilakukan satu kali pembacaan terhadap berkas yang akan dimampatkan balik. Penyandian data dilakukan secara bersamaan dengan proses penambahan string baru ke dalam kamus.

2.4.2 Algoritma pemampatan balik LZW

Algoritma pemampatan balik LZW sebagai berikut :

1. Kamus diinisialisasi dengan semua karakter dasar yang ada : {'A'..'Z','a'..'z','0'..'9'}.
2. Jika *CW* menunjukkan sandi pertama dari laju sandi (menunjuk ke salah satu karakter dasar). Lihat di kamus dan keluaran string dari sandi tersebut (string *CW*) ke laju karakter. Apakah String *PW* menggantikan string *CW*, dimana *CW* adalah sandi berikutnya dari laju sandi.
3. Apakah string *CW* terdapat dalam kamus ? Jika ada, maka :
   a. Keluaran string *CW* ke laju karakter.
   b. *P* menggantikan string *PW*.
   c. *C* menunjukkan karakter pertama dari string *CW*.
   d. Tambahkan string *(P+C)* ke dalam kamus. Jika tidak, maka :
      i. *P* menggantikan string *PW*.
      ii. *C* menggantikan karakter pertama dari string *PW*.
      iii. Keluaran string *(P+C)* ke laju karakter dan tambahkan string tersebut ke dalam kamus (sekarang berkorespondensi dengan *C*W). Apakah terdapat sandi lagi di laju sandi ? Jika ya, maka kembali ke langkah 2. Jika tidak, maka terminasi proses berhenti.

Tabel 2.2 Tahapan proses pemampatan balik LZW

Langkah	Sandi	Keluaran	Kamus
1.	[1]	A	- - -
2.	[2]	B	[4] A B
3.	[2]	B	[5] B B
4.	[4]	A B	[6] B A
5.	[7]	A B A	[7] A B A
6.	[3]	C	[8] A B A C

2.5 Steganografi

Kata steganografi berasal dari bahasa Yunani, yaitu dari kata *Steganós* (tersembunyi) dan *Graptos* (tulisan). Steganografi di dunia modern biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam suatu arsip citra digital, audio, atau video. Satu hal penting yang menjadi kelebihan steganografi adalah kemampuannya untuk menipu persepsi manusia, manusia tidak memiliki insting untuk mencurigai adanya arsip-arsip yang memiliki informasi yang tersembunyi di dalamnya, terutama bila arsip tersebut tampak seperti arsip normal lainnya.

2.5.1 Kegunaan Steganografi

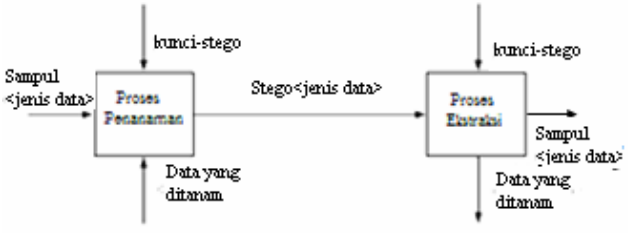
Seperti perangkat keamanan lainnya, steganografi dapat digunakan untuk berbagai macam alasan, beberapa diantaranya untuk alasan yang baik, namun dapat juga untuk alasan yang tidak baik. Untuk tujuan legitimasi dapat digunakan pengamanan seperti citra dengan tanda air dengan alasan untuk perlindungan hak cipta. Digital tanda air (yang juga dikenal dengan sidik jari, yang dikhususkan untuk hal-hal menyangkut hak cipta) sangat mirip dengan steganografi karena menggunakan metode penyembunyian dalam arsip, yang muncul sebagai bagian asli dari arsip tersebut dan tidak mudah di deteksi.

2.5.2 Metode Steganografi

Terdapat banyak metode yang digunakan dalam menyembunyian data ke dalam data lainnya. Berikut adalah penjelasan mengenai beberapa metode yang banyak digunakan dalam steganografi.

1. Metode Penanaman

Steganografi menyimpan pesan rahasia dalam suatu arsip yang biasanya diparameteri oleh suatu kunci-stego, dan pendeteksian atau pembacaan atas informasi tersembunyi tersebut dapat dilihat pada Gambar 2.2.



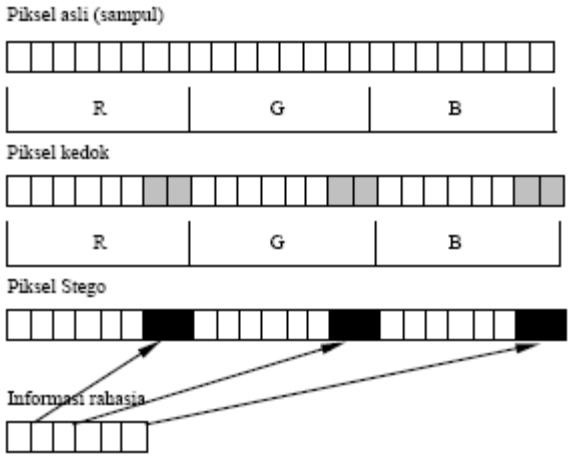
Gambar 2.2 Proses steganografi dengan metode penanaman

Metode penanaman ini juga biasa disebut dengan metode *injection* karena pesan rahasia disuntikkan langsung pada arsip lainnya dengan sedikit pengacakan atau enkripsi.[9]

2. Metode *Least Significant Bit*

Biasanya arsip 24-bit atau 8-bit digunakan untuk menyimpan citra digital. Penunjukkan warna dari piksel-piksel dapat diperoleh dari warna-warna primer yaitu merah, hijau, dan biru. Citra 24-bit menggunakan 3 bytes untuk masing-masing piksel, dimana setiap warna primer ditunjukkan dengan ukuran 1 byte. Penggunaan citra 24-bit memungkinkan setiap piksel ditunjukkan dengan nilai warna sebanyak 16.777.216 macam. Dua bit dari saluran warna ini dapat digunakan untuk menyembunyikan data, yang akan mengubah jenis warna untuk pikselnya menjadi 64-warna, namun hal ini akan mengakibatkan sedikit perbedaan yang dapat dideteksi secara kasat mata oleh manusia. Metode sederhana ini disebut dengan *Least Significant Bit* (LSB). [5] [10]

Dengan penggunaan metode ini, dimungkinkan adanya penambahan sejumlah besar informasi tanpa adanya degradasi tampilan dari citra itu sendiri. Gambar 2.3 menunjukkan proses kerja LSB.



Gambar 2.3 Proses steganografi dengan metode *Least Significant Bit* (LSB)

III. PERANCANGAN PROGRAM

Pada perancangan program ini dijelaskan spesifikasi perangkat-keras dan perangkat-lunak yang digunakan untuk membuat dan menjalankan program. Berikut penjelasan perangkat-keras dan perangkat-lunak yang digunakan.

3.1 Perangkat-keras

Perangkat-keras yang digunakan untuk membuat dan menjalankan program adalah satu set komputer jinjing (laptop). Berikut spesifikasi perangkat-keras yang digunakan.

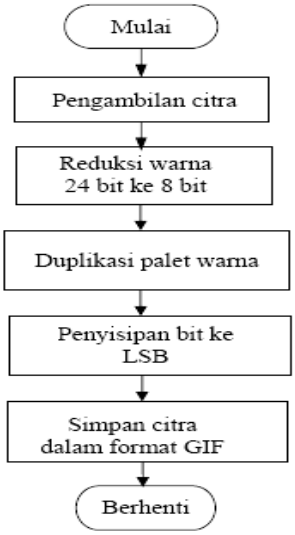
1. Merek : HP Compaq nx6120
2. Sistem Komputer : Intel Pentium Centrino M 1,87 GHz
3. Sistem Operasi : Microsoft Windows XP Profesional Service Pack 2
4. Media Tampilan : VGA (*true color* 32 bit,1024 × 800 piksel) dengan *chipset* intel Xtreme 128 Mb
5. Media masukan : *mouse*
6. Memori : 512 MB RAM.

3.1 Perangkat-lunak

Perangkat-lunak yang digunakan adalah Borland Delphi. Borland Delphi merupakan paket pemrograman yang bekerja dalam sistem operasi Windows, agar dalam pembuatan program dapat dikompilasi, persyaratan minimal pembuatan program menggunakan perangkat lunak Delphi dan sistem operasi Microsoft Windows XP Profesional harus dipenuhi. Hal ini untuk menghindari jika dalam perangkat lunak versi sebelumnya tidak terdapat beberapa fungsi yang ada pada versi sesudahnya.

3.2 Bagan-alir Program

Bagan-alir pembuatan program untuk melakukan Penyisipan data citra menggunakan teknik penyisipan bit LSB dimulai dari pengambilan citra sampai dengan mengambil data yang telah disisipkan ke dalam citra GIF. Pada Gambar 3.1 ditunjukkan bagan-alir yang digunakan dalam perancangan program Penyisipan data.



Gambar 3.1 Bagan-alir program.

3.3 Reduksi warna

Citra yang akan direduksi warnanya, pertama-tama diubah dulu ke dalam bit agar mudah untuk mengurangi ke dalaman bit citra tersebut. Program untuk merubah berkas citra ke dalam bit :

```
function datatobits(x:tdatarek):tbits;
var n,i,j:integer;
    b:^byte;
begin
     result:=tbits.create;
     b:=@x;
```

```
result.Size:=8*(sizeof(x)4)+8*(high(x.da
ta)+1);
```

Sebelum berkas diubah ke bit terlebih dahulu ditampung ke memori program. `setlength(d.data,filesize(f))` berfungsi menampung berkas yang akan disimpan. Untuk merubah ke bit diperlukan variabel `n, i, j` berupa integer. Sedangkan variabel `b` berfungsi untuk menyatakan ke dalam byte.

Format berkas header berisi sandi, nama berkas, jumlah d bit yang disisipkan. Untuk setiap 1 piksel diwakili 1 bit, jadi untuk 8 piksel berarti terdapat 1 byte. Dalam berkas citra yang akan disisipi data untuk mengetahui jumlah maksimum data yang akan disisipi dapat dihitung dengan persamaan :

$$\max = \left(\frac{w * h}{8} \right) \qquad (3.1)$$

Perintah program :

```
maxchar:=image1.Width*image1.Height  div
8;
```

3.4 Duplikasi palet

Pembentukan palet duplikat berfungsi untuk menciptakan palet 128 warna dasar yang sama dengan palet warna citra aslinya. Jadi warna ke 0 dan 128 adalah sama, 1 dan 129 sama dan demikian seterusnya sampai palet ke 255.

```
function mypalet:HPALETTE;
```

Penggunaan warna RGB pada program ini dipilih komposisi warna R = 3, G = 7, dan B = 3. Berarti ada 2^3 = 8 kombinasi warna merah, 2^7 = 128 kombinasi warna hijau, dan 2^3 = 8 kombinasi warna biru. Dengan demikian warna hijau yang paling dominan dan memiliki kombinasi yang paling banyak. Untuk membuat data palet dengan warna tertentu digunakan perintah sebagai berikut :

```
with (Pal.palPalEntry[i]) do begin
   peRed := r*255 div 3; // merah
   peGreen := g*255 div 7; // hijau
   peBlue  :=  b*255 div 3; // biru
   peFlags := PC_NOCOLLAPSE;
  end;
```

Setelah didapat warna yang sama dengan palet citra, kemudian dibuat data palet untuk warna yang sama dengan di atas namun untuk nomor warna +128

```
with (Pal.palPalEntry[i+128]) do begin
   peRed := r*255 div 3;
   peGreen := g*255 div 7;
   peBlue  :=  b*255 div 3;
   peFlags := PC_NOCOLLAPSE;
end;
```

3.5 Penyisipan data ke citra

Penyisipan data rahasia ke dalam citra pada tugas akhir ini menggunakan metode penyisipan LSB. Data yang akan disisipkan diubah terlebih dahulu ke bit agar mudah dalam melakukan penyisipan. Letak penyisipannya terletak pada posisi bit terendah.

Proses perubahan data ke bit :

```
data:=datatobits(dr);   for   i:=0   to
high(x.data) do begin
   for j:=0 to 7 do begin
   result.Bits[n]:=(x.data[i] and (1 shl
j))>0;
```

Proses penyisipan bit LSB :

```
for y:=0 to btmp.Height-1 do begin
    p:=btmp.ScanLine[y];
for x:=0 to btmp.Width-1 do begin
 if (n<data.size) and data.bits[n] then
p^:=p^+128;
    inc(n); // bits berikutnya
    inc(p); // picel berikutnya
pb1.Position:=n*100div(btmp.width*btmp.h
eight);
end;
```

3.6 Menyimpan hasil citra ke format GIF

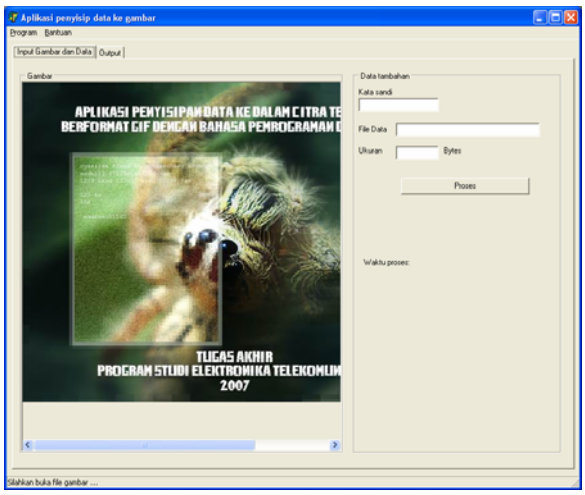
Setelah citra diproses, langkah terakhir adalah menyimpan hasil citra yang telah disisipi ke dalam format GIF. Hasil keluaran citra yang telah berformat GIF memiliki jumlah 2^8 warna = 256 warna (karena hanya terdapat 8 bit). Proses penyimpanan ke format GIF :

```
if SavePictureDialog1.execute then
mygif.SaveToFile(savepicturedialog1.file
name);
end;
```

IV. PENGUJIAN PROGRAM

Program yang telah dibuat dengan Delphi 2006 dapat dijalankan dengan langsung membuka berkas **penyusup data.exe** Pada Gambar 4.1 ditunjukkan tampilan jendela pembukaan. Setelah jendela pembuka muncul terdapat dua tombol yaitu tombol **Masuk** dan tombol **Keluar**. Tekan tombol **masuk** jika ingin melanjutkan program dan tekan tombol **keluar** jika tidak ingin meneruskan program. Tampilan jendela menu utama ditunjukkan pada Gambar 4.2.

Gambar 4.1 Tampilan jendela pembuka



Gambar 4.2 Tampilan menu utama

4.1 Uji Coba Dengan Data Sisipan di bawah Kapasitas Maksimal Citra

Pada uji coba ini diuji tingkat keberhasilan dan waktu yang dibutuhkan untuk melakukan steganografi yaitu setiap citra diuji dengan data uji berupa data teks, data audio dan video. Citra yang akan diujikan adalah **citra 1, citra2, citra 3, citra 4** seperti ditunjukkan pada Gambar 4.3

(a) Citra1

(b) Citra2

(c) Citra3

(d) Citra4

Gambar 4.3 Citra Uji.

4.1.1 Uji Coba Dengan Data Uji Teks Dokumen (*.doc, *.xls, *.ppt, *.pdf)

Citra2 berukuran 1371×886 piksel dengan ukuran berkas 76 KB diuji dengan 4 (empat) data uji berupa teks dokumen yaitu **Duji4. xls, Duji5.doc, Duji6.ppt, Duji7.pdf**. Hasil uji coba tersebut ditunjukkan pada Tabel 4.2.

Tabel 4.1 Hasil pengujian dengan data uji teks dokumen (***.doc, *.xls, *.ppt, *.pdf**)

Nama Citra uji	Pengujian	Nama Data Uji	Ukuran berkas	Waktu proses	Keterangan
Citra2.jpg	1	Duji4.xls	30 KB	1,73 detik	Berhasil
	2	Duji5.doc	40 KB	1,75 detik	Berhasil
	3	Duji6.ppt	101 KB	2,09 detik	Berhasil
	4	Duji7.pdf	120 KB	2 detik	Berhasil

Dari Tabel 4.1 terlihat bahwa **citra2.jpg** diuji dengan empat data uji dengan hasil semua data uji tersebut berhasil disisipkan ke dalam citra. Waktu yang dibutuhkan untuk mengolahnya juga cepat antara 1 sampai 2 detik. Untuk **Duji4** dan **Duji5** waktu yang dibutuhkan kurang dari 2 detik oleh ukuran berkas kurang dari 100 KB. Untuk **Duji6** dan **Duji7** waktu yang dibutuhkan lebih dari 2 detik. Walaupun citra uji memiliki ukuran 1371×886 piksel tetapi waktu yang dibutuhkan untuk menyisipkan cepat oleh data uji memiliki ukuran berkas tidak terlalu besar.

4.1.2 Uji Coba Dengan Data Uji Citra (*.jpg, *.GIF, *.PNG)

Citra 3 berukuran 1371×886 piksel dengan ukuran berkas 317 *KB* diuji dengan 3 (tiga) data uji berupa citra yaitu **Duji8.jpg, Duji9.GIF, Duji10.PNG.** Hasil uji coba tersebut seperti ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil pengujian dengan data citra (***.jpg, *.GIF, *.PNG**)

Nama Citra uji	Pengujian	Nama Data Uji	Ukuran berkas	Waktu proses	Keterangan
Citra3.jpg	1	Duji8.jpg	61 KB	3,28 detik	Berhasil
	2	Duji9.GIF	113 KB	3,56 detik	Berhasil
	3	Duji10.PNG	79 KB	3,21 detik	Berhasil

Dari Tabel 4.2 terlihat bahwa **citra3.jpg** diuji dengan tiga data uji berupa data citra dengan hasil semua data uji tersebut berhasil disisipkan ke dalam citra. Waktu yang dibutuhkan untuk mengolahnya antara 3 sampai 4 detik. Citra uji memiliki ukuran 1371×886 piksel termasuk ukuran citra di atas resolusi citra SVGA (800×600) piksel, maka waktu yang dibutuhkan cukup lama.

4.2 Uji Coba Dengan Data Sisipan Di atas Kapasitas Maksimal Citra

Pengujian dengan data uji melebihi kapasitas maksimal bertujuan menguji program dapat berlangsung atau tidak jika data uji yang akan disisipkan ke dalam citra melebihi kapasitas yang dapat ditampung citra uji. Citra uji yaitu **citra3.jpg** dengan data uji **Duji14.mp3, Duji15.jpg, Duji16.jpg.** Hasil uji coba ini ditunjukkan pada tabel 4.3.

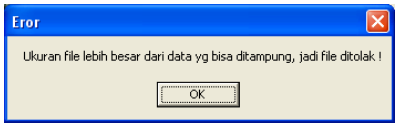
Tabel 4.3 Hasil pengujian dengan data uji di atas kapasitas daya tampung citra

Nama Citra uji	Pengujian	Nama Data Uji	Ukuran data	Keterangan
Citra3.jpg	1	Duji14.mp3	283 *KB*	Tidak berhasil
	2	Duji15.jpg	463 *KB*	Tidak berhasil
	3	Duji16.jpg	173 *KB*	Tidak berhasil

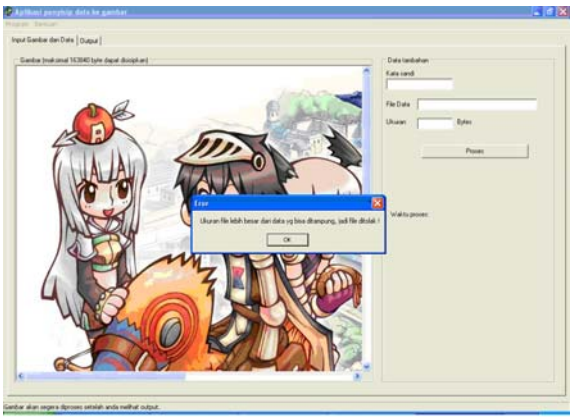
Dari tabel 4.3 terlihat bahwa program tidak berhasil menyisipkan data ke dalam citra. Program

akan menampilkan jendela informasi kesalahan seperti pada Gambar 4.6. Hal ini disebabkan oleh daya tampung berkas data yang disisipkan untuk **citra3.jpg** hanya sebesar 163 *KB* saja. Jadi program tidak bisa melanjutkan proses penyisipan data ke dalam citra.

Tanggapan program untuk pengujian kapasitas daya tampung citra, maka akan muncul jendela kesalahan seperti pada Gambar 4.4. Pada Gambar 4.5 merupakan tampilan utama program jika terdapat kesalahan.



Gambar 4.4 Tampilan kesalahan data melebihi kapasitas



Gambar 4.5 Tampilan tanggapan program utama data melebihi kapasitas

V. PENUTUP

5.1 Kesimpulan

Penelitian tentang aplikasi penyisipan data ke dalam citra termampat berformat GIF dengan bahasa pemrograman Delphi 2006 menghasilkan kesimpulan berikut ini.

1. Program dapat menyisipkan semua data, namun dengan syarat ukuran berkasnya tidak melebihi kapasitas yang mampu ditampung oleh citra uji.
2. Semakin besar ukuran piksel citra uji, maka dibutuhkan waktu yang lebih lama juga untuk menyisipkan data.
3. Citra yang mampu diproses program maksimal berukuran 6800×5000 piksel.

5.2 Saran

Berikut adalah saran-saran yang berkaitan dengan penelitian yang telah dilakukan.

1. Perlu dilakukan penelitian lanjutan untuk cira termampat tak berugi lainnya seperti **PNG** sehingga dapat diketahui perbedaan dalam keandalan penyisipan data.
2. Perlu dilakukan penelitian lanjutan untuk steganografi pada audio maupun video.

DAFTAR PUSTAKA

[1] Achmad, B. dan K. Firdausy, *Teknik Pengolahan Citra Digital*, Andi Publishing, Yogyakarta, 2005.

[2] Ziv. J. and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, September, 1978.

[3] Munir, R, *Pengolahan Citra Digital dengan Pendekatan algoritmik*, Informatika, Bandung, 2004.

[4] Robi'in, B, *Pemrograman Grafis Multimedia Menggunakan Delphi*, Andi Offset,Yogyakarta , 2004.

[5] Bender, W., D. Gruhl, and N. Morimoto, *Techniques for Data Hiding*, Massachusetts Institute of Technology, Media Laboratory Cambridge, Massachusetts USA, San Jose CA, February, 1995.

[6] Octaviano, S, *Aplikasi Steganografi Menggunakan Metode Pensinyalan LSB (Least Significant Bit) pada File Gambar dengan Format BMP*, Skripsi S-1, Universitas Diponegoro, Semarang, 2004

.[7] Linawati dan H. Panggabean, *Perbandingan kinerja algoritma kompresi huffman, LZW, dan DMC pada berbagai tipe file*, Integral, vol 9 no 1, Maret, 2004.

[8] Guillermito, *Easily Breaking a very weak Steganography Software*, http://www.guillermito2.net/stegano/camouflage.html, 16 September 2002.

[9] Armyta, D, *Studi Mengenai Aplikasi Steganografi Camouflage Beserta Pemecahan Algoritmanya*, http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/makalah1-2006/makalah1-007.pdf, 2006

[10] Tien, L., *Destegging Tutorial*, http://www.unfiction.com/dev/tutorial, 2002.

[11] Mangarae, A., *Steganography FAQ*, www.infosecwriters.com/text_resources/pdf/Steganography_AMangarae.pdf, 18 Maret 2006.

[12] Poli, L., *Penerapan Steganografi dengan Citra Digital Sebagai File Penampang*, Tugas Akhir Jurusan Teknik Informatika, 1998.

[13] Zhao, J. and E. Koch, *Embedding Robust Labels Into Images For Copyright Protection*, Germany,Vienna, Agustus 1995.

[14] Wikipedia,*Steganography*, http://en.wikipedia.org/wiki/Steganography, 3 Mei 2007.

Arif Nur Budi Setiyo[L2F304217]
Lahir di Semarang, 02 November 1981.
Mahasiswa Teknik Elektro 2004, Konsentrasi Elektronika dan Telekomunikasi, Universitas Diponegoro.
E-mail : Ayip_Jr02@yahoo.com.

Menyetujui dan Mengesahkan

Pembimbing I

Achmad Hidayatno, S.T., M.T.
NIP. 132 137 933
Tanggal……

Pembimbing II

R. Rizal Isnanto, S.T., M.M., M.T.
NIP. 132 288 515
Tanggal…….