PENYAMARAN PESAN PADA FILE SHOCKWAVE FLASH (.SWF) DENGAN MENGGUNAKAN METODE "*ZERO OPACITY*"

Abraham Ranardo Sumarsono – NIM : 13507056

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jln. Ganesha 10 Bandung
e-mail: abrahamranardo@gmail.com

Abstrak

Flash merupakan sebuah teknologi saat ini yang sering dimanfaatkan sebagai media untuk animasi gambar, audio, dan teks. Dalam sehari-hari, flash dapat dimainkan dengan cara meng-*embed* file flash pada web lalu dimainkan pada web browser ataupun dimainkan langsung dari dalam *disk* dengan menggunakan sebuah flash *player*.

Di dalam proses pembuatan sebuah file flash pun tidak rumit, hal-hal yang dilakukan hanyalah rekayasa dan manipulasi gambar, audio dan teks. Sehingga flash pun menjadi suatu alternatif utama di dalam membuat suatu animasi.

Dalam pembuatan suatu file flash, kita dapat memasukkan gambar, audio, dan teks ke dalam file dan memanipulasinya. Manipulasi yang dapat dilakukan pun bermacam-macam, mulai dari mengubah koordinat titik peletakan gambar/teks, mengecilkan suara audio, mengubah skala gambar, dan masih banyak lagi manipulasi yang dapat dilakukan.

Dari karakteristik tersebut, dirancanglah suatu mekanisme steganografi dimana file flash dimanipulasi untuk menyimpan suatu pesan rahasia. Metode ini dinamakan "*Zero Opacity*", dimana manipulasi yang dilakukan adalah dengan mengubah tingkat transparansi suatu teks ataupun gambar sehingga tidak dapat dilihat oleh mata.

Tools yang digunakan pun hanya *decompiler* flash ke notasi Flash Assembly, yaitu *flasm*. Teknik membuka pesan rahasia tersebut pun hanya dengan membuat suatu file baru dengan meng-*include* file flash yang menjadi media pesan rahasia dan menambahkan beberapa kode serta nama *MovieClip* (instansi pesan rahasia) sebagai kunci dari pembongkaran pesan rahasia tersebut.

Namun, untuk memudahkan teknik melihat pesan rahasia tersebut, penulis berkontribusi dengan membuat suatu program yang hanya membutuhkan masukan kunci berupa nama *MovieClip* pesan rahasia tersebut dan akan mengenerasi file flash yang baru dengan pesan rahasia terlihat.

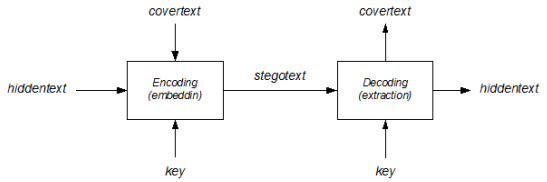
Kata kunci : *Zero Opacity,* flash, *decompiler*, *MovieClip*, tingkat transparansi, *flasm*.

1. Pendahuluan

Seiring dengan perkembangan teknologi informasi, internet menjadi salah satu kebutuhan penting di dalam kehidupan masyarakat sehari-hari. Pertukaran informasi pun kerap terjadi setiap saat di dunia maya. Sayangnya, pertukaran informasi melalui media ini tidaklah aman. Pesan penting yang kita kirimkan dapat saja disadap oleh orang lain. Bahkan transaksi yang penting pun dapat terganggu. Oleh karena itu, berkembanglah teknik-teknik kriptografi yang melindungi pesan yang dikirim. Dengan teknik-teknik kriptografi, walaupun makna suatu pesan menjadi hilang ataupun kacau, keberadaan pesan rahasia tersebut masih dapat diidentifikasi karena pesan akan terlihat ganjil atau aneh. Hal ini yang membuat penyadap informasi menyadari adanya pesan rahasia yang disembunyikan dan melakukan kriptanalis.

Berbeda dengan kriptografi, steganografi adalah salah satu metode yang dapat digunakan di dalam mengamankan akses terhadap suatu pesan tanpa menyebabkan kecurigaan. Penyembunyian ataupun penyamaran pesan ini dibuat sedemikian rupa sehingga pihak lain tidak curiga akan adanya pesan lain di dalam pesan yang dikirimkan. Hanya pihak penerima yang sah saja yang dapat mengetahui bahwa ada pesan lain yang disembunyikan di dalam pesan yang dikirim. Perbedaannya dengan kriptografi adalah kriptografi mengubah ataupun

mengacak karakter pesan menjadi bentuk lain yang tidak bermakna, sedangkan steganografi hanya mengaburkan ataupun menyembunyikan penyampaian pesan dengan berbagai cara dan tetap mempertahankan pesan.



Gambar 1. Langkah-langkah steganografi pada umumnya

Salah satu wadah yang dapat digunakan untuk berkirim pesan adalah melalui file flash. File flash ini berperan sebagai media penyampaian pesan dan pesan yang disampaikan adalah animasi, gambar, ataupun video yang ter-*embed* di dalam file flash itu sendiri. File flash ini pun dapat digunakan untuk menyamarkan suatu pesan dan kita dapat memanipulasi komponen flash tersebut untuk menyembunyikan pesan yang ada.

Flash merupakan sebuah platform multimedia yang terkenal dapat menambah animasi dan interaktivitas ke dalam halaman web. Pada awalnya, flash dikenalkan oleh Macromedia pada tahun 1996. Sekarang, flash dikembangkan dan didistribusikan oleh Adobe System inc.

Flash biasanya digunakan untuk membuat animasi, pengiklanan, dan bermacam-macam komponen flash di halaman situs, untuk mengintegrasikan video ke dalam halaman situs, dan saat ini untuk mengembangkan aplikasi internet yang bermacam-macam.

Flash dapat memanipulasi grafis berbasis vektor dan raster. Flash juga mendukung *streaming* dua arah dari video dan audio. Beberapa produk aplikasi, sistem dan alat-alat pun sekarang sudah dapat membuat ataupun menampilkan konten flash, termasuk *Adobe Flash Player*, yang bisa didapatkan secara gratis untuk browser web dan bermacam-macam peralatan *mobile*. Flash mengandung sebuah bahasa *scripting* yang disebut ActionScript.

ActionScript adalah sebuah bahasa *scripting* yang berbasis pada ECMAScript. Kegunaan utama dari ActionScript adalah untuk pengembangan situs web dan aplikasi yang menggunakan platform Flash (dalam format SWF yang ter-*embed* ke dalam halaman web). Selain itu, bahasa ini juga dapat digunakan di beberapa aplikasi basis data dan pada pemrograman dasar robot. Pada awalnya dikembangkan oleh Macromedia, tetapi sekarang dimiliki oleh Adobe.

ActionScript awalnya dirancang untuk mengendalikan animasi vektor 2D yang sederhana di Flash. Seiring dengan berkembangnya flash, versi-versi selanjutnya ditambahkan fungsionalitas yang mendukung komponen game berbasis web dan aplikasi internet dengan *streaming* media.

Tiga versi awal dari kakas Flash menyediakan fitur interaktivitas yang terbatas. Pengembang flash di awal hanya dapat menggunakan sebuah perintah sederhana yang disebut sebagai "action" pada sebuah tombol atau sebuah frame. Aksi-aksi yang ada antara lain kendali navigasi dasar, dengan perintah seperti misalnya, "play", "stop", "getURL", dan "gotoAndPlay".

Aksi-aksi ini pun berkembang menjadi sebuah bahasa *scripting* yang kecil. Kapabilitas yang ditambahkan termasuk variabel, ekspresi, operator, pernyataan "if then else", dan "loop". Dengan perkembangan yang pesat, bahasa ini sekarang pun dapat digunakan untuk memanipulasi berbagai objek-objek di dalam flash. Salah satu yang dapat dimanipulasi adalah MovieClip.

MovieClip merupakan sebuah kelas yang ada di bahasa *scripting* ActionScript. Objek yang merupakan instansi dari kelas MovieClip memiliki sebuah "timeline". Objek yang merupakan hasil instansiasi dari kelas MovieClip ini pun dapat berisi berbagai macam objek dari kelas-kelas lainnya termasuk kelas MovieClip itu sendiri. Karena dapat menampung banyak objek dan memiliki "timeline" sendiri, maka kelas ini kerap digunakan untuk mendefinisikan sebuah gambar, animasi, ataupun video.

Objek yang berada di dalam suatu MovieClip dapat dengan mudah diakses dengan bahasa *scripting* ActionScript. Cara pengaksesannya pun sama seperti mengakses atribut pada suatu objek.

2. Teknik Steganografi *Zero Opacity*

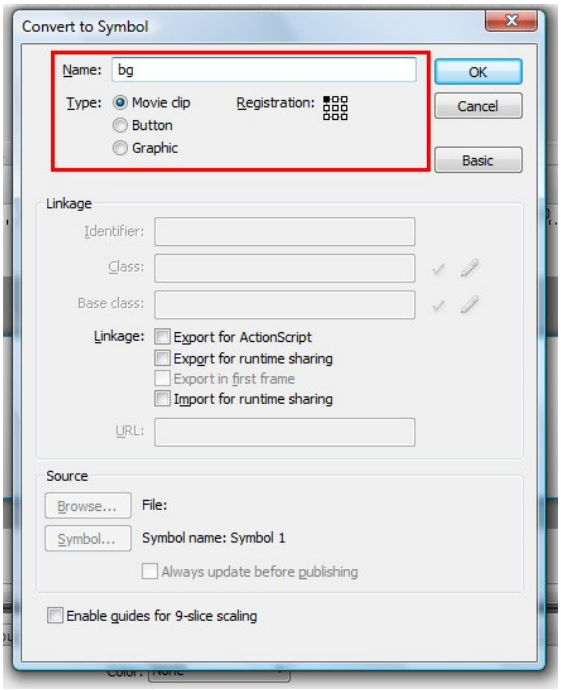
Teknik steganografi *Zero Opacity* ini berdasar pada suatu ide yang sederhana. Steganografi *Zero Opacity* mempergunakan manipulasi atribut "alpha" atau transparansi dari suatu objek di dalam file flash. Aspek dan karakteristik yang menjadi landasan teknik ini akan dijelaskan sebagai berikut.

2.1. Karakteristik Elemen

2.1.1. MovieClip

MovieClip adalah suatu kelas yang dapat menampung berbagai objek dari kelas lain serta kelasnya sendiri dan memiliki sebuah "timeline". Proses pembuatannya pun sangat mudah dengan adanya bantuan GUI Adobe. Objek apapun dapat

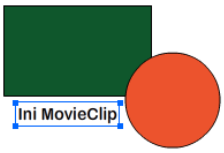
didefinisikan sebagai sebuah MovieClip hanya dengan meng-klik kanan objek tersebut dan memilih pilihan "Convert to Symbol". Selanjutnya pun dapat langsung memilih MovieClip sebagai hasil konversi.



Gambar 2. Screenshot proses konversi

Setelah menjadi sebuah MovieClip, pengguna dapat mengakses dan memanipulasi objek yang ada di dalamnya dengan meng-klik ganda MovieClip tersebut di panel "Library". Pengguna dapat membuat animasi ataupun mengubah atribut/properti dari sebuah objek di dalam MovieClip tersebut sesuai dengan kebutuhan masing-masing.

Atribut-atribut yang dapat dimanipulasi antara lain transparansi, mode warna, filter, tinggi, rotasi, skala absis dan ordinat, transformasi, lebar, visibilitas, letak absis, letak ordinat, dan masih banyak lagi properti yang dapat dimanipulasi.



Gambar 3. Contoh MovieClip

Objek MovieClip inilah yang menjadi pesan yang akan disembunyikan oleh metode steganografi *Zero Opacity*. Cara menyembunyikannya adalah dengan mengatur transparansi dari objek tersebut.

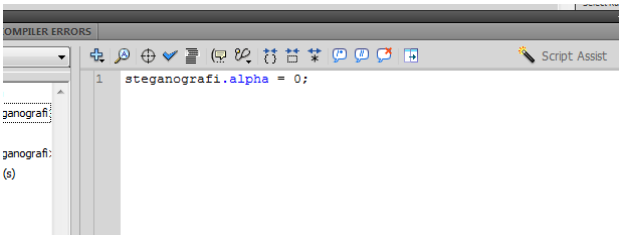
Nama objek hasil dari instansiasi MovieClip pun digunakan sebagai kunci dalam mengungkap pesan rahasia di metode steganografi *Zero Opacity*.

2.1.2. ActionScript 2.0

ActionScript 2.0 adalah suatu versi dari bahasa *scripting* ActionScript. Versi terbarunya adalah 3.0. Namun, meskipun sudah banyak yang beralih menggunakan versi 3.0, versi 2.0 masih banyak digunakan oleh masyarakat karena sintaksnya yang sederhana dan tidak kompleks seperti bahasa ActionScript 3.0.

Pengguna dapat menambahkan kode ActionScript pada "frame" manapun di dalam timeline dengan menekan "F8" atau masuk ke panel "Action". Bahkan, pengguna pun dapat membuat suatu file eksternal berekstensi .as yang berisi kode-kode ActionScript yang kemudian dapat dipanggil di dalam file proyek flash.

Berikut adalah gambar dari cara penambahan kode ActionScript dengan melalui panel "Action".



Gambar 4. Penambahan kode ActionScript melalui panel "Action"

Kode yang disoroti dalam metode ini adalah pengubahan nilai atribut transparansi dari suatu MovieClip. Hal ini dikarenakan pengubahan atribut transparansi pada metode steganografi ini dilakukan dengan kode ActionScript agar dapat dimanipulasi ketika file flash dibongkar ulang. Berikut adalah kode yang digunakan untuk menyembunyikan objek MovieClip rahasia tersebut:

```
nama-objek-kunci._alpha = 0;
```

2.1.3. Flasm

Flasm adalah suatu aplikasi yang dapat membongkar file SWF termasuk semua "timeline" dan "events". Dengan aplikasi ini, pengguna dapat belajar bagaimana compiler Flash bekerja, yang juga dapat meningkatkan kemampuan ActionScript.

Pengguna juga dapat melakukan optimisasi pada kode hasil bongkaran dari aplikasi ini secara

manual. Setelah dimodifikasi, aplikasi ini dapat mengaplikasikan hasil modifikasi tersebut ke file SWF yang asli, menimpa aksi yang orisinil (aksi awal yang belum dimodifikasi).

Yang perlu diperhatikan di sini adalah flasm dapat membongkar file SWF dan mengubah "bytecodes" yang ada menjadi representasi yang dapat dibaca oleh manusia bukan kode ActionScript.

Flasm adalah sebuah kakas yang berbasis pada baris perintah (*command line tool*). Untuk menggunakannya, pengguna harus membuka DOS terlebih dahulu. Untuk menjalankannya, pengguna cukup mengetikkan "flasm *command filename*" (pada pengguna Windows). Berikut adalah perintah-perintah yang dapat dieksekusi di flasm.

```
flasm command filename

command
-d  Disassemble SWF file to the console
-a  Assemble Flasm project
-u  Update SWF file, replace Flasm macros
-b  Assemble actions to __bytecode__()
instruction or byte sequence
-z  Compress SWF with zlib
-x  Decompress SWF
```

Setiap pernyataan ActionScript dikompilasi oleh Flash menjadi sepasang aksi "bytecode" yang sederhana. Sebagai contohnya, `a=b*b;` ditranformasi menjadi:

```
constants 'a', 'b'
push 'a', 'b'
getVariable
push 'b'
getVariable
multiply
setVariable
```

Kode-kode inilah yang disimpan di dalam file SWF dalam bentuk biner. Kode-kode tersebut diinterpretasikan oleh *virtual machine* di Flash Player. Kode yang ditampilkan sebelumnya adalah representasi visual dari "bytecode" yang digenerasi oleh Flasm. Kode "bytecode" yang asli merupakan kode mesin yang susah untuk dibaca oleh manusia.

Flasm bekerja mengikuti prinsip mesin virtual flash. Mesin virtual flash berbasis *stack*, sehingga pengguna tidak dapat mereferensikan ke suatu lokasi memori. Stack adalah sebuah tempat di memori dimana data bisa disimpan dengan aturan LIFO (*Last In First Out*). Sehingga operasi yang dilakukan pada "bytecode" di flasm pun berupa "push" dan "pop".

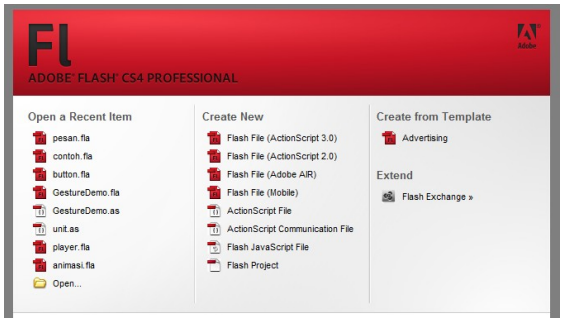
Batasan dari aplikasi flasm ini antara lain versi bahasa ActionScript yang didukung. Versi yang didukung adalah versi 2.0. Sedangkan pada saat ini, ActionScript sudah berkembang ke versi 3.0.

Salah satu pertimbangan penulis menggunakan aplikasi ini di dalam mendukung metode yang dirancang adalah aplikasi ini merupakan *freeware* dan tersedia untuk berbagai sistem operasi. Selain itu, flasm dapat menyusun ulang file SWF yang telah dipecah menjadi kode-kode "bytecode" baik yang asli maupun yang sudah dimodifikasi.

2.2. Langkah Menyembunyikan Pesan

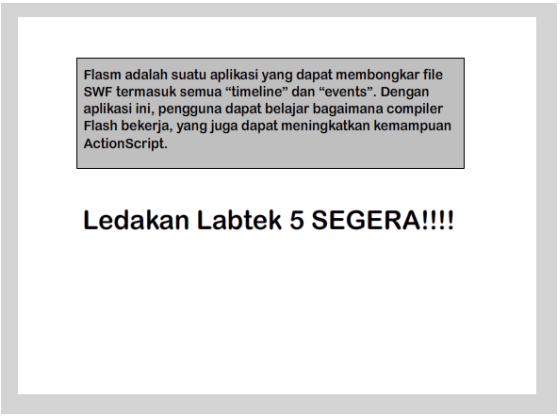
Untuk menyembunyikan pesan, penulis menganjurkan beberapa langkah sebagai berikut:

1. Buat file proyek flash dengan menggunakan bahasa ActionScript 2.0, lalu rancang animasi dan gambar sesuai dengan keinginan masing-masing sebagai cover teks.



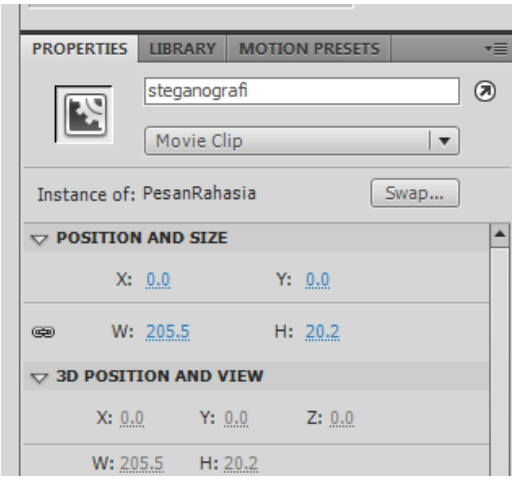
Gambar 5. Screenshot awal aplikasi pembuat flash

2. Buat objek MovieClip yang memuat pesan rahasia yang ingin disembunyikan. Contohnya pada gambar di bawah ini, pesan rahasia yang ingin disembunyikan adalah teks "Ledakan Labtek 5 SEGERA!!!!".



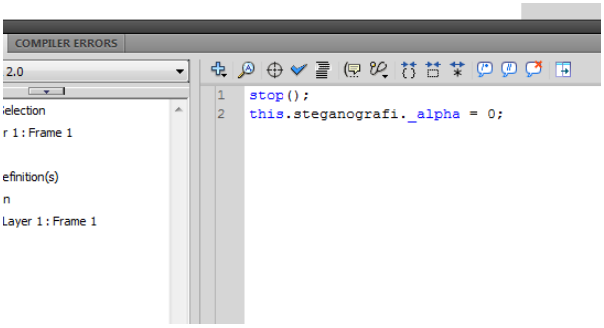
Gambar 6. Screenshot susunan MovieClip

3. Beri nama objek hasil instansiasi MovieClip yang menjadi pesan rahasia dalam file flash ini seperti gambar di bawah ini.



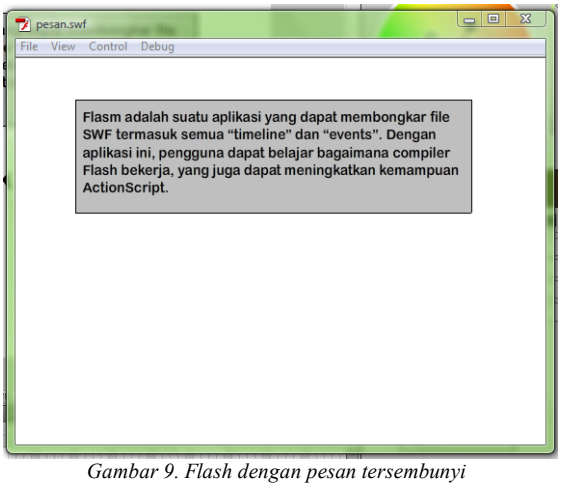
Gambar 7. Pemberian nama untuk MovieClip

4. Selanjutnya, ketikkan kode ActionScript 2.0 "`this.namamovieclip._alpha=0;`" pada panel "Action" di frame untuk menyembunyikan pesan rahasia.



Gambar 8. Penulisan kode ActionScript 2.0

5. *Publish* file proyek flash menjadi file SWF yang siap untuk dikirim.



Gambar 9. Flash dengan pesan tersembunyi

Hasil dari langkah-langkah tersebut adalah sebuah file SWF yang berisi pesan rahasia tapi tersamarkan dengan adanya pesan lainnya sehingga tidak menimbulkan kecurigaan.

Dengan teknik ini, penulis dapat menyembunyikan berbagai macam bentuk objek, seperti gambar, teks, animasi, ataupun video.

Proses ekstraksi pesan rahasia pun melibatkan kakas flasm dalam membongkar file SWF dan menghasilkan file kode-kode berisi "bytecode". Selanjutnya, akan dilakukan proses modifikasi kode "bytecode".

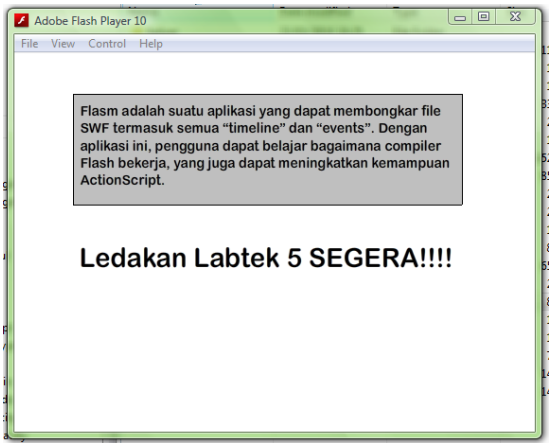
Proses selanjutnya, dicari baris kode yang memuat perubahan transparansi objek MovieClip dengan mencocokkan nama objek MovieClip yang memuat pesan rahasia. Di sini terlihat bahwa nama objek MovieClip yang memuat pesan rahasia tersebutlah yang menjadi kunci dari teknik steganografi ini.

```
  defineMovieClip   5   //   total
frames: 1
  end // of defineMovieClip 5

  defineMovieClip   6   //   total
frames: 1
    frame 0
      stop
      push 'this'
      getVariable
      push 'steganografi'
      getMember
      push '_alpha', 0.0
      setMember
    end // of frame 0
  end // of defineMovieClip 6
end
```

Setelah menemukan baris kode yang mengatur transparansi objek MovieClip tersebut, proses selanjutnya adalah mengubah nilai dari "_alpha" yang ada menjadi 100.

Proses terakhir adalah menyusun ulang file yang berisi kode "bytecode" yang sudah dimodifikasi menjadi file SWF dengan bantuan kakas flasm kembali.



Gambar 12. Flash dengan pesan rahasia yang terungkap

2.3. Algoritma *Zero Opacity*

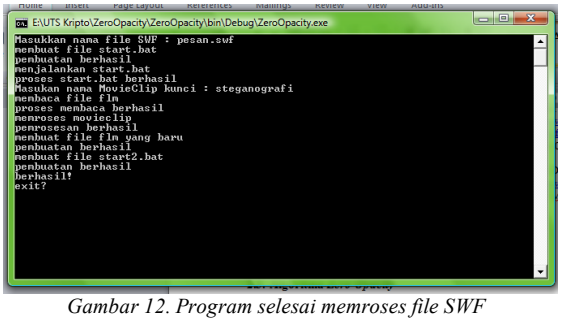
Teknik ini memiliki proses yang sulit dalam mencari nama objek MovieClip yang memuat pesan rahasia di antara banyaknya kode-kode "bytecode" yang ada.

Oleh karena itu, selain merancang teknik steganografi ini, penulis juga membuat suatu program sederhana yang berbasis baris perintah (*command line*). Program ini menerima masukan berupa nama file SWF dan nama objek MovieClip yang memuat pesan rahasia. Program ini pun dapat menjalankan kakas flasm dengan otomatis sehingga pengguna tidak perlu pusing menjalankan program flasm.

Gambar 11.Program memroses file SWF

Proses yang dijalankan pada program yang dibuat antara lain:

1. Menerima masukan berupa nama file SWF
2. Menjalankan flasm untuk memperoleh file "bytecode" dari file SWF yang diacu
3. Membuka file "bytecode" dan memasukkan isi file ke dalam suatu variabel sementara
4. Meminta masukan berupa nama objek MovieClip yang memuat pesan rahasia
5. Mencari baris kode yang memuat pengaturan transparansi objek MovieClip yang memuat pesan rahasia
6. Mengubah nilai transparansi objek MovieClip dari "0.0" menjadi "100"
7. Menyimpan hasil modifikasi dan menjalankan kakas flasm untuk menyusun ulang file SWF



Gambar 12. Program selesai memroses file SWF

Program yang dirancang masih berupa program yang sederhana tetapi memiliki fungsionalitas meningkatkan optimalitas dalam proses ekstraksi pesan rahasia dengan metode *Zero Opacityi* ini.

Batasan dari program yang dibuat antara lain, program harus dijalankan satu direktori dengan kakas flasm dan file SWF yang ingin diungkap pesan rahasianya.

3. Hasil dan Pembahasan

Agar pesan rahasia tidak dapat diungkap dengan mudah, maka pihak yang mengirimkan pesan rahasia tersebut harus menyampaikan nama objek MovieClip yang menjadi kunci ke penerima saja. Proses pengiriman kunci pun dapat dikreasikan sesuai dengan pihak yang terlibat. Proses kriptografi pun dapat digunakan untuk mengirimkan kunci ini ke pihak penerima. Bahkan, kunci pun bisa diselipkan sebagai salah satu pesan yang terlihat namun ditempatkan tersembunyi dan tidak mencurigakan.

Seperti yang yang dipaparkan sebelumnya, pihak penerima cukup mengetahui kunci jika ingin mengungkap pesan rahasia yang ada pada file SWF tersebut. Selanjutnya pihak penerima hanya cukup menjalankan program yang dibuat oleh penulis ataupun melakukan langkah-langkah ekstraksi pesan seperti yang dipaparkan sebelumnya.

Algoritma yang penulis rancang pada makalah ini hanyalah prototipe fondasi dari metode steganografi *Zero Opacity*. Untuk mencapai tingkat keamanan yang lebih baik, algoritma ini dapat dikolaborasikan dengan algoritma steganografi lainnya serta metode-metode kreatif dari pihak yang akan memanfaatkan algoritma ini.

Selain itu file yang dihasilkan adalah file SWF yang membutuhkan sebuah Flash Player agar dapat dimainkan. Yang perlu disoroti di sini adalah kompabilitas file SWF dengan Flash Player yang dimiliki antar pihak pengirim dan pihak penerima. Hal ini dikarenakan ketika membuat suatu file SWF, diatur agar file tersebut dapat dimainkan di Flash Player versi tertentu. Sehingga, perlu diadakan perjanjian awal dalam menentukan kompabilitas file SWF dan Flash Player yang dimiliki oleh pihak pengirim dan pihak penerima.

Penulis berencana mengembangkan mekanisme yang lebih baik dan lebih optimal agar integritas metode *Zero Opacity* yang ada dapat meningkat.

4. Simpulan

Berikut ini adalah beberapa simpulan yang dapat ditarik dari pemaparan-pemaparan yang ada pada makalah ini:

1. Algoritma Zero Opacity ini cukup menarik untuk digunakan sebagai salah satu alternatif di dalam mekanisme penyembunyian pesan berupa gambar, ataupun teks di dalam file flash shockwave. Bahkan algoritma ini dapat

digunakan untuk menyembunyikan pesan berupa animasi dalam flash.
2. Untuk meningkatkan keamanan dari metode ini, algoritma metode ini dapat dikolaborasikan dengan berbagai algoritma enkripsi pesan lainnya.
3. Program yang dibuat dengan metode steganografi ini hanya terbatas pada file flash Shockwave dengan bahasa ActionScript 2.0. Hal ini dikarenakan decompiler yang digunakan hanya dapat mendukung bahasa ActionScript 2.0.

5. Daftar Pustaka

[1] Bahan Kuliah IF3054
Munir, Rinaldi. 2010. Bahan Kuliah IF3054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Dokumentasi Flash
http://www.adobe.com/support/documentation/en/flash/ (diakses pada tanggal 23 Maret 2010 pada pukul 19:53 WIB)

[3] Format File Flash SWF
http://www.the-labs.com/MacromediaFlash/SWF-Spec/SWFfileformat.html (diakses pada tanggal 3 Maret 2010 pada pukul 19.47 WIB)

[4] Tools Decompiler Flasm
http://www.nowrap.de/flasm.html (diakses pada tanggal 3 Maret 2010 pada pukul 19.47 WIB)