STEGANOGRAFI DIGITAL CITRA BERGERAK

ANIMATED GIF

Inas Luthfi – NIM : 13506019
Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : nassdzr@gmail.com

Abstrak

Makalah ini membahas tentang steganografi digital citra bergerak *Animated Graphics Interchange Format* (*Animated GIF*) dimana file *animated gif* tersebut memiliki beberapa frame gambar sebagai media penyisipan *plain text*. Terdapat dua algoritma steganografi berkas citra digital yang akan dibahas dalam makalah ini yaitu LSB (*Least Significant Bit)* dan GifShuffle. Format *animated GIF* sering dipakai dalam dunia web untuk memberikan animasi yang sederhana seperti iklan ataupun pemanis navigasi. Format GIF membatasi jumlah warnanya sebanyak 256 warna dan merupakan format berkas multimedia yang *lossless* sehingga sesuai untuk media steganografi.

Algoritma LSB merupakan salah satu algoritma steganografi dimana penyisipan pesan dilakukan dengan cara merubah bit LSB lebih tinggi atau lebih rendah satu bit dari sebelumnya. Karena hanya bit LSB yang dirubah, maka menurut penglihatan manusia, gambar tidak akan banyak berubah.

Algoritma GifShuffle yang dikembangkan oleh Matthew Kwan adalah salah satu algoritma steganografi yang menggunakan berkas citra dengan format GIF. Algoritma ini melakukan penyisipan pesan dengan cara mengganti susunan palet warna yang ada dalam sebuah berkas citra dalam format GIF.

Menyembunyikan pesan dalam kumpulan frame gambar memiliki keuntungan tersendiri yang akan penulis bahas dalam makalah ini. Keuntungan yang paling utama adalah batasan panjang pesan yang disembunyikan tidak terbatas dalam satu frame gambar saja, serta proses steganalisis yang digunakan untuk memecahkan gambar berisi pesan rahasia tentu akan lebih lama.

***Kata kunci**:* Steganografi citra, *Animated GIF*, *LSB, GifShuffle, GIF Palette*

1. Pendahuluan

Terdapat banyak cara untuk menyampaikan pesan secara aman kepada orang lain, misalnya dengan cara menenkripsi pesan dan hanya orang yang dituju yang dapat mendekripsi pesan tersebut, namun cara ini tentu akan membuat orang lain curiga terdapat pesan rahasia dalam pesan terenkripsi tersebut. Untuk menghilangkan kecurigaan tersebut, dikembangkanlah Steganografi, yaitu ilmu dan seni menyembunyikan (*embedded*) informasi dengan cara menyisipkan pesan di dalam pesan lain [1]. Pesan lain dalam artian ini tidaklah harus merupakan sebuah teks seperti layaknya pesan yang akan disembunyikan, namun dapat merupakan sebuah media seperti citra, video, suara, dan lain-lain. Karena media yang disisipi pesan tidak mengalami banyak perubahan berdasarkan indra manusia (mendengar, melihat), orang lain tidak akan curiga bahwa ada pesan tersembunyi yang disisipkan.

Steganografi sendiri terus dikembangkan, terutama dalam bidang steganografi digital. Apabila ada media digital baru yang memungkinkan untuk disisipi pesan tersembunyi, misal video dengan *codec*MPEG, besar kemungkinan akan muncul algoritma untuk menyembunyikan pesan dalam media tersebut.

2. Steganografi

Istilah steganografi berasal dari bahasa Yunani yaitu *steganos* yang artinya adalah penyamaran atau penyembuyian dan *graphein* yang artinya adalah tulisan. Pada tahun 400 SM di Yunani, dikenal istilah *wax tablet*. Pesan ditulis di atas media kayu kemudian disembunyikan dengan

melapisi lilin sebagai penutupnya. Ketika Perang Dunia II, untuk menulis dokumen yang tersembunyi, pesan ditulis dengan menggunakan tinta yang tak terlihat, untuk melihat apa yang ditulis, dokumen harus dipanaskan agar tinta tak terlihat tersebut menjadi gelap dan pesan yang disembunyikan dapat dibaca.

Dalam steganografi modern, dengan kemajuan teknologi komputer digital, fokus steganografi cenderung kearah menyembunyikan pesan dalam media digital, misalnya citra digital dengan format BMP, GIF, suara dan musik digital, bahkan dengan algoritma steganografi tertentu, tidak hanya media digital, namun berkas HTML dapat pula disisipi pesan dengan cara merubah urutan *tag*-nya. Steganografi ini disebut dengan steganografi digital.

Beberapa contoh media yang digunakan dalam Steganografi Digital:

- Citra Digital
  Citradigital merupakan sebuah media digital yang paling sering dipertukarkan, ukurannya yang relatif kecil merupakan pilihan yang tepat untuk menyembunyikan pesan singkat

- *Well Formed* Teks
  Dokumen HTML dan XML merupakan sebuah dokumen yang memiliki cara pemformatan menggunakan *tag-tag*. Urutan dari *tag* dan atribut *tag* tersebut dapat diatur kembali yang bertujuan untuk menyembunyikan pesan rahasia

- Audio
  Sebuah file audio, biasanya memiliki durasi yang cukup panjang, dengan memanipulasi *timing* dan frekuensi suara menggunakan algoritma steganografi spesifik, pesan dapat disisipkan tanpa membuat curiga orang yang mendengarkan file audio.

- Video
  Video memiliki algoritma sendiri (codec) untuk memperkecil ukuran berkas video tanpa mengorbankan kualitas gambar. Steganografi dapat memanfaatkan algoritma tersebut untuk memasukkan pesan rahasia kedalam video.

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

- Imperceptible
  Keberadaan pesan dalam media penampung tidak dapat dideteksi.

- Fidelity
  Media digital yang disisipi pesan tidak banyak mengalami perubahan dalam hal kualitas.

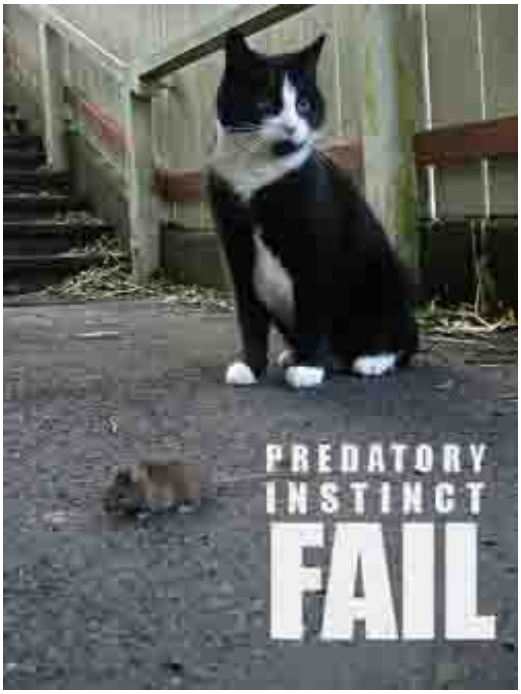
- Recovery
  Pesan rahasia yang telah disisipkan dalam media penampung harus dapat diungkap kembali. Hal ini merupakan syarat mutlak dalam sebuah algoritma steganografi.

3. Format Citra Digital

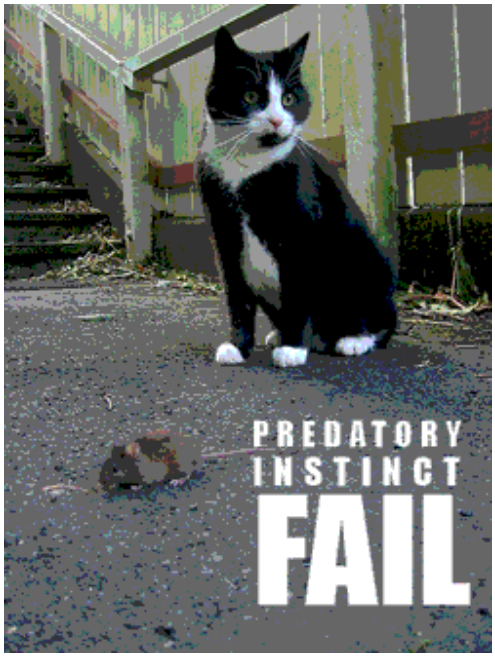
Graphics Interchange Format disingkat GIF adalah sebuah format berkas citra yang diperkenalkan pada tahun 1987 oleh CompuServe untuk menggantikan format RLE yang hanya mampu menampilkan gambar dengan warna hitam dan putih saja. Spesifikasi yang terkenal dari GIF adalah spesifikasi GIF87 dan GIF89 yang maksimum memiliki 256 warna dalam *palette*-nya[2]. Kedua format tersebut dapat diketahui dari header berkas citra GIF tersebut. *Palette* dalam citra digital berformat GIF merupakan informasi warna yang dimiliki oleh sebuah citra. Apabila sebuah citra berformat GIF hanya memiliki dua buah warna yaitu hitam dan putih, tentu *palette* yang berukuran optimal akan terdapat dua informasi warna saja. Warna yang dipilih dalam sebuah *Pallete* diambil dari warna RGB 24-bit, namun karena dibatasi menjadi 256 warna dan biasanya dioptimalkan dengan warna *websafe* saja, maka pada citraGIF sulit untuk menampilkan gradasi warna yang halus, lebih terkesan terkotak-kotak gradasinya.

Format citra yang terkenal sekarang adalah *Portable Network Graphic* (PNG), GIF, dan *Joint Photographic experts Group* (JPG). Karena GIF beberapa tahun yang lalu terkena biaya lisensi kompresi LZW yang dipakai didalam spesifikasi GIF, muncullah format baru citra yang menyerupai GIF, yaitu PNG. Dalam spesifikasinya, PNG tidak memiliki batasan 256 warna seperti halnya GIF, bahkan kedalaman warna PNG dapat mencapai 32-bit termasuk dengan transparansi yang halus. PNG merupakan format citra yang *lossless* seperti halnya GIF, namun PNG tidak dapat melakukan animasi per-frame gambar. Terdapat format lain yang merupakan turunan PNG seperti Motion PNG (MPNG), sayangnya format ini tidak populer

dibandingkan *Animated GIF*. Lain halnya dengan JPG, JPG muncul sebagai format citra yang *lossy* artinya sebagian informasi citra akan hilang ketika kualitas citra diturunkan (dikompres misalnya). Format JPG sesuai digunakan untuk foto-foto digital, karena sering berukuran besar dan detail foto tidak perlu terlalu detail. Dengan detail yang relatif sama, format citra GIF berukuran lebih besar daripada format citra JPG



Gambar 1 Gambar berformat JPG dengan kompresi sangat tinggi(26,1kb)



Gambar 2 Gambar berformat GIF dengan palette WebSafe (21,6 kb)

3. 1. Detail Format GIF

Format *Animated GIF* memiliki perbedaan dalam struktur penyimpanan berkasnya. Berikut ini adalah berkas citra berformat GIF89 yang standar[2]:

```
GIF Header
Graphic Control Extension
Image Block
Trailer
```

dan berikut ini adalah format *Animated* GIF:

```
GIF Header
Application Extension
[
  Graphic Control Extension
  Image Block
]*
Trailer
```

Berikut pengaturan detail format GIF89:

Byte Order: Little-endian

GIF Header

```
Offset   Length   Contents
  0      3 bytes  "GIF"
  3      3 bytes  "87a" or "89a"
  6      2 bytes  <Logical Screen Width>
  8      2 bytes  <Logical Screen Height>
 10      1 byte
                   bit 0:     Global Color Table Flag
(GCTF)
```

```
        bit 1..3: Color Resolution
        bit 4:    Sort Flag to Global Color Table
                    bit  5..7:  Size  of  Global  Color
Table: 2^(1+n)
 11       1 byte   <Background Color Index>
 12       1 byte   <Pixel Aspect Ratio>
 13         ? bytes   <Global  Color  Table(0..255  x  3
bytes) if GCTF is one>
          ? bytes  <Blocks>
          1 bytes  <Trailer> (0x3b)
```

Image Block

```
Offset   Length   Contents
  0      1 byte   Image Separator (0x2c)
  1      2 bytes  Image Left Position
  3      2 bytes  Image Top Position
  5      2 bytes  Image Width
  7      2 bytes  Image Height
  8      1 byte
                    bit 0:     Local  Color  Table  Flag
(LCTF)
        bit 1:    Interlace Flag
        bit 2:    Sort Flag
        bit 2..3: Reserved
        bit 4..7: Size of Local Color Table: 2^(1+n)
         ? bytes  Local Color Table(0..255 x 3 bytes)
if LCTF is one
          1 byte   LZW Minimum Code Size
[ // Blocks
          1 byte   Block Size (s)
         (s)bytes  Image Data
]*
          1 byte   Block Terminator(0x00)
```

Graphic Control Extension Block

```
Offset   Length   Contents
  0      1 byte   Extension Introducer (0x21)
  1      1 byte   Graphic Control Label (0xf9)
  2      1 byte   Block Size (0x04)
  3      1 byte
                    bit 0..2: Reserved
        bit 3..5: Disposal Method
        bit 6:    User Input Flag
        bit 7:    Transparent Color Flag
  4      2 bytes  Delay Time (1/100ths of a second)
  6      1 byte   Transparent Color Index
  7      1 byte   Block Terminator(0x00)
```

Comment Extension Block

```
Offset   Length   Contents
  0      1 byte   Extension Introducer (0x21)
  1      1 byte   Comment Label (0xfe)
[
          1 byte   Block Size (s)
         (s)bytes  Comment Data
]*
          1 byte   Block Terminator(0x00)
```

Plain Text Extension Block

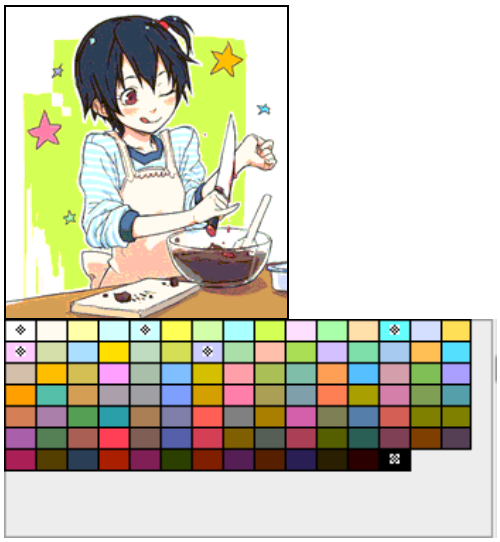
```
Offset   Length   Contents
  0      1 byte   Extension Introducer (0x21)
  1      1 byte   Plain Text Label (0x01)
  2      1 byte   Block Size (0x0c)
  3      2 bytes  Text Grid Left Position
  5      2 bytes  Text Grid Top Position
  7      2 bytes  Text Grid Width
  9      2 bytes  Text Grid Height
 10      1 byte   Character Cell Width(
 11      1 byte   Character Cell Height
 12      1 byte   Text Foreground Color Index(
 13      1 byte   Text Background Color Index(
[
          1 byte   Block Size (s)
         (s)bytes  Plain Text Data
]*
          1 byte   Block Terminator(0x00)
```

Application Extension Block

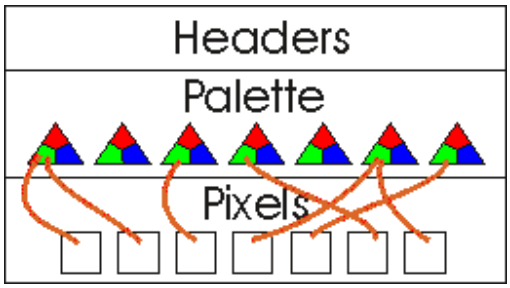
```
Offset   Length   Contents
  0      1 byte   Extension Introducer (0x21)
  1      1 byte   Application Label (0xff)
  2      1 byte   Block Size (0x0b)
  3      8 bytes  Application Identifire
[
          1 byte   Block Size (s)
         (s)bytes  Application Data
]*
          1 byte   Block Terminator(0x00)
```

3. 2. *Palette* Warna

Terlihat bahwa perbedaan antara format GIF biasa dan *Animated GIF* adalah berulangnya daerah Graphic Control Extension dan Image Block. Selain itu pada *Animated GIF*[3], tiap frame gambar dapat memiliki *palette* warna terpisah yang disebut *local palette*. Pada aplikasi pengolah citra sekarang, rata-rata menyarankan untuk menghilangkan *local palette* diganti dengan menggunakan satu *global palette* yang dapat diakses oleh setiap frame dalam citra bergerak *Animated GIF*. Namun karena sifat algoritma GifShuffle yang akan dibahas [5], menggunakan satu *global palette* berarti mengurangi panjang pesan rahasia yang disisipkan kedalam citra bergerak secara signifikan, meskipun apabila menggunakan LSB, *palette* warna tidaklah berpengaruh.



Gambar 3 Gambar berformat GIF dan Palette Warnanya (111 warna/256 warna)



Gambar 4 Multi *Palette Animated* GIF

Dengan detail tersebut, kita dapat membaca byte demi byte dari sebuah berkas citra GIF untuk kemudian kita sisipkan dengan pesan rahasia melalui algoritma LSB maupun dengan GifShuffle

4. Algoritma LSB

LSB atau *Least Significant Bit* sesuai namanya, menyembunyikan pesan rahasia dengan cara merubah bit paling rendah dari sebuah pixel warna pada citra.

[1] Misalkan penyisipan pada citra 24-bit. Setiap pixel panjangnya 24 bit (3 x 3 byte, masing-masing komponen R (1 byte), G (1 byte), dan B (1 byte))

00110011 10100010 11100010
(misal pixel berwarna merah)

Misalkan embedded message: 010

Encoding:
00110010 10100011 11100010
(pixel berwarna "merah berubah sedikit", tidak dapat dibedakan secara visual dengan citra aslinya)

Untuk memperkuat teknik penyembunyian data, bit-bit data rahasia tidak digunakan mengganti *byte-byte* yang berurutan (pixel yang berurutan), namun dipilih susunan *byte* secara acak, dengan bantuan Pembangkit bilangan acak-semu (*PRNG*, *pseudo-random-number-generator*)

Umpan (*seed*) untuk bilangan acak berlaku sebagai kunci (*stego-key*). Misal citra berukuran 40 byte yang dan ada 3 bit yang akan disembunyikan, dapat dipilih byte 12,32,5 untuk menyembunyikan pesan

Kunci sendiri dapat dibuat secara otomatis misalnya dari ukuran file, banyaknya warna dalam *palette*, dan lain-lain.

5. Algortima GifShuffle

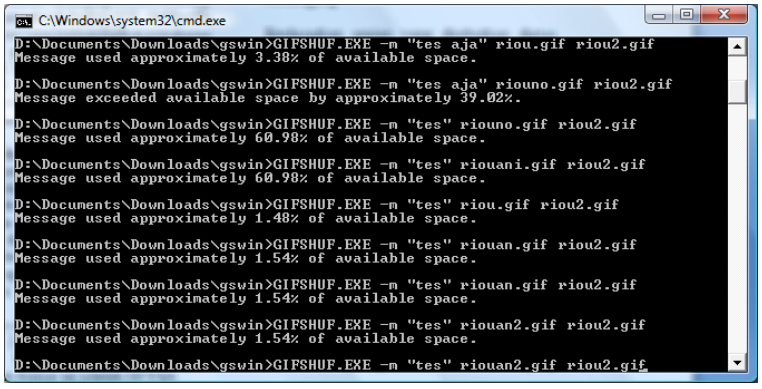
GifShuffe adalah sebuah algoritma steganografi yang digunakan untuk menyembunyikan pesan dalam berkas citra format GIF. Algoritma ini ditemukan oleh *Matthew Kwan*. Sarjana ilmu komputer lulusan dari University of Melbourne.

Dalam situsnya[5] beliau memberikan sebuah penjelasan singkat tentang algoritma ini dan juga memberikan program yang menerapkan algoritma ini, lengkap dengan source codenya.

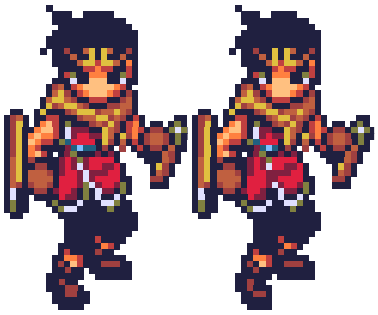
Algoritma GifShufle pada intinya memanfaatkan header file GIF yang menyimpan palet warna sebagai media penyisipan pesan. Dalam algoritma ini tidak terjadi perubahan apapaun dalam data bitmapnya. GifShuffle akan melakukan "Shuffle" atau perputaran terhadap palet warna dari sebuah berkas GIF. GifShuffle adalah algoritma yang memanfaatkan penukaran posisi ke 256 palet warna dalam berkas citra berformat GIF, berhubung dua buah berkas GIF dengan palet warna yang berbeda akan ditampilkan secara sama persis, jadi tidak masalah bagaimana urutan warna dalam *palette*.

Dengan dilakukannya penukaran posisi maka akan dapat diperoleh sebuah informasi berkaitan dengan perbedaan 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.

Karena berkas dengan format GIF89 maksimal mengandung 256 palet warna maka dapat disimpulkan bahwa total penyimpanan maksimum dari format ini adalah 1675 bit.



Gambar 5 Program GIFSHUF.EXE Untuk menyisipkan pesan dengan algoritma GifShuffle



Gambar 6 Citra *Animated Gif* yang disisipi pesan, sebelum dan sesudah

6. Steganalisis

Steganalisis adalah ilmu yang mempelajari karakteristik penyembunyian suatu data pada media (steganografi) dan bagaimana cara untuk mendeteksi bahkan sampai membongkar data tersembunyi tersebut. Steganalisis secara teori cukup mudah, namum implementasinya cukup sulit dan memakan waktu dan ketelitian tinggi.

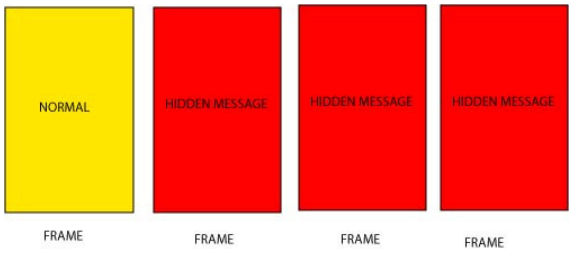
Metode paling mendasar dari Steganalisis adalah mencari perbedaan dari dua buah media, media yang normal dan media yang diyakini terdapat pesan rahasia. Apabila terdapat pesan rahasia yang disembunyikan, pasti akan terdapat perbedaan yang cukup terlihat.

Dari kedua algoritma yang telah disebutkan sebelumnya, untuk algoritma LSB, terdapat metode steganalisis khusus yang disebut ***Noise Floor Consistency Analysis*** dimana tiap pixel dalam gambar akan dicek *Higher-Order Bits*-nya. Pesan rahasia akan membentuk *noise.* Apabila diyakini ada pesan rahasia, maka proses selanjutnya adalah memecahkan metode LSB apa yang dipakai.

Untuk algoritma Gifshuffle, karena metode yang dipakai adalah penukaran posisi warna dalam palet, maka metode steganalisis yang paling sesuai adalah metode statistik.

6.1 *Bluffing*

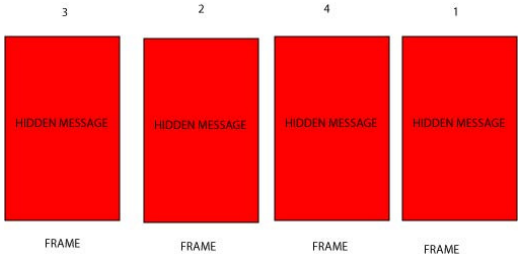
Berdasarkan uraian yang disebutkan diatas, penulis memiliki ide untuk mengecoh proses steganalisis. Teknik ini penulis sebut *Bluffing* dimana dalam sebuah berkas citra *Animated GIF*. Steganografi tidak akan dilakukan pada frame pertama, namun dilakukan pada frame-frame selanjutnya. Dengan frame pertama merupakan frame normal, kemungkinan proses steganalisis akan terkecoh dan menganggap berkas *Animated GIF* yang diteliti merupakan berkas citra normal.



Gambar 7 Steganografi *Animated* GIF *Bluffing*

6.2 PRNG *Frame Order*

Seperti pada algoritma LSB yang membuat *Pseudo-Random-Number-Generator* yang *dependant* terhadap variabel khusus (besar file, jumlah warna pallete, dan lain-lain). Penulis memikirkan hal yang sama dapat pula diterapkan pada proses steganografi *Animated GIF*. Pesan akan disisipkan pada frame yang ditunjuk oleh PNRG, misalkan terdapat 5 buah frame animasi pada satu berkas *Animated GIF*, maka pesan akan dituliskan dengan urutan frame ke 2-3-1-5-4



Gambar 8 Steganografi *Animated* GIF *Frame Order*
(Urutan pesan dimasukkan di frame 4-2-1-3)

7. Keuntungan Steganografi Animated GIF

Steganografi pada *Animated GIF* sebenarnya sama saja dengan steganografi pada gambar yang tidak bergerak atau statis namun perbedaannya, karena terdapat banyak frame gambar, maka panjang pesan yang dapat disisipkan akan menjadi lebih banyak.

- Algoritma LSB

  Pada algoritma ini, ukuran citra sangat mempengaruhi besar gambar, dengan format *Animated GIF* yang memiliki banyak

frame, tentunya lebih banyak lagi pesan yang dapat disisipkan kedalam gambar.

- Algoritma GifShuffle

Pada algoritma ini, *palette* warna adalah media yang dipakai untuk menyembunyikan pesan, oleh karena itu panjang pesan yang dapat disisipkan kedalam citra tidak terpengaruh pada ukuran citra, oleh karena itu hal ini tentu sangat membatasi panjang pesan yang dapat disisipkan. *Animated GIF* sangat membantu menambah batasan pesan yang dapat disisipkan, asalkan citra *Animated GIF* menggunakan *local palette* pada setiap framenya, bukan menggunakan satu *global palette*

8. Kesimpulan

Kesimpulan yang dapat dimbil dari studi steganografi pada *Animated GIF* adalah:

1. Format berkas GIF adalah format file yang sesuai digunakan untuk steganografi karena berukuran kecil dan bersifat *lossless*.
2. *Animated GIF* adalah format media steganografi yang memiliki kelebihan tersendiri, contohnya adalah batasan pesan rahasia yang lebih panjang.
3. *Animated GIF* memiliki multiple *frame* yang merupakan kelebihan dibandingkan format GIF yang haya memiliki satu buah frame.
4. LSB adalah algoritma yang sering dipakai dalam steganografi citra digital
5. GifShuffle adalah algoritma yang tidak merubah *bitmap* berkas citra digital, namun hanya mampu menampung pesan rahasia yang sedikit.
6. Proses steganalisis dapat dihambat dengan teknik yang diusulkan penulis yaitu *Bluffing* dan *PNRG Frame Order*. Metode tersebut dapat menipu proses steganalisis

DAFTAR PUSTAKA

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

[2] Format Resmi File GIF89 dari W3C, *http://www.w3.org/Graphics/GIF/spec-gif89a.txt*
Tanggal akses: 29 Maret 2009 pukul 17:10.

[3] Detail Header GIF
http://www.onicos.com/staff/iz/formats/gif.html#header
Tanggal akses: 29 Maret 2009 pukul 18:10.

[3] Pengenalan Format Animated GIF
http://www.botos.com/anigif.html
Tanggal akses: 29 Maret 2009 pukul 18:10

[4] Tutorial Codeproject, *Steganography In Picture Noise*
http://www.codeproject.com/KB/security/steganodotnet.aspx
Tanggal akses: 29 Maret 2009 pukul 17:10

[5] Website Resmi GifShuffle, *http://www.darkside.com.au/gifshuffle*
Tanggal akses: 29 Maret 2009 pukul 17:00.

[6] Tutorial Codeproject, *Steganography Order Shuffling*
http://www.codeproject.com/KB/security/steganodotnet14.aspx
Tanggal akses: 29 Maret 2009 pukul 17:20

[7] Tutorial Codeproject, *Steganography Indexed Image Pallete Shuffling*
http://www.codeproject.com/KB/security/steganodotnet11.aspx
Tanggal akses: 29 Maret 2009 pukul 17:50

[8] Kumpulan tools dan informasi steganografi
http://stegano.net/
Tanggal akses: 30 Maret 2009 pukul 08:00

[9] Artikel tentang Steganalisis
http://andreastjong.wordpress.com/2008/09/20/steganalisis-aka-steganalysis/
Tanggal akses: 30 Maret 2009 pukul 10:00