Penggunaan Teknik Obfuskasi Dengan Algoritma obfLBS Untuk Mengacak Kode Sumber File SWF

Wawan Wardiana
Pusat Penelitian Informatika – LIPI
wawan@informatika.lipi.go.id

Ana Heryana
Pusat Penelitian Informatika – LIPI
aheryana@informatika.lipi.go.id

Abstract

Duplicated or modified source code by crackers or competitors is often done for software piracy. The obfuscation technique can be used to prevent it. It works by changing or randomizing the source code of an application software while still maintaning semantic with various algorithms. This study uses an obfLBS algorithm, this algorithm is the change of variable that exist in the program with he numbers. In this paper, obfuscation technique is implemented to a flash program, namely Small Web Format files (swf extension). Using the Java programming language version 1.6 and Netbeans IDE 6.5. This software aims to randomize the source code of the swf files with the first related to disassemble the file, and then it re-assembled into swf file extesion, so that the source code is protected. Results of testing the swf file generated after the obfuscating process have the same quality as the previous swf file.

Keywords : Obfuscation, Randomizing, Source Code, obfLBS Algorithm, SWF File

Abstrak

Pembajakan terhadap perangkat lunak sering dilakukan oleh cracker maupun para pesaing untuk digandakan maupun diubah kode sumbernya. Teknik yang bisa digunakan untuk mencegah hal tersebut adalah dengan teknik obfuskasi. Cara kerjanya dengan mengubah / mengacak kode sumber suatu aplikasi namun tetap memelihara semantiknya, dalam mengacak kode sumber tersebut dapat menggunakan berbagai algoritma. Penelitian ini menggunakan algoritma obfLBS, obfuskasi point fungsi yaitu mengubah variabel yang ada pada program dengan bilangan hexa . Pada tulisan ini teknik obfuskasi diimplementasikan pada program flash, yaitu file Small Web Format (berekstensi swf). Perangkat lunak dengan menggunakan bahasa pemrograman Java versi 1.6 dan IDE Netbeans 6.5. Perangkat lunak ini bertujuan untuk mengacak kode sumber dari file ekstensi swf, dengan terlebih dahulu me-disassemble file terkait, kemudian me-assemble ulang menjadi file ekstensi swf sehingga kode sumber terlindungi. Secara keseluruhan perangkat lunak yang dibuat berjalan sesuai dengan rancangan. Hasil dari pengujian file swf yang dihasilkan setelah proses obfuscating memiliki kualitas yang sama dengan file swf sebelumnya.

Kata Kunci : Obfuscation, Mengacak, Kode sumber, Algoritma obfLBS, File SWF

1. Pendahuluan

Dewasa ini banyak sekali file analog dalam berbagai aplikasi telah tergantikan dengan file digital, baik yang berbentuk video, audio, image termasuk perangkat lunak. File digital memiliki banyak kelebihan seperti mudahnya digandakan, diubah, disimpan dalam berbagai media elektronik, namun kelebihan tersebut sekaligus menjadi kekurangannya.

Perlindungan terhadap karya cipta digital (video, game, edutainment dan perangkat lunak lainnya) menjadi sangat penting jika dilihat dari dampak kerugian yang dirasakan oleh produsen. Produsen berusaha untuk melindungi karya cipta digitalnya dengan berbagai cara, salah satunya adalah dengan melakukan pengacakan terhadap kode sumbernya. Dari

sekian banyak file digital, file yang kini populer di dunia web adalah flash atau adobe flash (macromedia flash). Flash dapat menampilkan animasi dan interaksi di web, juga dapat dijalankan dengan mudah pada komputer *stand-alone* di berbagai *platform* sistem operasi.

Saat ini Flash tidak hanya sebagai perangkat lunak dengan merek dagang Adobe Flash, tetapi juga merupakan suatu teknik animasi di web. Untuk membuat satu animasi di web dengan format flash tidak harus menggunakan Adobe Flash namun dapat juga menggunakan perangkat lunak lain seperti SwishMax, Vecta 3D, Swift 3D, Amara, Kool Moves dan lain-lain.

2. Teknik Obfuskasi

Menurut Clark Thomborson [1] "*Obfuscation is a semantics-preserving transformation of computer code that renders it more secure against confidentiality attacks*". Beberapa aspek yang seharusnyanya tetap menjadi rahasia dalam sebuah aplikasi komputer yaitu :

a. Algoritma, sehingga kompetitor tidak dapat membangun hal yang sama kecuali membangun dari awal.
b. Constant, seperti kunci enkripsi.
c. Fungsi internal yang penting, seperti fungsi untuk mengecek lisensi "if (not licensed) exit()".
d. Antarmuka eksternal, untuk menolak akses dari penyerang dan kompetitor sehingga dapat masukan dari "pintu belakang" atau "lubang".

Cara-cara untuk meng-obfuskasi perangkat lunak dapat digolongkan menjadi :

a. Obfuskasi Leksikal :
   Kacaukan nama *variable*, *constant*, *method*, *class*, antar muka dan sebagainya.
b. Obfuskasi Data :
   Kacaukan nilai *variable* (misalnya dengan meng-kode *Boolean* menjadi int, meng-kode int pada flat, meng-kode nilai-nilai pada *graph*).
c. Obfuskasi Kendali :
   Kacaukan penyataan-pernyataan kendali (if, while, for).

Pada intinya *obfuscation* digunakan untuk mencegah *reverse engineering*. Teknik *obfuscation* umumnya mengubah sintaks skrip tanpa mengubah semantiknya sehingga walaupun tulisan skrip menjadi susah untuk dibaca namun ketika dieksekusi masih tetap dapat dijalankan seperti sebelum di-obfuskasi.

Macam – macam Algoritma Obfuskasi yang dapat digunakan, menurut Christian Collberg [2] algoritma-algoritma yang dapat digunakan untuk obfuskasi ini adalah Algoritma obfLBS : yakni obfuskasi dengan Point Fungsi, Algoritma obfNS: yakni obfuskasi *Database,* Algoritma obfPP: yakni Enkripsi Homomorphic, dan Algoritma obfCEJO: yakni algoritma Whitebox DES.

Pada penelitian ini teknik obfuskasi menggunakan algoritma obfLBS yakni dengan cara mengubah atau menggganti variabel-variabel yang ada pada *action script* file SWF dengan nilai-nilai numerik (hexa). Di bawah ini salah satu contoh aplikasi program yang menggunakan password di dalam kode sumbernya kemudian password tersebut diubah menjadi bilangan-bilangan hexa agar tidak mudah diketahui, seperti terlihat di bawah ini.

```
boolean isValidPassword(String
password){
    if
(password.equals("yellowblue"))
       then return true;
       else return false;
    }
```

Menjadi

```
boolean isValidPassword_PF(String
password){
if
(sha1(password).equals("642fb031ad5e9
46766bc9a25f35dc7c2"))
then return true;
else return false;
}
```

3. Metode Penelitian

Dalam penelitian ini metodologi yang digunakan adalah *object oriented* dengan UML *(Unified Modelling Language)* sebagai model visualisasinya saat melakukan analisis dan perancangannya. UML adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, analisis dan perancangan serta dokumentasi sistem piranti lunak [3]. Dengan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dan aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

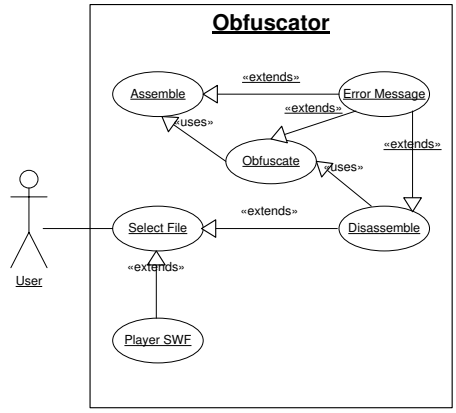
Metode yang digunakan dalam pengembangan perangkat lunak ini adalah dengan menggunakan *Software Development Life Cycle* (SDLC) *Waterfall* [4]. Berikut adalah langkah-langkah yang ada pada penelitian ini:

a. Diawali dengan menganalisis kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk perangkat lunak obfuscator.
b. Dilanjutkan dengan analisis sistem, kemudian dilanjutkan dengan perancangan perangkat lunak dengan mengacu pada kebutuhan-kebutuhan sistem sebelumnya dengan menggunakan algoritma obfLBS untuk mengubah atau mengganti variabel-variabel yang ada dalam sebuah skrip dengan bilangan-bilangan hexa.
c. Implementasi sistem, diimplementasikan dengan menggunakan bahasa pemrograman, kemudian dilakukan pengujian terhadap hasil implementasinya.

Perangkat lunak ini dikembangkan dengan tujuan untuk dapat mengacak kode sumber dari file ekstensi swf, dengan terlebih dahulu me-*disassemble* file terkait. Kemudian me-*assemble* ulang menjadi file ekstensi swf. Pengembangan perangkat lunak ini juga bertujuan untuk ketahanan terhadap proses *reverse engineering* dan *cracking*. Perangkat lunak ini diberi nama Obfuscator.

4. Hasil dan Pembahasan

Perangkat lunak Obfuscator SWF ini memiliki empat proses utama yaitu: *disassemble*, *obfuscate*, *assemble* dan *playerSWF*. *Aktor* pada perangkat lunak ini ada satu. *Aktor* ini akan melakukan obfuskasi ataupun *playing* pada file data yang dimiliki. Seperti terlihat pada gambat 1 di bawah ini.



Gambar 1 Use Case Diagram Obfuscator SWF

4.1. Perancangan Sistem

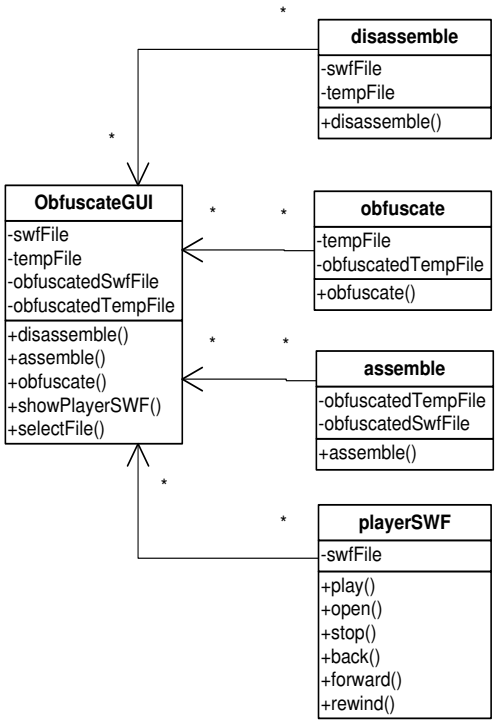
Perangkat lunak Obfuscator SWF dirancang dengan menggunakan beberapa kelas. Kelas-kelas tersebut sebagai berikut :

1. Disassemble
   Kelas ini berfungsi untuk melakukan proses disassemble pada file ekstensi swf yang telah dipilih pengguna.
2. Assemble
   Kelas ini berfungsi untuk melakukan proses assemble ulang pada file yang telah diacak menjadi file ekstensi swf yang terproteksi.
3. Obfuscate
   Kelas ini berfungsi untuk mengacak file file hasil disassemble.
4. Obfuscator GUI

Kelas ini berfungsi untuk mempermudah interaksi pengguna dengan perangkat lunak Obfuscator SWF ini.

5. Player SWF
   Kelas ini berfungsi untuk menjalankan / eksekusi file ekstensi swf.

Penggambaran class-class yang ada pada perancangan perangkat lunak ini seperti terlihat pada gambar 2 di bawah ini

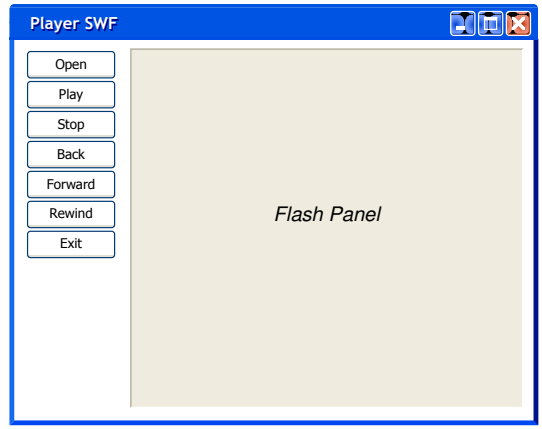


Gambar 2. Class Diagram Obfuscator SWF

Berikut ini adalah layar saji yang dirancang untuk memudahkan interaksi antara perangkat lunak dengan pengguna. Rancangan layar saji utama dapat dilihat pada Gambar 3. Layar saji utama tersebut digunakan untuk proses utama yaitu *obfuscation*. Layar ini dirancang untuk memberikan kemudahan kepada pengguna dalam menggunakan perangkat lunak Obfuscator SWF ini. Di dalam tampilan tersebut disediakan pula pilihan untuk menyajikan tampilan berikutnya yang sesuai dengan pilihan pengguna.

Layar saji *Player* ini digunakan sebagai antarmuka untuk melakukan memainkan file ekstensi swf. Tampilan layar saji *Player* ini dapat dilihat pada Gambar 4.

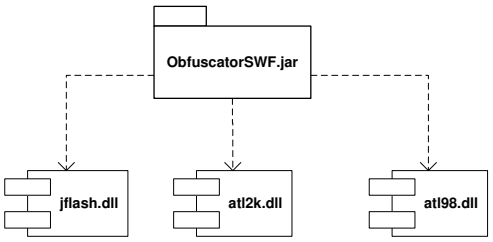
Gambar 3. Desain Layar Saji Utama



Gambar 4. Desain Layar Saji Player

Bahasa pemrograman yang digunakan untuk membangun Obfuscator SWF adalah Java versi 1.6. Sedangkan kompilator sekaligus IDE yang digunakan untuk memudahkan pengembangan perangkat lunak adalah NetBeans 6.5, karena penggunaannya cukup mudah dan memiliki fitur-fitur yang lengkap.

Aplikasi Obfuscator SWF memerlukan 3 komponen yaitu jflash.dll, atl2k.dll dan atl98.dll. Ketiga komponen tersebut digunakan dalam kelas Player SWF, yang berfungsi untuk *environment* flash. Seperti terlihat pada gambar 5.



Gambar 5. Component Diagram Obfuscator SWF

4.2. Pengujian

4.2.1. Bentuk Pengujian

Untuk mendapatkan perangkat lunak yang sesuai dengan kebutuhan, maka harus dilakukan pengujian-pengujian yang relevan. Pengujian yang dilakukan pada penelitian ini baru sebatas pengujian blackbox dalam beberapa tahapan, meliputi pengujian kebenaran dan pengujian kinerja. Berikut akan disajikan masing-masing pengujian dan hasil pengujiannya.

a. Pengujian Kebenaran

Pengujian kebenaran perangkat lunak dilakukan untuk mengetahui kebenaran program dengan menjalankan file SWF yang belum dan sudah di obfuskasi. Tahapan pengujiannya sebagai berikut :

1. Mainkan (Play) file ekstensi swf sebelum diobfuskasi. Langkah pengujiannya adalah sebagai berikut :
   a. Open nama file ekstensi swf yang akan diobfuskasi.
   b. Play file ekstensi swf.
2. Obfuskasi file ekstensi swf yang dipilih. Langkah pengujiannya adalah sebagai berikut :
   a. Open nama file ekstensi swf yang akan diobfuskasi.
   b. Lakukan proses obfuskasi.
3. Play file ekstensi swf yang telah diobfuskasi. Langkah pengujiannya adalah sebagai berikut :
   a. Open nama file ekstensi swf yang telah diobfuskasi.
   b. Play file ekstensi swf tersebut.

b. Pengujian Kinerja

Pengujian kinerja menggunakan file ekstensi swf seperti yang terlihat pada Tabel 1. File ekstensi swf tersebut dipilih berdasarkan variasi ukuran file yang dimiliki.

Tabel 1 File ekstensi swf yang diujikan

No	Nama File (swf)	Tipe	Ukuran (KB)
1	Deal_or_no_deal	Game	494
2	Fast_and_the_furious	Game	292
3	Video_1	Video	39
4	Video_2	Video	7.105
5	Ruralracer	Game	306

Pengujian kinerja perangkat lunak untuk mengukur:

1. Kecepatan pengacakan *action script*.
2. Kualitas gambar dan suara file ekstensi swf yang telah diobfuskasi dibandingkan dengan file sebelum diobfuskasi.

4.2.2. Hasil Pengujian

Masing-masing pengujian dilaksanakan minimal sebanyak satu kali dan hasilnya dicatat. Apabila pada sebuah kasus uji terdapat hasil yang meragukan, maka kasus uji tersebut akan diulang kembali proses pengujiannya. Untuk suatu kasus uji yang pelaksanaan pengujiannya dilakukan berulang kali, maka hasil pengujian yang dicatat hanyalah hasil pengujian yang terbaik.

a. Hasil Pengujian Kebenaran

Hasil pengujian kebenaran perangkat lunak dapat dilihat pada Tabel 2. Keberhasilan proses memainkan file ekstensi swf dinilai dari dapat tidaknya file ekstensi swf tersebut dimainkan, sedangkan keberhasilan proses obfuskasi dinilai dari perbandingan *action script* hasil obfuskasi dengan *original* file.

Tabel 2 Hasil pengujian kebenaran Obfuscator SWF

No.	Proses	Hasil
1	Memainkan file ekstensi swf sebelum diobfuskasi	Berhasil
2	Proses obfuskasi pada sebuah file ekstensi swf	Berhasil
3	Memainkan file ekstensi swf setelah diobfuskasi	Berhasil

b. Hasil Pengujian Kinerja

Hasil pengujian kinerja perangkat lunak dapat dilihat pada Tabel 3. Penjelasan parameter tabel tersebut yaitu :

No. : Nomor file yang diuji sesuai dengan Tabel 5.1
Waktu obfuskasi : Lama proses obfuskasi dalam satuan milidetik
Kualitas gambar : Kualitas gambar yang ditampilkan
Kualitas suara : Kualitas suara yang dihasilkan

Tabel 3 Hasil pengujian kinerja Obfuscator SWF

No.	Waktu Obfuscating	Kualitas Gambar	Kualitas Suara
1	1828 ms	Sama	Sama
2	1984 ms	Sama	Sama
3	2734 ms	Sama	Sama
4	3500 ms	Sama	Sama
5	2578 ms	Sama	Sama

Perangkat lunak Obfuscator SWF secara umum bekerja relatif baik. Kinerja yang dihasilkan oleh perangkat lunak ini juga relatif baik, ditinjau dari kecepatan proses obfuskasi dan file ekstensi swf yang dihasilkan.

5. Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Perangkat lunak yang dibangun dapat mempersulit atau mencegah aksi *cracking* dan *reverse engineering* dengan melakukan pengacakan / *obfuscation* pada kode sumber file swf.
2. Ukuran file sebelum dan sesudah proses obfuskasi relatif berubah karena adanya perubahan *variable*.
3. Kualitas gambar dan suara yang dihasilkan saat memainkan file swf yang belum dan sudah diobfuskasi hasilnya tidak ada perbedaan.

6. Ucapan Terima Kasih

Penelitian ini didanai oleh Program Insentif KNRT Tahun 2009. Pada kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada Sdr. Sudino Asih yang telah membantu penulis dalam penelitian ini.

7. Daftar Pustaka

[1] Collberg Christian and Jasvir Nagra. 1st edition., *Surreptitious Software: Obfuscation, Watermarking and Tamperproofing for Software Protection*, Addison Wesley, 2009

[2] Thomborson Clark. *Methods for Software Protection, Keynote Address on International Forum on Computer Science and Advanced SoftwareTechnology*, Jianxi Normal University, 2007

[3] Lethbridge C. Timothy and Laganiere Robert. *Object-Oriented Software engineering: Practical software development using UML and Java,* McGraw Hill, London, 2001

[4] Pressman, Roger*: Software Engineering A Practitioner's Approach 6th Edition,* McGraw Hill, 2005.