Steganografi Pada Citra dengan Format GIF Menggunakan Algoritma GifShuffle

Ronald Augustinus Penalosa – NIM : 13503117

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13117@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang algoritma steganografi pada citra dengan format GIF. Graphics Interchange Format(GIF) adalah sebuah format yang sering digunakan dalam dunia web maupun dalam dunia citra digital. Format ini sering digunakan karena ukurannya yang relati kecil dan juga banyaknya software editor gambar yang telah mendukung citra ini. GIF berukuran kecil karena membatasi jumlah warnanya sebanyak 256 warna sehingga dapat menghemat ukuran berkas.

Algoritma GifShuffle yang dikembangkan oleh Matthew Kwan adalah salah satu algoritma steganografi yang menggunakan berkas citra dengan format GIF. Akan dibahas bagaimana proses *encoding* dan *decoding* pesan dalam citra dengan menggunakan algoritma GifShuffle. Algoritma ini melakukan penyisipan pesan dengan cara mengganti susunan palet warna yang ada dalam sebuah berkas citra dengan format GIF.

Selain itu akan dilakukan pula pengujian terhadap citra yang telah disisipi oleh pesan. Pengujian ini dilakukan dengan menggunakan software paranoia Hal ini dilakukan untuk memperoleh kesimpulan bagaimana ketahanan pesan yang telah disisipkan ke dalam berkas GIF tersebut. Pengujian terhadap berbagai ukuran berkas citra dan jumlah warna yang dikandung dalam berkas tersebut pun dilakukan untuk memperoleh bagaimana kriteria berkas citra dengan format GIF.

***Kata kunci**: Graphics Interchange Format(GIF),Encoding,Decoding*Steganografi,Paranoia

1. Pendahuluan

Dewasa ini peyembunyian pesan tidak hanya dapat dilakukan dengan menyamarkan pesan tersebut. Melainkan dapat pula menyisipkan pesan tersebut dalam media lain.Sehingga orang tidak akan curiga terhadap pesan yang kita kirimkan, karena pesan tersebut tidak terlihat, yang terlihat hanyalah media penampung pesan kita tersebut.

Sebagai contoh, kita ingin mengirimkan pesan kepada seseorang yang jauh melalui email. Karena takut pesan kita diketahui orang lain maka kita menyisipkan pesan tersebut dalam sebuah media lain yang berukuran lebih besar. Misalnya dalam media berkas citra. Sehingga orang lain tidak akan curiga akan gambar yang kita kirimkan. Hal ini tentunya akan lebih praktif ketimbang kita mengirimkan pesan dalam bentuk berkas yang terenkripsi. Hal ini tentunya akan membuat orang lain curiga dan mulai melakukan *attack* untuk mengetahui isi pesan yang kita kirimkan.

Teknik penyisipan pesan dalam media lain yang lebih besar ini dinamakan Steganography. Media penyimpan yang dapat digunakan dalam steganography dapat berupa berkas lagu, citra, atau berkas-berkas lain yang berukuran besar dan dapat dimasukkan pesan yang ingin kita sembunyikan.

Dengan menggunakan steganografi maka orang tidak akan menjadi curiga kalau ternya kita mengirimkan pesan rahasia. Tetapi di ada harga

yang harus dibayar dalam steganografi yaitu besarnya ukuran file yang disisipi pesan rahasia. Kita perlu mentransfer data dalam jumlah besar padahal data penting yang diperlukan hanya berukuran kecil.

2. Steganography

Seperti telah disebutkan sebelumnya steganography adalah sebuah teknik penyisipan pesan dalam media yang lebih besar. Dalam bahasa Yunani kata ”Steganography” diterjemahkan sebagai ”Steganos” yang berarti tulisan tersembunyi. Sehingga dapat diartikan dalam terjemahan bebas bahwa Steganography adalah ilmu dan seni menyisipkan informasi dengan cara menyisipkan pesan dalam pesan lain.

Sejarah mencatat bahwa steganography pertama kali digunakan pada tahun 440 sebelum masehi. Ketika itu Demeratus mengirimkan berita tentang penyerangan yang akan dilakukan ke Yunani. Beliau mengirimkan pesan tersebut dalam sebuah kayu dan menutupinya dengan lilin. Sehingga pesan yang dikirimkannya tidak terlihat mencurigakan dan kerahasian pesan tetap terjaga.

Tentunya metode Steganography tersebut sudah sangat usang jika digunakan dalam dunia modern seperti sekarang ini. Teknik Steganography yang digunakan dalam dunia moderrn sekarang ini sudah amat beragam. Beragam mulai dari algoritma yang digunakannya sampai pada media yang digunakannya.

Beberapa contoh media penyisipan pesan rahasia yang digunakan dalam teknik Steganography antara lain adalah :

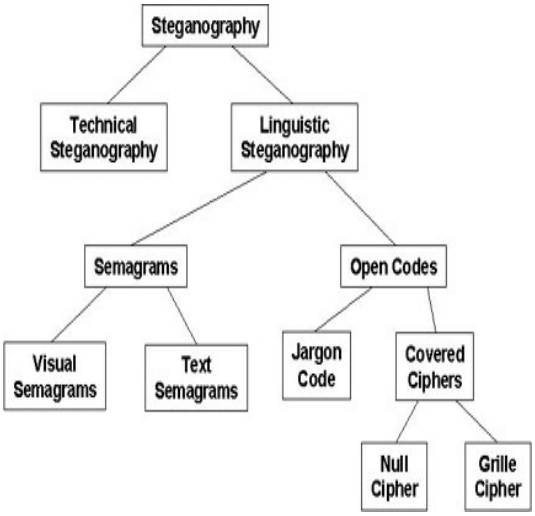
1. Teks
   Dalam algoritma Steganography yang menggunakan teks sebagai media penyisipannya biasanya digunakan teknik NLP sehingga teks yang telah disisipi pesan rahasia tidak akan mencurigakan untuk orang yang melihatnya.

2. Audio
   Format ini pun sering dipilih karena biasanya berkas dengan format ini berukuran relatif besar. Sehingga dapat menampung pesan rahasia dalam jumlah yang besar pula.

3. Citra
   Format pun paling sering digunakan, karena format ini merupakan salah satu format file yang sering dipertukarkan dalam dunia internet. Alasan lainnya adalah banyaknya tersedia algoritma Steganography untuk media penampung yang berupa citra.

4. Video
   Format ini memang merupakan format dengan ukuran file yang relatif sangat besar namun jarang digunakan karena ukurannya yang terlalu besar sehingga mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

Selain berdasarkan format berkas penampung yang digunakan steganogaphy pun menurut Bauer dapat dikelompokkan menjadi sebuah taxonomy sebagai berikut :



1. Technical Steganography
Teknik ini menggunakan metode sains untuk menyembunyikan pesan. Contohnya adalah penyembunyian pesan dalam chip mikro.

2. Linguistic Steganography
Teknik ini menyembunyikan pesan dalam cara yang tidak lazim. Teknik ini terbagi menjadi 2 bagian yaitu Semagrams dan Open Codes.

3. Semagrams
Teknik ini menyembunyikan informasi dengan mengunakan simbol atau tanda.

Contoh penggunaan adalah dengan mengganti ukuran teks atau mengganti ukuran font. Pergantian ukuran atau tipe tersebutlah yang digunakan sebagai media penyisipan pesan. Sebagai informasi algoritma gifshuffle yang dibahas dalam makalah ini termasuk ke dalam tipe ini.

4. Open Codes
Teknik ini menyembunyikan pesan cara yang tidak umum namun tetap tidak mencurigakan. Teknik ini terbagi menjadi 2 bagian yaitu Jargon Code dan Covered Ciphers

5. Jargon Code
Teknik ini sesuai dengan namanya mengunakan bahasa yang hanya dimengerti oleh sebagian orang. Sebagai contoh adalah warchalking, underground terminology atau percakapan biasa yang mengandung fakta yang hanya diketahui oleh pembicara.

6. Covered Ciphers
Teknik ini menyembunyikan pesan dalam media pembawa sehingga pesan kemudian dapat diekstrak dari media pembawa tersebut oleh pihak yang mengetahui bagaimana pesan tersembunyi tersebut disembunyikan.

Seperti telah dibahas sebelumnya bahwa teknik steganography yang menggunakan penampung berformat citra adalah teknik terbanyak yang diterapkan selain format audio.

Salah satu algoritma yang sering dipakai untuk menyembunyikan pesan dalam citra adalah dengan mengganti bit paling tidak berarti dari data media penampung. Algoritma ini sering disebut dengan metode LSB.

Namun algoritma ini tidak mangkus karena media citra yang dijadikan penampung tersebut rentan terhadap perubahan. Sebga

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

1. Imperceptible
Keberadaan pesan dalam media penampung tidak dapat dideteksi.

2. Fidelity
Mutu media penampung setelah ditambahkan pesan rahasia tidak jauh berbeda dengan mutu media penampung sebelum ditambahkan pesan

3. Recovery
Pesan rahasia yang telah disisipkan dalam media penampung harus dapat diungkap kembali. Hal ini merupakan syarat mutlak dalam sebuah algoritma steganography, kerena ada banyak cara penyisipan pesan yang tidak terdeteksi namun sulit dalam pembacaan kembali.

Ketiga syarat diatas lah yang akan digunakan untuk pengujian algoritma gifshuffle yang dibahas dalam makalah ini.

3. Citra berformat GIF

Graphics Interchange Format atau yang sering disingkat GIF adalah sebuah format berkas citra yang diperkenalkan pada tahun 1987 oleh CompuServe untuk menggantikan format RLE yang hanya mampu menampilkan gambar dengan warna hitam dan putih saja.

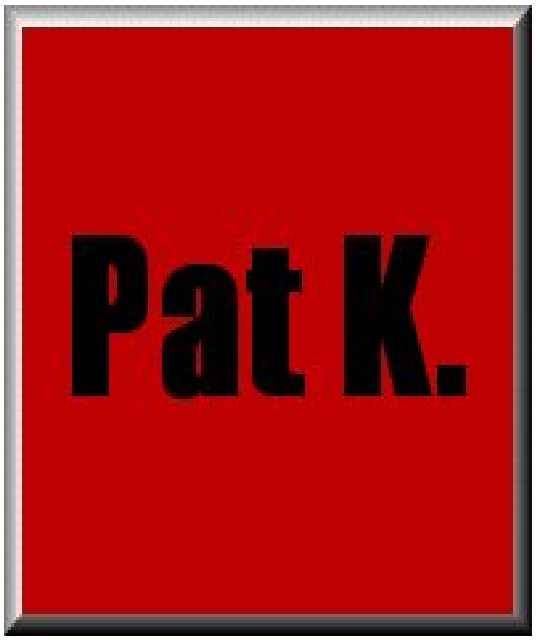
GIF adalah salah satu format berkas citra yang paling sering ditemui di dunia digital.Hal ini terjadi karena format ini berukurang relatif kecil. Sebagai contoh untuk citra yang sama, berkas dengan format GIF dapat berukuran lebih kecil jika dibandingkan dengan format JPG .

Hal ini disebabkan karena file GIF hanya menggunakan 256 palet warna. Sehingga tentunya ukuran file akan lebih kecil. Namun 256 palet warna tersebut tidak mutlak hanya 256 warna tertentu. Namun warna tersebut dapat dipilih dari 24-bit palet warna RGB. Sehingga dengan singkat kata dapat disimpulkan bahwa berkas dengan format GIF akan membuang palet warna yang tidak diperlukan dan mengambil hanya 256 palet warna yang diperlukan.

Ukuran palet sebesar 256 warna adalah standar GIF’89 dan 87. Beberapa versi dari gif sekarang telah dapat menampilkan warna dengan lebih dari 256 warna. GIF dengan format GIF’89 dan GIF’87 dapat dibedakan melalui header file.

Menurut sebuah sumber sekarang ini 3 format file yang digemari adalah JPG,GIF dan juga PNG. Format PNG memang sekarang ini

merupakan format yang dinilai lebih favorit ketimbang format lainya namun format GIF tetap merupakan salah satu format yang umum digunakan. Karena salah satu kelebihan format ini adalah adanya dukungan untuk penampilan gambar bergerak.



Gambar 1 Citra dengan format JPG (6KB)

Gambar 2 Citra dengan format GIF(5 KB)

Dari perbandingan dua buah gambar diatas terlihat jelas bahwa citra dengan format GIF memiliki ukuran berkas yang lebih kecil 1 KB untuk citra yang sama. Bahkan jika dilihat secara lebih teliti lagi maka dapat dilihat bahwa citra pada gambar 2 memiliki hasil yang lebih baik dari pada citra pada gambar pertama.

Hal ini terjadi karena GIF menggunakan tipe kompresi data yang berbeda dengan JPG. GIF menggunakan tipe kompresi yang sering disebut “lossless”. Hal ini berarti bahwa citra tidak mengalami kehilangan kualitas ketika dikompresi. Sedangkan pada JPG tipe kompresi yang digunakan adalah “lossy” sehingga citra mengalami pengurangan kualitas ketika dikompresi.
Memang berkas JPG dapat berukurang lebih kecil lagi ketimbang file GIF namun kualitas citra yang dihasilkan akan sangat menurun seperti pada gambar dibawah ini .

Gambar 3 Citra dengan format GIF (4 KB)

Tentunya setelah melihat beberapa perbandingan diatas orang menjadi memiliki kecenderungan untuk menggunakan format GIF sebagai format yang dipilih untuk format berkas citranya.

Namun format GIF bukannya format yang tidak memiliki kekurangan. Akibat dari jumlah warna yang hanya 256 warna saja maka format ini jarang sekali digunakan untuk citra fotografi. Karena seringkali citra fotografi menggunakan warna yang lebih dari 256 warna. Sehingga jika citra fotografi direpresentasikan dalam format GIF akan mengalami penurunan kualitas yang banyak. Hal tersebut dapat dilihat dalam perbandingan dibawah ini :

Gambar 4 Citra dengan format GIF

Gambar 5 Citra dengan format JPG

Terlihat jelas dalam 2 gambar diatas bahwa citra dengan format GIF memiliki hasil yang buruk jika dibandingkan dengan format JPG pada gambar 2.

4. Algoritma GifShuffle

GifShuffe adalah sebuah algoritma steganography yang digunakan untuk menyembunyikan pesan dalam berkas citra dengan format GIF. Algoritma ini ditemukan oleh Matthew Kwan. Beliau adalah seorang sarjana ilmu komputer lulusan dari University of Melbourne. Beliau pun sekarang menjadi pemilik dan pendiri dari Unicypt Pty Ltd, sebuah perusahaan yang bergerak di bidang enkripsi email.

Dalam situsnya, beliau memberikan sebuah penjelasan singkat tentang algoritma ini dan juga memberikan program yang menerapkan algoritma ini, lengkap dengan source codenya. Sehingga memudahkan dalam penjelasan algoritma gifshuffle.

Algoritma gifshufle pada intinya memanfaatkan header file GIF yang menyimpan palet warna sebagai media penyisipan pesan. Dalam algoritma ini tidak terjadi perubahan apapaun dalam data berkas dengan format GIF. Sehingga menambah aspek robustness dari algoritman ini.

Sesuai dengan namanya GifShuffle akan melakukan "Shuffle" terhadap palet warna dari sebuah berkas gif. "Shuffle" jika diterjemahkan ke dalam bahasa Indonesia berari memutar. 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.

Dengan dilakukannya penukaran posisi maka akan dapat diperoleh sebuah informasi berkaitan dengan perbedaap posisi dengan posisi awal. Sebagai contoh jika kita mempunyai 52 kartu remi maka kita akan dapat mengurutkan kartu-kartu tersebut dalam 52! cara. Dengan kata lain jika kita diberikan n buah kartu maka kita dapat menyimpan $\log_2(n!)$ bit informasi berdasarkan pengurutannya. Sebagai contoh adalah jika kita mengurutkan 4 kartu menurut besarnya sebagai berikut :

1♣	2♣	3♣	4♣

Kemudian karena kita ingin menyimpan informasi maka kita dapat mengganti pengurutannya sebagai berikut :

4♣	3♣	2♣	1♣

Pergantian urutan dari sebuah kartu dapat diasosiasikan dengan informasi. Sebagai contoh dalam ilustrasi diatas. Jika kartu *1*♣ bergeser 3 kali ke kiri maka dapat diartikan bahwa informasi yang terkandung dalam pergeseran tersebut adalah huruf A. Atau jika kartu *2*♣ bergeser sejauh 1 kartu dapat diartikan huruf C. Hal tersebut tergantung kepada kesepakatan di awal.

Ilustrasi diatas adalah sedikit ilustrasi untuk menggambarkan bagaimana cara algoritma GifShuffle menyisipkan pesan dalam sebuah berkas GIF yaitu dengan mengganti susunan palet warna dan mengasosiasikan sebuah informasi berkaitan dengan pergeseran tersebut.

4.1 Source Encoding dalam C

```
static BOOL
    GIFINFO     *gi,
    EPI     *epi
) {
    int     i, ncols = 0;
    CMAP_INFO   ci_array[256];

    ncols = unique_colours (gi->gi_colours, gi->gi_num_colours, ci_array)

    if (encrypting_colourmap ()) {
        for (i=0; i<ncols; i++) {
        CMAP_INFO   *ci = &ci_array[i];

        encrypt_colour (ci->rgb.r, ci->rgb.g, ci->rgb.b, ci->ct&xt);
        }
        qsort (ci_array, ncols, sizeof (CMAP_INFO), cmap_encrypt_cmp);
    } else
        qsort (ci_array, ncols, sizeof (CMAP_INFO), cmap_cmp);

    for (i=0; i<ncols; i++)
        ci_array[ncols - i - 1].pos = epi_divide (epi, i + 1);

    if (epi->epi_high_bit > 0) {
        fprintf (stderr, "Error: remainder of %d bits.\n",
                         epi->epi_high_bit);
        return (FALSE);
    }

    for (i=0; i<ncols; i++) {
        CMAP_INFO   *ci = &ci_array[ncols - i - 1];
        int     j, pos = ci->pos;

        for (j=i; j>pos; j--)
        gi->gi_colours[j] = gi->gi_colours[j - 1];

        gi->gi_colours[pos] = ci->rgb;
    }

        /* Pad out the rest of the colourmap. */
    for (; i<gi->gi_num_colours; i++)
        gi->gi_colours[i] = gi->gi_colours[gi->gi_num_colours - 1];

    return (TRUE);
}
```

4.2 Source Decoding dalam C

```
static void
colourmap_decode (
    const GIFINFO   *gi,
    EPI     *epi
) {
    int     i, ncols = 0;
    CMAP_INFO   ci_array[256];

    ncols = unique_colours (gi->gi_colours, gi->gi_num_colours, ci_array);
    for (i=0; i<ncols; i++)
        ci_array[i].pos = i;

    if (encrypting_colourmap ()) {
        for (i=0; i<ncols; i++) {
        CMAP_INFO   *ci = &ci_array[i];

        encrypt_colour (ci->rgb.r, ci->rgb.g, ci->rgb.b, ci->ct&xt);
        }
        qsort (ci_array, ncols, sizeof (CMAP_INFO), cmap_encrypt_cmp);
    } else
        qsort (ci_array, ncols, sizeof (CMAP_INFO), cmap_cmp);

    epi_init (epi);

    for (i = 0; i < ncols - 1; i++) {
        int     j, pos = ci_array[i].pos;
        EPI     epi_pos;

        epi_multiply (epi, ncols - i);
        epi_set (&epi_pos, pos);
        epi_add (epi, &epi_pos);

        for (j = i + 1; j < ncols; j++)
        if (ci_array[j].pos > pos)
            ci_array[j].pos--;
    }
}
```

4.1 Skema encoding GifShuffle

Sebagai mana telah dibahas sebelumnya bahwa algoritma ini akan mengganti susunan palet warna dari sebuah berkas GIF. Karena berkas dengan format GIF mengandung 256 palet warna maka dapat disimpulkan bahwa total penyimpanan maksimum dari format ini adalah 1675 bit.

Langkah-langkah algoritma GifShuffle adalah sebagai berikut :

1. Dimulai dengan dengan pesan yang akan disisipkan. Pesan tersebut akan diubah kedalam sebuah bentuk biner dengan representasi 1/0.

2. Anggap kumpulan representasi biner yang tadi diperoleh sebagai sebuah angka. Biasanya langkah ini akan menghasilkan sebuah bilangan yang sangat besar karena konversi dari biner yang besar . Namakan bilangan yang diperoleh ini sebagai M.
3. Hitung jumlah warna yang terkandung dalam berkas GIF yang ingin disisipkan. Namakan jumlah yang diperoleh ini sebagai N. Apabila M > N!-1 maka pesan yang ingin disisipkan berukuran terlalu besar sehingga proses penyisipan tidak dapat dilakukan.
4. Urutkan warna dalam palet warna sesuai dengan urutan yang "natural". Setiap warna dengan format RGB dikonversikan ke bilangan integer dengan aturan (merah* 65536 + hijau * 256 + biru). Kemudian diurutkan berdasarkan besar bilangan interger yang mewakili warna tersebut.
5. Lakukan iterasi terhadap variabel i dengan nilai I dari 1 sampai N. Setiap warna dengan urutan N-i dipindahkan keposisi baru yaitu M mod i, kemudian M dibagi dengan i.
6. Kemudian palet warna yang baru hasil iterasi pada langkah 5 dimasukkan ke dalam palet warna berkas GIF. Apabila ada sebuah tempat yang diisi oleh 2 buah warna maka warna yang sebelumnya menempati tempat tersebut akan digeser satu tempat ke samping.
7. Apabila ternyata besar dari palet warna yang baru lebih kecil dari 256 maka palet warna akan diisi dengan warna terakhir dari palet warna sebelumnya.
8. Kemudian berkas GIF ini akan dikompresi ulang dengan palet warna yang baru untuk menghasilkan berkas yang baru dengan ukuran dan gambar yang sama namun telah disisipi pesan.

Contoh singkat dari langkah-langkah diatas adalah apabila kita memiliki sebuah pesan yang ingin disisipkan. Anggap bilangan dari hasil konversi pesan tersebut adalah 22. Sehingga M adalah 22 dan media penyisipan pesan berukuran 4. Maka susunan awal media penyisipan adalah sebagai berikut :

A	B	C	D

Kemudian dilakukan iterasi sebagai mana yang ada dalam langkah ke 5 untuk mengetahui perubahan posisi.

1. D
   22 mod 1 = 0
   M = 22 div 1
   M = 22
   Sehingga D menempati posisi ke-0 pada susunan yang baru.

2. C
   22 mod 2 = 0
   M = 22 div 2
   M = 11
   Sehingga C menempati posisi ke-0 pada susunan yang baru.

3. B
   11 mod 3 = 2
   M = 11 div 3
   M = 3
   Sehingga B menempati posisi ke-2 pada susunan yang baru.

4. A
   3 mod 4 = 3
   M = 3 div 4
   M = 0
   Sehingga A menempati posisi ke-3 pada susunan yang baru

Setelah diperoleh susunan yang baru maka diterapkanlah prosedur yang sama seperti pada langkah ke 6. hal ini perlu diterapkan karena huruf C dan D menempati posis yang sama yaitu posisi ke-0. Sehingga sekarang posisi hiruf D digeser 1 blok ke kanan untuk menempati posisi ke-1. Hasil akhir dari penggantian susunan huruf-huruf ini adalah :

C	D	B	A

4.2 Skema decoding GifShuffle

Sama seperti kebanyakan algoritma dalam bidang kriptografi proses decoding hanyalah berupa pembalikan dari proses encoding.

Contoh pembalikan dari susunan yang telah dihasilkan pada skema encoding adalah :

1. A
   M = 1 + 1*0
   M = 1

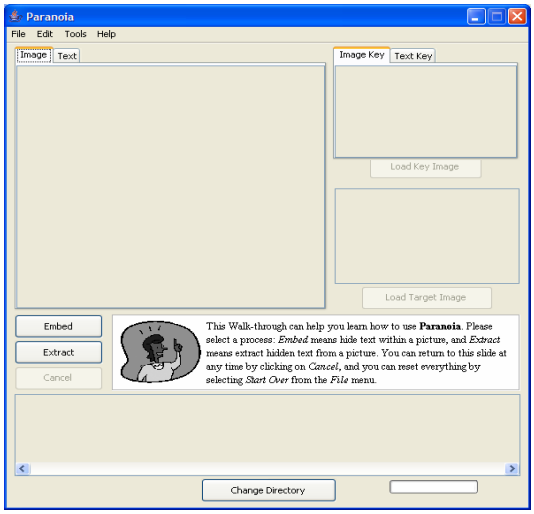
2. B
   M = M + 1+ 2*1
   M = 4

3. C
   M = M + 2 +3*2
   M = 12

4. D
   M = M + 2+4*2
   M = 22

Dari proses pembalikan tersebut maka diperoleh bahwa hasil pembalikan adalah M = 22, sama seperti pesan yang disisipkan. Bilangan integer M ini pun hanya tinggal dikonversikan kembali untuk memperoleh pesan.

5. Software Paranoia



Gambar 6 Software Paranoia

Paranoia adalah sebuah software opensource yang dikembangkan oleh Ubiquitous Software Engineers pada musim gugur 2003 di Stanford University. Software ini selain memberikan source namun juga turut melampirkan dokumentasi-nya sehingga sangat memudahkan dalam pengujian algoritma GifShuffle yang dipakai dalam software ini.

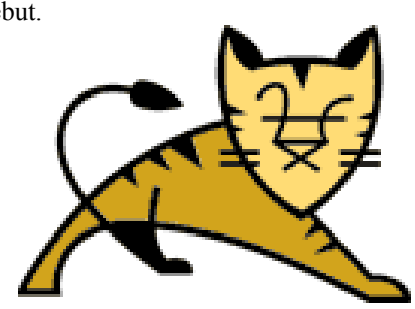
Sebagai informasi, software ini dibangun dengan menggunakan bahasa Java. Software ini merupakan sebuah project dalam sebuah mata kuliah yang dilaksanakan di stanford university.

Paranoia mempunyai fitur untuk menyembunyikan sebuah pesan dalam beberapa jenis format citra. Jenis format yang didukung adalah format GIF dan JPEG. Dalam software ini algoritma GifShuffle digunakan untuk melakukan steganografi dalam berkas citra berformat GIF.

6. Pengujian Algoritma GifShuffle

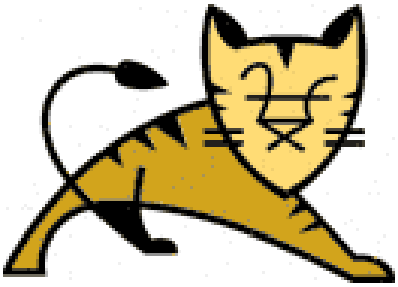
Sebagaimana telah disebutkan sebelumnya bahwa sebuah algoritma stegenografi harus memenuhi 3 faktor. 2 faktor telah dipenuhi yaitu kita dapat membaca kembali pesan yang ditulis oleh algoritma ini dan orang pun tidak akan curiga akan gambar yang disisipi pesan.

Dalam upabab ini akan dibahas tentang pengujian mutu berkas citra yang telah disisipi pesan. Apakah mengalami penurunan kualitas atau tidak, dan apakah informasi masih dapat diperoleh dari citra yang telah diubah-ubah tersebut.



Gambar 7 Gambar yang digunakan

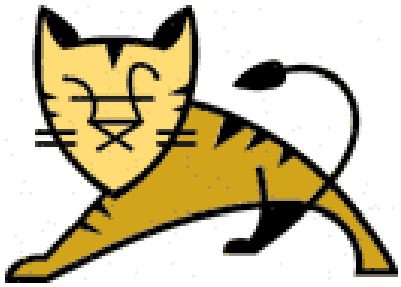
Setelah dilakukan penyisipan pesan maka citra dengan format GIF tidak terlalu banyak megalami perubahan hal ini terlihat dari gambar 8 yang telah mengalami penyisipan pesan. Tidak terlilhat perbedaan sama sekali.



Gambar 8 Gambar yang telah disisipi pesan

Pengujian kemudian dilanjutkan untuk melihat apakah pesan yang disisipkan masih dapat diekstrak meskipun gambar mengalami beberapa perubahan. Beberapa pengujian yang dilakukan adalah :

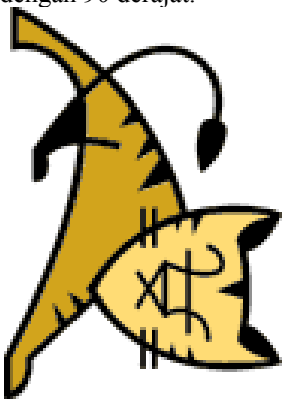
1. Horizontal Flip
   Horizontal Flip adalah teknik untuk membalik sebuah gambar secara horizontal.



Gambar 9 Gambar di-Horizontal Flip

Setelah dilakukan perubahan secara horizontal flip ternyata pesan tidak dapat diekstrasi lagi.

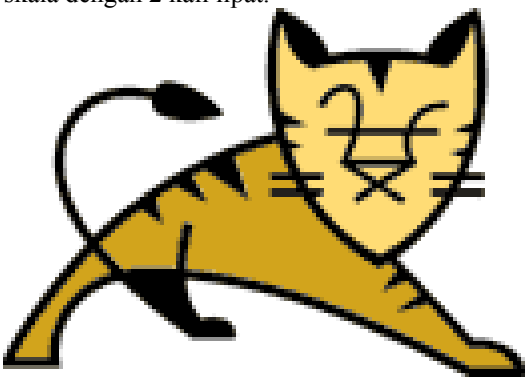
2. Rotation
   Rotasi adalah teknik pembalikan gambara dengan derajat tertentu . Gambar di bawah ini mengunakan rotasi dengan 90 derajat.



Gambar 10 Gambar Di- Rotasi

Setelah dilakukan perubahan secara Rotasi ternyata pesan tidak dapat diekstrasi lagi.

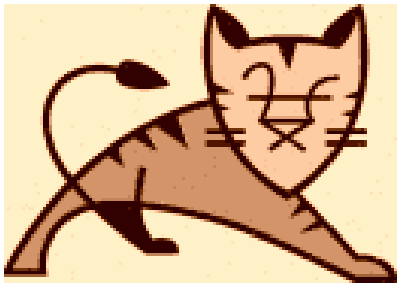
3. Scaling
   Scaling adalah perubahan ukuran gambar dengan skala tertentu. Dalam contoh dibawah ini adalah penggunaaan skala dengan 2 kali lipat.



Gambar 11 Gambar di Scale 200%

Setelah dilakukan perubahan secara Scaling ternyata pesan tidak dapat diekstrasi lagi.

4. Penambahan efek Sephia
   Sephia adalah perubahan warna agara mendekati cokelat. Contohnya adalah gambar dibawah ini



Gambar 12 Gambar diberi efek Sephia

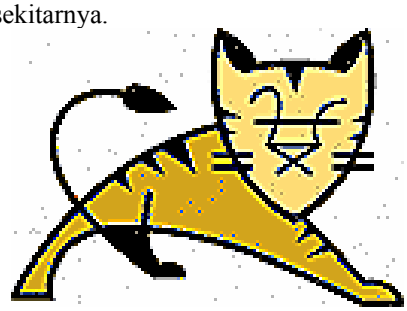
Setelah dilakukan penambahan efek Sephia ternyata pesan tidak dapat diekstrasi lagi.

5. Penambahan efek Blur
   Blur adalah perubahan warna tiap-tiap pixel dalam gambar agar membentuk distribusi normal. Hasil dari efek ini adalah gambar menjadi terlihat kabur.

Gambar 13 Gambar diberi efek Blur

Setelah dilakukan penambahan efek Sephia ternyata pesan tidak dapat diekstrasi lagi.

6. Penambahan efek Sharpen
   Sharpen adalah sebuah efek yang menonjolkan sebuah elemen pixel sehingga terlihat lebih menonjol dari sekitarnya.



Gambar 14 Gambar diberi efek Sharpen

Setelah dilakukan penambahan efek Sephia ternyata pesan tidak dapat diekstrasi lagi.

Dari beberapa percobaan diatas diperoleh informasi bahwa ternya pesan yang disisipkan gagal untuk diekstraksi apabila media gif yang digunakan dikenai perubahan. Hal ini terjadi karena palet wana pun ikut dirubah ketika sebuah berkas citra GIF disunting.

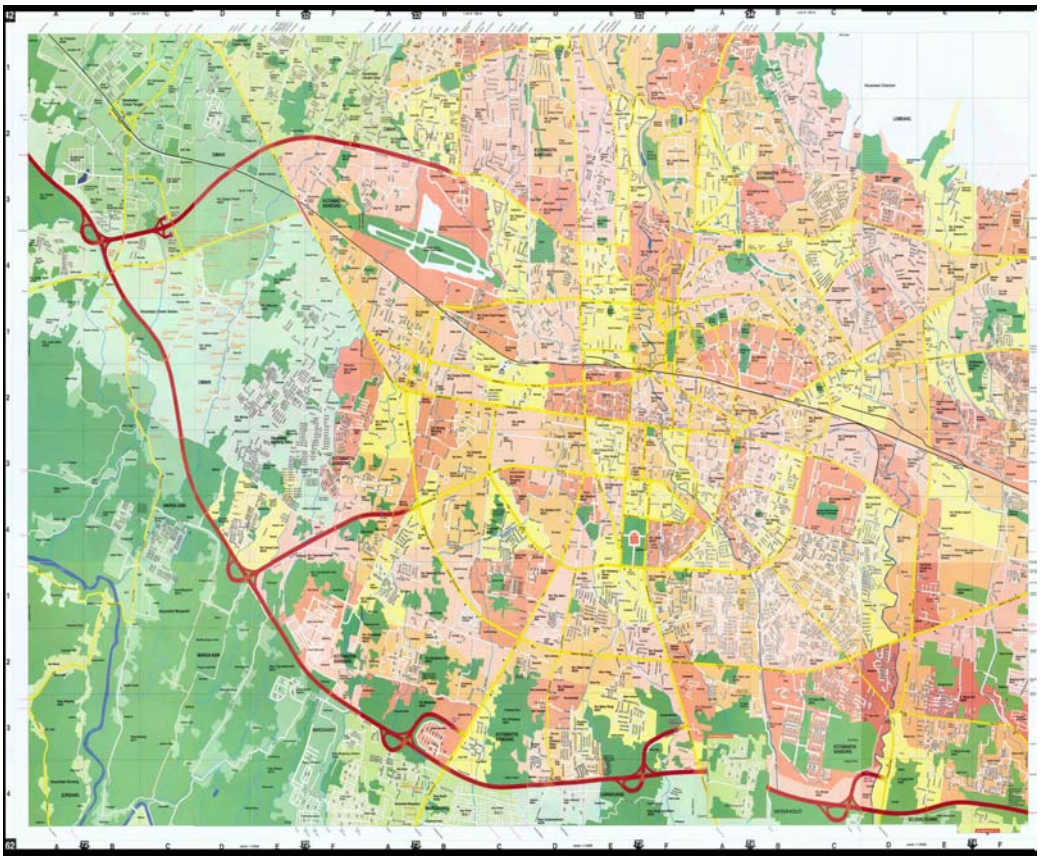
Pengujian pun dilanjutkan untuk mencari kriteria citra apakah yang cocok untuk disisipkan pesan.

Gambar 15 citra 1

Gambar 16 citra 2

Gambar 17 citra 3

Gambar 18 Citra 4



Gambar 19 Citra 5

Gambar 20 Citra 6

Untuk pengujian citra dengan kriteria mana yang terbaik untuk menyisipkan pesan adalah dengan dilakukan pengujian terhadap 3 citra. Citra-citra tersebut mempunyai spesifikasi sebagai berikut :

1. Citra 1
   Nama File : gifshuffle.gif
   Ukuran : 100 * 200
   Warna : 8 bit

2. Citra 2
   Nama File : angsa.gif
   Ukuran : 1024 * 800
   Warna : 24 bit

3. Citra 3
   Nama File : pics23.gif
   Ukuran :800 *600
   Warna : 8 bit

4. Citra 4
   Nama File : lena.gif
   Ukuran : 512*512
   Warna : 8 bit

5. Citra 5
   Nama File : map.gif
   Ukuran : 4813 * 3425
   Warna : 24 bit

6. Citra 6
   Nama File : tiger.gif
   Ukuran : 2048 * 1536
   Warna : 24 bit

Dari pengujian terhadap 6 buat citra tersebut ternyata diperoleh kesimpulan bahwa ternyata semua citra memberikan batas penampungan yang sama yaitu 209 bytes, bahkan citra sebesar 4813*3425 piksel pun tidak memberikan tambahan ruang untuk penyisipan pesan. Hal ini terjadi karena algoritma gifshuffle hanya melakukan penukaran pada 256 palet warna. Sehingga ukuran berkas dan jumlah warna pada media pengampung tidak menjadi masalah dalam algoritma ini. Karena sebesar apapun ukuran file dan sebanyak apapun warna yang dipakai tetap saja ukuran palet warna yang digunakan dalam berkas GIF adalah 256 buah.

Perlu dicatat bahwa waktu yang digunakan oleh paranoia dalam menyelesaikan tugasnya menyisipkan pesan adalah sekita 70 ms. Hal ini tergolong cepat karean ketika meakukan penyisipan pada gambar dengan format JPG waktu dapat mencapai 1 detik lebih.

7. Kesimpulan

Kesimpulan yang diperoleh dari studi tentang algoritma GifShuffle dalam format GIF ini adalah :

1. Algoritma ini cukup baik untuk diterapkan pada sebuah file gif. Karena penyisipan pesan sama sekali tidak menimbulkan adanya perbedaan dalam gambar.
2. Algoritma ini memiliki kelemahan yaitu citra yang telah disisipi pesan tidak tahan terhadap perubahan. Karena setelah dilakukan beberapa perubahan terhadap citra diperoleh hasil bahwa pesan tidak lagi dapat diekstrak.
3. Algoritma ini cukup mudah untuk diterapkan karena hanya mengandung langkah-langkah singkat yang tidak rumit.
4. Format file gif adalah format file yang cocok dalam dunia web karena berukuran kecil. Format ini sekarang hanya kalah dari format PNG (Portable Network Graphics).
5. Algoritma ini hanya cocok untuk menyisipkan pesan-pesan pendek. Karena berdasarkan pengujian algoritma ini hanya dapat disisipkan 209 bytes pesan atau 209 karakter.
6. Penambahan penggunaan algoritma kompresi pesan akan menambah jumlah pesan yang dapat disisipkan

8. Daftar Pustaka

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Penjelasan singkat steganografi, *http://en.wikipedia.org/wiki/Steganography*. Tanggal akses: 12 Oktober 2005 pukul 08:00.

[3] Penjelasan singkat sejarah herodotus *http://en.wikipedia.org/wiki/The_HIstories Of Herodotus* Tanggal akses: 12 Oktober 2005 pukul 08:00.

[4] Klasifikasi algoritma steganografi *http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004_03_research01.htm*. Tanggal akses: 12 Oktober 2005 pukul 08:25.

[5] Penjelasan singkat tentang isu-isu dalam dunia steganografi, *http://technolog.it.umn.edu/technolog/issues/fall2004/steg.htm* Tanggal akses: 10 Oktober 2005 pukul 16:00.

[6] Penjelasan sejarah format GIF, *http://www.olsenhome.com/gif/* Tanggal akses: 12 Oktober 2005 pukul 08:00.

[7] Perbandingan Format GIF dan JPG, *http://www.siriusweb.com/tutorials/gifvsjpg/* Tanggal akses: 12 Oktober 2005 pukul 08:25.

[8] Format Resmi File GIF, *http://ptolemy.eecs.berkeley.edu/eecs20/sidebars/images/gif.html* Tanggal akses: 1 Oktober 2005 pukul 16:00.

[9] Website resmi GifShuffle, *http://www.darkside.com.au/gifshuffle* Tanggal akses: 1 Oktober 2005 pukul 16:00.

[10] Websitedari Paranoia, *http://ccrma.stanford.edu/~Eeberdhahl/Projects/Paranoia* Tanggal akses: 1 Oktober 2005 pukul 16:00.

[11] Metode steganografi untuk citra *http://www.ws.binghamton.edu/fridrich/Research/pics99.ps* Tanggal akses: 1 Oktober 2005 pukul 16:00.

[12] Kutter, M., A Fair benchmark for image watermarking system.,Kutter 99 fair

[13] Beberapa resource tentang GIF, *http://www.compression-links.info/GIF* Tanggal akses: 1 Oktober 2005 pukul 16:00.

[14] Masalah patent dari GIF *http://lpf.ai.mit.edu/Patents/patents.html#GIF* Tanggal akses: 11 Oktober 2005 pukul 13:10.

[15] Masalah patent dari GIF *http://www.freesoftwaremagazine.com/node/1772* Tanggal akses: 11 Oktober 2005 pukul 13:00.

[16] Perbandingan beberapa format gambar *http://en.wikipedia.org/wiki/Comparison_of_graphics_file_formats* Tanggal akses: 11 Oktober 2005 pukul 13:30.

[17] Website dari Compuserve *http://www.compuserve.com/ Tanggal akses: 12 Oktober 2005 pukul 08:10.*

[18] Pembahasan format gambar *http://en.wikipedia.org/wiki/Image_file_formats* Tanggal akses: 11 Oktober 2005 pukul 13:30.

[19] Perbandingan beberapa format gambar *http://en.wikipedia.org/wiki/Comparison_of_graphics_file_formats* Tanggal akses: 11 Oktober 2005 pukul 13:00.

[20] Software Steganografi *http://www.jjtc.com/Security/stegtools.htm* Tanggal akses: 11 Oktober 2005 pukul 14:30.

[21] Riset terkait oleh FBI
http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004_03_research01.htm
Tanggal akses: 11 Oktober 2005 pukul 13:30.

[22] Kumpulan tools steganografi
http://stegano.net/
Tanggal akses: 11 Oktober 2005 pukul 14:45.

[23] Steganalisis analisis
http://www.jjtc.com/Steganalysis/
Tanggal akses: 11 Oktober 2005 pukul 13:30.