

KARYA TEKNOLOGI RANCANG BANGUN
PROTOTIPE ANTIVIRUS DENGAN METODE MD5 dan
HEURISTICS ARRS



Perancang :
Agus Nursikuwagus
Rita Rahmawati

Progra Studi Manajemen Informatika
Fakultas Teknik dan Ilmu Komputer
Universitas Komputer Indonesia
Bandung
2016

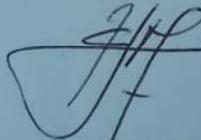
LEMBAR PENGESAHAN

KARYA TEKNOLOGI RANCANG BANGUN
PROTOTYPE ANTIVIRUS DENGAN METODE MD5 dan HEURISTICS ARRS

Telah disetujui dan disahkan sebagai
Karya Teknologi rancang bangun
Perangkat Lunak Prototipe Antivirus Dengan Metode Md5 dan Heuristics Arrs

Disahkan : Bandung
Pada Tanggal : Januari 2016

Disiapkan Oleh,
Perancang Aplikasi



Agus Nursikuwagus, MT.,MM
NIP. 197507092005011003

Rita Rahmawati.ST

Disahkan oleh,
Ketua Program Studi Manajemen Informatika
Fakultas Teknik dan Ilmu Komputer



Dr. Marlana Budhiningtias Winanti, S.Si.,M.Si
NIP. 4127.70.26.020

KATA PENGANTAR

Alhamdulillahirobbil'alamin, allohummasholi 'ala syaidina Muhammad wa'ala ali syaidina Muhammad. Segala puji bagi Allah SWT, perancangan prototipe ini selesai dibuat. Prototipe ini memperkenalkan tatacara membuat antivirus dengan Metode MD5 dan Heuristik Arrs. Prototipe ini dibuat untuk diujicobakan pada komputer berbasis Windows. Prototipe ini berisikan menu :

1. Proses menu utama
2. Proses pemilihan file yang akan di baca
3. Proses menampilkan nama virus yang menginfeksi file serta membersihkan
4. Menu tambahan mengenai antivirus

Prototipe ini diharapkan bisa digunakan sebagai alat pemahaman mengenai tatacara membuat antivirus. Demikian rancang bangun prototipe ini dibuat. Semoga bisa dikembangkan lebih lanjut.

Bandung, Juli 2012
Perancang

Agus Nursikuwaqus, MT.,MM
Rita Rahmawati, ST

DAFTAR ISI

LEMBAR PENGESAHAN	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
I. PENGENALAN APLIKASI.....	1
II. RANCANGAN APLIKASI.....	3
2.1 Rancangan Proses	3
2.1.1 Use Case Antivirus.....	7
2.1.2 State Diagram.....	12
2.1.3 Sequence Diagram.....	19
2.2. Perancangan Data	24
III. IMPLEMENTASI.....	26
3.1 Antarmuka Prototipe	26

I. PENGENALAN APLIKASI

Virus adalah salahsatu malware yang dapat mengakibatkan kerugian terhadap sebuah sistem, penanggulangan serta tindakan pencegahan perlu dilakukan untuk meminimalisir terjadinya kerusakan sistem yang disebabkan oleh virus komputer tersebut. Program antivirus yang dirancang pada penelitian ini menerapkan algoritma MD5 sebagai pengidentifikasian sebuah Virus dengan menggunakan 32 digit Cheksum hexa desimal dan sebuah pendekatan Heuristik dari sifat antivirus yang memanfaatkan file Autorun sebagai starting point pengaktifan virus dalam sebuah sistem.

Tujuan dari pembuatan prototipe ini adalah menerapkan metode MD5 dan Heuristik ArrS pada sebuah antivirus sehingga dapat memberikan fleksibilitas pada proses identifying, scanning dan deleting sebuah file yang terinfeksi virus. Tools yang digunakan untuk merancang program ini adalah menggunakan UML (Unified Modeling language) dengan memanfaatkan Use case Diagram, State Diagram, Sequence Diagram dan Class Diagram sebagai penggambaran sistem yang dirancang.

Keberadaan PC (Personal Computer) sebagai teknologi yang membantu manusia modern untuk mempermudah pekerjaannya merupakan sebuah fakta yang tidak dapat disanggah lagi. Kebutuhan akan PC sebagai media (storage) untuk menyimpan informasi-informasi yang penting menjadi salah satu poin yang mempengaruhi seseorang memerlukan PC. Namun, pengguna PC akan merasa terganggu apabila PC mereka tidak dapat bekerja sebagaimana mestinya. Hal tersebut bisa diakibatkan oleh gangguan dari dalam sistem (hardware failure, bad sector) atau gangguan yang berasal dari luar sistem sehingga menyebabkan hilangnya data dan informasi yang telah tersimpan.

Gangguan dari luar sistem dapat disebabkan oleh program-program malware yang memiliki kelebihan tertentu yang dapat merusak sistem. Beberapa contoh program malware tersebut adalah : virus, worm, trojan, backdoor, dll.

Program malware yang merusak dan sering dikeluhkan para pengguna PC adalah virus file. Kelebihan virus tersebut salah satunya adalah bisa menggandakan diri dan memakai service sebuah program untuk membuat auto-run maka Program untuk scanning dan deleting virus tersebut mulai bermunculan, mulai dari antivirus,

malware detected, virus scanning, dll. Seiring dengan perkembangan teknologi, virus yang bermunculan semakin bervariasi. Begitupula dengan antivirus, para pembuat antivirus akan semakin meningkatkan kemampuan antivirus mereka agar dapat digunakan untuk mengidentifikasi dan melakukan aksi penyelesaian terhadap virus saat ini.

Berbagai macam metode digunakan untuk identifikasi virus yang menyerang sebuah sistem komputer. Salah satunya adalah metode MD5 yaitu metode yang dapat mengidentifikasi virus dengan melihat checksum dari setiap file dengan menghasilkan checksum 32 digit hexa. File-file virus akan memiliki checksum tersendiri dan hal itulah yang akan menjadi kunci identifikasi sebuah virus. Selain dengan penggunaan checksum sebagai identifikasi Virus, diperlukan juga sebuah pendekatan lain untuk mempersempit area identifikasi virus sesuai dengan pendekatan yang diinginkan, sehingga proses scanning virus pada program ini akan ditambahkan dengan penggunaan Heuristik ArrS.

Pembuatan prototipe antivirus difungsikan sebagai media identifikasi Virus berjenis Virus File yang berextensi EXE, DLL, VBS, VMX, DB, COM, SCR dan BAT. prototipe ini hanya difungsikan sebagai media *identifying*, *scanning* dan *deleting* Virus file yang dapat digunakan pada Sistem Operasi Windows. Program yang dibuat merupakan sebuah *prototype* yang mengimplementasikan Metode MD5 dan Heuristik ArrS pada Antivirus

II. RANCANGAN APLIKASI

2.1 Rancangan Proses

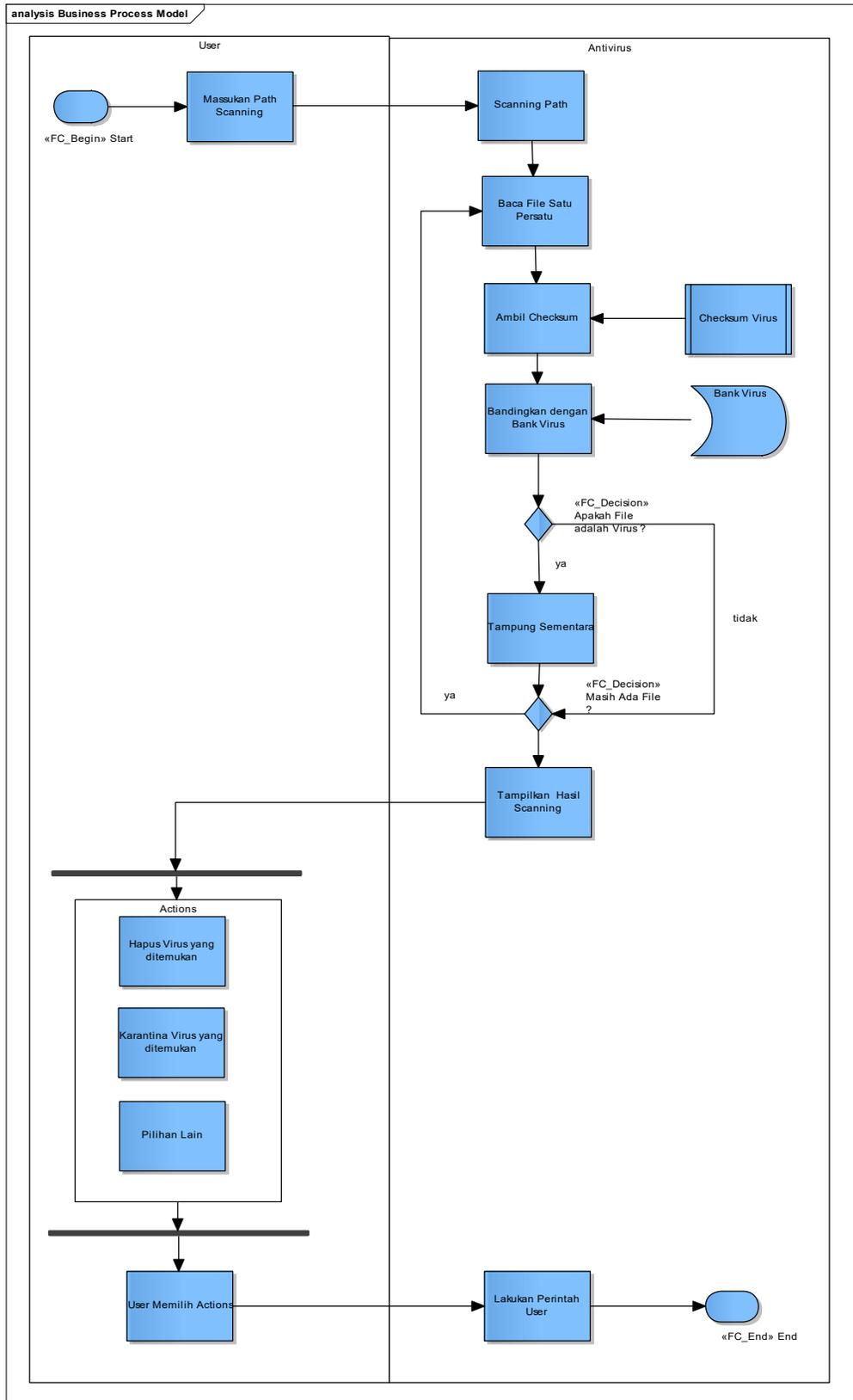
Antivirus adalah sebuah program yang berfungsi melindungi, baik mencegah maupun membasmi virus sebelum dan sesudah masuk dalam sebuah Sistem Operasi yang ada dalam sebuah komputer.

Sebuah program disebut antivirus apabila memenuhi syarat sebagai berikut :

- a. Mampu mencari seluruh file dalam sebuah path.
- b. Memiliki checksum sebagai pengingat virus beserta databasenya.
- c. Mampu *men-terminate* / membunuh proses virus standar.
- d. Mampu menghapus virus yang ditemukannya dari *memory fixed drive* maupun *removable drive*.

Antivirus memanfaatkan Checksum yang dimiliki oleh sebuah file virus dan mencocokkannya dengan database yang dimilikinya. Jika checksum yang didapat sama dengan data yang ada di database virus, maka file tersebut akan teridentifikasi sebagai virus.

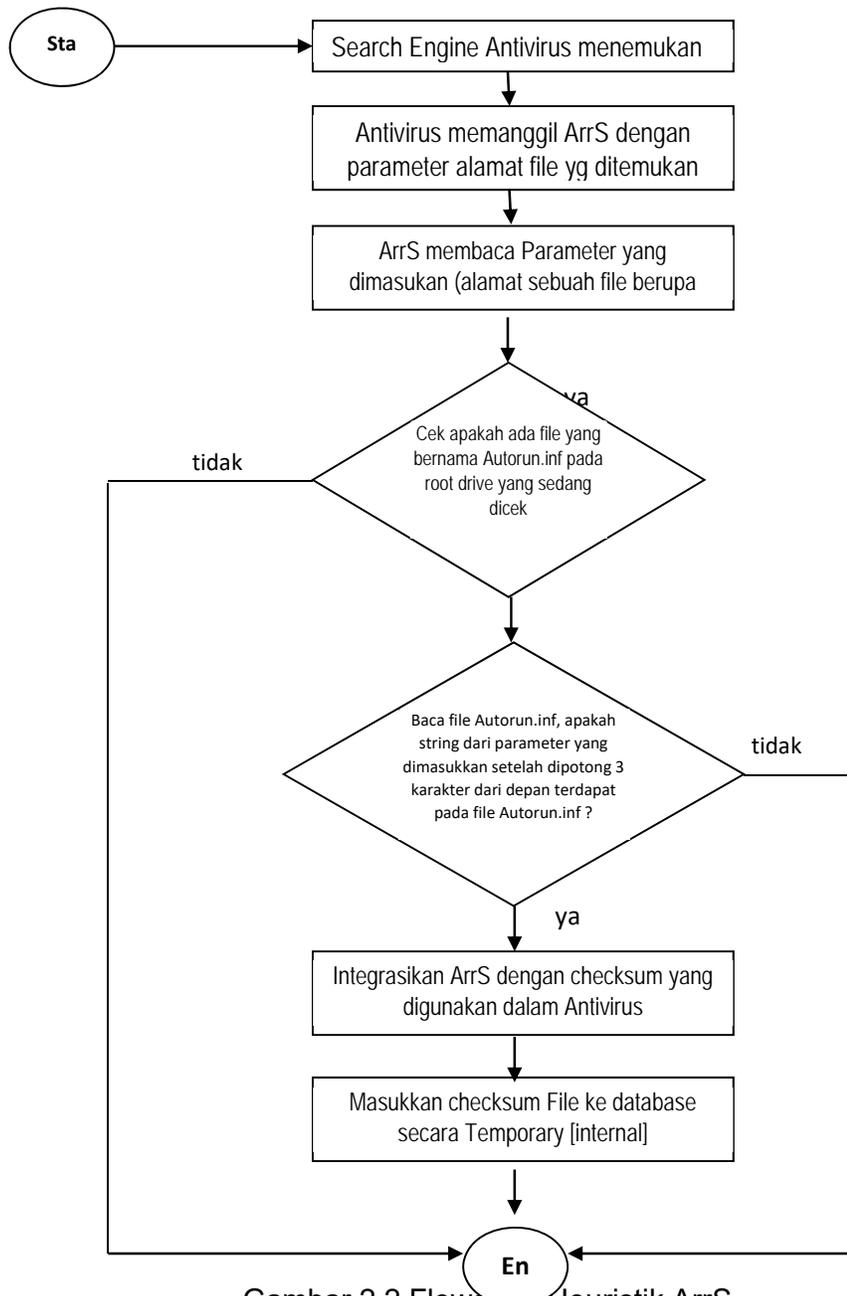
Secara umum Antivirus memiliki fungsi untuk menjelajahi suatu *path* di dalam sebuah komputer kemudian menemukan file yang dicurigai sebagai virus dan melakukan proses lanjutan seperti : menghapus file virus, menormalkan kembali atribut file yang telah dimodifikasi virus, dan atau proses pilihan lainnya yang disediakan. Sehingga *Flowchart* sebuah Antivirus umum bisa digambarkan sebagai berikut:



Gambar 2.1 Flowchart Umum Antivirus

gambar diatas merupakan *flowchart* secara umum bagaimana sebuah Antivirus mengidentifikasi dan melakukan proses tambahan setelah suatu virus teridentifikasi. Identifikasi sebuah virus tersebut didasarkan kepada sebuah metode pengambilan checksum yang telah ditentukan. Salah satu metode identifikasi Virus yang digunakan pada penelitian ini adalah Metode Checksum : MD5.

Selain dengan memanfaatkan checksum sebuah file, Antivirus juga memerlukan sebuah metode tambahan untuk mengidentifikasi virus tertentu yang tidak bisa didapatkan dengan metode checksum biasa. Pada penelitian ini, pendekatan lain yang digunakan untuk mengidentifikasi sebuah virus adalah melalui celah otomatisasi akses file virus dengan memanfaatkan file Autorun.inf yaitu Heuristic ArrS. Heuristic yang digunakan sebagai metode pelengkap dalam identifikasi virus pada penelitian ini adalah Heuristik Arrs atau kependekan dari *Autorun Read System* adalah heuristic yang dibuat untuk mengeksploitasi informasi yang diciptakan oleh virus/program itu sendiri (untuk virus yang biasanya ada di dalam Flashdisk). Hal ini didasarkan kepada sebgaiian besar teknik virus lokal yang diamati peneliti yang masih mengeksploitasi file Autorun.inf, heuristic Arrs memanfaatkan hal tersebut dengan cara membaca file Autorun.inf tersebut dan mencari file yang dicurigai sebagai induk virus yang membuat Autorun tersebut. Berikut *Flowchart* cara kerja Heuristik Arrs :



Gambar 2.2 Flowchart Heuristik ArrS

Dari gambar diatas dapat dilihat bahwa pada saat antivirus dijalankan, selain memakai checksum sebuah file dengan metode checksum tertentu pada antivirus dilakukan juga pengecekan secara logika. Heuristik memanfaatkan file Autorun.inf yang biasanya digunakan virus untuk berjalan secara otomatis saat sebuah drive dibuka.

Cara kerja heuristik dalam mengidentifikasi sebuah suspect virus adalah :

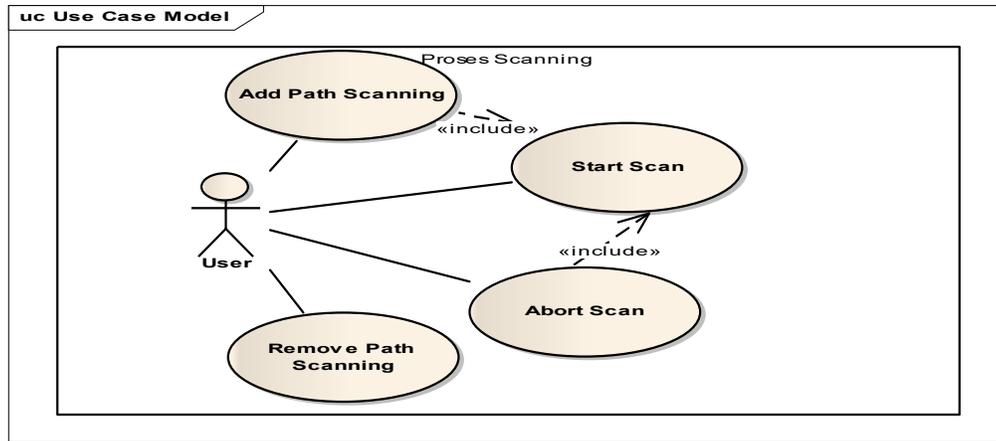
1. *Search engine* pada sebuah antivirus akan melakukan Scanning pada path tertentu saat antivirus tersebut dijalankan dan akan menemukan file.
2. Antivirus akan memanggil ArrS dengan parameter alamat file yang ditemukan.
3. Arrs Akan membaca parameter yang dimasukkan dan memotong 3 karakter dari depan dari parameter tersebut.
4. *Scanning* ArrS-pun dijalankan dengan tujuan menemukan file yang bernama Autorun.inf pada root drive yang dicek.
5. Jika ada, ArrS akan membaca file Autorun.inf tersebut dan menyelidiki apakah parameter (setelah dipotong 3 karakter dari depan) terdapat pada File Autorun.inf. jika tidak, maka proses *Scanning* ArrS akan berhenti. Jika ya, lanjut ke no 6.
6. Integrasikan ArrS dengan Ceksum yang digunakan dalam antivirus, sehingga menghasilkan checksum baru.
7. Masukkan ceksum file ke database secara temporary dan disimpan dalam database internal.
8. Untuk proses scanning yang dilakukan antivirus selanjutnya, maka checksum tersebut akan dipertimbangkan dan di ikutkan dalam database virus antivirus.

2.1.1 Use Case Antivirus

Antivirus yang dirancang memiliki beberapa fungsi dan fitur yang dapat dimanfaatkan oleh user untuk menemukan dan membersihkan virus. Berikut akan dijelaskan proses – proses yang terjadi pada Antivirus yang sedang dirancang.

1. Proses Scanning

Proses Scanning adalah proses yang utama bagi sebuah antivirus, pada proses ini sistem menelusuri *Path* yang diberikan dan membaca file satu persatu untuk didapatkan checksumnya dan dibandingkan dengan bank Virus Antivirus. Berikut adalah *Use Case Proses Scanning* :



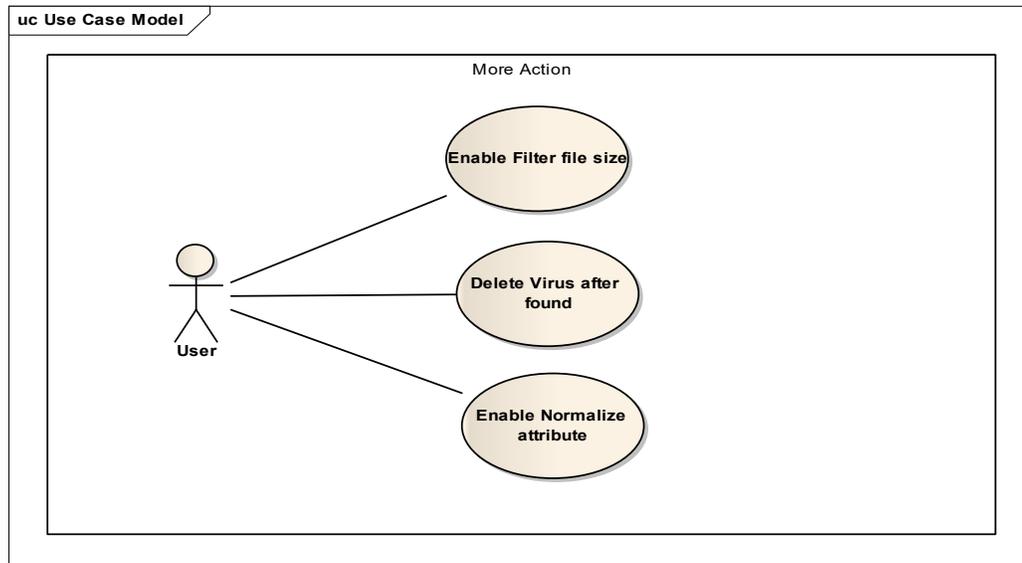
Gambar 2.3 Use Case Diagram : Proses Scanning

Tabel 2.1 Use Case Diagram Proses Scanning

Uses case	Proses Scanning
Deskripsi	<p>User memilih salah satu fungsi :</p> <ul style="list-style-type: none"> - <i>Add Path Scanning</i> : fungsi untuk menambahkan Path Folder yang akan di Scanning oleh Antivirus - <i>Remove Path Scanning</i> : fungsi untuk menghapus Path Folder dan di keluarkan dari range Scanning oleh Antivirus. - <i>Start Scan</i> : fungsi untuk menjelajahi Path Folder yang ditambahkan, membaca File satu-persatu dan menemukan File Virus sampai file di path tersebut telah dibaca semuanya. - <i>Abort Scan</i> : Fungsi untuk menghentikan proses scanning ketika sedang berjalan.
Pre-condition	User memilih fungsi
Post-condition	Fungsi yang dipilih melakukan proses
Flow of event	<ol style="list-style-type: none"> a. User berada pada Form Utama b. User memilih salah satu fungsi c. Sistem memulai proses fungsi yang dipilih user

2. More Action

More action adalah fungsi-fungsi tambahan yang disediakan sistem sebagai tindak lanjut terhadap virus yang ditemukan. *Use Case more action* dapat dilihat pada gambar berikut :



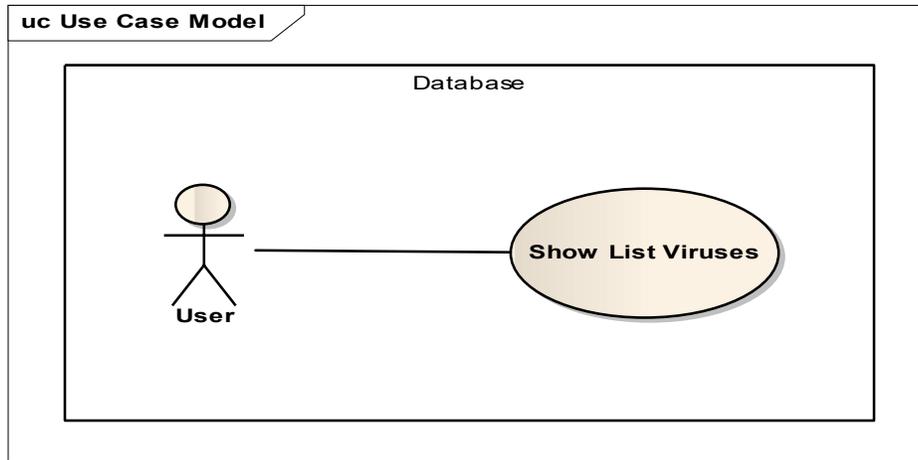
Gambar 2.4 Use Case Diagram : More Action

Tabel 2.2 Use Case Diagram More Action

Uses case	More Action
Deskripsi	<p>User memilih salah satu fungsi :</p> <ul style="list-style-type: none"> - <i>Enable Filter File Size</i> : fitur yang memungkinkan modifikasi pada proses scanning standar, yaitu dengan cara membuat penyaringan akan size sebuah file. - <i>Delete Virus After Found</i> : fitur yang memungkinkan penghapusan terhadap file Virus yang ditemukan secara otomatis. - <i>Enable Normalize Attribute</i> : fitur yang memungkinkan pengembalian Attribute File yang telah dimodifikasi Virus sebelumnya menjadi normal kembali.
Pre-condition	User memilih fungsi
Post-condition	Fungsi yang dipilih melakukan proses
Flow of event	<ol style="list-style-type: none"> a. User berada pada Form Options b. User memilih salah satu fungsi c. Sistem memulai proses fungsi yang dipilih user

3. List Virus

List Viruses digunakan sebagai informasi kepada user mengenai daftar virus yang ada pada Bank Antivirus yang dirancang. *Use Case List Virus* adalah :



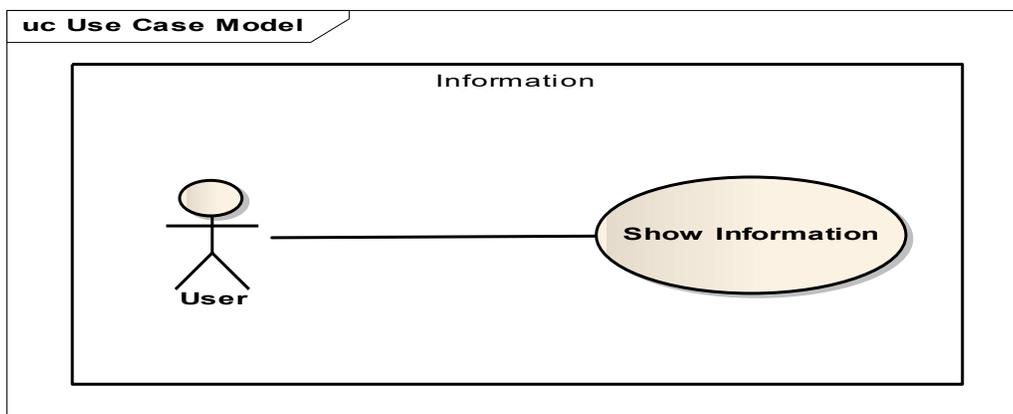
Gambar 2.5 Use Case Diagram : Show List Virus

Tabel 2.3 Use Case Diagram List Viruses

Uses case	Show List Virus
Deskripsi	User ada pada Form Utama
Pre-condition	User memilih List Viruses
Post-condition	Sistem menampilkan List Virus yang ada pada bank virus
Flow of event	a. User berada pada Form Utama b. User memilih List Viruses c. Sistem menampilkan List Virus

4. Antivirus Information

Antivirus yang dirancang menggunakan model pengembangan *Prototype*, sehingga sangat besar kemungkinan kedepannya akan dilakukan perubahan pada Antivirus ke arah yang lebih baik lagi. Oleh karena itu, Informasi mengenai versi dan pembuat Antivirus dapat dilihat melalui About.



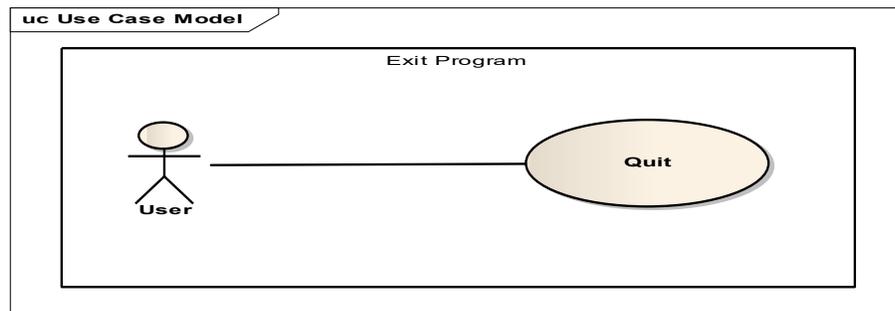
Gambar 2.6 Use Case Diagram : Information

Tabel 2.4 Use Case Diagram Information

Uses case	Show Information
Deskripsi	<i>User</i> ada pada Form Utama
Pre-condition	<i>User</i> memilih About
Post-condition	Sistem menampilkan informasi mengenai Antivirus, seperti : Versi, pembuat, dll.
Flow of event	a. <i>User</i> berada pada Form Utama b. <i>User</i> memilih About c. Sistem menampilkan Informasi mengenai Antivirus

5. Exit

Setelah proses *scanning* selesai dan telah dilakukan proses terhadap virus, maka *User* dapat keluar dari sistem.



Gambar 2.7 Use Case Diagram : Exit Program

Tabel 2.5 Use Case Diagram Exit Program

Uses case	List Virus
Deskripsi	<i>User</i> ada pada Form Utama
Pre-condition	<i>User</i> memilih Quit
Post-condition	<i>User</i> keluar dari program
Flow of event	a. <i>User</i> berada pada Form Utama b. <i>User</i> memilih Quit c. <i>User</i> keluar dari program

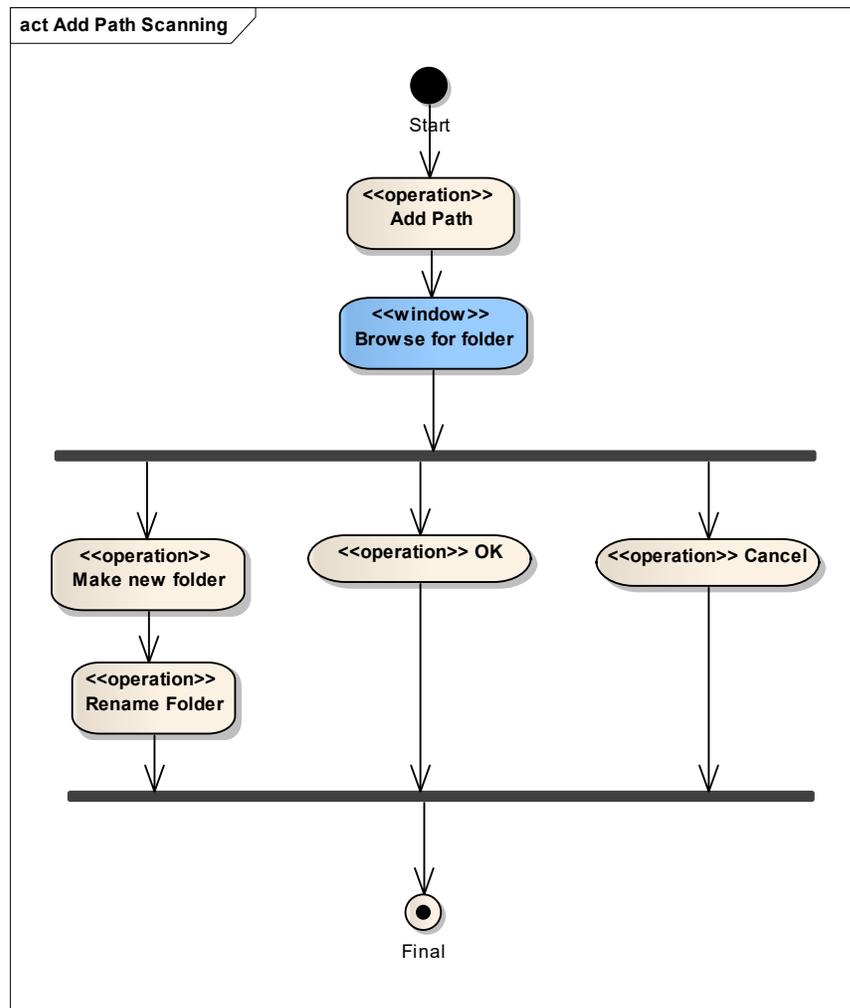
Tabel 2.6 Implementasi Use Case Diagram

NO	Layanan	State
1	3.4.1.1 Proses Scanning	
	• Add Path Scanning	3.4.2.1 Add Path Scanning
	• Remove Path Scanning	3.4.2.2 Remove Path
	• Start Scan	3.4.2.3 Scanning
2	3.4.1.2 More Action (Options)	
	• Enable Filter File Size	3.4.2.4 Enable Filter File Size
	• Delete Virus After Found	3.4.2.5 Delete Virus After Found

	<ul style="list-style-type: none"> • Enable Normalize Attribute 	3.4.2.6 Enable Normalize Attribute File
3	3.4.1.3 List Viruses	
	<ul style="list-style-type: none"> • List Virus 	3.4.2.7 List Viruses
4	3.4.1.4 Information	
	<ul style="list-style-type: none"> • About 	3.4.2.8 About
5	3.4.1.5 Exit	
	<ul style="list-style-type: none"> • Quit 	3.4.2.9 Quit

2.1.2 State Diagram

1. Add Path Scanning

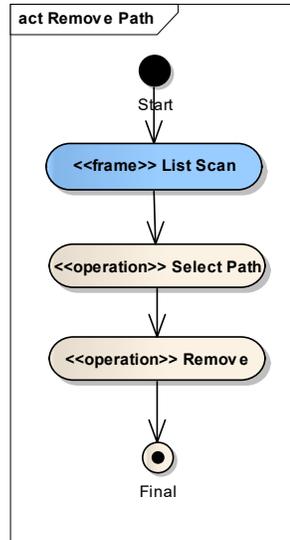


Gambar 2.8 State Diagram : Add Path Scanning

Layanan *add Path* dimaksudkan agar *user* dapat menentukan Folder mana yang akan di *Scan*, *user* dapat memilih *New Folder* untuk membuat sebuah Folder Baru,

atau memilih folder yang ada di dalam *root drive* komputer *user* kemudian **OK**, atau **Cancel** untuk membatalkan perintah.

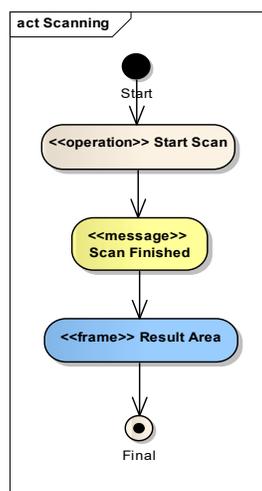
2. Remove Path



Gambar 2.9 State Diagram : Remove Path Scanning

Layanan *Remove Path* berfungsi untuk menghapus *Path* yang tidak akan diikuti sertakan dalam proses *Scanning*. *Path* yang tidak akan diikuti tersebut dipilih pada *List Scan* kemudian pilih perintah **Remove**, maka *Path* yang dipilih tadi akan dihapus dari *List Scan*.

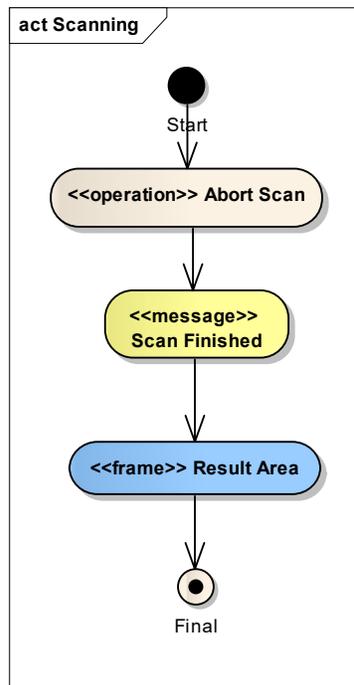
3. Scanning



Gambar 2.10 State Diagram : Scanning

Proses *Scanning* adalah proses utama pada sebuah Antivirus, *Scanning folder* dimulai dari *Path* yang telah ditambahkan oleh User sebelumnya. Namun, apabila User tidak menambahkan *Path* untuk di *Scan*, maka sistem akan menggunakan *Path %RootSystem%\Windows* sebagai default inisialisasi proses *scanning*. Antivirus akan membaca File satu persatu sampai File pada Folder tersebut habis, kemudian Antivirus akan menampilkan pesan bahwa proses *Scanning* telah selesai dan apabila ada File Virus yang ditemukan maka File tersebut akan disimpan dan ditampilkan pada **Result Area**.

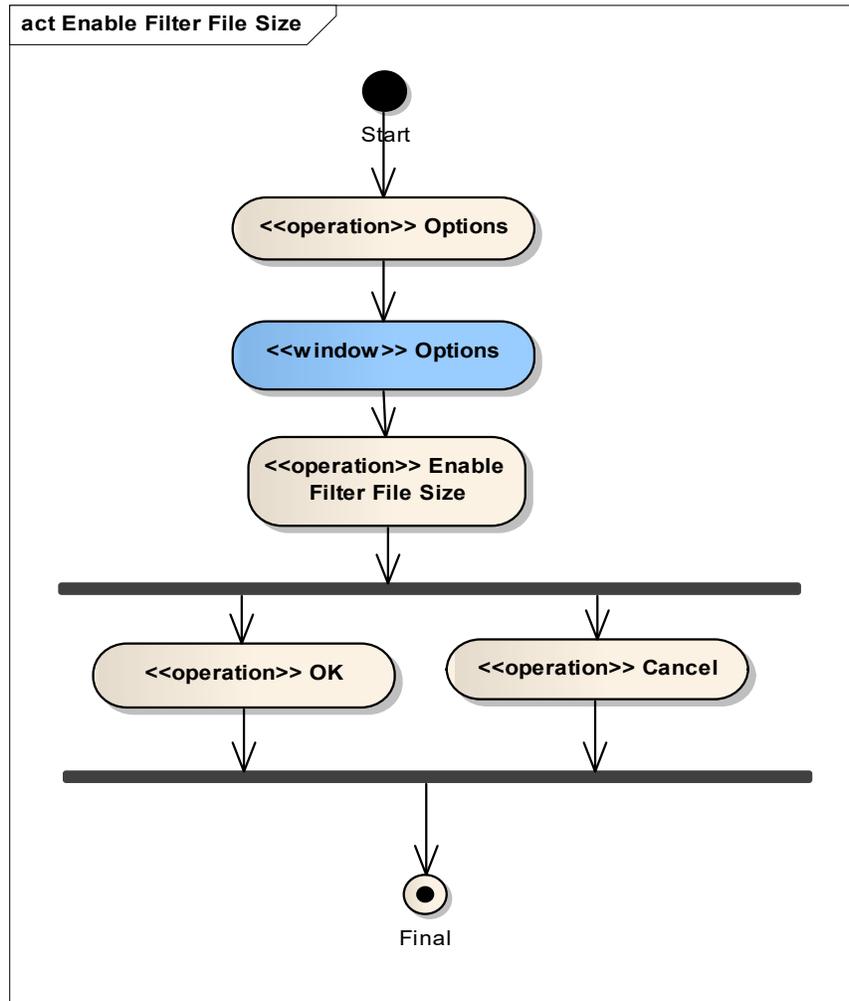
4. Abort Scan



Gambar 2.11 State Diagram : Abort Scan

Layanan *Abort Scan* digunakan jika *user* memilih untuk menghentikan proses *scanning* yang tengah berjalan, sehingga memunculkan pesan bahwa *scan* telah selesai dilaksanakan. Hasil dari proses *scanning* akan ditampilkan pada **Result Area**.

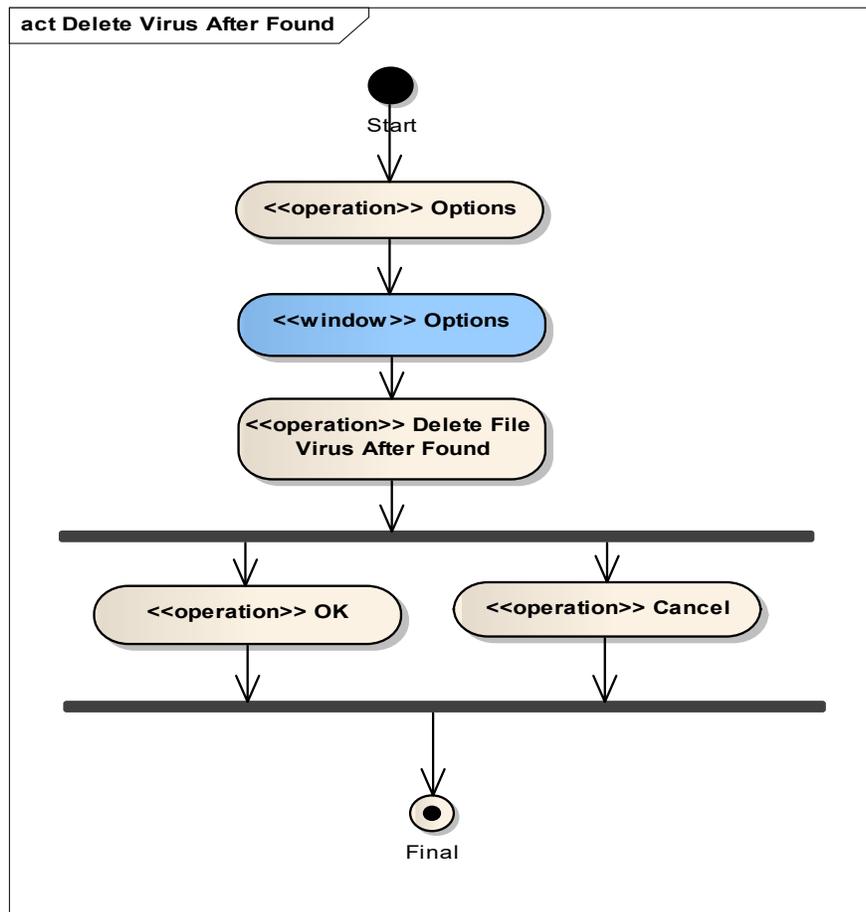
5. Enable Filter File Size



Gambar 2.12 State Diagram : Enable Filter Size

Layanan *Enable Filter File Size* dapat diakses oleh user pada menu Options, pemilihan dianjurkan dilakukan sebelum proses *scan* dimulai sebagai patokan dan inisialisasi *Role Scan*. Fitur *Enable Filter File Size* dimunculkan dalam bentuk **Checkbox** dan *user* dapat memilih **OK** untuk melanjutkan proses atau **Cancel** untuk membatalkan proses.

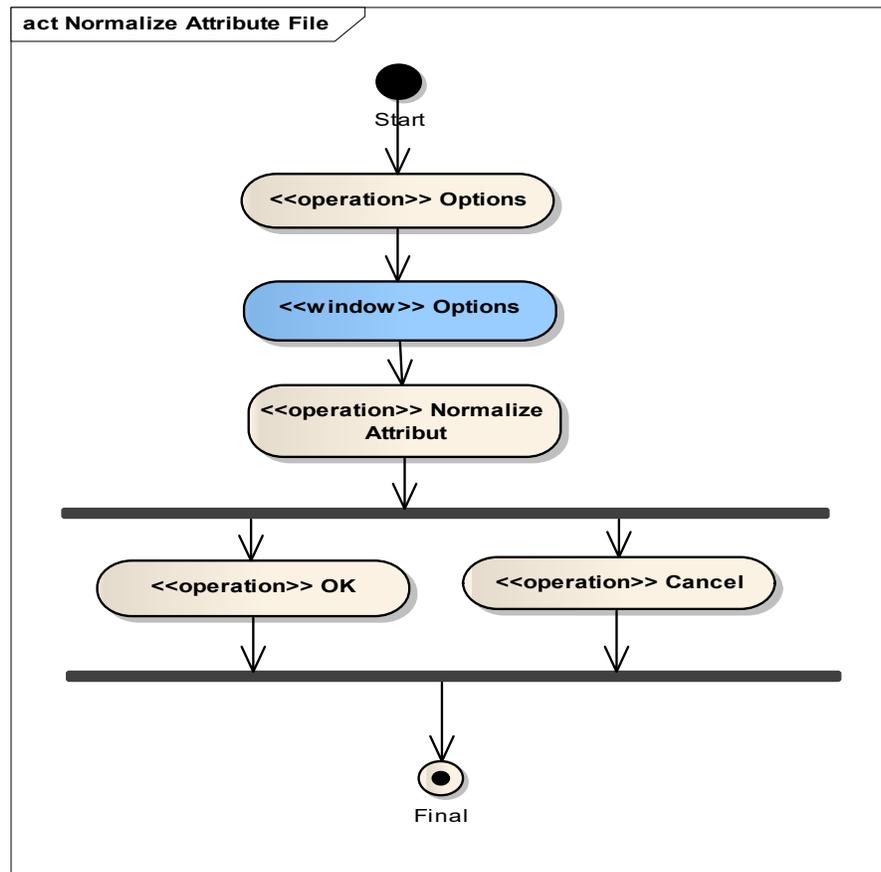
6. Enable Delete Virus



Gambar 2.13 State Diagram : Enable Delete Virus

Layanan *Delete Virus After Found* dapat diakses oleh *user* pada menu **Options**, pemilihan dianjurkan dilakukan sebelum proses *scan* dimulai sebagai patokan dan inialisasi *Role Scan*. Fitur Delete Virus After Found dimunculkan dalam bentuk **Checkbox** dan *user* dapat memilih **OK** untuk melanjutkan proses dengan akibat File Virus yang ditemukan pada saat proses *Scanning* akan langsung dihapus atau **Cancel** untuk membatalkan proses.

7. Enable Normalize Attribute File

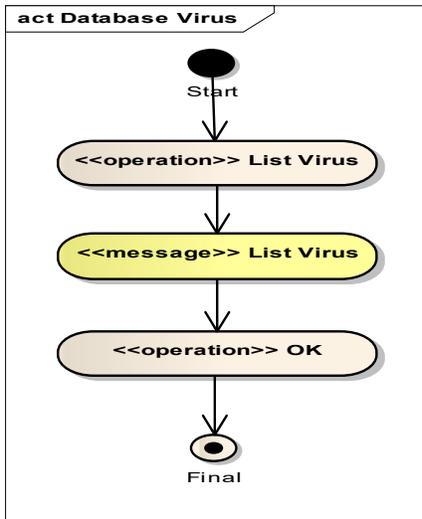


Gambar 2.14 State Diagram : Enable Normalize Attribute File

Layanan *Enable Normalize Attribute File* dapat diakses oleh user pada menu **Options**, pemilihan dianjurkan dilakukan sebelum proses *scan* dimulai sebagai patokan dan inialisasi *Role Scan*. Fitur *Enable Normalize Attribute File* dimunculkan dalam bentuk **Checkbox** dan user dapat memilih **OK** untuk melanjutkan proses dengan akibat File Virus yang ditemukan akan secara otomatis dinormalkan kembali Attributnya (yang sebelumnya telah dimodifikasi oleh Virus) atau **Cancel** untuk membatalkan proses.

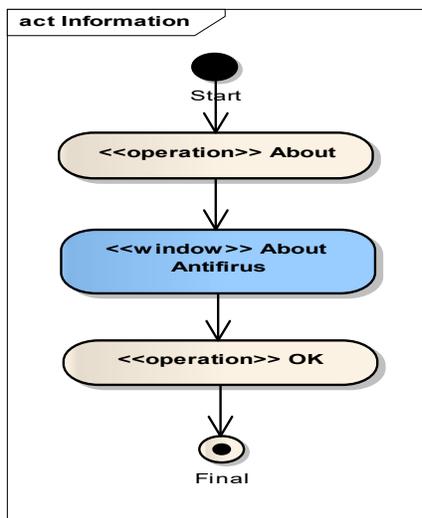
8. List Viruses

Layanan List Virus disediakan bagi User yang ingin mengetahui Daftar Virus apa saja yang telah masuk ke dalam Bank Checksum Virus pada Antivirus. Layanan ini dapat diakses pada menu List Virus kemudian sistem akan menampilkan pesan berupa List Virus yang telah terdaftar Checksumnya pada Antivirus. User dapat memilih **OK** untuk keluar dari pesan List Virus dan kembali ke **MainForm**.



Gambar 2.15 State Diagram : List Viruses

9. About

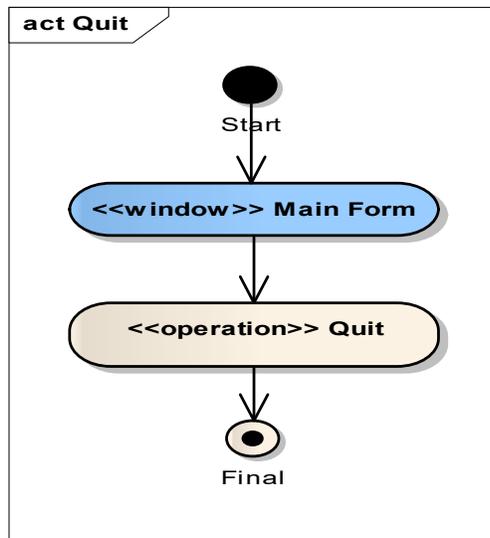


Gambar 2.15 State Diagram : About

Layanan *About* dimaksudkan agar user dapat mengetahui informasi mengenai Antivirus seperti : Versi, Pembuat dan Hak Cipta. Layanan ini dapat diakses pada menu About dan sistem akan memunculkan Window baru berisi informasi mengenai Antivirus, pilih **OK** untuk keluar dari menu About.

10. Quit

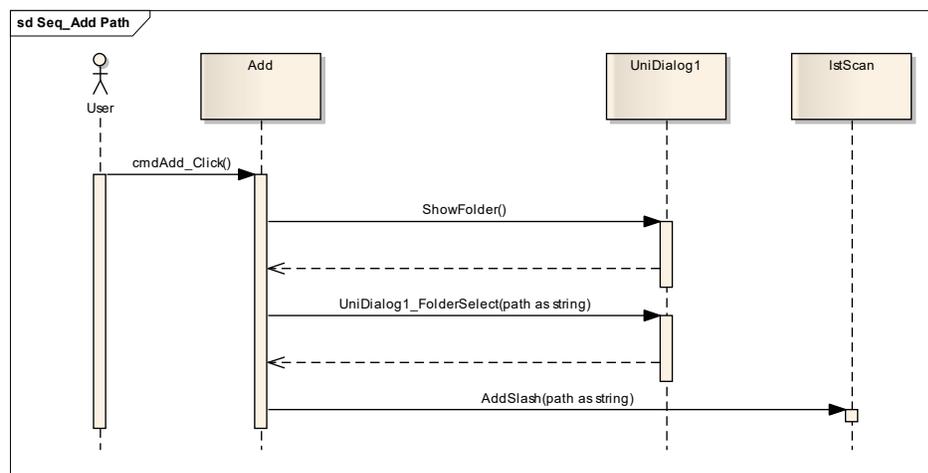
User dapat memilih menu Quit untuk keluar dari aplikasi Antivirus.



Gambar 2.16 State Diagram : Quit

2.1.3 Sequence Diagram

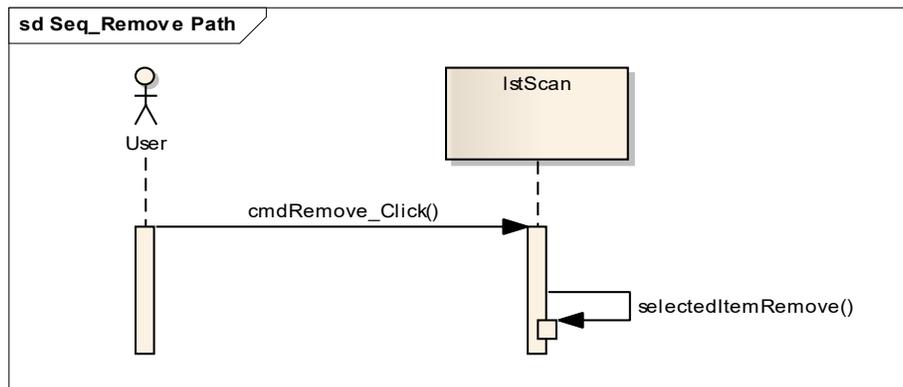
1. Add Path



Gambar 2.16 Sequence Diagram : Add Path

Saat user memilih *Add Path* maka akan dikirimkan fungsi **ShowFolder** ke UniDialog1 untuk memunculkan folder yang ada pada komputer sebagai nilai balikan, kemudian *user* akan memilih *path folder* dan hasilnya akan di tampilkan pada **IstScan**.

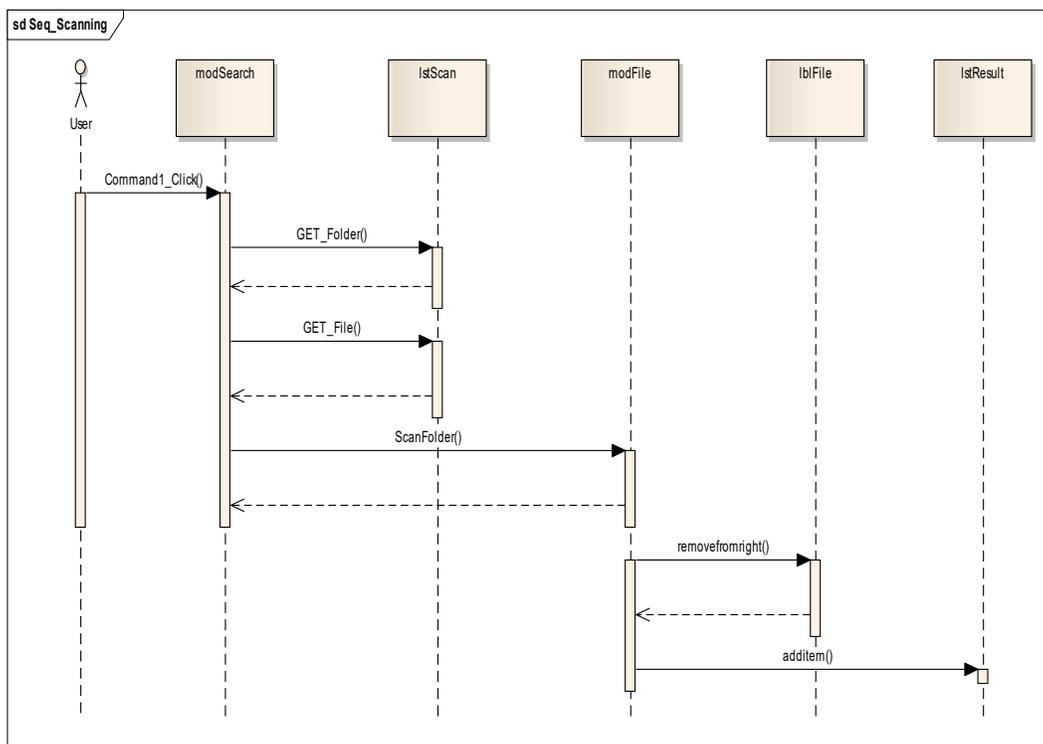
2. Remove Path



Gambar 2.17 Sequence Diagram : Remove Path

Jika *user* memilih perintah *Remove* maka **IstScan** akan menghapus item path yang telah dipilih oleh *user*.

3. Scanning

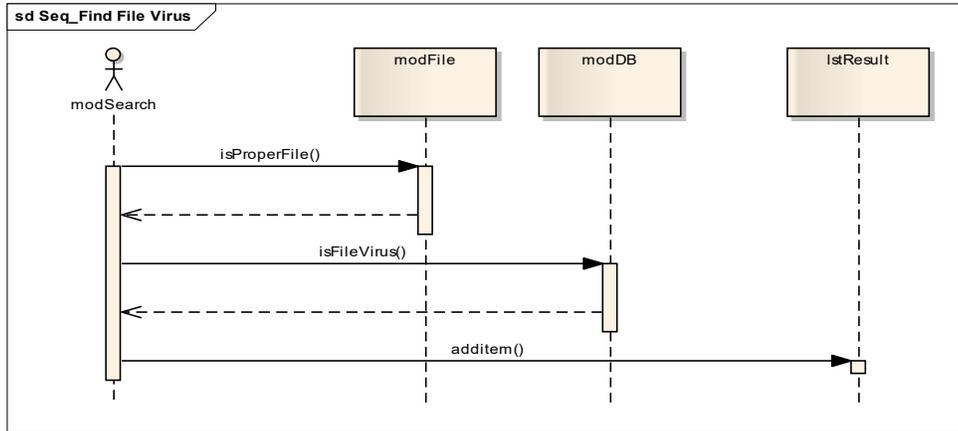


Gambar 2.18 Sequence Diagram : Scanning

Saat *user* menekan tombol *Start Scan*, maka pencarian akan dimulai dengan menggunakan **modSearch** dimulai dari folder dan file yang berada pada **IstScan** setiap file yang ditemukan akan dilakukan proses pembacaan oleh **modFile** dan

lbfFile akan memunculkan *path* yang sedang di *scanning*. [dilanjutkan ke **Sequence Diagram Find Virus File**]

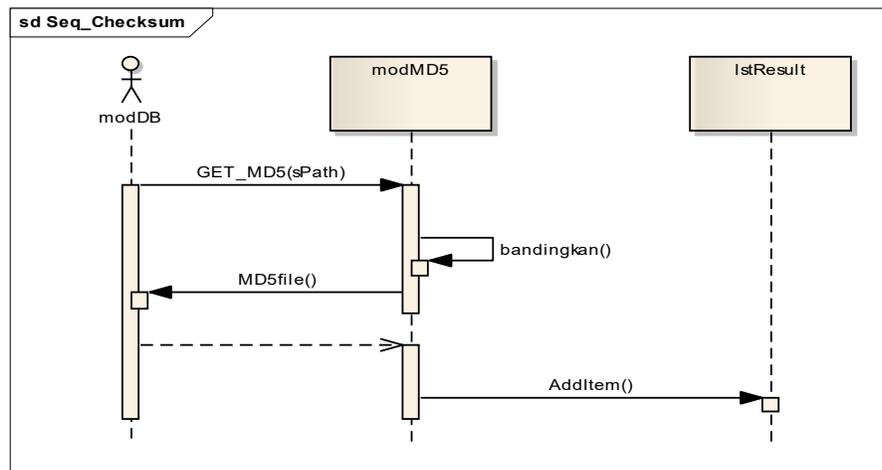
4. Find File Virus



Gambar 2.19 Sequence Diagram : Find Virus File

Setelah pembacaan dilakukan terhadap file yang berada pada **IstScan**, file-file tersebut akan dicek dari extensinya dahulu (dalam penelitian ini extensi File Virus dibatasi hanya berupa : EXE, DLL, VBS, VMX, DB, COM, SCR dan BAT) oleh **modFile**, jika cocok maka pengecekan dilanjutkan dengan pengecekan Checksum dan pendekatan Autorun serta dibandingkan dengan database virus yang dimiliki antivirus. Jika terjadi kecocokan, maka **isFileVirus** akan bernilai **TRUE**, sebaliknya jika tak ada kecocokan data MD5, nilai fungsi **isFileVirus** akan diatur sebagai **FALSE**. Hasilnya akan ditampilkan di **IstResult**. [dilanjutkan ke **Sequence Diagram Checksum MD5 implementation**]

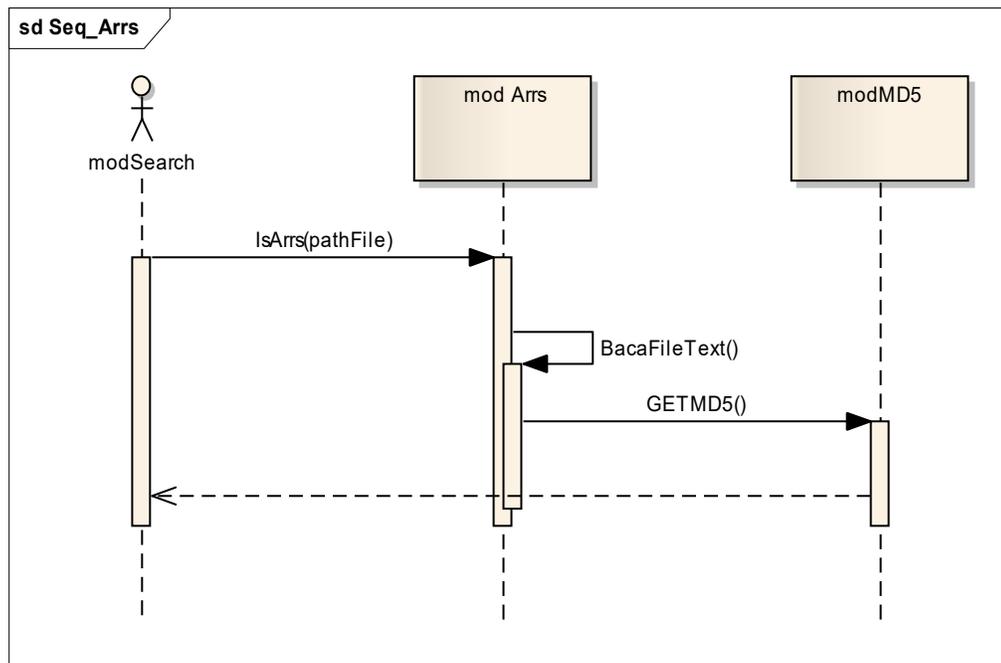
5. Checksum MD5 implementation



Gambar 2.20 Sequence Diagram : MD5 implementation

Di dalam **modDB**, file yang di *scan* akan didapatkan checksum MD5-nya dengan menggunakan fungsi **GET_MD5()**. Setelah didapat checksumnya file-file tersebut akan dibandingkan dengan database virus yang dimiliki antivirus. pengecekan dilakukan dengan melakukan perulangan sebanyak jumlah virus pada database, ketika proses perulangan ada kecocokan nilai pada *Variable* masterMD5 dengan nilai MD5 file yang sedang diuji, informasi akan dimunculkan pada **IstResult**. [dilanjutkan ke **Sequence Diagram Heuristic ArrS Implementation**].

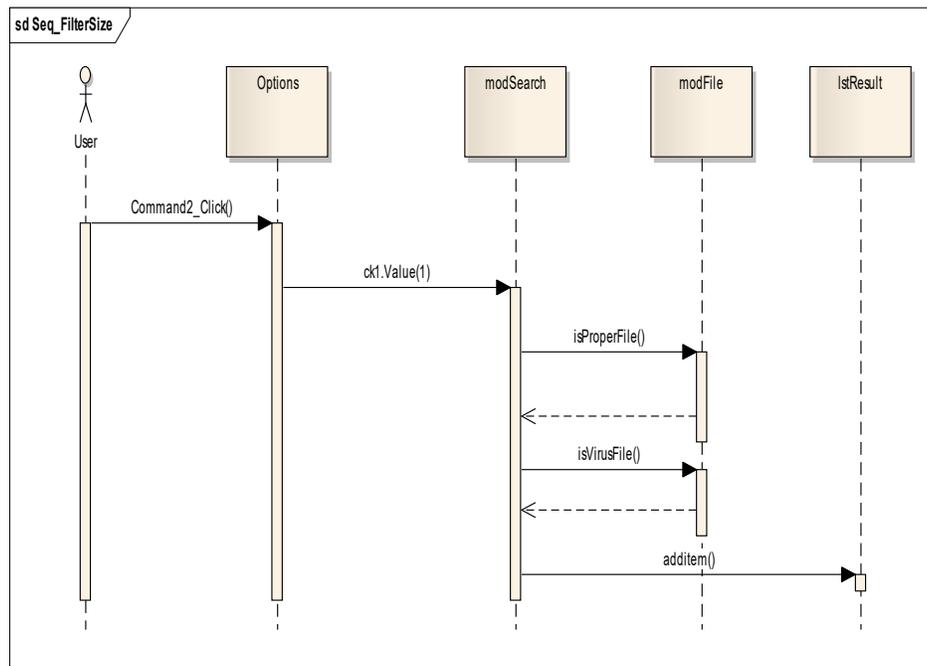
6. Heuristic ArrS Implementation



Gambar 2.21 Sequence Diagram : ArrS Implementation

Sama seperti pengecekan file pada **modMD5**, file-file yang di *Scan* juga akan diperiksa ekstensi nya. Jika terdapat ekstensi "Autorun" maka **modArrS** akan mengirimkan file tersebut ke **modMD5** untuk didapatkan checksum MD5nya sehingga checksum dari file autorun yang baru akan dibandingkan dengan **masterChecksumAutorun** pada *database* dan diikutsertakan dalam perbandingan checksum file dan bank virus.

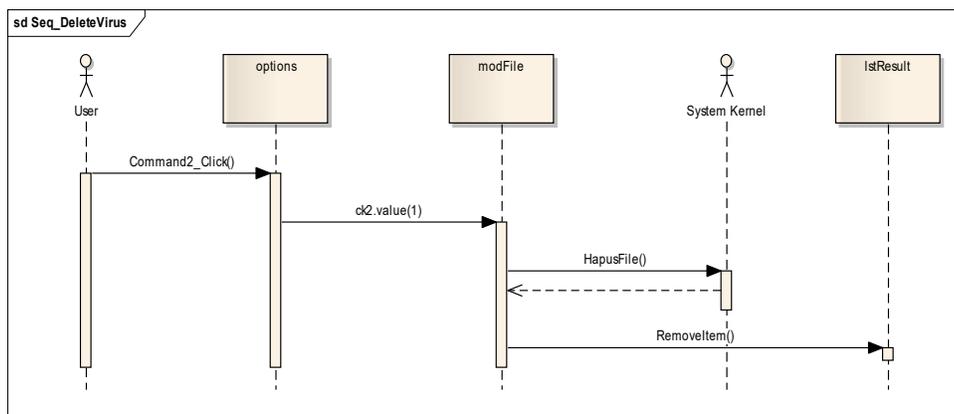
7. Filtering File Size



Gambar 2.22 Sequence Diagram : Filtering File Size

Saat *user* melakukan pilihan *Enable Filter File Size*, maka nilai **Checkbox1** akan bernilai 1, hal tersebut akan merubah *role scanning* yang awalnya pengecekan dilakukan terhadap semua size file menjadi scanning file yang memiliki size diatas atau dibawah *default size File Scanning*. Scanning dilakukan dengan memanggil class **modFile** dan hasilnya akan ditampung di **IstResult**.

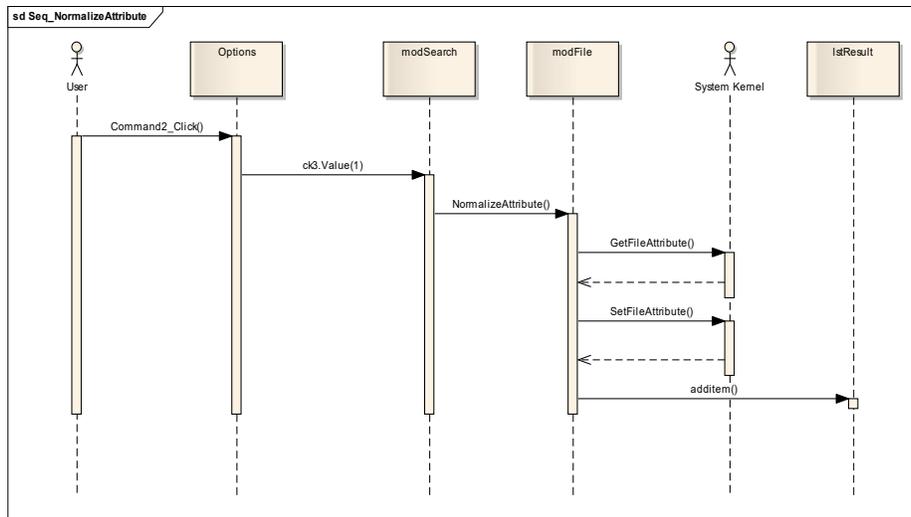
8. Delete Virus File



Gambar 2.23 Sequence Diagram : Delete Virus File

Delete virus secara otomatis saat file virus ditemukan akan dilakukan apabila user mengaktifkan pilihan **Delete Virus after Found** pada menu **Options**. Sehingga nilai **ck2** akan menjadi 1 dan mengakibatkan penghapusan file tersebut dengan memanfaatkan fungsi **HapusFile()** pada sitem kernel, hasilnya akan ditampilkan di **IstResult**.

9. Normalizing File Attributes



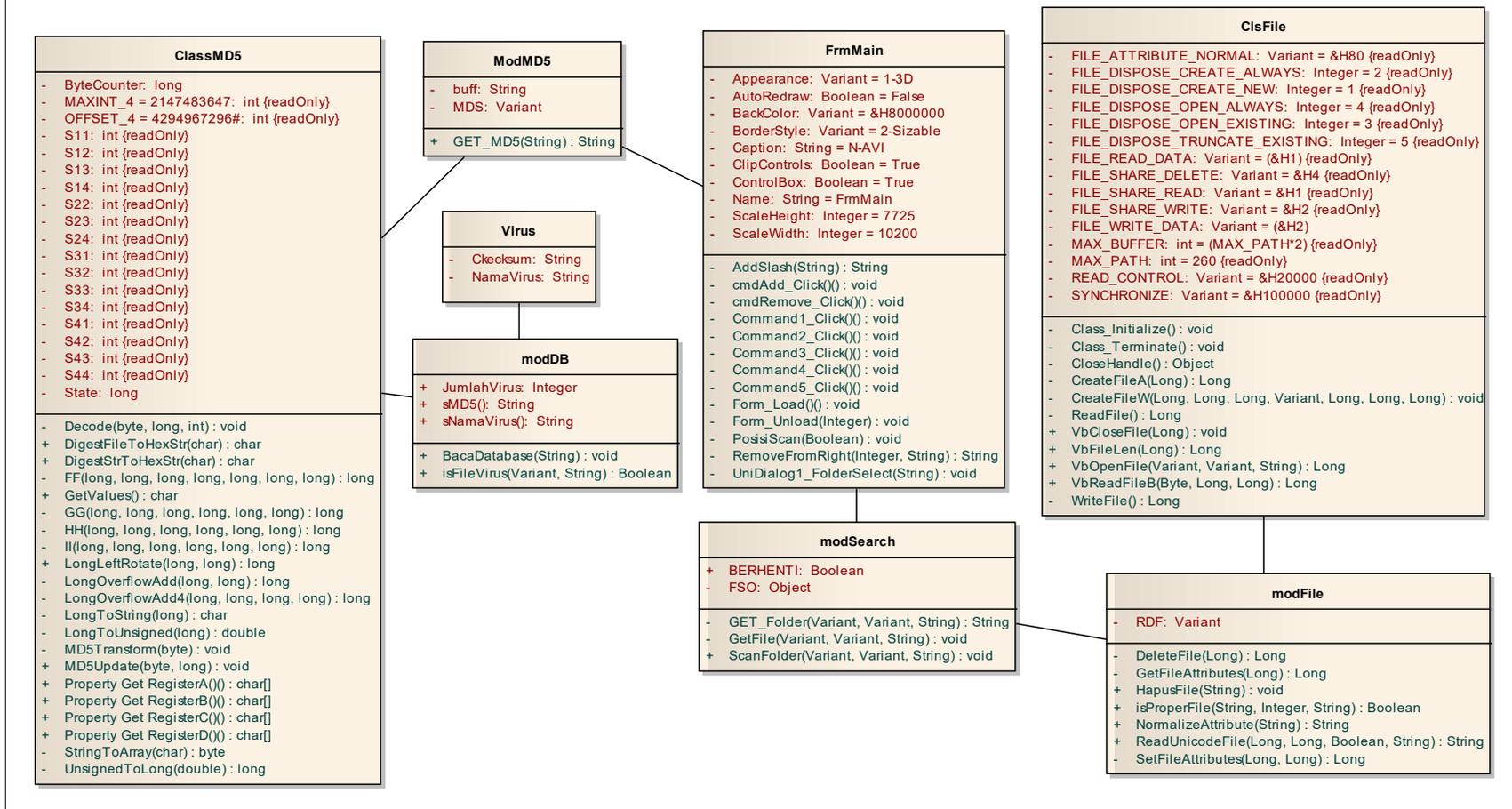
Gambar 2.23 Sequence Diagram : Normalizing File Attribute

Jika user memilih pilihan *Enable Normalize attribute file* pada pilihan **Options**, **ck3** akan bernilai 3 sehingga saat proses *scanning* terjadi **modFile** akan mengakses **SystemKernel** untuk mendapatkan *attribute* dari file yang sedang *discanning* (**GetFileAttribute()**) dan mengirimkan nilai balikan yang bisa saja bernilai 4 (file system), 2 (hidden) atau 6 (hidden + file system). Kemudian attribut file tersebut akan dinormalkan kembali (default value : 0) dan hasilnya akan ditampung pada **IstResult**.

2.2. Perancangan Data

Perancangan data yang dibuat disajikan dalam bentuk Class diagram. Berikut gambar Class Diagram yang dibuat.

class Class Model



Gambar 2.24. Class Diagram

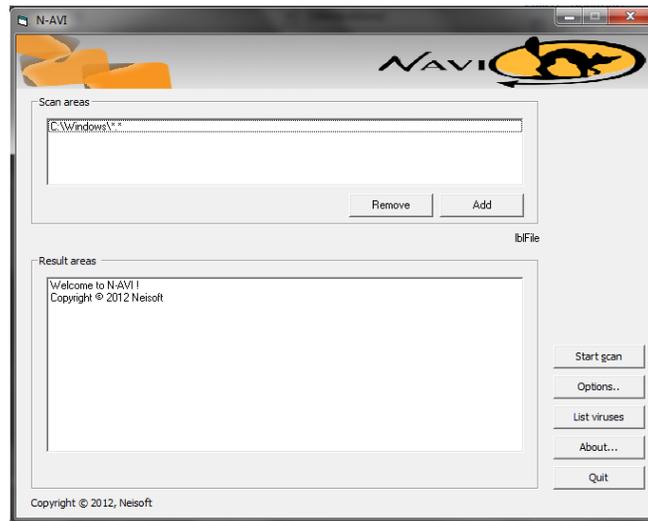
III. IMPLEMENTASI

3.1 Antarmuka Prototype

Implementasi sistem merupakan tahap menterjemahkan hasil perancangan berdasarkan pada analisis kedalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Seluruh kode program perangkat lunak sistem ditulis dengan menggunakan bahasa Visual Basic.

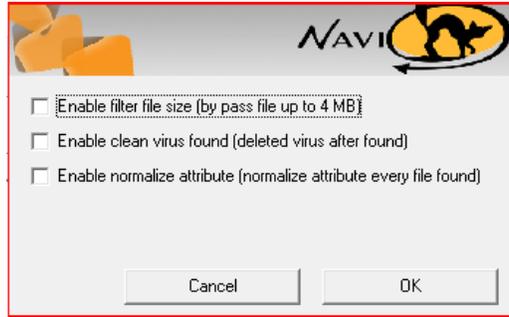
Kelancaran didalam perancangan dan implementasi aplikasi tidak terlepas dari dukungan perangkat keras, perangkat lunak dan pengguna yang akan mengoperasikan aplikasi.

1. Tampilan Utama Antivirus



Tampilan antarmuka di atas adalah tampilan utama dari Antivirus yang dirancang, Tombol **Start Scan** berfungsi untuk memulai proses *Scanning*. Progress *scanning* akan di tampilkan pada Label **Scanning Files ...** dan hasil temuan virus akan ditampilkan pada Result Area. Jika user ingin melihat pilihan/fitur tambahan yang disediakan oleh antivirus, *user* dapat menekan tombol Options. Tombol **List Virus** digunakan untuk melihat Daftar Virus yang ada pada bank data Antivirus. Dan jika *user* ingin melihat informasi mengenai antivirus, user dapat menekan tombol **About** dan tombol **Quit** untuk keluar dari Aplikasi Antivirus.

2. Form Options

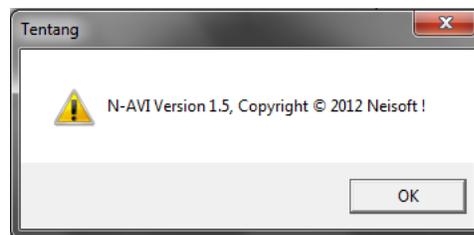


Tampilan layar diatas merupakan rancangan tampilan Form **Options** yang digunakan *user* untuk memilih pilihan yang disediakan oleh antivirus yaitu : **Enable Filter File Size, Enable Deleted Virus After Found, Enable Normalize Attribute.** Tombol **OK** untuk melanjutkan atau **Cancel** untuk membatalkan perintah

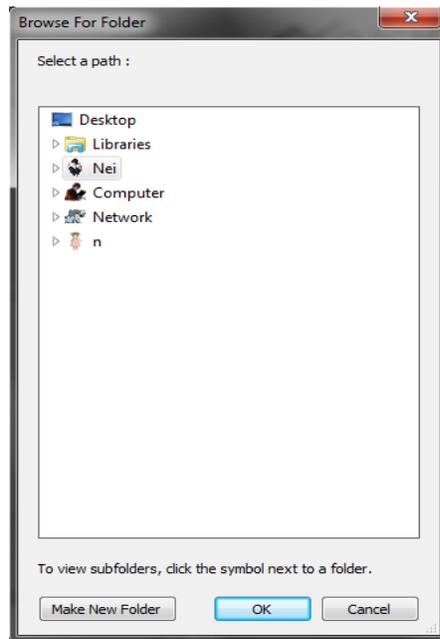
3. Form List Viruses



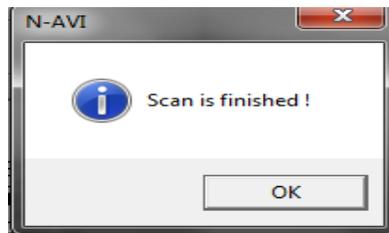
4. Form About



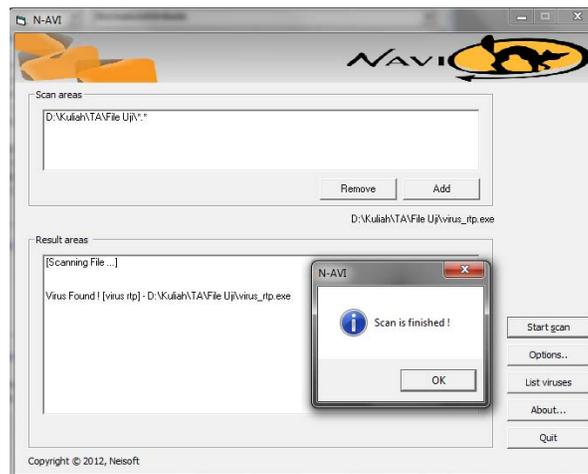
5. Form Browse For Folder



6. Message Box



7. Proses scanning dan deteksi virus



Tampilan layar diatas adalah tampilan Antivirus yang melakukan operasi *Scanning* dan *Identifying* sebuah Virus. Dengan syarat : Ada *Path* yang di *Scan*, kemudian di ambil Checksum File dan bandingkan dengan bank checksum Antivirus jika cocok Virus yang teridentifikasi akan di tampilkan di **Result Area** dan program akan menampilkan **message box** bahawa proses *Scanning* telah selesai dilaksanakan.