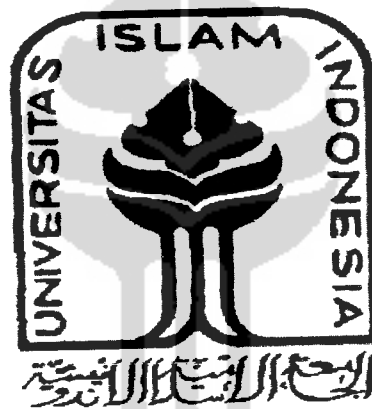


**PERANCANGAN DAN IMPLEMENTASI KEAMANAN
FILE *.SWF DENGAN MENGGUNAKAN
ACTION SCRIPT**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



oleh :
Nama : Andesita Prihantara
No. Mahasiswa : 02523098

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**

LEMBAR PERNYATAAN KEASLIAN

HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

Nama : Andesita Prihantara

No. Mahasiswa : 02 523 098

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

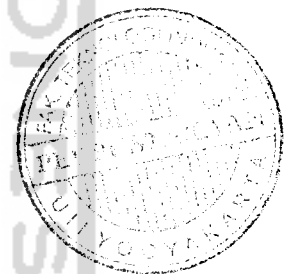
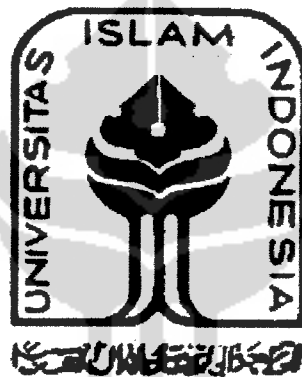
Yogyakarta, Maret 2007

(Andesita Prihantara)

LEMBAR PENGESAHAN PEMBIMBING

PERANCANGAN DAN IMPLEMENTASI KEAMANAN
FILE *.SWF DENGAN MENGGUNAKAN
ACTION SCRIPT

TUGAS AKHIR

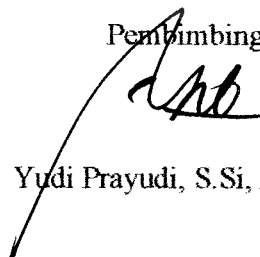


Oleh:

Nama : Andesita Prihantara
No. Mahasiswa : 02 523 098

Yogyakarta,

Pembimbing,


Yudi Prayudi, S.Si, M.Kom.

LEMBAR PENGESAHAN PENGUJI

PERANCANGAN DAN IMPLEMENTASI KEAMANAN FILE *.SWF DENGAN MENGGUNAKAN ACTION SCRIPT

TUGAS AKHIR

oleh:

Nama : Andesita Prihantara
No. Mahasiswa : 02 523 098

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 7 Maret 2007

Tim Penguji

Yudi Prayudi, S.Si, M.Kom

Ketua

Hendrik, ST

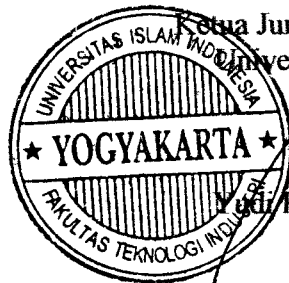
Anggota I

Syarif Hidayat, S.Kom

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia



Yudi Prayudi, S.Si, M.Kom.

HALAMAN PERSEMBAHAN

Kupersembahkan Te A ku ni kepada

Bapak dan Ibuku yang telah memberikan nasehat, semangat, kasih sayang, fasilitas, dan tak henti-hentinya berdoa demi kebaikan ku.



*Yayang-ku Atin yang selalu memberiku dukungan
Semoga tiada kebosanan dalam menghadapiku*

Halaman Motto

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya...”

(Q.S. Al-Baqarah: 286)

“Kemenangan dan keberhasilan hanya dapat dicapai dengan kesabaran”

(HR. At Tirmidzi)

“Allah selalu membantu hambanya, selama hamba itu membantu saudaranya.”

(HR. Bukhari dan Muslim)

“jangan kau sia-siakan waktumu, karena waktu yang berlalu tak kan bisa kembali”

وَلَا تُسْرِفُوا إِنَّهُ يَرْسُدُ السَّرْفَ كَمَا يَرْسُدُ الْوَشْكَ إِنَّهُ كَانَ يُفْسِدُ الْوَشْكَ كَمَا يَفْسِدُ الْبَالِغُ الْوَشْكَ إِنَّهُ كَانَ يُفْسِدُ الْبَالِغُ الْوَشْكَ

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah swt. Tuhan seru sekalian alam. Berkat hidayah dan izin-Nya penyusun dapat menyelesaikan laporan penelitian yang berjudul "PERANCANGAN DAN IMPLEMENTASI KEAMANAN FILE *.SWF DENGAN MENGGUNAKAN ACTION SCRIPT" sebagai salah satu syarat untuk memperoleh gelar sarjana.

Penulisan laporan ini bertujuan untuk mendokumentasikan beberapa hal yang berkaitan dengan penelitian yang dilakukan oleh penyusun, sehingga memungkinkan untuk dikembangkan lebih lanjut di kemudian hari.

Selama proses penyusunan berlangsung, penulis mendapat banyak dukungan dan bantuan dari berbagai pihak, sehingga penelitian ini dapat terlaksana sesuai dengan yang diharapkan. Oleh karena itu penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Bapak Yudi Prayudi, S.Si, M.Kom., yang telah membimbing penulis dari awal sampai akhir pelaksanaan tugas akhir.
2. Bapak Yudi Prayudi, S.Si, M.Kom., selaku Ketua Jurusan Teknik Informatika.
3. Bapak Fathul Wahid, ST., MSC., selaku dekan Fakultas Teknologi Industri.

4. Bapak Prof. Dr. Edy Suandi Hamid, M.Ec., selaku Rektor Universitas Islam Indonesia.
5. Bapak dan Ibu di Cilacap, terima kasih atas motivasi dan doa yang selama ini selalu tercurah.
6. Adik-adikku yang kusayang, Rizki dan Kelik.
7. Yayangku Atin di Cilacap yang setia menungguku dan mendukungku sepenuhnya.
7. Teman-teman seperjuanganku semasa kuliah, agung, ayong, dana, hendra dan semua teman-teman angkatan 2002, terima kasih atas semua bantuannya.
8. Teman-teman kos, mas Roby, mas Rifki, Gatot, mas Hendro, Rivan, Agung, Priyo dan Banu, yang selalu memberikan bantuan materi maupun non-materi.

Penulis menyadari bahwa masih banyak kekurangan dalam penelitian maupun penyusunan laporan ini, untuk itu penulis mengharapkan kritik dan saran yang membangun agar penelitian ini menjadi lebih berguna di masa yang akan datang.

Akhirnya, penulis berharap semoga penelitian ini dapat bermanfaat serta memperkaya khazanah ilmu pengetahuan di Indonesia.

Wassalam

Yogyakarta, Februari 2007

Penyusun

SARI

Menciptakan karya animasi dua atau tiga dimensi tidak mudah, selain dibutuhkan kemampuan dalam menggunakan perangkat lunak untuk animasi juga dibutuhkan kreatifitas yang tinggi. Seseorang yang mampu menciptakan sebuah animasi tentulah orang yang kreatif dan mempunyai jiwa seni. Pada penelitian ini mengamankan file *.swf bertujuan agar file tersebut tidak dapat di *compile* ulang ke bentuk *.fla, sehingga apabila dalam bentuk *.fla sebuah file animasi dua dimensi dengan menggunakan perangkat lunak Macromedia Flash MX dapat diedit dan dihapus.

Kata kunci : *compile* ulang, animasi



TAKARIR

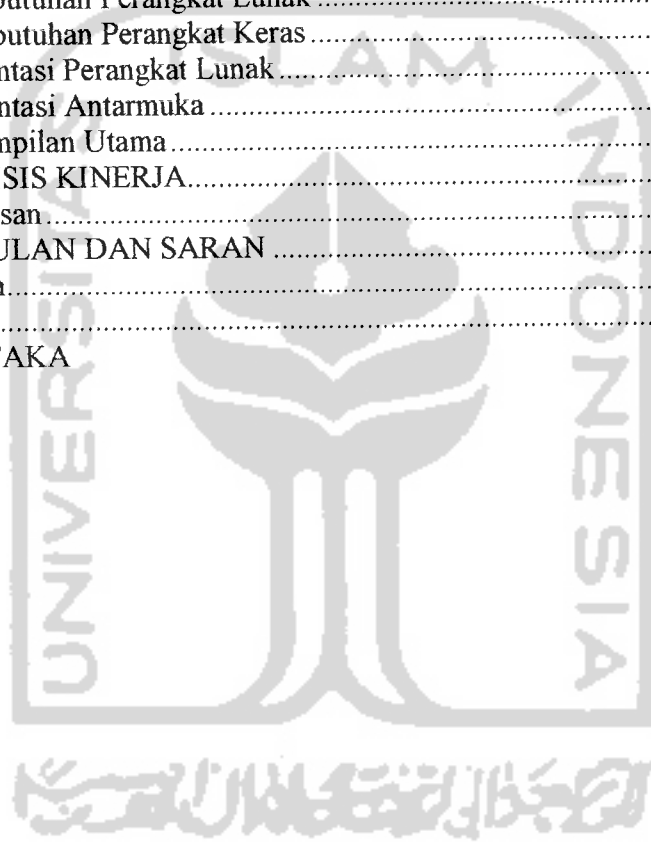
attribute	sifat, perlengkapan
children	anak
execute	menjalankan
flowchart	diagram alir
hardware	perangkat keras
interface	antarmuka
movie	gambar hidup
parent	orang tua, induk
path	garis edar
reference	referensi, rekomendasi
root	akar
simulation test	tes tiruan
software	perangkat lunak
user	pemakai
visible	tampak



DAFTAR ISI

JUDUL	i
LEMBAR PERNYATAAN KEASLIAN	ii
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
SARI	ix
TAKARIR	x
DAFTAR ISI	xi
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
1.7 Sistematika Penelitian	5
BAB II LANDASAN TEORI	7
2.1 Tinjauan pustaka	7
2.1.1 Pengenalan Action Script	8
2.1.2 WIN32 – API dan Windows Registry	19
2.1.3 Pengenalan Visual Basic	24
BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK	30
3.1 Metode analisis	31
3.2 Metode Pengumpulan Data	32
3.2.1 Pengumpulan Data Primer	32
3.2.2 Pengumpulan Data Sekunder	32
3.3 Hasil Analisis	33
3.3.1 Masukan (Input)	36
3.3.2 Proses	37
3.3.2.1 Browse File ke Sistem	37
3.3.2.2 Manipulasi File	37
3.4 Keluaran (Output)	37
3.5 Antarmuka Sistem	38

BAB IV PERANCANGAN PERANGKAT LUNAK	39
4.1 Metode Perancangan	39
4.2 Prosedur Sistem	40
4.3 Hasil Perancangan	40
4.3.1 Desain Arsitektur Perangkat Lunak	41
4.3.2 Analisis Software SoThink SWF Decompiler	44
4.3.3 Desain Antarmuka	50
BAB V IMPLEMENTASI	52
5.1 Batasan Implementasi	52
5.1.1 Kebutuhan Perangkat Lunak	52
5.1.2 Kebutuhan Perangkat Keras	53
5.2 Implementasi Perangkat Lunak	53
5.3 Implementasi Antarmuka	54
5.3.1 Tampilan Utama	54
BAB VI ANALISIS KINERJA	60
6.1 Pembahasan	60
BAB VII SIMPULAN DAN SARAN	66
7.1 Simpulan	66
7.2 Saran	66
DAFTAR PUSTAKA	



DAFTAR GAMBAR

Gambar 2.1 Flowchart if ... else action	13
Gambar 2.2 Movie clip dengan method drag.....	14
Gambar 2.3 Flash Interface	16
Gambar 2.4 Keyframe mengindikasikan ke sebuah frame action.....	18
Gambar 2.5 Path salah satu file .dll (user32)	20
Gambar 2.6 Struktur Hieraki database Registry.....	24
Gambar 3.1 File methods drag movie clip.swf yang akan di compile ulang	33
Gambar 3.2 File .swf yang siap di edit.....	34
Gambar 3.3 Hasil file yang telah di edit.....	35
Gambar 3.4 Hasil dari proses manipulasi.....	36
Gambar 4.1 Flowchart secara umum.....	41
Gambar 4.2 Flowchart eksekusi file.....	42
Gambar 4.3 Tampilan utama dari SoThink SWF Decompiler.....	45
Gambar 4.4 Desain antarmuka	51
Gambar 5.1 Tampilan utama	54
Gambar 5.2 Tampilan saat tipe file telah dipilih.....	55
Gambar 5.3 Dialog box yang muncul saat tombol browse ditekan	56
Gambar 5.4 Detail file yang ditampilkan	56
Gambar 5.5 Ukuran file setelah manipulasi	57
Gambar 5.6 Tampilan help.....	58
Gambar 6.1 Pesan saat tombol cancel pada kotak dialog	62
Gambar 6.2 Tampilan saat tombol open ditekan.....	63
Gambar 6.3 Tampilan saat user salah memilih file.....	63
Gambar 6.4 Hasil sebelum program dijalankan.....	64
Gambar 6.5 Hasil program setelah program dijalankan.....	65

DAFTAR TABEL

Tabel 4.1 Daftar perintah fscommand.....	49
--	----



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Multimedia mengalami perubahan yang cepat dan merupakan salah satu skill yang harus dimiliki pada abad ini. Kenyataannya multimedia berubah secara alami, sebagai gantinya pembatasan presentasi dengan menggunakan teks yang dicetak pada kertas, multimedia membuat dinamik presentasi lebih menarik. Semua ini dilakukan tidak hanya menampilkan teks tetapi dengan menambah suara, gambar, musik dan video.

Flash seperti software gado-gado dimana di dalamnya terdapat semua kelengkapan yang dibutuhkan. Mulai dari fitur menggambar, ilustrasi, mewarnai, animasi dan *programming*. Software ini bisa menghasilkan sebuah karya animasi dua dimensi yang indah, kreatif dan sangat dibutuhkan sebagai ajang untuk promosi sebuah perusahaan, organisasi atau perorangan, sehingga karya hasil ini banyak dijual dan ditawarkan di internet yang harganya dengan satuan puluhan bahkan ratusan dolar untuk satu karya animasi yang menggunakan flash.

Saat ini tidak dipungkiri lagi bahwa teknologi informasi semakin berkembang, tidak ada yang tidak mungkin bagi teknologi informasi. Setiap system yang paling sempurna pun tetap memiliki kelemahan. Jika pembuat system tidak dapat mengatakan kelemahan system tersebut maka pihak lain akan mengambil keuntungannya.

Sejak ditemukannya software decompiler flash, karya template animasi flash yang dipasang di internet dapat dengan mudah di ambil dan di edit. Dari sini dapat disimpulkan bahwa yang menggunakan software ini tidak perlu bersusah payah merancang karya animasi dari awal juga tanpa harus membayar karya tersebut.

Penelitian ini mencoba mencari cara bagaimana melindungi beberapa animasi dengan menggunakan Action Script. Action Script adalah bahasa pemrograman pada Flash yang digunakan untuk mengontrol sebuah *movie* dan objek. Contohnya, action script dapat digunakan untuk mengontrol navigasi pada *movie*, membuat interaksi, menggandakan objek dan lain-lain.

L2 Rumusan Masalah

Perancangan dan implementasi keamanan file *.swf dengan menggunakan Action Script. Dari latar belakang masalah di atas, maka topik yang akan di bahas meliputi :

- 1) Bagaimana mengatasi dan melindungi file *.swf agar tidak dapat di-*compile* ulang menggunakan software SoThink SWFDecompiler.
- 2) Bagaimana Action Script pada flash dapat mengatasi masalah tersebut, sehingga karyanya tidak dapat dimanfaatkan atau di-*compile* ulang dengan software decompiler ini.
- 3) Bagaimana menemukan solusi lain dari *development software* seperti Delphi dan Visual Basic agar dapat melindungi file *.swf.

I.3 Batasan Masalah

Batasan masalah pada penelitian ini akan dibatasi pada :

- 1) Animasi yang dipakai untuk eksperimen hanya animasi sederhana atau mengambil dari flash template di internet.
- 2) Penelitian ini dilakukan hanya menggunakan *software decompiler*, yaitu SoThink SWFDecompiler.
- 3) Sistem ini hanya diujikan pada Sistem Operasi Windows XP ServicePack II.
- 4) Penelitian ini menggunakan Visual Basic untuk membuat file *.exe.

I.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari kegiatan penelitian ini adalah sebagai berikut :

- 1) Agar peneliti mempunyai dasar pengetahuan bagaimana mengamankan file *.swf.
- 2) Mengurangi kerugian perancang dan pembuat karya animasi dengan menggunakan Macromedia Flash.
- 3) Memanfaatkan Action Script untuk mencegah *decompiler* flash dapat bekerja sehingga dengan mudah memanfaatkan karya animasi yang telah dibuat.
- 4) Untuk melindungi animasi dari *software decompiler*.

I.5 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini adalah :

- 1) Dapat mengerti lebih dalam Action Script pada Macromedia Flash MX 2004.
- 2) Dapat menguntungkan perancang animasi yang menggunakan flash sehingga tidak ada pihak yang bisa memanfaatkan karyanya secara ilegal.
- 3) Dapat membuat file dalam bentuk *.exe menggunakan Visual Basic 6.0 dengan sederhana.

I.6 Metodologi Penelitian

- 1) Data yang diperlukan
 - a. Dasar pengetahuan tentang Action Script Flash MX 2004.
 - b. Meneliti cara kerja SoThink SWFDecompiler.
- 2) Metode pengumpulan data
Observasi
Pengamatan dilakukan dengan cara mengamati cara kerja perangkat lunak decompiler flash.
- 3) Metode kepustakaan
 - a. Mencari referensi lewat internet.
 - b. Literatur dan buku-buku yang merupakan sumber pengetahuan sebuah dasar pemrograman Action Script Flash MX 2004.

I.7 Sistematika Penulisan

Sistematika di sini maksudnya yaitu memberikan gambaran tentang ringkasan dan penjelasan dari kerja praktek ini, yang terbagi dalam beberapa bab dan subbab, yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Memuat latar belakang penulisan laporan, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian dan metodologi penelitian, serta sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bagian ini memuat teori-teori yang berhubungan dengan penelitian, yaitu mengenai Action Script, Microsoft Visual Basic 6.0 serta WIN 32 API (*Application Programming Interface*).

BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK

Berisi penjelasan mengenai metode apa yang dibutuhkan dalam pembuatan sistem, analisis masukan serta keluaran, analisis perangkat lunak, antarmuka.

BAB IV PERANCANGAN PERANGKAT LUNAK

Bagian ini memuat uraian tentang metode perancangan perangkat lunak yang dipakai. Beberapa metode perancangan yang dapat dipakai adalah *flowchart*.

BAB V IMPLEMENTASI PERANGKAT LUNAK

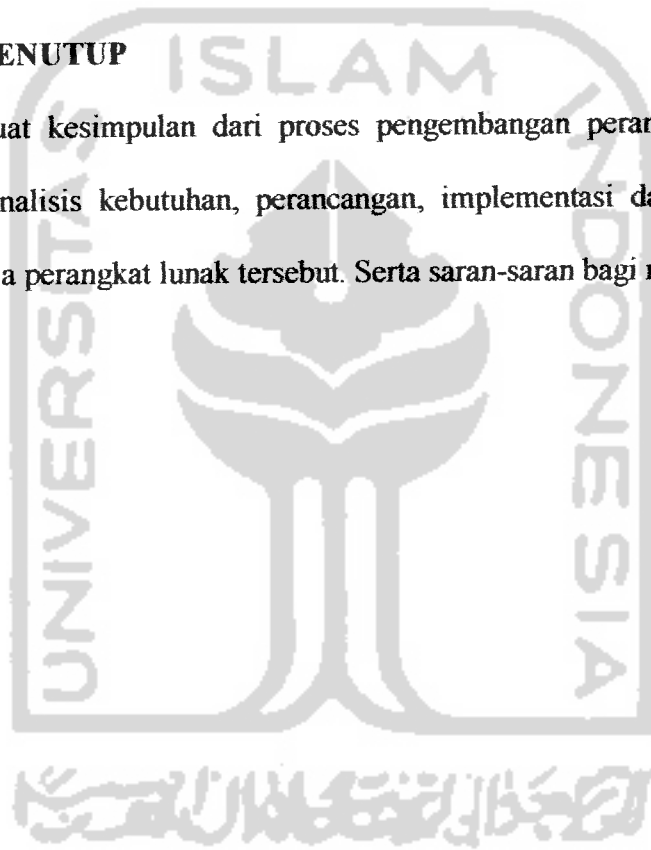
Bagian ini memuat dokumentasi implementasi perangkat lunak, juga berisi keterangan tentang implementasi dari rancangan perangkat lunak, yang meliputi prosedur-prosedur dalam bahasa pemrograman yang dipilih.

BAB VI ANALISIS KERJA PERANGKAT LUNAK

Pengujian terhadap sistem yang telah dibuat termasuk penanganan kesalahan algoritma dan fungsi-fungsi sehingga dapat diperbaiki untuk kesempurnaan sistem. Metode pengujian yang digunakan adalah metode simulation test dimana test dilakukan dengan mencoba meniru keadaan pada lingkungan pada saat sistem digunakan.

BAB VII PENUTUP

Memuat kesimpulan dari proses pengembangan perangkat lunak, baik pada tahap analisis kebutuhan, perancangan, implementasi dan terutama pada analisis kinerja perangkat lunak tersebut. Serta saran-saran bagi mahasiswa.



BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Action Script seperti bahasa pemrograman lainnya yang menggunakan terminologi khusus berdasarkan aturan sintaks. Pada bab ini akan menjelaskan dasar pemrograman Action Script, API (*Application Programming Interface*) dan Visual Basic.

Action Script adalah bahasa pemrograman pada perangkat lunak Macromedia Flash yang dipakai untuk mengontrol movie dan objek. Contohnya, *action script* dipakai untuk navigasi control pada sebuah movie, membuat interaksi, menggandakan objek, mempercantik tampilan mouse dan lain-lain. [SYA05]

Untuk keperluan pengembangan aplikasi berbasis windows yang handal, Microsoft menciptakan sebuah interface pemrograman berupa sekumpulan prosedur dan rutin yang memiliki kemampuan mengakses kekuatan system. Sekumpulan prosedur dan rutin tersebut dikenal dengan istilah *Application Programming Interface* atau yang lebih familiar dengan sebutan API.

Visual Basic merupakan salah satu *development tools* yang dibuat oleh Microsoft. Visual Basic lebih bersifat fleksibel dibanding dengan development tool lainnya, karena Visual Basic tidak terlalu bergantung terhadap fungsi dari API (*Application Programming Interface*) pada windows. API sudah dimasukkan

oleh Microsoft ke dalam Visual Basic sebagai file .dll. Visual Basic juga dibuat dengan interface yang sangat mudah dipahami, sehingga jika orang awam atau yang belum mengerti tentang pemrograman, dapat membuat sebuah project aplikasi yang sederhana. Dengan kata lain interface Visual Basic sangat *userfriendly*. [SHA06]

Program virus yang digunakan bukan untuk merusak system tetapi untuk melindungi file *.swf agar tidak dapat di-compile ulang oleh software decompiler.

2.1.1 Pengenalan Action Script

Perangkat lunak ini digunakan untuk menciptakan karya animasi 2 dimensi yang di dalamnya dapat ditambah suara, membuatnya bergerak, terciptanya interaksi antara manusia dan komputer dan lain sebagainya. Agar sebuah objek dapat bergerak diperlukan bahasa pemrograman dan bahasa pemrograman tersebut adalah action script.

Action Script adalah bahasa pemrograman pada perangkat lunak Macromedia Flash yang dipakai untuk mengontrol movie dan objek. Contohnya, action script dipakai untuk navigasi control pada sebuah movie, membuat interaksi, menggandakan objek, mempercantik tampilan mouse dan lain-lainnya.

[AMS05]

a. Elemen-Elemen Action Script

1. Actions

Pernyataan untuk membuat movie sesuai dengan perintah, contohnya action dibawah ini memberi perintah kepada movie untuk lompat ke frame 10 dan setelah itu movie ditampilkan.

```
gotoAndPlay(10);
```

2. Boolean

Data dengan dua nilai yaitu benar (true) dan salah (false).

3. Classes

Tipe data yang digunakan untuk mendefinisikan tipe baru suatu objek

4. Constant

Elemen yang bersifat mutlak dan tidak dapat dirubah, contohnya kunci

TAB akan selalu mengarah ke tombol Tab pada keyboard.

5. Constructor

Fungsi yang dipakai untuk menggambar property dan class method.

6. Data Types

Sebuah nilai dan operasi yang dipakai untuk tipe data yang terdiri dari string, Boolean, objek, movie clip,function, null dan undefined.

7. Event

Aksi yang terjadi pada saat movie dimainkan, misalnya aksi klik pada button, yaitu release

8. Expression

Kombinasi symbol action script yang merepresentasikan suatu nilai.

Expression dibedakan menjadi dua, yaitu operator dan operand.

Sebagai contoh expression dibawah ini, x dan 5 adalah operand, sedangkan tanda (+) adalah operator.

$$x + 5$$

9. Function

Blok atau kode yang dapat digunakan berulang-ulang, yang juga dapat diikuti parameter dan dapat mengembalikan suatu nilai.

10. Identifier

Nama yang digunakan untuk mengindikasikan variable, objek, function, atau metode. Untuk karakter pertama harus dimulai dengan huruf atau garis bawah (`_`) atau symbol dolar (`$`), sedangkan karakter berikutnya bisa berupa hiruf atau angka.

11. Instance

Objek yang masuk dalam kelas tertentu, sebagai contoh semua movie clip merupakan instance dari kelas movie clip.

12. Method

Fungsi yang diterapkan pada objek.

13. Properties

Atribut yang mendefinisikan suatu objek, sebagai contoh `_visible` adalah property untuk semua movie clip yang menentukan ditampilkannya atau disembunyikannya movie clip di dalam panggung.

14. Variable

Pelacak identitas yang menampung semua tipe data.

b. Menggunakan Perintah Action Script

Action Script mempunyai aturan penulisan dan ketepatan script, aturan tersebut menentukan karakter dan kata tertentu yang digunakan untuk membuat arti dan susunan pada saat dituliskan. Action Script juga bersifat case-sensitive, yang berarti kapitalisasi di dalam penulisan huruf berpengaruh. Sebagai contoh bola dengan Bola didefinisikan sebagai objek yang berbeda. Oleh karena itu, sangat disarankan untuk selalu memperhatikan kapitalisasi penulisan huruf.

1. Dot Syntax

Dalam action script, tanda titik (.) digunakan untuk mengindikasikan properties atau method yang mempunyai hubungan dengan objek atau movie clip. Tanda ini juga untuk mengidentifikasi target path ke sebuah movie clip setelah itu tanda titik (.) dan diakhiri dengan property, method atau variable yang dituju. Contohnya :

```
Berat.ballerina = 50  
Berat.ballerina._x = 50  
Flintstone.berat = 65
```

Pernyataan diatas menunjukkan nilai 50 untuk property berat dari ballerina dan nilai 65 untuk berat dari flintstone.

Syntax titik juga mempunyai dua alias, yaitu `_root` yang menunjuk pada timeline utama dan `_parent` yang menunjuk pada suatu movie clip yang didalamnya tersarang movie clip lain. Kedua alias tersebut bersifat absolute, dengan kata lain membuat target path yang absolute.

Perhatikan contoh sebagai berikut :

```
_root.besar._x =100;
```

contoh movie clip ballerina berada di dalam movie clip berat. Di dalam movie clip ballerina dituliskan kode dibawah ini yang berarti memerintahkan movie clip ballerina untuk berhenti.

```
_parent.stop();
```

2. Menggunakan tanda titik koma

Di dalam aturan bahasa Indonesia, suatu kalimat diakhiri dengan tanda titik, sedangkan di dalam action script, pernyataan diakhiri dengan tanda titik koma (;), seperti contoh kode di atas.

3. Menggunakan tanda kurung kurawal

Pernyataan action script di masukkan ke dalam satu group atau blok yang diawali dengan tanda kurung kurawal buka ({} dan diakhiri dengan tanda kurung kurawal tutup (}).

4. Menggunakan tanda kurung di dalam action script adalah :

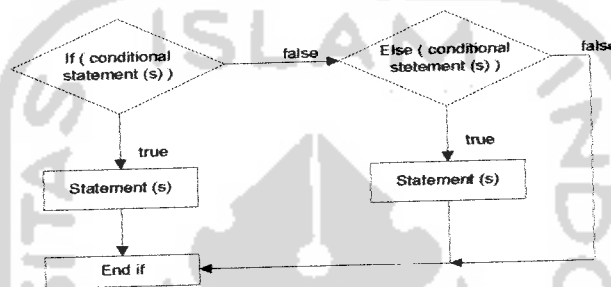
- a) Sebagai tempat parameter pada saat mendefinisikan fungsi.
- b) Sebagai tempat parameter pada saat memanggil fungsi.
- c) Sebagai tempau untuk mengevaluasi ekspresi jika diletakkan sebelum tanda titik.

5. Menggunakan dua tanda garis miring (comments)

Penulisan pernyataan yang diawali dua tanda garis miring akan didefinisikan sebagai komentar atau catatan. Komentar tidak menghasilkan suatu perintah, melainkan hanya sebagai pengingat maksud dan tujuan pernyataan dituliskan.

c. Flowchart Action Script

Flash mengeksekusi statement action script diawali dengan statement pertama dan dilanjutkan berurutan sampai statement terakhir atau sampai dengan yang memberikan instruksi untuk ke movie berikutnya, yaitu statement if, do ... while, loops dan if ... else pada flash.



Gambar 2.1 Flowchart if ... else action

Statement if disebut statement kondisi atau “logical branch” karena mengontrol arus dari script berdasarkan evaluasi dari beberapa kondisi. Contohnya, sebuah kode memeriksa apakah nilai tersebut sama dengan 10, jika pemeriksaan benar maka nilai dari variable sama dengan 10. jika salah variable akan memberikan peringatan yang menampilkan “nilai input tidak sama dengan 10”.

```
if (angka <=10) {  
  alert = "nilai input tidak sama dengan 10";  
}
```

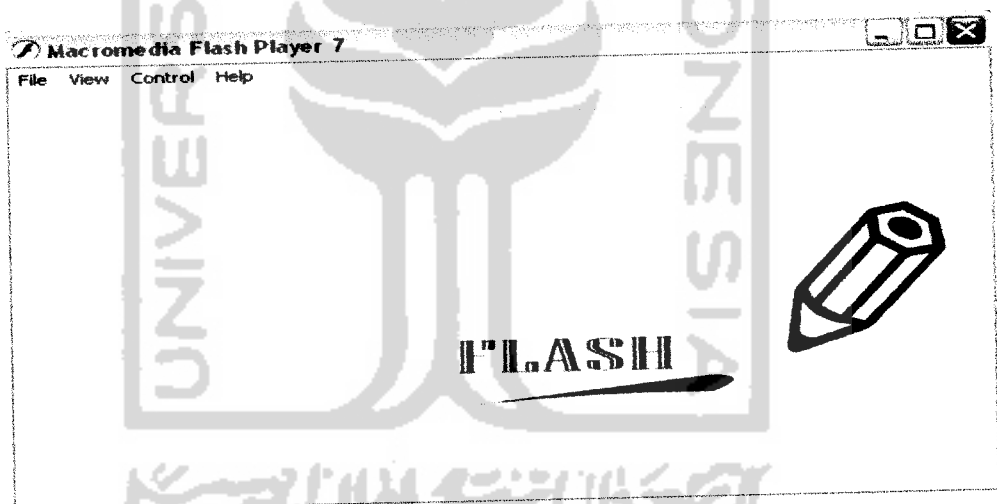
Selain statement di atas dapat ditambahkan statement lain untuk menciptakan statement kondisi yang lebih kompleks. Contoh di bawah ini, jika kondisinya benar (nilai yang dimasukkan adalah 3) statement yang

dijalankan adalah statement diantara kurung kurawal pertama “angka tidak sama dengan 10”. Jika kondisi salah (nilai yang dimasukkan 10), baris pertama di abaikan dan menuju statement selanjutnya. Berikut contoh program kondisi if ... else

```
if (number <= 10) {  
  alert ="the number is less than or equal to 10";  
} else {  
  alert ="the number is greater than 10";  
}
```

d. Membuat Action Script Sederhana

Berikut ini adalah movie yang didalamnya terdapat interaksi yang dapat menyeret (dragging) movie clip ke posisi yang baru.



Gambar 2.2 movie clip dengan method drag

Pada movie diatas animasi bisa diseret ke posisi yang baru.

```
onClipEvent (mouseDown) {  
  this.startDrag(1);  
  this.gotoAndStop(1);  
}
```

Actions di atas berarti jika tombol kiri mouse ditekan, movie clip dapat diseret (methods *startDrag(1)*, parameter 1 berarti pointer akan

berada di posisi center movie clip), dan animasi akan berhenti karena playhead di dalam timeline movie clip diperintahkan melompat ke frame 1 dan berhenti (*gotoAndStop(1)*).

```
onClipEvent(mouseUp) {  
    this.stopDrag( ) ;  
    this.play( ) ;  
}
```

Actions di atas untuk memainkan animasi di dalam movie di posisinya yang baru, jika tekanan pada tombol kiri mouse dilepaskan, penyeretan movie clip akan berhenti, actions methods *stopDrag()*, dan timeline movie clip akan dimainkan, actions methods *play()*.

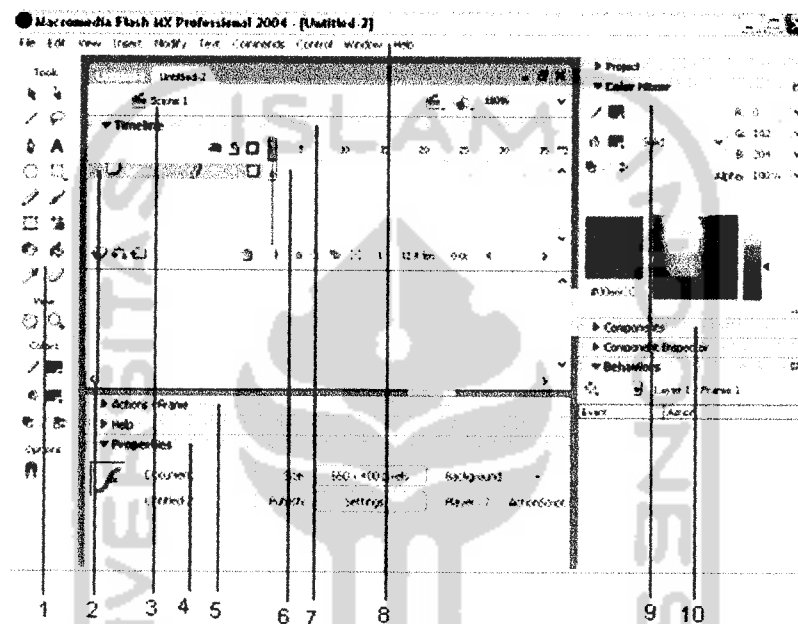
e. Menggunakan Panel Action

Panel Action adalah tempat untuk menulis perintah atau script yang dilengkapi oleh berbagai fasilitas dan kecanggihan khusus. [ZEE 05] Panel action merupakan tempat untuk menciptakan dan mengedit sebuah objek atau frame menggunakan dua jenis model editing yang berbeda. Pada action panel dapat memilih action yang telah disediakan pada tool box, action drag dan drop, juga dapat menggunakan tombol untuk menghapus atau mengatur action-action secara default, panel action ditampilkan di bagian bawah jendela kerja bersama dengan panel properties dan dapat mengatur posisi panel action dengan menyeret symbol di bagian kiri atas panel ke posisi baru. Sedangkan untuk expert action panel dapat menulis dan mengedit action langsung dari text box, lebih banyak menulis script langsung dengan text editor. Untuk menampilkan action panel :

Pilih Window > Development Panel > Actions

Panel action terdiri dari 3 kolom, yaitu :

1. kolom yang berisikan daftar action.
2. kolom tempat menuliskan kode action script.
3. kolom yang berisikan informasi objek, layer dan scene yang terseleksi.



Gambar 2.3 Flash Interface

1. Tool Box : Bagian yang digunakan sebagai alat Bantu dalam menggambar dan memformat gambar.
2. Layer : Bagian yang digunakan untuk mengatursusunan dari objek yang tampak di stage.
3. Scene / Stage : Layar yang digunakan untuk menyusun objek, gambar, teks, animasi dan lain-lain.

4. Panel Properties : Bagian yang digunakan untuk memberikan perintah tambahan dari objek yang sedang dipilih.
5. Panel Action : Bagian yang digunakan untuk memberikan perintah Script pada objek yang sedang dipilih.
6. Frame : Bagian dari Layer yang digunakan untuk mengatur pergerakan animasi.
7. Timeline : Bagian yang digunakan untuk mengatur susunan Layer.
8. Menu : Sekelompok perintah yang digunakan untuk mengatur pembuatan objek, animasi, layer dan lain-lain.
9. Panel Color Mixer : Bagian yang digunakan untuk mewarnai suatu objek, baik warna solid maupun warna gradasi.
10. Components : Suatu tombol yang digunakan untuk menampung data-data.

f. Menuliskan Action pada Frame

Action dapat diterapkan pada frame, button atau movie clip.

Penulisan actions dapat dilakukan dengan memilih objeknya, kemudian melakukan cara sebagai berikut :

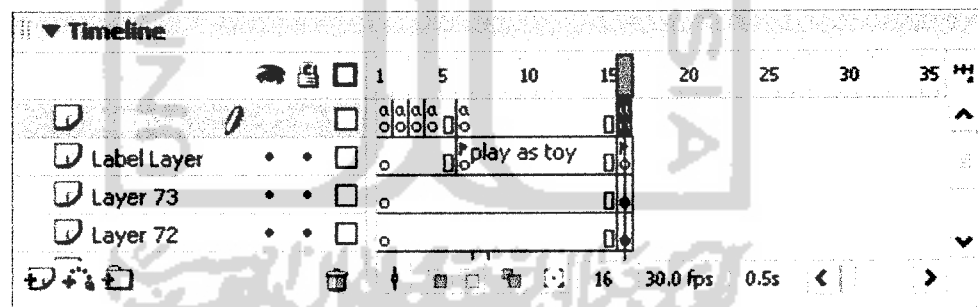
- 1) Pilih sebuah keyframe pada timeline dan pilih Window > Development Tool > Action. Jika frame yang dipilih bukan merupakan keyframe, action mengarah ke keyframe berikutnya.
- 2) Didalam kolom daftar action, klik ganda maka kode action akan ditampilkan didalam kolom penulisan action.

- 3) Didalam kolom penulisan script klik tombol add (+) kemudian klik action didalam menu yang tampil, maka kode action akan tampil didalam penulisan action.
- 4) Didalam kolom penulisan action, ketikkan secara langsung kode action.
- 5) Untuk menjalankan frame action gunakan Ctrl + Enter.

Untuk membuat movie melakukan pada keyframe, dapat dimasukkan frame action ke keyframe. Contohnya, untuk menciptakan looping pada timeline antara frame 20 dan 10, dapat ditambahkan frame action ke frame 20.

`GotoAnd Play (10);`

Sangat disarankan untuk meletakkan frame action di layer terpisah. Frame dengan action script menampilkan huruf a pada timeline.



Gambar 2.4 Keyframe mengindikasikan ke sebuah frame action

2.1.2 WIN32 – API dan Windows Registry

Pada saat Windows diciptakan oleh Microsoft, sesungguhnya Microsoft menempatkan sejumlah besar kode siap pakai pada pustaka sistemnya sehingga dapat diakses oleh programmer lainnya. Kode siap pakai tersebut berupa kumpulan rutin dan fungsi yang dikenal sebagai “API Function”. Programmer Windows menulis rutin dan fungsi dalam bentuk file *.dll* (*dynamic link library*) untuk melakukan tugas-tugas yang bersifat pemeliharaan system dan perintah-perintah computer tingkat rendah (*low-level system task*), agar dapat digunakan oleh berbagai aplikasi yang ter-instal pada Windows.

Terdapat berbagai file *.dll* yang berada pada direktori *%SystemRoot%* dan umumnya berada pada *C:\Windows\System32*. Beberapa diantaranya yang sering digunakan adalah :

a. User32.dll

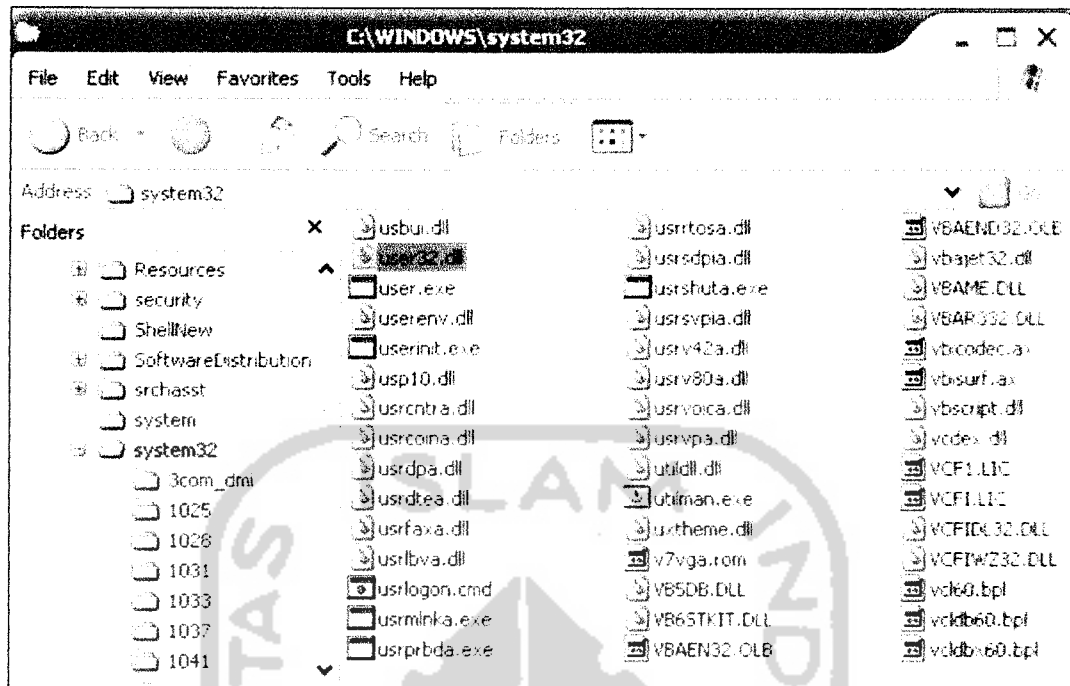
File ini dipakai untuk mengontrol objek-objek yang terlihat pada layar.

b. Gdi32.dll

Merupakan gudang kode pemrograman berorientasi grafik dari API.

c. Kernel32.dll

Di dalamnya terdapat fitur-fitur untuk mengakses sistem operasi tingkat bawah.



Gambar 2.5 Path salah satu file .dll (user32)

Pendeklarasian fungsi API harus ditulis secara utuh dalam satu baris pernyataan. Apabila tidak memungkinkan, dapat dipisah dengan sebuah karakter underscore (`_`) setelah spasi. Deklarasi fungsi dapat dilakukan dengan awalan Private atau Public. Awalan Public selalu di tempatkan dalam sebuah modul (*.bas) dan fungsi yang didklarasikan pada jendela modul selalu bersifat global yang dapat dipakai pada setiap form atau listing program yang lain, sedangkan fungsi yang dideklarasikan pada jendela modul, selalu diawali dengan Private. Sebagai contoh :

```
Public Declare Function ExitWindowsEx Lib "user32" (ByVal _
    uFlags As long, ByVal dwReserved As Long) As Long
```

Deklarasi fungsi seperti di atas dengan awalan Public, selalu ditempatkan pada jendela modul dan bersifat global, sedangkan jika di deklarasikan selain pada

jendela model, maka penulisannya diawali kata `Private` dan pemanggilan fungsi hanya berlaku terbatas pada listing, dimana fungsi tersebut dideklarasikan.

Penulisan nama fungsi atau prosedur bersifat *case sensitive*, artinya Visual Basic membedakan antara penulisan `ExitWindowsEx` dengan `exitwindowsex`. Secara default deklarasi pada jendela modul adalah `Public`. Pendeklarasian fungsi API jika tidak diawali dengan kata `Public` dan hanya diawali kata `Declare` pada jendela modul, maka hal ini tidak akan mengubah apa-apa.

Fungsi-fungsi dasar WIN32-API :

- a. `FindWindow` : berfungsi untuk mencari handle suatu Parent Window.
- b. `FindWindowsEx` : berfungsi untuk mencari handle suatu Child Window.
- c. `GetCursorPos` : berfungsi untuk mendapatkan koordinat cursor pada layar.
- d. `WindowsFromPoint` : berfungsi untuk mendapatkan Window Caption suatu jendela.
- e. `SetWindowText` : berfungsi untuk mengubah Window Caption suatu jendela.
- f. `GetClassName` : berfungsi untuk mendapatkan Class Name dari suatu jendela.
- g. `SendMessage` : berfungsi untuk mengirimkan pesan kepada sebuah jendela untuk melakukan task tertentu.

Untuk dapat menyisipkan sebuah file virus harus dapat mengerti registry pada system operasi Windows. Registry adalah database yang biasanya digunakan untuk menyimpan informasi setting semua hardware, software dan pengguna yang ada di dalam system operasi Windows. Jika user mengubah setting control panel,

system atau software yang telah terinstal akan mempengaruhi dan akan disimpan pada registry. Perangkat lunak ini merupakan free software yang didalamnya terdapat pengetahuan tentang registry windows.

Database registry secara hierarki terdiri dari 5 jenis key dan masing-masing mempunyai fungsi tertentu. Kelima key tersebut adalah :

1. HKEY_CLASSES_ROOT

Terdiri dari peta kumpulan file untuk mendukung drag dan drop, informasi OLE, windows shortcut dan aspek inti dari windows user interface.

2. HKEY_CURRENT_USER

Merupakan tempat penyimpanan seluruh konfigurasi atau user profile, seperti konfigurasi control panel, warna screen, folder tiap user.

3. HKEY_LOCAL_MACHINE

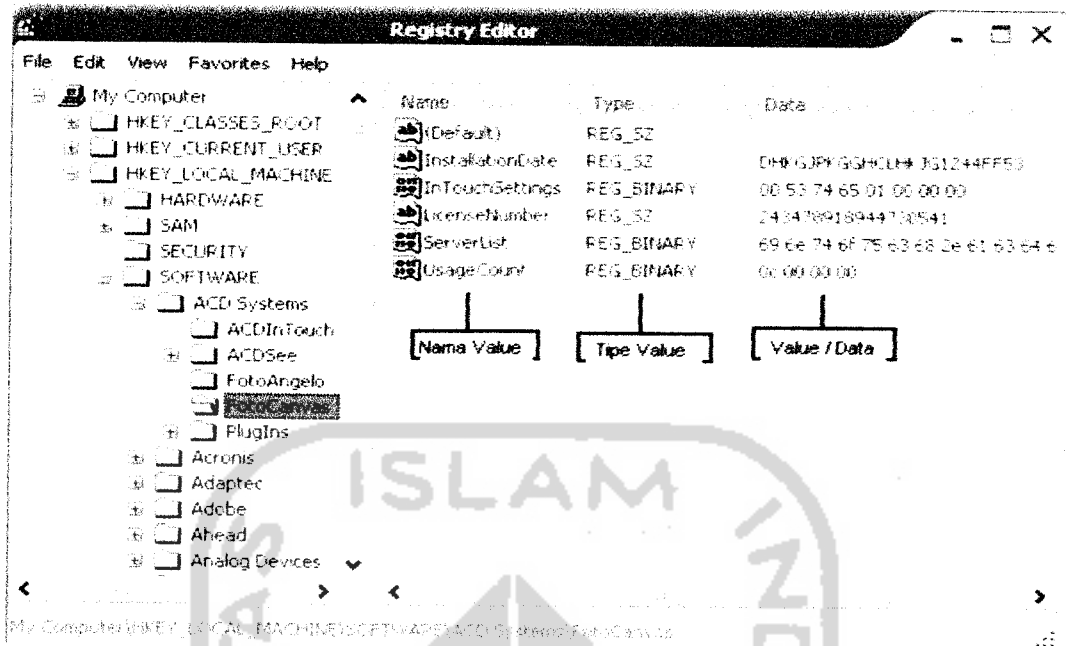
Merupakan kumpulan informasi tentang tipe perangkat keras, perangkat lunak dan preferences lainnya pada komputer, informasi ini dipakai untuk semua pengguna yang login ke komputer tersebut.

4. HKEY_USERS

Merupakan bagian yang terdiri dari preferences untuk masing-masing pengguna komputer, masing-masing pengguna diwakili sub_key SID.

5. HKEY_CURRENT_CONFIG

Merupakan HKEY_LOCAL_MACHINE yang sesuai untuk konfigurasi perangkat keras.



Gambar 2.6 Struktur hierarki database registry

2.1.3 Pengenalan Visual Basic

Visual Basic merupakan salah satu development tools yang dibuat oleh Microsoft. Visual Basic lebih bersifat fleksibel di banding dengan development tools lainnya, Visual Basic tidak terlalu tergantung terhadap fungsi dari API pada Windows. API sudah dimasukkan oleh Microsoft ke dalam VB sebagai file .dll.

Untuk dapat membuat aplikasi virus, harus mempunyai *software development*. Salah satu software development yang digunakan adalah Visual Basic dari Microsoft. Lebih baik jangan salah paham karena virus yang dipakai bukan bertujuan untuk merusak sistem, tetapi untuk melindungi file *.swf dengan menyembunyikan file *.exe (virus).

A. Malicious Code

Kode jahat/perusak (*malicious code* atau disingkat *malcodes*) didefinisikan sebagai semua macam program makro atau *script* yang dapat dieksekusi dan dibuat dengan tujuan untuk merusak sistem computer. Oleh karena itu, bug yang dibuat secara tidak sengaja oleh programmer tidak termasuk dalam kategori ini, tetapi untuk bug yang benar-benar mengganggu, banyak orang mengkategorikannya sebagai *malcode*. [RAH05]

Kode perusak dapat digolongkan dalam 3 macam golongan, yaitu virus, worm dan Trojan Horse.

1) Virus

“A program that can infect other programs by modifying them to include a slight altered copy of it self. A virus can spread throughout a computer system or network using the authorization of every user using it to infect their programs. Every programs that gets infected can also act as a virus that infection grows.” [DJO99]

Virus komputer bisa diartikan sebagai suatu program komputer biasa, tapi memiliki perbedaan yang mendasar dengan program-program lainnya, yaitu virus dibuat untuk menulari program-program lainnya, mengubah, memanipulasinya bahkan sampai merusaknya. Ada yang perlu dicatat disini, virus hanya akan menulari apabila program pemicu atau program yang telah terinfeksi tadi dieksekusi, disinilah perbedaannya dengan “worm”.

B. Kriteria Virus

Suatu program dapat disebut suatu virus apabila memenuhi beberapa kriteria berikut :

1) Kemampuan mendapatkan informasi penting

Virus didesain untuk menginfeksi banyak file atau program. Misalnya, virus akan menuluri seluruh file .txt atau .htt, maka virus tersebut akan mencari informasi tentang file-file atau program-program yang ada. Setelah virus mendapatkan informasi tersebut, maka akan disimpan di memori atau di file tertentu yang telah di-set atributnya menjadi hidden (tersembunyi).

2) Kemampuan untuk memeriksa program atau file

Ketika menginfeksi sebuah file, virus harus bisa memeriksa apakah file tersebut telah terinfeksi atau belum. Biasanya virus akan memeriksa byte yang dipakai oleh virus tersebut, atau dengan nama lain penanda. Apabila penanda tersebut telah ada maka virus tidak akan menginfeksi file tersebut. Tetapi apabila file tersebut tidak ada penandanya, maka virus akan menuluri file yang sehat itu.

3) Kemampuan menggandakan diri

Kemampuan ini harus (wajib) dimiliki oleh virus, karena kemampuan ini digunakan untuk perkembangbiakkan virus tersebut. Biasanya virus akan memasang penanda dan menyisipkan program untuk memperbanyak virus tersebut.

4) Kemampuan untuk memanipulasi

Kemampuan ini digunakan untuk memunculkan pesan, gambar, menghapus file, mencuri data/file, dan lain sebagainya.

5) Kemampuan untuk menyembunyikan diri dan data yang dibuat.

Seluruh aktifitas yang dilakukan virus haruslah tidak kelihatan, baik dari user ataupun dari computer. Hal ini menjadi keindahan sebuah virus. Biasanya kita bisa melihat seluruh proses yang ada di computer (khususnya yang berbasis windows) hanya dengan menekan Ctrl+Alt+Delete, maka akan muncul kotak yang berisikan proses-proses yang sedang berjalan pada computer. Sebuah virus yang bagus adalah yang tidak tertangkap prosesnya oleh trapping tersebut. Apalagi jika dapat menghindari deteksi sebuah antivirus. [SHA06]

C. Siklus Hidup Virus

Siklus hidup virus secara umum melalui empat tahap, yaitu :

1) Dormant Phase (fase istirahat / tidur)

Pada fase ini virus tidaklah aktif. Virus akan diaktifkan oleh suatu kondisi tertentu semisal : tanggal yang ditentukan, kehadiran program lain / dieksekusinya program lain dan sebagainya. Tidak semua virus melalui fase ini.

2) Propagation Phase (fase penyebaran)

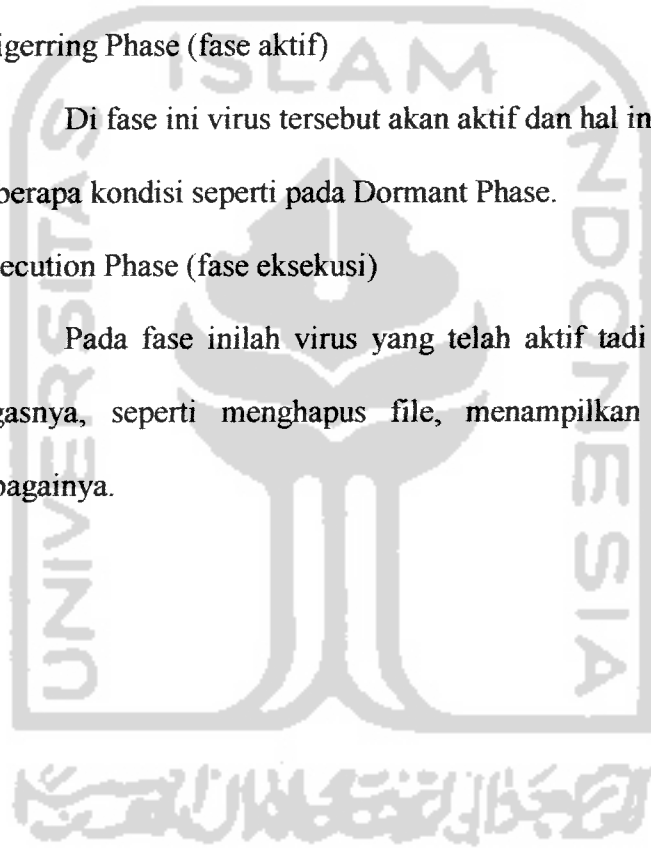
Pada fase ini virus akan meng-copy dirinya kepada suatu program atau ke suatu tempat dari media storage (baik harddisk, RAM dan sebagainya). Setiap program yang terinfeksi akan menjadi hasil “klonng” virus tersebut (tergantung cara virus tersebut menginfeksinya).

3) Triggerring Phase (fase aktif)

Di fase ini virus tersebut akan aktif dan hal ini juga dipicu oleh beberapa kondisi seperti pada Dormant Phase.

4) Execution Phase (fase eksekusi)

Pada fase inilah virus yang telah aktif tadi akan melakukan tugasnya, seperti menghapus file, menampilkan pesa-pesan dan sebagainya.



BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

Menciptakan karya animasi dua atau tiga dimensi tidak mudah, selain dibutuhkan kemampuan dalam menggunakan perangkat lunak untuk animasi juga dibutuhkan kreatifitas yang tinggi. Seseorang yang mampu menciptakan sebuah animasi tentulah orang yang kreatif dan mempunyai jiwa seni.

Tentunya akan merasa kecewa apabila sesuatu yang telah dibuat dengan susah payah dapat dimanfaatkan orang lain tanpa adanya timbal balik yang diberikan. Dalam penelitian ini mengamankan file *.swf bertujuan agar file tersebut tidak dapat di-*compile* ke bentuk *.fla, sehingga apabila dalam bentuk *.fla sebuah file animasi dua dimensi dengan menggunakan perangkat lunak Macromedia Flash MX dapat di edit dan di hapus.

Menggunakan file ini tidak begitu sulit, dibutuhkan ketekunan untuk mempelajarinya. Metode yang akan digunakan untuk mengamankan file *.swf ini adalah dengan menggunakan sebuah *development software*, dengan perangkat ini akan menghasilkan file yang berekstensi *.exe. File tersebut berisi source code program pembantu untuk melindungi file *.swf yang nantinya akan dimasukkan ke dalam flash dengan menggunakan bahasa pemrograman Action Script yang ada pada Macromedia Flash MX.

Untuk dapat menyisipkan sebuah file virus harus dapat mengerti registry pada sistem operasi Windows. Registry adalah *database* yang biasanya digunakan

untuk menyimpan informasi *setting* semua *hardware*, *software* dan pengguna yang ada di dalam sistem operasi Windows. Jika user mengubah *setting control panel*, sistem atau *software* yang telah terinstal akan mempengaruhi dan akan disimpan pada *registry*. Perangkat lunak ini merupakan *free software* yang didalamnya terdapat pengetahuan tentang *registry windows*.

3.1 Metode Analisis

Metodologi pengembangan sistem merupakan kesatuan metode-metode, konsep-konsep pekerjaan dan aturan-aturan ataupun prosedur-prosedur yang akan digunakan untuk mengembangkan sistem, karenanya metode ini adalah metode yang dibutuhkan untuk dapat melakukan suatu pengembangan sistem perangkat lunak (*software development*).

Tahap analisis dilakukan untuk mengetahui dan menterjemahkan permasalahan dan kebutuhan perangkat lunak serta kebutuhan sistem yang dirancang dengan mendapatkan data-data yang diperlukan untuk membangun sistem.

Metode yang digunakan adalah metode analisis berarah data, yaitu metode analisis secara terstruktur dengan membuat diagram arus data yang akan mendeskripsikan input, proses dan output berdasarkan aliran informasi yang telah ada. Dengan metode ini akan dihasilkan sistem yang lebih terstruktur karena metode ini memberikan gambaran terperinci mengenai komponen-komponen yang digunakan.

3.2 Metode pengumpulan data

3.2.1 Pengumpulan data primer

Dalam mengumpulkan data primer yang diperlukan untuk melakukan penelitian ini, digunakan metode observasi, yaitu suatu metode pengumpulan data dengan cara mengamati secara langsung terhadap objek yang ditinjau untuk mendapatkan data-data yang berhubungan dengan penelitian yang dilakukan.

Pengamatan yang dilakukan pada penelitian ini adalah mengamati cara kerja software decompiler yang dapat meng-compile file yang sudah dalam bentuk *.swf, sehingga dapat dibuat algoritma program yang akan dibuat sehingga dapat memenuhi tujuan dari penelitian ini.

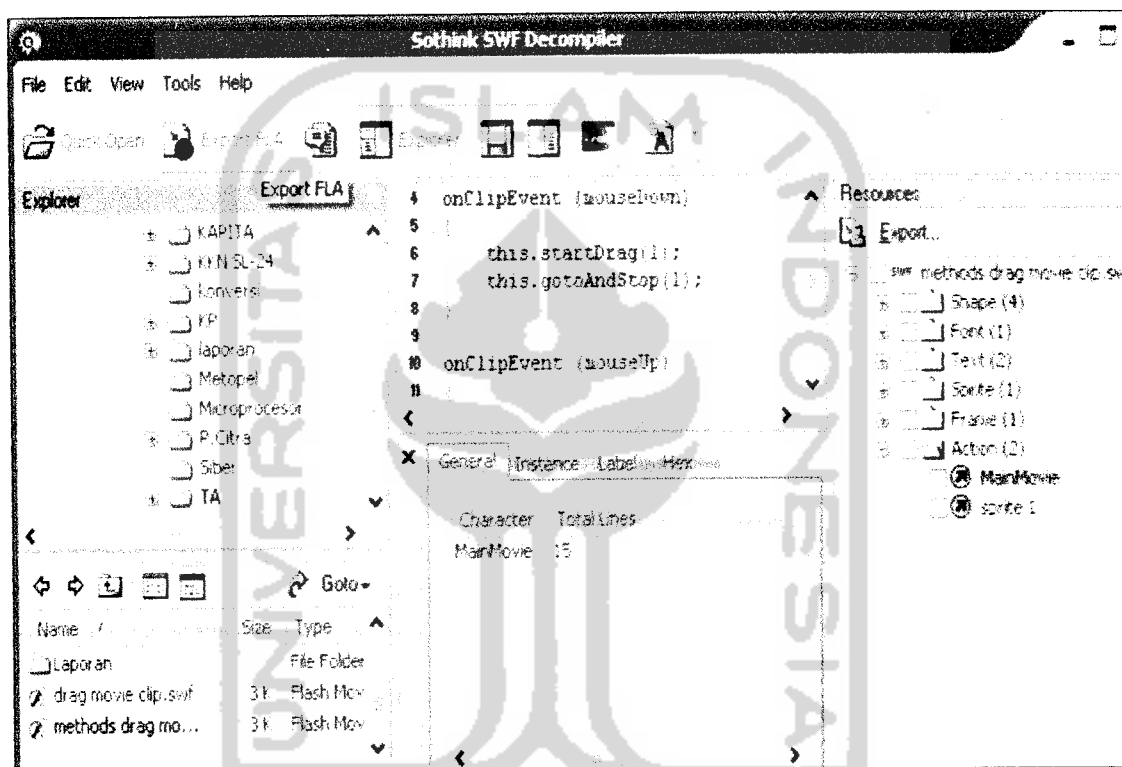
3.2.2 Pengumpulan data sekunder

Pengumpulan data sekunder ini penulis banyak membahas tentang aplikasi yang menggunakan perangkat lunak flash dan development software untuk membuat file *.exe.

Pencarian data bertumpu pada materi yang membahas bagaimana menjalankan file *.swf dengan menggunakan action script dan tutorial yang membahas membuat file *.exe sederhana dengan menggunakan perangkat lunak Visual Basic. File *.exe ini nantinya akan menjadi program yang melindungi file *.swf dari decompiler software.

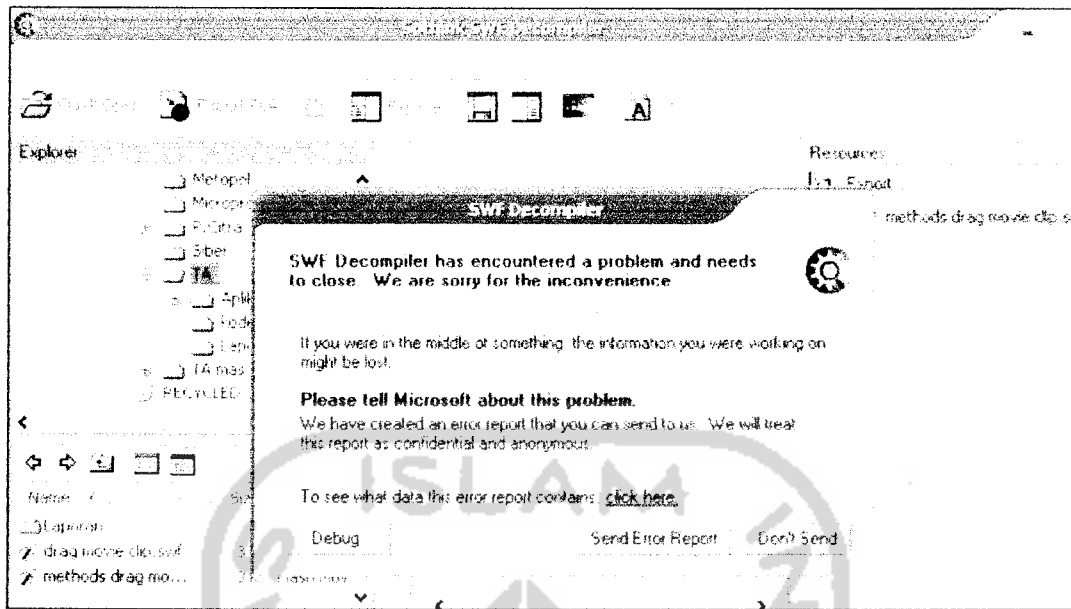
3.3 Analisis SoThink SWF Decompiler

SoThink SWF Decompiler adalah salah satu perangkat lunak yang digunakan untuk mengubah file *.swf menjadi *.fla. Program yang dibuat diharapkan mampu melumpuhkan SoThink SWF Decompiler. Berikut ini adalah salah satu contoh yang bisa dilakukan oleh SoThink SWF Decompiler.



Gambar 3.1 File methods drag movie clip.swf yang akan di compile ulang

Setelah file methods drag movie clip.swf yang sudah dapat terlihat *source codenya* di dalam SoThink SWF Decompiler, maka dengan mudah user dapat meng-export file tersebut ke dalam file .fla. Untuk pengeditan user dapat menggunakan Macromedia Flash MX 2004. Berikut ini adalah tampilan saat file methods drag movie clip.swf yang sudah berada dalam Flash MX 2004.



Gambar 3.4 Hasil dari proses manipulasi

SoThink SWF Decompiler yang telah di manipulasi tidak dapat digunakan lagi sebagai alat yang digunakan untuk mengubah file .swf menjadi file .fla, sehingga file .swf yang telah dibuat bisa di amankan dari orang-orang yang mengambilnya begitu saja dari internet.

Hasil dari analisis kebutuhan adalah kebutuhan perangkat lunak yang hendak dibangun. Kebutuhan yang dimaksud adalah kebutuhan masukan, keluaran, kebutuhan antarmuka, dan kebutuhan proses atau fungsi. Perangkat lunak yang dihasilkan dalam penelitian ini bernama *Perancangan dan Implementasi Keamanan File *.SWF dengan Menggunakan Action Script*.

3.3.1 Masukan (Input)

Masukan yang dibutuhkan oleh perangkat lunak yang dibangun dalam penelitian ini adalah sebuah file ActiveX yang berada dalam sistem, yaitu flash.ocx.

3.3.2 Proses

Terdapat dua buah proses yang dibutuhkan oleh perangkat lunak, yaitu browse file ke sistem dan manipulasi file flash.ocx.

3.3.2.1 Browse file ke sistem

Proses ini dijalankan untuk membuka dan memilih file pada dialog box sistem operasi untuk dilanjutkan dengan proses berikutnya. Ketika file tersebut telah dipilih melalui kotak dialog yang muncul, maka detail dari file tersebut akan diperlihatkan di text area, yaitu ukuran file, tipe file, nama file, tanggal pembuatan dan tanggal modifikasi file.

3.3.2.2 Manipulasi file

Proses ini berjalan dengan cara menambahkan sebuah string ke dalam sebuah file tertentu yang digunakan oleh flash player pada decompiler, sehingga software decompiler tersebut tidak dapat digunakan dengan maksimal. Pada proses ini file akan dimanipulasi, terdapat beberapa langkah dalam manipulasi file, yaitu mencari file yang terdapat di register Sistem Operasi kemudian file tersebut di manipulasi.

3.4 Keluaran (Output)

Hasil dari manipulasi sebuah file ini dapat diketahui dengan adanya penambahan ukuran file yang telah berhasil dimanipulasi. Manipulasi file ini dikatakan berhasil jika software decompiler yang ada pada komputer tidak dapat

digunakan. Hal ini dikarenakan salah satu item yang ada pada decompiler tersebut, yaitu flash player telah dimanipulasi filenya. Dengan demikian keamanan file .swf dapat terpenuhi dengan tidak berjalannya software decompiler sehingga file .swf yang didapat tidak dapat di ubah menjadi .fla.

3.5 Antarmuka Sistem

Antarmuka merupakan penghubung antara pengguna dengan sistem. Agar pengguna dapat mengeksplorasi sistem dengan mudah maka diperlukan sebuah antarmuka yang tidak membingungkan (*user friendly*).



BAB IV

PERANCANGAN PERANGKAT LUNAK

Sistem komputer terdiri dari 3 aspek, yaitu aspek perangkat keras (hardware), perangkat lunak (software) dan manusia (brainware). Ketiga aspek tersebut harus dapat bekerja sama agar sebuah sistem komputer dapat bekerja dengan sempurna. [SAN97]

4.1 Metode Perancangan

Setelah diperoleh spesifikasi kebutuhan aplikasi perangkat lunak pada tahap analisis kebutuhan, langkah selanjutnya adalah perancangan perangkat lunak. Tahap ini merupakan langkah awal untuk implementasi dan pengembangan perangkat lunak. Dalam rekayasa perangkat lunak, tahap perancangan merupakan tahap yang sangat menentukan kualitas perangkat lunak yang dihasilkan. Oleh karena itu diusahakan tahap perancangan ini dibuat dan didokumentasikan sebaik mungkin agar perangkat lunak yang dihasilkan sesuai dengan yang diharapkan.

Flowchart merupakan bagan aliran sistem yang menggambarkan urutan kerja prosedur atau fungsi yang ada dalam sistem. Pada tahap perancangan sistem, penggunaan simbol membantu komunikasi antara pengguna dengan sistem, sehingga diharapkan pengguna dapat memahami secara logika. Dengan alat desain flowchart memungkinkan penggambaran sistem secara keseluruhan, mulai awal sistem, pemasukkan data hingga keluaran hasil sesuai dengan yang diharapkan.

Aplikasi perangkat lunak ini didesain berdasarkan metode desain berorientasi pada diagram alur. Sistem dideskripsikan sebagai proses atau kumpulan proses transformasi yang mengubah data masukan menjadi data keluaran. Flowchart digunakan untuk menggambarkan aliran informasi dan transformasi data yang terjadi di dalam sistem.

4.2 Prosedur Sistem

Program yang dibuat diharapkan mampu melumpuhkan SoThink SWF Decompiler. Maka cara yang diambil untuk melumpuhkan software tersebut adalah dengan menghapus file *.exe dari SoThink SWF Decompiler. Dari hasil mempelajari registry diketahui bahwa agar file *.exe tidak mudah untuk dilumpuhkan, dengan cara mengunci registry windows maka file *.exe yang telah dihapus tidak dapat digunakan kembali kecuali dengan proses penginstalan ulang. Selain itu pada registry dapat diketahui alamat path semua program yang telah terinstal pada Sistem Operasi.

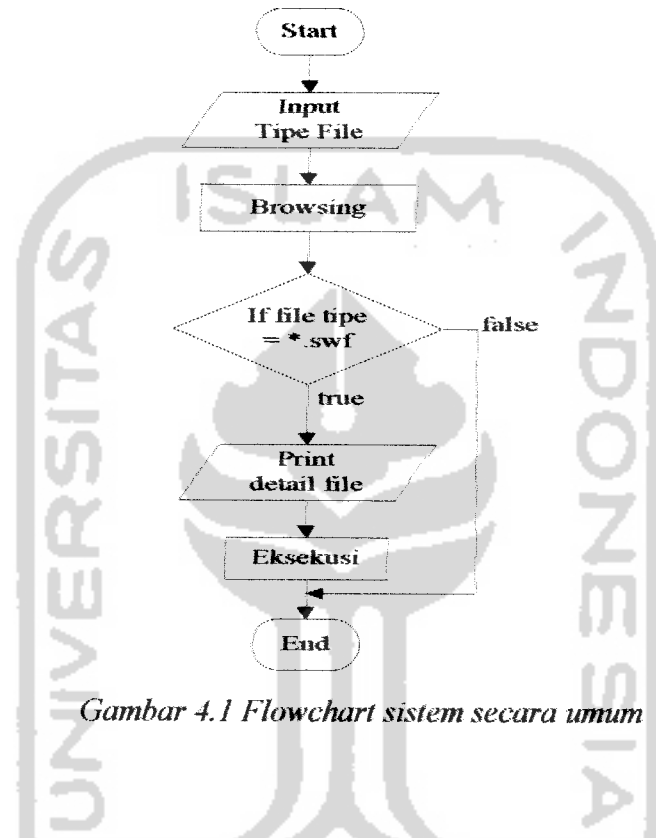
Program yang akan dibuat untuk mengamankan file *.swf diperlukan data path file pada registry lokasi dari “Installer Location” yang berguna untuk mengetahui lokasi software SoThink SWF Decompiler.

4.3 Hasil Perancangan

Hasil perancangan merupakan terjemahan kebutuhan perangkat lunak yang dibangun dalam penelitian, yang meliputi dokumentasi desain arsitektur

perangkat lunak, analisis software SoThink decompiler dan desain antarmuka aplikasi yang hendak dibangun.

4.3.1 Desain Arsitektur Perangkat Lunak



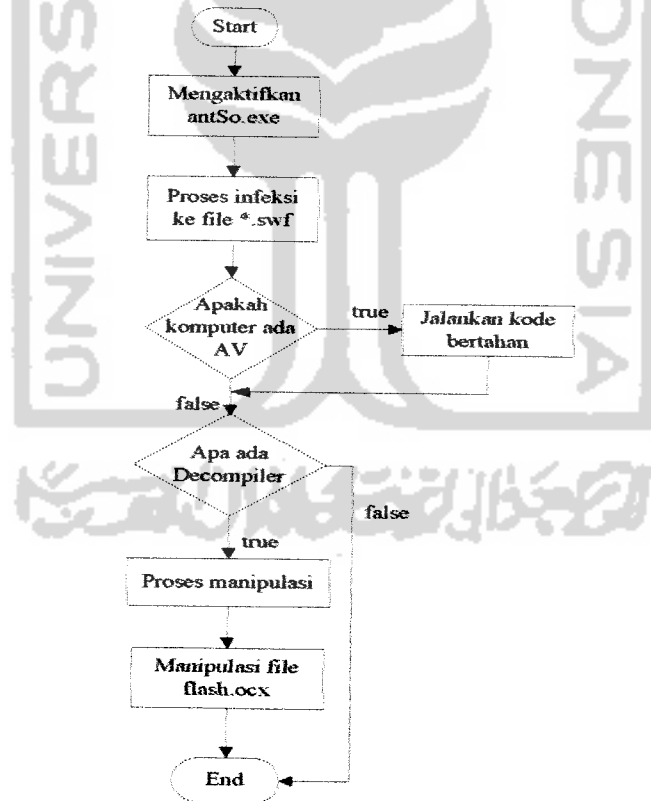
Gambar 4.1 Flowchart sistem secara umum

Algoritma dari flowchart diatas dapat dijabarkan sebagai berikut :

1. Mulai.
2. Memasukkan input, yaitu dengan memilih tipe file yang ada dalam combo box. Tipe file tersebut adalah SWF (*Shock Wave Flash*) yang file tersebut mempunyai ekstensi file .swf.
3. Tipe file telah terpilih, kemudian tombol browse aktif sehingga user dapat mencari file-file *.swf di dalam komputer melalui kotak dialog yang muncul.

4. Jika user memilih selain file.swf maka sistem akan menampilkan pesan error dan jika user memilih flash.ocx maka sistem akan meneruskannya.
5. File.swf terpilih, maka akan muncul pada system message beberapa detail dari file tersebut, yaitu ukuran file, tipe file, nama file, tanggal pembuatan file dan tanggal modifikasi file.
6. Tombol eksekusi akan menyala dan sistem sudah siap melakukan manipulasi file.
7. Selesai.

Berikut ini gambar 4.2 yang menunjukkan bagaimana eksekusi berjalan.



Gambar 4.2 Flowchart eksekusi file

Algoritma dari flowchart diatas dapat dijabarkan sebagai berikut :

1. Mulai.
2. Sistem memerlukan pemicu agar file .exe yang disisipkan dapat diaktifkan, yaitu dengan meng-klik atau membuka file yang telah disisipkan file .exe.
3. Proses infeksi ke file yang dituju mulai dilakukan, sistem akan menginfeksi file *.swf.
4. Proses pendeteksian anti virus, apakah komputer tersebut terdapat anti virus yang aktif ?
 - a. Jika ada, sistem akan mengaktifkan kode bertahan, kemudian akan melanjutkan ke proses nomer 6.
 - b. Jika tidak ada, maka akan diteruskan ke proses nomer 6.
5. Proses pengecekan software decompiler, apakah ada software decompiler di dalam komputer ?
 - a. Jika ada, sistem akan melakukan proses nomer 8.
 - b. Jika tidak ada, maka akan langsung menuju proses 10.
6. Proses manipulasi.
7. Manipulasi file flash.ocx.
8. Selesai.

4.3.2 Analisis Software SoThink SWF Decompiler

Perangkat lunak yang dapat mendeteksi semua file *.swf yang ada di harddisk, folder yang ada pada internet, didalam jaringan local. Dengan kata lain semua file *.swf yang terdeteksi oleh perangkat lunak ini dapat dikenali, selain itu juga dapat berfungsi untuk export file *.swf menjadi *.fla sehingga jika telah dirubah menjadi *.fla maka sebuah animasi yang dibuat menggunakan flash dapat diedit, hapus dan ditambahkan.

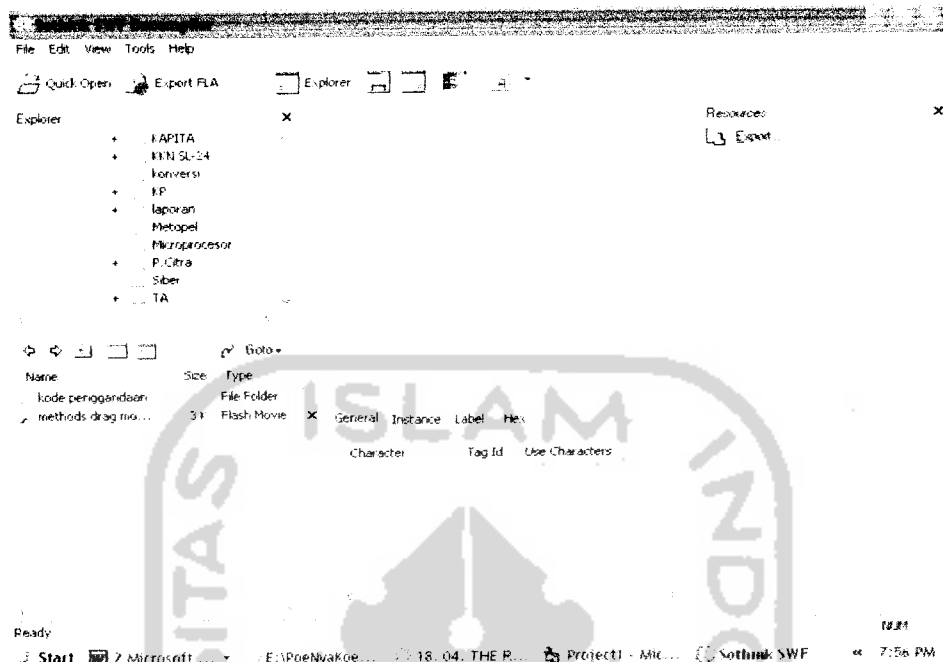
SoThink SWF Decompiler adalah salah satu perangkat lunak yang digunakan untuk mengubah file *.swf menjadi *.fla. Perangkat ini diproduksi oleh Guandong Software & Technology Industry Park, Wuhan Cina.

Setelah proses penginstalan selesai maka software sudah dapat digunakan, SoThink akan membuat folder sourceTec pada registry dengan alamat registry :

- HKEY_LOCAL_MACHINE/Software/SourceTec/SoThink SWF
Decompiler/2.0
- HKEY_LOCAL_USERS/Software/SourceTec/SoThink SWF
Decompiler

Sedangkan pada program file akan membentuk folder *SourceTec/SoThink SWF Decompiler/SWFDecompiler.exe*

Berikut ini adalah tampilan awal dari SoThink SWF Decompiler.



Gambar 4.3 Tampilan utama dari SoThink SWF Decompiler

Keterangan :

- Export fla : tombol yang digunakan untuk mengekspor file *.swf ke *.fla.
- Path folder : berfungsi untuk memilih dimana file *.swf disimpan.
- Kolom file : didalamnya terdapat semua file *.swf yang tersimpan.
- Source file : bagian ini berfungsi menampilkan folder source action script.
- Flash player : bagian ini merupakan tampilan dari file *.swf.

Dari tampilan diatas apabila ditekan tombol “Extract Fla” maka akan mengekspor file *.swf ke dalam bentuk *.fla. Dalam bentuk *.fla maka file tersebut dapat di edit dan di buka dengan menggunakan Flash MX. Software ini mudah untuk dipelajari tidak perlukan seseorang yang ahli untuk dapat mengerti kegunaan dari masing-masing tombol yang disediakan.

Untuk menjalankan file *.exe diperlukan pemicu agar file dapat dijalankan sebagaimana mestinya. Pemicu yang digunakan untuk menjalankan program ini adalah dengan meng-klik atau membuka file *.swf yang didalamnya sudah disisipkan file *.exe. Decompiler tidak mudah untuk dilumpuhkan, dari analisis yang telah dilakukan untuk melumpuhkan software ini adalah sebagai berikut :

- Mencoba untuk menghapus nilai dari SoThink SWF Decompiler yang ada pada registry, setelah dicoba decompiler tersebut masih bisa dijalankan dengan normal.
- Setelah itu mencoba menghapus file *.dll yang ada pada SoThink, dan software ini masih bisa dijalankan dengan normal. Tidak semua komputer meng-copy file *.dll, oleh karena itu ketika file tersebut dihapus maka software itu masih tetap bisa berjalan.
- Percobaan mendekrip software tersebut dengan menggeser binary bitnya sehingga membuat program tersebut error karena mengacak source file yang ada di dalamnya. Namun percobaan ini juga masih belum dapat melumpuhkan SoThink. Hasilnya adalah dekripnya berhasil dijalankan namun dalam hitungan detik Sothink dapat memperbaiki dirinya sendiri.
- Percobaan menghapus file SWF Decompiler pada default pathnya pada saat penginstallan. Percobaan ini sudah dapat membuat software tersebut tidak bisa dijalankan namun kelemahannya adalah pada programnya hanya disetting untuk menghapus file SWF Decompiler pada drive dimana default pathnya pada saat penginstallan yaitu tersimpan pada `C:\ProgramFiles\SourceTec\SoThinkSWFDecompiler.exe`.

Apabila default pathnya tidak berada pada C:\ maka software tersebut masih bisa digunakan. Selain itu kelemahan percobaan ini hanya dapat menghapus file *.exe dengan nama SWF Decompiler.exe sehingga apabila nama tersebut diganti maka software tersebut masih bisa dijalankan.

- Analisis selanjutnya adalah mencari logika lain selain menghapus dan mendekrip software tersebut. Pada bab sebelumnya dijelaskan tampilan menu utama dan kegunaannya. SoThink SWF Decompiler tidak dapat dijalankan apabila tidak memiliki flash player dadalamnya. Sehingga dalam percobaan selanjutnya mencari file shockwave dari flash player. Flash player memiliki file flash *.ocx dan flash *.oca. Apabila file flash *.ocx dibaca dalam bentuk text lalu disimpan dengan nama yang lain meka akan memberi dampak pada SoThink SWF Decompiler yaitu tidak dapat digunakan secara maksimal.

Cara kerja dari sistem yang akan dibuat adalah sebagai berikut :

1. Mencari file flash *.ocx, mencari file ini adalah untuk melumpuhkan SoThink agar tidak dapat menjalankan file *.swf dengan semaksimal mungkin. File flash *.ocx yang terletak pada alamat C:\WINDOWS\System32\Macromed\Flesh\Flesh.ocx pada windows, sedangkan alamat pada registry HKLM\Software\Microsoft\Windows\ _ CurrentVersion\Installer\InstallerLocation.

4.

Dibawah ini adalah source program yang mencari file tersebut

```
keyinstall
ReadKey("HKLM\Software\Microsoft\Windows\CurrentVersion\In
staller\InstallerLocation")
sIn = FileText (keyinstal & "\Macromed\Flash\
Flash.ocx")
```

2. File dalam bentuk *.exe yang dibuat hasil yang dijalankan pertama kali adalah menghentikan proses "SWF Decompiler.exe" yang terletak di C:\ProgramFiles\SourceTec\SoThinkSWFDecompiler.exe,

Source program untuk mematikan proses adalah sebagai berikut :

```
shell "TaskKill/F/IM SWFDecompiler.exe",vbHide
```

3. Proses telah dihentikan setelah itu program akan membaca registry untuk mencari file flash *.ocx dengan source program sebagai berikut :

```
Public Sub bacafile flash()
Dim sIn asString
Dim swords() asString
Dim X asLong
Dim Data asString
Dim tik asString
Dim tuk asString

tik = ""
tuk = "&_"

keyinstall
ReadKey("HKLM\Software\Microsoft\Windows\CurrentVersion\I
nstaller\InstallerLocation")
sIn = FileText (keyinstal &
"\Macromed\Flash\Flash.ocx")
swords = Split (sIn,vbCrLf)
'ReDim Preserve swords (Ubound(swords)-1)
sIn = ""
for x = Lbound(swords) to Ubound(swords)
data = swords(X)
log tik & data & tik & tuk
next x
exit sub
```

men;

meru

untul

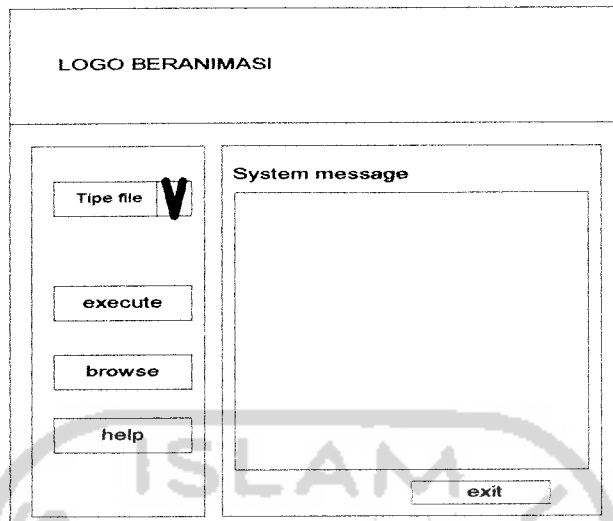
Tab

Pe

Full

Allo

Show



Gambar 4.4 Desain antarmuka

Browse merupakan salah satu komponen dalam sistem aplikasi yang dibuat, berfungsi untuk menampilkan kotak dialog yang terhubung dengan sistem operasi, user dapat memilih jenis file yang ada di dalam sistem. File yang akan dimanipulasi adalah file flash.ocx, file ini dapat ditemukan berada di lokasi *C:\WINDOWS\System32\Macromed\flash*.

Eksekusi file yang dilakukan oleh sistem adalah proses manipulasi file flash.ocx, apabila file yang dituju telah dipilih. Proses manipulasi ini dilakukan dengan menambahkan sejumlah string ke dalam file flash.ocx yang terlebih dulu telah dipilih. Perubahan pada ukuran file merupakan sebuah indikasi bahwa flash.ocx telah berhasil di manipulasi. Secara tidak langsung aplikasi telah mematikan salah satu fungsi dari software decompiler, yaitu flash player, sehingga software decompiler tersebut tidak dapat bekerja dengan maksimal.

BAB V

IMPLEMENTASI

5.1 Batasan Implementasi

Berikut batasan-batasan implementasi perangkat lunak dari sisi *hardware* dan *software* yang digunakan dalam pengembangan perangkat lunak serta spesifikasi *hardware* maupun *software* yang diperlukan untuk menjalankan perangkat lunak yang dihasilkan dalam penelitian ini.

5.1.1 Kebutuhan Perangkat Lunak

Penelitian ini pada dasarnya bertujuan untuk melindungi sebuah karya animasi yang dihasilkan dari sebuah perangkat lunak decompiler, sehingga perangkat lunak mempunyai peranan yang penting dalam penelitian ini. Adapun software yang digunakan adalah sebagai berikut :

- a. Microsoft Windows XP Professional 2002
- b. Microsoft Visual Basic 6.0
- c. Microsoft Word 2002
- d. Microsoft Visio 2002
- e. Macromedia Flash Profesional 8
- f. Windows Registry Guide 2003

Berikut adalah perangkat lunak yang dibutuhkan untuk menjalankan aplikasi yang dihasilkan:

- a. Sistem operasi Windows 98 dan sesudahnya
- b. Macromedia Flash MX 2004
- c. SoThink SWF Decompiler

5.1.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras untuk pengembangan aplikasi dalam penelitian ini adalah sebagai berikut:

- a. *Processor* Intel Pentium Celeron 4 1.7 GHz
- b. Memori DDRAM 128 MB
- c. VGA GeForce 2 - 32 MB
- d. Hard Disk Seagate 20 GB
- e. *Monitor, Keyboard, dan mouse*

Spesifikasi minimal perangkat keras yang diperlukan untuk menjalankan aplikasi tersebut adalah sebagai berikut:

- a. Processor pentium III 800 MHz atau yang setara
- b. Memori minimal DDRAM 128 MB atau yang setara
- c. Ruang Hardisk 40 MB
- d. *Monitor, Keyboard, dan mouse*

5.2 Implementasi Perangkat Lunak

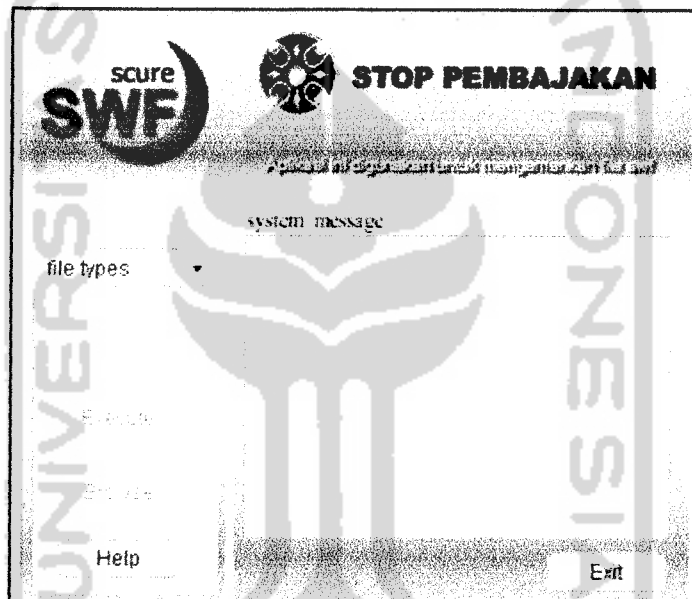
Bab ke lima ini akan menjelaskan mengenai implementasi perangkat lunak berdasarkan pada analisis kebutuhan dan perancangan yang sudah diterangkan

dalam dua bab terdahulu. Implementasi akan menggunakan perangkat lunak pengembangan dan bahasa Visual Basic 6.0 serta action script.

5.3 Implementasi Antarmuka

5.3.1 Tampilan Utama

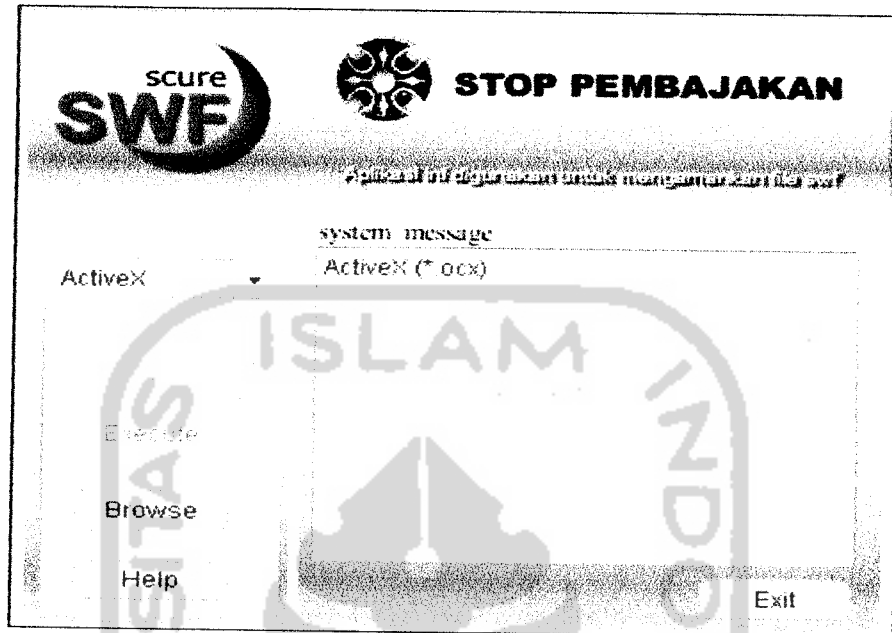
Tampilan utama dari aplikasi yang dikembangkan dapat diilustrasikan pada *Gambar 5.1* sebagai berikut :



Gambar 5.1 Tampilan utama

Pada tampilan utama terdapat dua bagian penting, yaitu tombol-tombol operasi dan system message. Combo box tipe file berisi jenis file ActiveX yang akan dimanipulasi, dalam hal ini file yang akan dimanipulasi adalah flash.ocx. Tipe file ini nantinya akan memberikan batasan pada kotak dialog yang akan muncul untuk hanya menampilkan file-file ActiveX yang mempunyai ekstensi *.ocx saja, pada system message juga terdapat keterangan bahwa jenis file yang

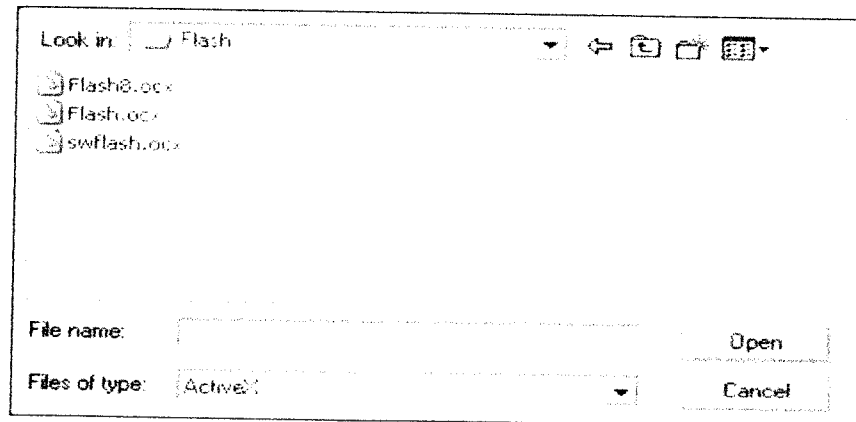
dipilih hanya file-file ActiveX yang berekstensi *.ocx, seperti yang ada dalam Gambar 5.2 berikut ini.



Gambar 5.2 Tampilan saat tipe file telah dipilih

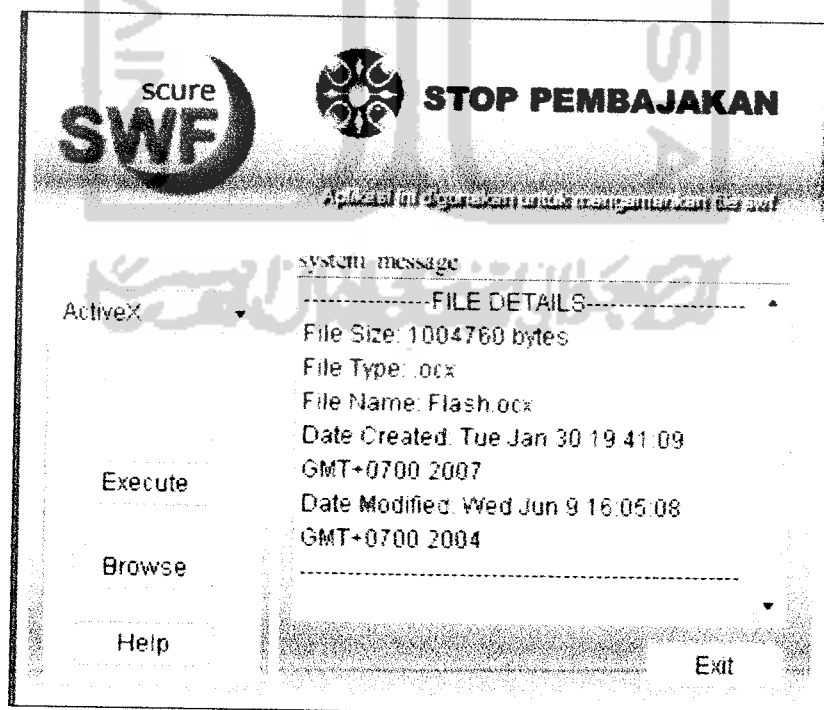
Pada awal program dijalankan hanya terdapat dua tombol yang aktif, yaitu help dan exit. Tombol browse tidak akan muncul jika user tidak memilih tipe file terlebih dulu.

Tombol browse berfungsi untuk menampilkan kotak dialog yang didalamnya terdapat file-file lokal yang ada dalam komputer, kotak dialog ini dimaksudkan agar user dapat memilih file flash.ocx yang berada di dalam *local file*, seperti pada Gambar 5.3 berikut ini.



Gambar 5.3 Dialog box yang muncul saat tombol browse ditekan

Saat browse selesai diproses dan user telah memilih file flash.ocx, maka pada *system message* akan ditampilkan beberapa atribut yang di miliki oleh file tersebut. Atribut yang ada pada pada file flash.ocx yang ditampilkan adalah tipe file, ukuran file, nama file, tanggal pembuatan file dan tanggal modifikasi file tersebut.



Gambar 5.4 Detail file yang ditampilkan

Setelah detail file ditampilkan pada system message, maka tombol execute yang tadinya tidak aktif pada awal sistem dijalankan dapat digunakan untuk melakukan proses berikutnya. Di dalam detail file terdapat salah satu atribut yang menjadi acuan apabila sistem berhasil melakukan tugasnya dalam memanipulasi file tersebut, yaitu ukuran file akan berubah, seperti pada *Gambar 5.5* berikut ini.



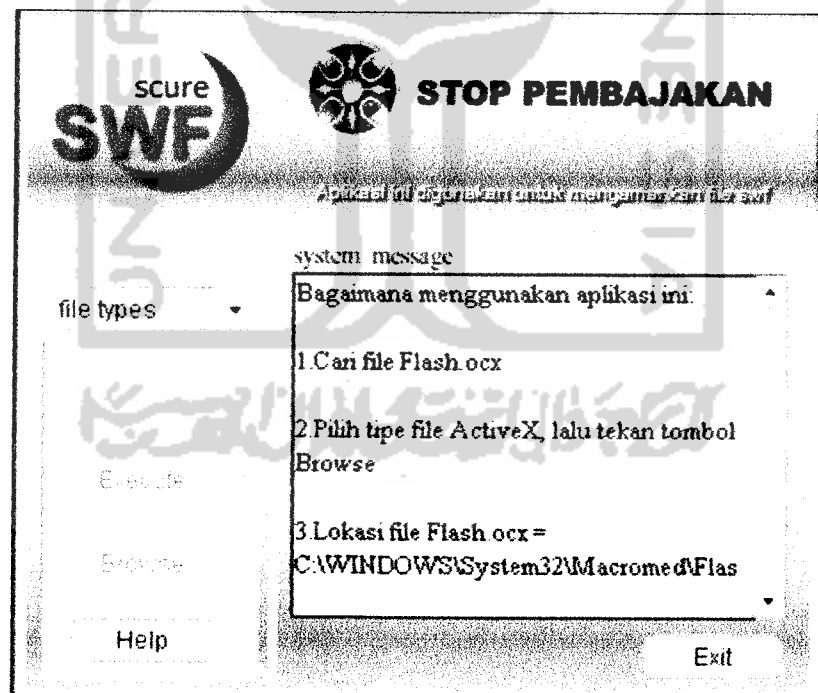
Gambar 5.5 Ukuran file setelah manipulasi

Dengan berubahnya ukuran file tersebut, seperti yang ditunjukkan *Gambar 5.5* maka sistem telah berhasil sehingga apa yang diharapkan dalam pembuatan sistem ini dapat tercapai, yaitu software decompiler dapat di non-aktifkan sehingga file .swf dapat di amankan dari software decompiler yang tugasnya meng-convert dari file .swf ke file .fla.

SoThink SWF Decompiler merupakan perangkat lunak yang dapat mendeteksi semua file *.swf yang ada di harddisk, folder yang ada pada internet,

di dalam jaringan local. Dengan kata lain semua file *.swf yang terdeteksi oleh perangkat lunak ini dapat dikenali, selain itu juga dapat berfungsi untuk export file *.swf menjadi *.fla sehingga jika telah dirubah menjadi *.fla maka sebuah animasi yang dibuat menggunakan flash dapat diedit, hapus dan ditambahkan.

Help pada sistem ini merupakan fungsi yang harus ada dalam setiap sistem tidak terkecuali pada sistem yang sekarang dikembangkan, help mempunyai peranan yang tidak kalah penting dengan fungsi utama dari sistem itu sendiri. Di dalam help sendiri terdapat beberapa langkah-langkah dalam penggunaan sistem yang dibuat, selain itu di dalamnya juga terdapat bagaimana caranya fungsi flash.ocx dapat berfungsi seperti semula.



Gambar 5.6 Tampilan help

Karya animasi dibuat dengan tujuan yang berbeda-beda namun bila dilihat dari tampilannya, pada dasarnya sebuah animasi dibuat untuk mempercantik dan menyampaikan pesan menjadi lebih menarik. Untuk dapat membuat animasi tentunya tidak mudah, dibutuhkan ketelitian dan ide yang segar. Rasanya akan sangat kecewa jika sebuah karya animasi yang dibuat dapat dirubah tanpa adanya kompensasi. Untuk itulah pada penelitian ini akan membahas bagaimana mengamankan file *.swf agar sebuah decompiler tidak dapat mengubahnya.

Menyisipkan sebuah file yang berekstensi *.exe yang dibuat dengan menggunakan *development software*, tetapi virus tersebut bukan bertujuan untuk merusak sistem operasi Windows. Cara kerja dari program itu sendiri disesuaikan untuk dapat melindungi file *.swf dari *software decompiler* tersebut. Program di atas dapat *dicompile* sehingga menghasilkan file *.exe, setelah file *dicompile* maka dapat digunakan untuk melindungi file *.swf.

Memasukkan file *.exe pada flash juga tidak begitu sulit, yaitu dengan menggunakan action script bahas pemrograman yang ada di Flash MX. Pada action script terdapat syntax yang berfungsi untuk memanggil file *.exe agar dapat dijalankan di flash. Syntax tersebut adalah *fscommand()*. Algoritma dari penggunaan *fscommand()* adalah sebagai berikut :

```
fscommand("nama perintah",parameter);
```

BAB VI

ANALISIS KINERJA

Dalam tahap ini akan dijelaskan tentang pengujian program aplikasi yang digunakan pada sistem pengamanan file .swf dengan menggunakan action script. Dari hasil pengujian diharapkan dapat diketahui kekurangan-kekurangan dari sistem untuk kemudian diperbaiki sehingga kesalahan dari sistem dapat di minimalisasi atau bahkan dihilangkan

Pengujian terhadap sistem yang telah dibuat termasuk penanganan kesalahan algoritma dan fungsi-fungsi sehingga dapat diperbaiki untuk kesempurnaan sistem. Metode pengujian yang digunakan adalah metode *simulation test* dimana test dilakukan dengan mencoba meniru keadaan pada lingkungan pada saat sistem digunakan.

6.1 Pembahasan

Pada tahap ini akan dilakukan ujicoba untuk mengetahui apakah sistem berjalan dengan baik sesuai dengan yang diharapkan atau sebaliknya. Melalui tahap implementasi dapat dievaluasi jalannya sistem dan kendala-kendala yang dapat muncul dalam sistem sehingga dapat diketahui apakah perlu perbaikan lebih lanjut atau tidak.

Hal yang akan diuji pada saat sebuah file *.swf yang sudah disisipkan file *.exe dijalankan menggunakan Flash Player adalah:

- a. Setelah file *.swf dijalankan apakah ada perubahan atau kekurangan karena file *.exe telah disisipkan.
- b. Program ini dapat terdeteksi oleh antivirus atau tidak? File *.exe yang dimasukkan ke dalam file *.swf adalah sebuah virus, oleh karena itu agar tidak terdeteksi oleh antivirus diperlukan kode pertahanan.
- c. Mencoba untuk meng-*compile* ulang file *.swf dengan menggunakan SoThink SWF Decompiler.
- d. Melihat registry perubahan yang diberikan oleh program yang dibuat.

Dari pengujian sistem diharapkan dapat diketahui :

- a. Apakah file *.exe yang telah disisipkan dapat melindungi file *.swf dari SoThink SWF Decompiler ?
- b. Apakah system telah berjalan sesuai dengan flowchart yang dirancang ?
- c. Kelemahan dan kelebihan system yang telah dibuat.

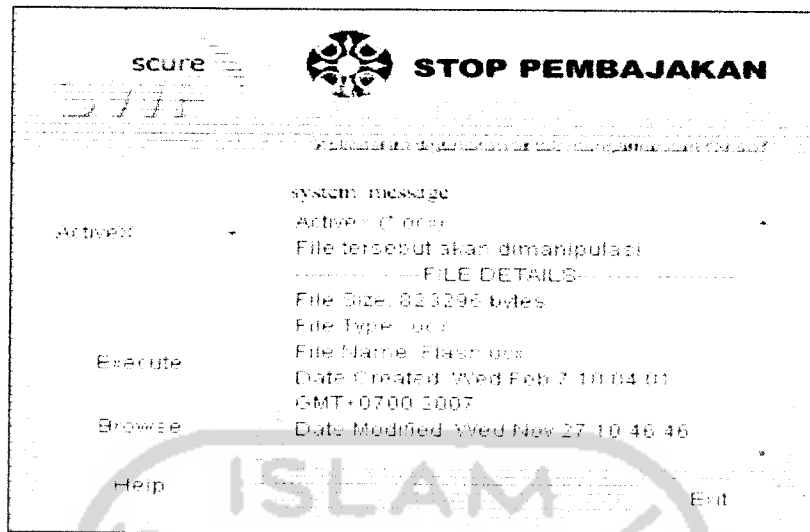
Pengujian ini dilakukan dengan menggunakan beberapa kemungkinan yang akan dilakukan oleh pengguna dari sistem ini. Terdapat beberapa pesan yang ditampilkan pada *system message* bila user menggunakan sistem ini. Kesalahan-kesalahan yang kemungkinan dilakukan oleh user sesungguhnya telah diminimalkan oleh pembuat sistem dengan tidak mengaktifkan semua tombol yang ada pada saat sistem muncul pertama kali. .



Gambar 6.1 Pesan saat tombol cancel pada kotak dialog

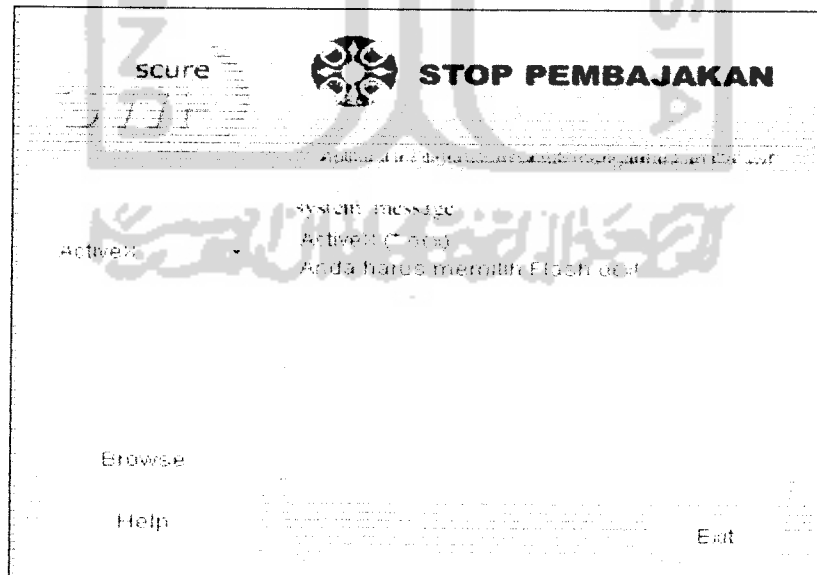
Ketika user menekan tombol Browse, maka sistem akan menampilkan kotak dialog yang berisi file-file yang ada dalam komputer lokal. Kotak dialog tersebut memiliki tombol open dan tombol cancel, pada Gambar 6.1 di atas menunjukkan bahwa user menekan tombol cancel, yang artinya user tidak jadi memilih file yang ada dalam komputer lokal.

Apabila user memilih tombol open yang ada dalam kotak dialog, maka pada system message menampilkan pesan dan atribut file yang dipilih oleh user. Pada Gambar 6.2 dapat terlihat pesan yang muncul saat tombol open pada kotak dialog di tekan.



Gambar 6.2 Tampilan saat tombol open ditekan

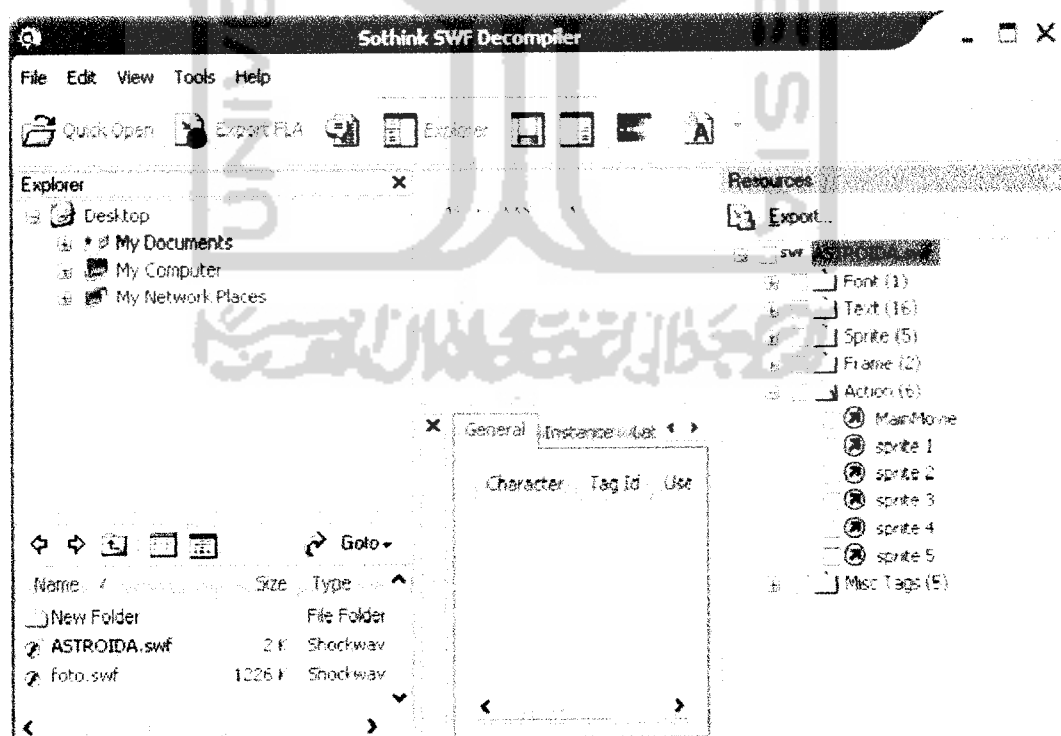
Pada gambar di atas terlihat bahwa tombol execute berubah menjadi aktif, yang tadinya ketika user menjalankan sistem aplikasi ini pertama kali tidaklah aktif. Ini merupakan usaha dari pembuat sistem aplikasi ini untuk mengurangi kesalahan yang mungkin akan dilakukan oleh user.



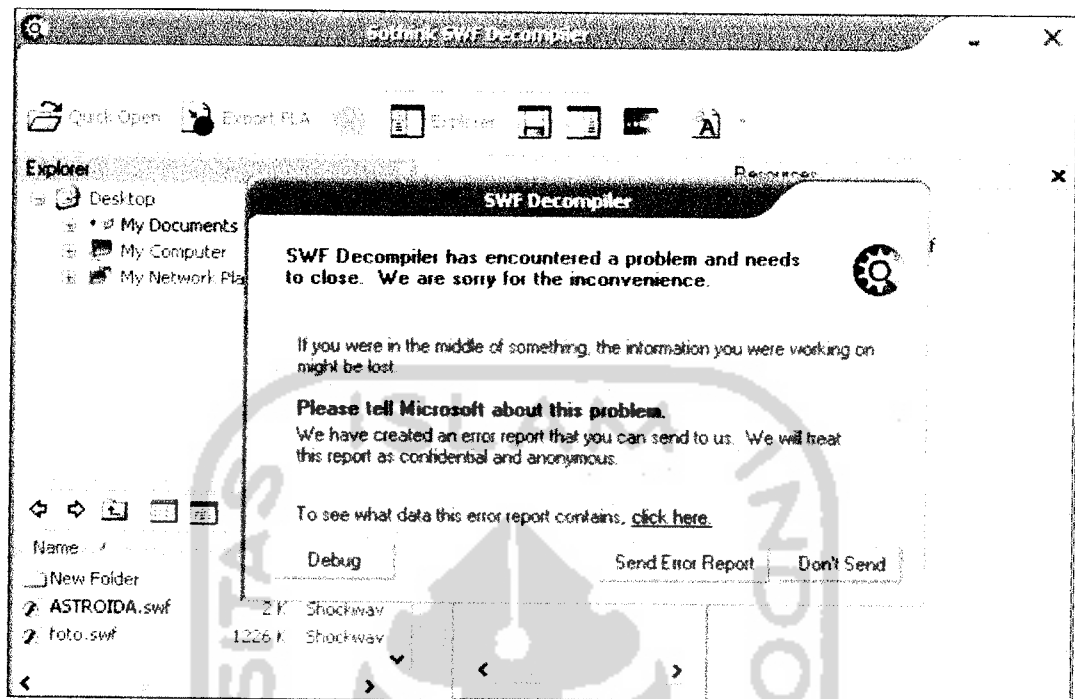
Gambar 6.3 Tampilan saat user salah memilih file

Selain itu terdapat kesalahan yang merupakan kesalahan paling banyak dilakukan oleh user, yaitu saat user salah memilih file yang ada dalam komputer lokal. Kesalahan ini menyebabkan tombol execute yang merupakan tombol proses manipulasi tidak dapat aktif dan terdapat pesan yang ditampilkan oleh sistem, seperti pada *Gambar 6.3* di atas.

Setelah file *.exe dimasukkan pada file *.swf dengan menggunakan action script maka dapat dilihat sistem sudah berjalan sesuai dengan yang diharapkan atau sebaliknya. Berikut ini adalah gambaran sebuah file *.swf setelah diberikan program virus, hasilnya ketika SoThink dijalankan akan error. Dengan ketentuan user menjalankan file *.swf terlebih dahulu sebelum SoThink, maka program akan membuat SoThink tidak dapat dijalankan seperti gambar dibawah ini :



Gambar 6.4 Hasil sebelum program dijalankan



Gambar 6.5 hasil program setelah program dijalankan

Gambar 6.5 menunjukkan bahwa SoThink tidak dapat menjalankan file *.swf karena flash player yang digunakan untuk mendeteksi seluruh file *.swf telah dilumpuhkan, sehingga pada saat file *.swf akan diekspor ke *.fla program harus ditutup. Dengan begitu file *.swf tidak dapat diekspor ke *.fla

Seperti yang sudah dijelaskan pada bab sebelumnya bahwa file *.exe (virus) yang akan dibuat bukan merupakan file yang akan merusak sistem namun tugas dari program tersebut adalah melindungi file *.swf agar tidak dapat di-compile oleh software decompiler. Program yang dibuat dengan menggunakan Visual Basic 6.0 sangat sederhana dan ukuran filenya kecil.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Perancangan dan Implementasi Keamanan file .swf dengan menggunakan action script ini menggunakan bahasa pemrograman Visual Basic 6.0 dalam membuat file .exe yang nantinya akan disisipkan pada file .swf dan Macromedia Flash dalam membuat *interface* aplikasi ini, sehingga user dapat mengerti bagaimana sebenarnya aplikasi ini berjalan.

Tujuan dari dibuatnya sistem ini adalah untuk melindungi file .swf agar tidak dapat di-*compile* ulang dengan menggunakan SoThink swf decompiler. Melindungi file .swf sangat diperlukan untuk mencegah pencurian karya animasi milik para flash designer. Dengan menggunakan aplikasi ini file flash.ocx dapat di manipulasi dan sebagai dampaknya yaitu, Sothink SWF Decompiler sebagai software decompiler tidak dapat digunakan dengan maksimal.

7.2 Saran

Mengingat berbagai keterbatasan terutama masalah waktu berdampak pada munculnya kelemahan yang ditemui pada aplikasi ini, maka saran penulis untuk kesempurnaan aplikasi ini pada masa mendatang adalah :

1. Sistem yang dikembangkan ini dapat dimanfaatkan semaksimal mungkin, sehingga nantinya pembuatan system tersebut dapat bermanfaat bagi masyarakat yang membutuhkannya.

2. System dapat digunakan untuk mengamankan file .swf, hasil program yang dibuat hanya dapat berjalan apabila user telah meng-klik file .swf yang telah di amankan.
3. Pengembangan system lebih lanjut diharapkan mampu mengamankan aplikasi Macromedia lainnya yang berbentuk multimedia seperti, Swish, PowerPoint dan lain-lain.



DAFTAR PUSTAKA

- [AMP02] Amperiyanto, Tri, *Bermain main dengan Registry Windows Tenggelam dalam Windows 98*, Jakarta : PT Elex Media Komputindo, 2002
- [DJO99] Djojo, Minnarto , *Konsep Perlindungan Komputer Terhadap Virus*, Arcle Technologies (1999), <http://www.arclenet>, Diakses tanggal 5 Januari 2007
- [RAH05] Rahardjo, Budi, *Virus Komputer*, <http://it.sbm.itb.ac.id/?p=14>, Diakses tanggal 5 Januari 2007
- [SAN97] Santoso, P. Insap, *Interaksi Manusia dan Komputer Teori dan Praktek*, Yogyakarta : Penerbit Andi, 1997
- [SAP05] Saputra, Johan, *Eksplorasi Kekuatan WIN32-API dengan Visual Basic*, Jakarta : PT Elex Media Komputindo, 2005
- [SHA06] Shadewa, Aat, *Seni Pemrograman Virus menggunakan Visual Basic 6.0*, Yogyakarta : DSI Publishing, 2006
- [SYA05] Syarif, Ary Maulana, *Mastering Action Script Macromedia Flash MX 2004*, Jakarta : PT Elex Media Komputindo, 2005
- [WAH04] Wahana Komputer, *Tutorial membuat Program dengan Visual Basic*, Semarang : Salemba Infotek, 2004
- [ZEE05] Zeembry, *123 Tip & Trik Action Script Flash MX 2004*, PT Elex Jakarta : Media Komputindo, 2005