JURNAL

TEKNOLOGI DAN INFORMATIKA

Volume : 8, No : 2 - November 2012 ISSN : 1412 – 9361

Diterbitkan oleh :
Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM)
SEKOLAH TINGGI SAINS DAN TEKNOLOGI INDONESIA
(ST-INTEN)
Jalan Ir.H.Juanda No.126C Telepon : 022-2504523, Fax : 022- 2510390
Bandung 40132

Volume 8, No. 2 – November 2012 ISSN 1412 – 9361

JURNAL TEKNOLOGI DAN INFORMATIKA ST-INTEN

Alamat
Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Sekolah Tinggi Sains dan Teknologi Indonesia
(ST-INTEN)
Jl.Ir.H.Juanda No.126C Bandung
Telp. 022-2504523 Fax. 022-2510390

Jurnal Teknologi dan Informatika ST-INTEN diterbitkan secara berkala dua kali setahun bulan Mei dan November oleh LPPM ST-INTEN. Redaksi mengundang partisipasi para dosen dan peneliti untuk penyumbangkan tulisan berupa hasil penelitian, tinjauan ilmiah, konsepsi dalam ilmu pengetahuan, yang berkaitan dengan bidang teknologi dan informatika.

PENGANTAR REDAKSI

Dengan gembira kami hadir kembali menjumpai Anda pada penerbitan Jurnal Teknologi dan Informatika nomor terakhir untuk volume yang kedelapan. Kali ini kami menghadirkan lima buah artikel yang berasal dari bidang teknik informatika, teknik elektro dan teknik arsitektur.

Artikel pertama dari bidang teknik informatika hasil karya dari Heri Purwanto yang membahas tentang IT blueprint yang dapat diimplementasikan pada Rencana Induk Teknologi Informasi Pendataan Pendidikan.

Artikel kedua mengenai penelitian perancangan mikrokontroler AT89S51 sebagai komponen utama, untuk putaran motor stepper yang mempunyai presisi tinggi dengan tingkat kesalahan 2,5 %. Artikel ini ditulis oleh Iksal Rachman dari Jurusan Teknik Elektro ST-INTEN.

Pada artikel yang ketiga dari bidang teknik informatika yang ditulis oleh Peti Savitri dengan judul Database Design For System Of Production Floor Layout Designing At “CAR for MFLaSh” Prototype.

Artikel berikutnya Yuda Wastu bersama mahasiswanya menuliskan hasil penelitiannya tentang penggunaan beton pracetak pada pembangunan gedung apartemen. Pengembang masih memilih metoda beton konvensional karena berkaitan dengan nilai jual gedung, lokasi dan pasar kota Bandung.

Jurnal Teknologi dan Informatika volume kedelapan diakhiri dengan sebuah artikel penelitian dari teknik informatika. Artikel ini ditulis oleh Rima Rahmawati dan Agus Nursikuwagus, dalam penelitiannya ini bertujuan mengimplementasikan algoritma MD5 dan Heuristik ARRS untuk mengidentifikasi virus yang menyerang sistem komputer.

Demikian kelima artikel yang dapat Anda simak dalam Jurnal Teknologi dan Informatika kali ini. Kami menunggu konstribusi artikel ilmiah dari anda semua. Selamat Membaca.

Redaksi

DAFTAR ISI

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

PERANCANGAN ANTIVIRUS DENGAN MENGGUNAKAN METODE MD5 DAN HEURISTIK ARRS

[1]Rita Rahmawati, [2]Agus Nursikuwagus
Jurusan Teknik Informatika, Sekolah Tinggi Sains dan Teknologi Indonesia
Email :neitahyuga@yahoo.com; agus235032@yahoo.com

Abstrak

Penelitian ini bertujuan untuk mengimplementasikan algoritma MD5 dan Heuristik ARRS. Algoritma MD5 dan Heuristik ARRS digunakan untuk mengidentifikasi virus yang menyerang sistem komputer. Virus yang dikenali adalah bersifat malware yang dapat mengakibatkan kerugian terhadap sebuah system. Penanggulangan serta tindakan pencegahan dilakukan untuk meminimalisir terjadinya kerusakan sistem yang disebabkan oleh virus. Penanganan MD5 menggunakan pendekatan checksum 32 bit. Sedangkan Heuristik ARRS mengenali sifat antivirus yang memanfaatkan file Autorun sebagai starting point pengaktifan virus dalam sebuah sistem. Pemanfaat model objek digunakan sebagai pendekatan perancangan program antivirus. Adapun diagram yang digunakan adalah Use case Diagram, State Diagram, Sequence Diagram dan Class Diagram. Use Case diagram digunakan untuk menggambarkan proses yang terjadi di dalam program antivirus. State diagram digunakan untuk menggambarkan kejadian system, Sequence diagram digunakan untuk melihat keterurutan proses system. Sedangkan class diagram digunakan untuk menunjukan objek yang digunakan dalam system. Hasil akhir dari penelitian ini adalah program antivirus yang dapat mengenali file yang mengandung virus, serta tindakan yang dilakukan terhadap virus yang dikenali.

Kata kunci : *Antivirus, Checksum, MD5, Heuristik ArrS*

1. Pendahuluan

1.1 Latar Belakang

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

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

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. 41

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.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah yaitu : Bagaimana Virus dapat teridentifikasi dan menetukan tindakan selanjutnya terhadap file yang terinfeksi virus File.

1.3 Maksud dan Tujuan

Adapun Maksud dari penelitian ini adalah membangun sebuah program antivirus yang dapat mengidentifikasi virus dan melakukan fungsi *scanning* dan *deleting* virus File. Sedangkan tujuan dari penelitian ini adalah : menerapkan metode MD5 dan Heuristik ArrS pada antivirus untuk proses *identifiying*, *scanning* dan *deleting* sebuah file yang terinfeksi virus file.

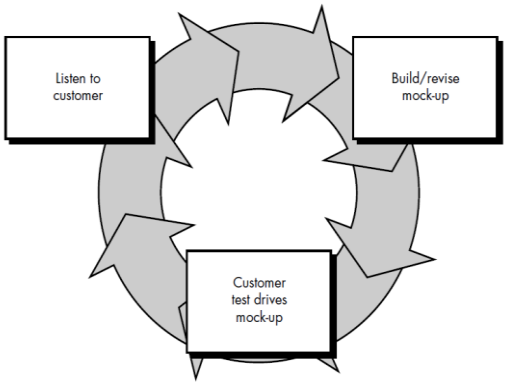
1.4 Batasan Masalah

Pembuatan program antivirus ini diperlukan adanya batasan agar mendapatkan hasil yang baik yaitu: 1) program ini difungsikan sebagai media identifikasi Virus berjenis Virus File yang berextensi EXE, DLL, VBS, VMX, DB, COM, SCR dan BAT.
2) program ini hanya difungsikan sebagai media *identifying, scanning* dan *deleting* Virus file yang dapat digunakan pada Sistem Operasi Windows.
3) program yang dibuat merupakan sebuah *prototype* yang mengimplementasikan Metode MD5 dan Heuristik ArrS pada Antivirus.

1.5. Metode Pengembangan Perangkat Lunak

Pengembangan suatu perangkat lunak, sangat dibutuhkan metode. Metode yang digunakan untuk penelitian ini adalah prototype modeling. Gambar 1, merupakan model prototipe yang dibuat digambarkan oleh Pressman.

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361



Gambar 1. Prototype Modeling (Pressman-2001)

Tahapan yang dilakukan pada prototype modeling yaitu : 1) *Listen to customer* : *developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya; 2) *Build/Revise Mockup* yaitu perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*; 3) *Customers test drives mock-up*: klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*.

2. Teori

2.1. Virus

Berikut definisi virus menurut para ahli : virus adalah program yang dapat menginfeksi program lain dengan memodifikasi program tersebut untum memasukkan kemungkinan penggandaan dari dirinya sendiri (Cohen (1984, 6: section 1.3]. Sedangkan definisi lain, virus komputer adalah segala bentuk program komputer yang bersifat merusak ataupun tidak yang datang dengan sendirinya tanpa diinginkan oleh pengguna computer serta memiliki kemampuan untuk bertahan hidup. [2 : 23].

Berbagai bentuk jenis virus sudah dikembangkan, antara lain *malware*. Ciri– ciri sebuah *malware* yang dapat dikategorikan sebagai Virus : 1) Dapat melakukan infeksi terhadap file yang dapat dijadikan inang seperti exe, scr, com dll. Sehingga bila file dijalankan akan mengaktifkan virus. 2) Manipulasi yang dilakukan lebih tinggi dari worm, sehingga lebih bertahan dalam sebuah sistem dibandingkan dengan sebuah worm. 3) Hampir tidak ada duplikasi karena dilengkapi dengan teknik infeksi file lain. 4) Memakai ikon standar *executable* [1 : 4].

2.2. Antivirus

Untuk membasmi atau membersih file yang terinfeksi virus, maka diperlukan antivirus. Antivirus adalah sebuah program yang berfungsi melindungi, baik mencegah maupun membasi virus sebelum dan sesudah masuk dalam sebuah Sistem Operasi yang ada dalam sebuah komputer. [5]. Sebuah program disebut antivirus apabila memenuhi syarat antara lain: a) mampu mencari seluruh file dalam sebuah path; b) memiliki checksum sebagai pengingat virus beserta databasenya; c) mampu men-*terminate* / membunuh proses virus standard; d) mampu menghapus virus yang ditemukannya dari *memory fixed drive* maupun *removable drive*. [1 : 26].

Model pencarian bisa dilakukan dengan checksum dan heuristik. *Checksum* adalah suatu nilai untuk membedakan suatu file dengan cepat. [1 : 23]. Sebuah file dapat

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

mengalami kerusakan berupa pengurangan data (*data corrupt*), hal ini bisa diketahui dengan menggunakan checksum error. Checksum error memiliki sensitivitas yang sangat tinggi, sehingga kemungkinan dua file yang berbeda tidak akan memiliki checksum eror yang sama. Checksum yang digunakan pada penilitian ini adalah Checksum MD5 yaitu checksum yang memiliki kecepatan hash yang lebih cepat dan mudah dimodifikasi sesuai keinginan dengan nilai standar hash 32 digit hex.

Heuristik adalah teknik yang dipakai setelah penggunaan checksum dalam pendeteksian virus. Teknik ini merupakan teknik pendekatan untuk mencurigai bahwa sebuah file adalah virus atau bukan. Algoritma pencarian virus yang mengguakan heuristic adalah *ArrS Heuristic*, terlahir di 2008. Heuristic ini diperkenalkan oleh A. M. Hirin seorang programer antivirus. Heuristik ini dibuat berdasarkan exploitasi virus-virus lokal terhadap file autorun. [2]

2.3. Model Objek

Untuk memudahkan dalam penyelesaian suatu penelitian, maka digunakan bantuan model objek. Model objek yang digunakan seperti yang disampaikan oleh Geofrry Spark. Sedangkan diagram yang digunakan adalah use case diagram, class diagram, dan state diagram. *(Geoffrey Spark : 2010).*

Uses case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah ”apa” yang diperbuat sistem, dan bukan ”bagaimana”. User / pengguna aplikasi di lambangkan dengan *actor* sedangkan apa yang bisa dilakukan oleh pengguna pada aplikasi dilambangkan dengan bulatan-bulatan yang terkoneksi dengan *actor*. *State diagram* menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Class* adalah sebuah spesifikasi yang jika menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

3. Analisis dan Perancangan

3.1. Analisa Proses Kerja Antivirus

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 standard; d) mampu menghapus virus yang ditemukannya dari *memory fixed drive* maupun *removable drive*. [5 : 26]

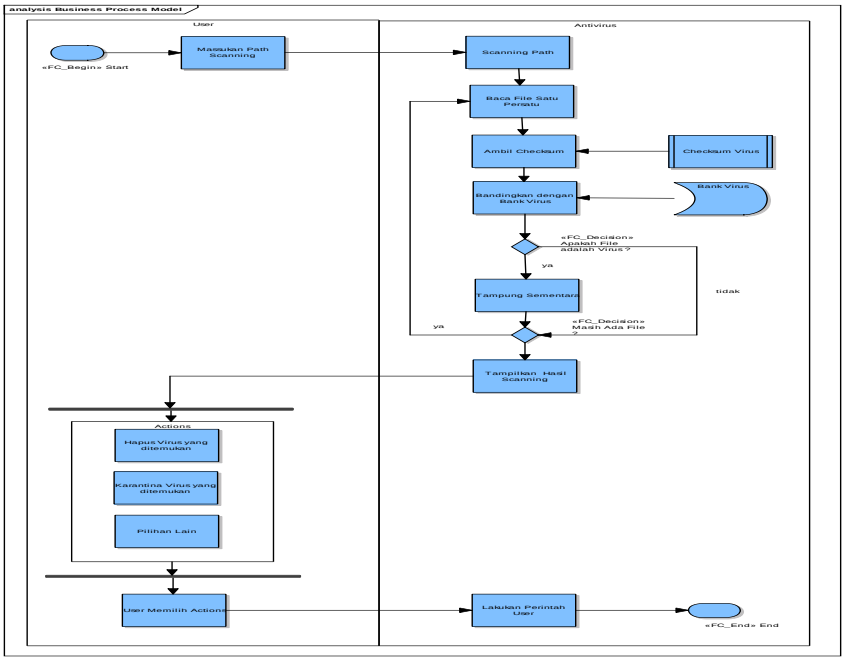
Antivirus memanfaatkan *Checksum* yang dimiliki oleh sebuah file virus dan mencocokannya dengan

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

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. *Flowchart* Antivirus umum digambarkan pada gambar 2.

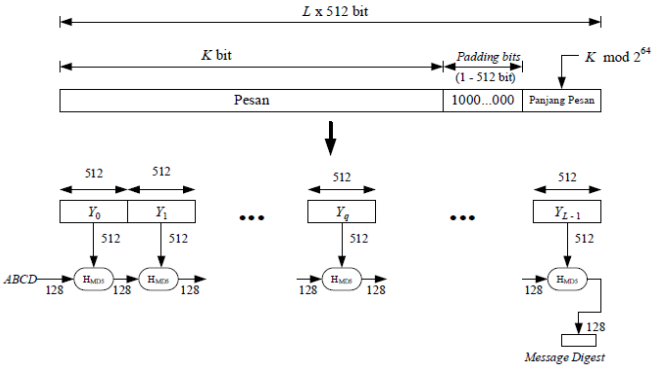
3.2. Implementasi Algoritma MD5 pada Antivirus

Karakteristik dari metode ini adalah : 1) Penambahan padding bits; 2) penambahan nilai panjang pesan semula; 3) inisialisasi penyangga MD; 4) pengolahan pesan dalam blok berukuran 512 bit. Algoritma umum dari MD5 digambarkan pada gambar 3.



Gambar 2. Flowchart Umum Antivirus

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

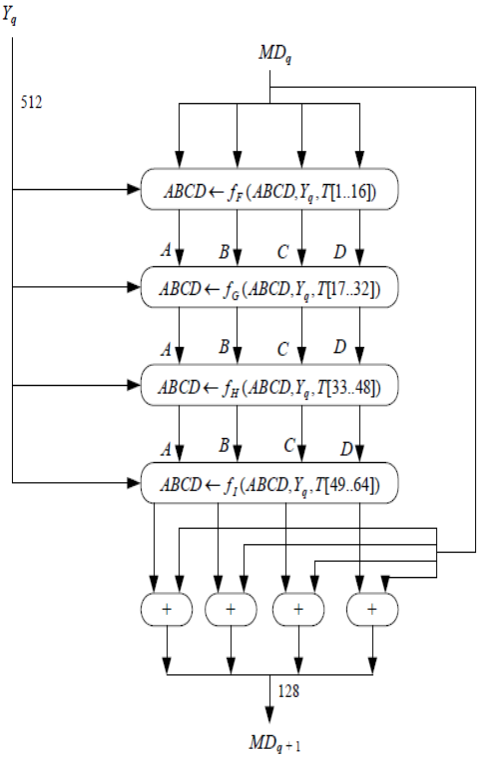


Gambar 3. Algoritma Umum MD5 [3]

Proses *scanning* dan *identifying* sebuah virus oleh antivirus dilakukan dengan menggunakan metode tertentu, disesuaikan dengan kebutuhan dan algoritma yang di yakini dapat mendeteksi virus dengan cepat dan tepat. Dengan pemisahan blok data sebesar 512 bit, proses untuk mendapatkan nilai Checksumnya pun (*hash function*) menjadi sangat cepat. Algoritma MD5 yang utama beroperasi pada kondisi 128-bit, dibagi menjadi empat word 32-bit, menunjukkan A, B, C dan D. Operasi tersebut di inisialisasi dijaga untuk tetap konstan. Algoritma utama kemudian beroperasi pada masing-masing blok pesan 512-bit, masing-masing blok melakukan pengubahan terhadap kondisi. Pemrosesan blok pesan terdiri atas empat tahap, batasan putaran; tiap putaran membuat 16 operasi serupa berdasar pada fungsi non-linear F, tambahan modular, dan rotasi ke kiri. Ada empat macam kemungkinan fungsi F, berbeda dari yang digunakan pada tiap-tiap putaran:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z) \quad \text{(3.1)}$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z) \quad \text{(3.2)}$$
$$H(X,Y,Z) = X \oplus Y \oplus Z \quad \text{(3.3)}$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z) \quad \text{(3.4)}$$

$\oplus, \wedge, \vee, \neg$ menunjukkan operasi logikan XOR, AND, OR dan NOT. Sehingga cara kerja MD5 dapat digambarkan sebagai berikut :



Gambar 4. Cara Kerja MD5

Pada gambar 4, Yq adalah satu blok berukuran 512 bit tadi yang merupakan bagian dari pesan yang telah ditambahkan padding bit dan tambahan nilai panjang semula. MD adalah *message*

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

diggest 128 bit yang dihasilkan. Proses HMD5 terdiri dari 4 buah putaran, yang masing-masing putaran melakukan operasi dasar MD5 sebanyak 16 kali dan setiap operasi dasar memakai sebuah elemen T. Fungsi-fungsi fF, fG, fH, dan fI masing-masing berisi 16 kali operasi dasar terhadap masukan menggunakan tabel T. Hasil akhir atau string keluaran dari algoritma MD5 merupakan gabungan / penyambungan dari bit-bit di A, B, C, dan D. Modifikasi yang dilakukan pada algoritma MD5 agar dapat digunakan sebagai checksum sebuah virus adalah pada saat di implementasikannya algoritma MD5 tersebut pada proses Scanning Antivirus yang dibuat. Ketika antivirus membaca sebuah file kemudian membandingkannya dengan database virus yang ada, maka disitulah letak penyesuaian algoritma MD5 terhadap file yang sedang di identifikasi. MD5 akan menentukan berapa banyak byte yang akan dibaca dari kiri pada sebuah file, serta penentuan role Limit Size file yang akan dicurigai dan langsung di dapatkan Check sumnya dan kemudian dilakukan pencocokan terhadap database virus yang digunakan. Hal yang perlu diingat, metode yang dipakai untuk mendapatkan checksum sebuah virus pada antivirus haruslah sama dengan metode yang digunakan untuk mendapatkan checksum virus pada database virus, karena jika tidak akan dikhawatirkan terjadinya ketidak cocokan checksum yang didapat dan berujung pada gagalnya sebuah Antivirus dalam mengidentifikasi sebuah virus.

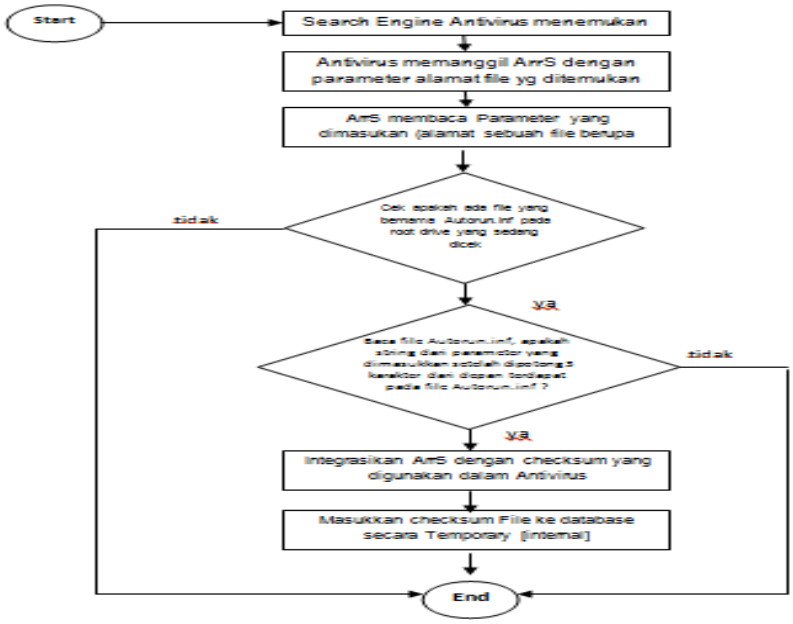
3.3 Implementasi Heuristic ArrS pada Antivirus

Selain dengan memanfaatkan checksum sebuah file, Antivirus juga memerlukan sebuah metode tambahan untuk mengidentifikasi virus tertentu yang tidak bisa didapatkan dengan metode check sum 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 mengexploitasi informasi yang diciptakan oleh virus/program itu sendiri (untuk virus yang biasanya ada di dalam Flashdisk). Hal ini didasarkan kepada sebagaian besar teknik virus lokal yang diamati peneliti yang masih mengeksploitasi file Autorun.inf, heuristik Arrs memanfaatkan hal tersebut dengan cara membaca file Autorun.inf tersebut dan mencari file yang dicurigai sebagai induk virus yang membuat Autorun tersebut.

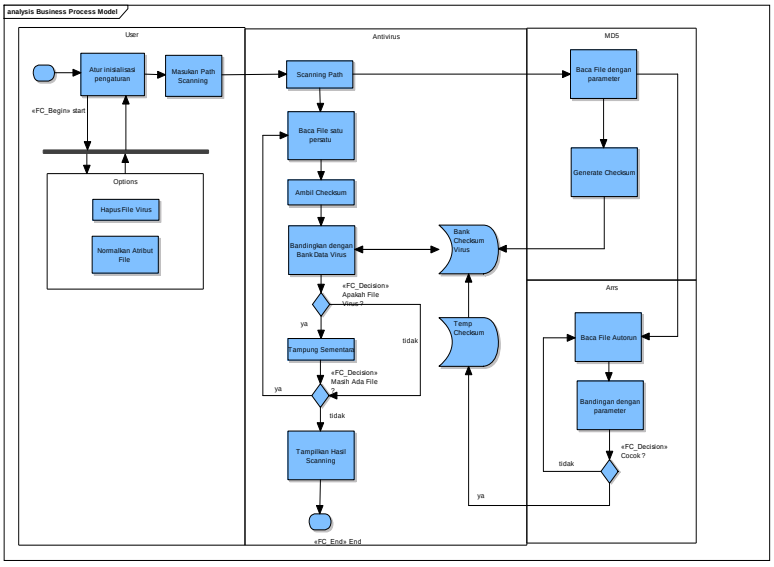
Pada gambar 5, dapat dilihat bahwa pada pada saat antivirus dijalankan, selain memakai check sum sebuah file dengan metode checksum tertentu pada antivirus dilakukan juga pengecekan secara logika. Heuristik memanfaatkan file Autorun.inf yang biasanya digunakan virus untuk berjalan secara otomatis saat sebuah drive dibuka. Dari gambar 5, maka apabila MD5

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

dan Heuristik diimplementasikan pada sebuah antivirus, maka *flowchart* yang didapat seperti pada gambar 6.



Gambar 5 Flowchart Heuristic ArrS



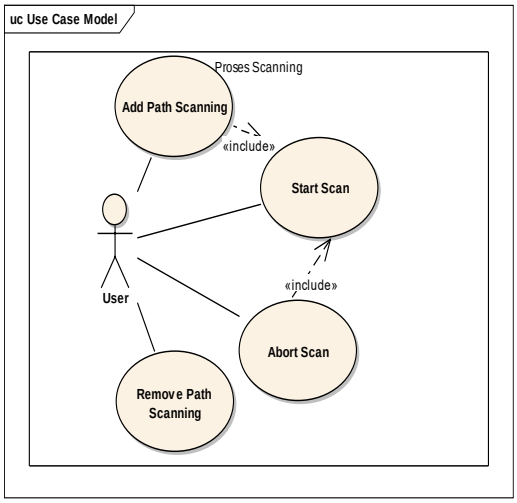
Gambar 6. Flowchart Antivirus yang diajukan

3.4. Perancangan Sistem

3.4.1. Use Case Antivirus

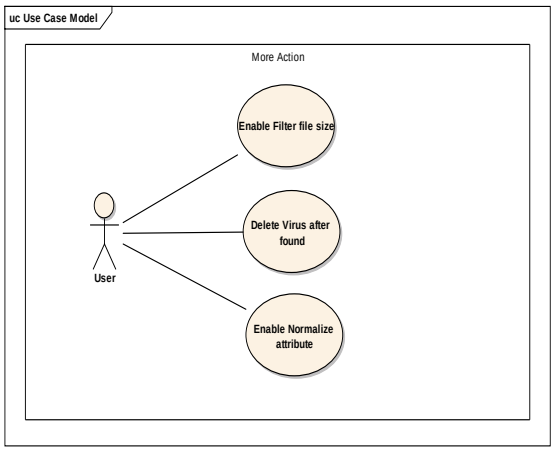
Proses Scanning adalah proses yang utama bagi sebuah antivirus, pada proses ini sistem menelusuri *Path* yang diberikan dan membaca file satu persatu untuk didapatan checksumnya dan dibandingkan dengan bank Virus Antivirus. Proses scanning dapat dilihat pad gambar 7.

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361



Gambar 7. Use Case Diagram : Scanning

More action adalah fungsi-fungsi tambahan yang disediakan sistem sebagai tindak lanjut terhadap virus yang ditemukan. *Use Case more action* dapat dilihat pada gambar 8.



Gambar 8. Use Case Diagram :More Action

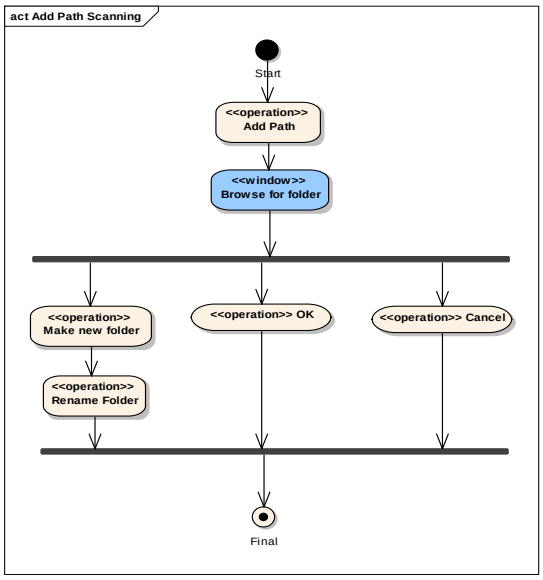
3.4.2. State Diagram

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 Pada gambar 9 merupakan contoh salah satu dari state diagram.

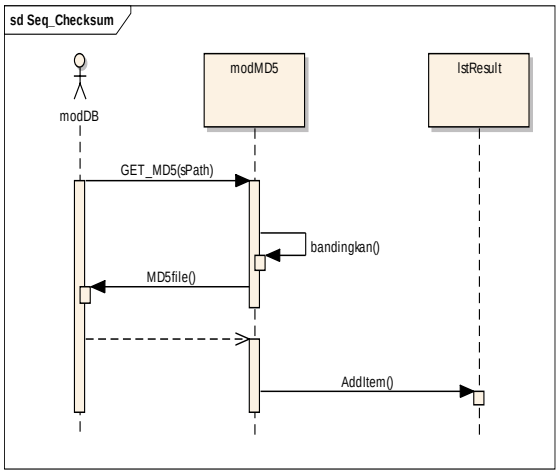
3.4.3. Sequence Diagram

Saat user menekan tombol Start Scan, maka pencarian akan di-mulai dengan menggunakan modSearch dimulai dari folder dan file yang berada pada lstScan setiap file yang ditemukan akan dilakukan proses pembacaan oleh modFile dan lblFile akan me-munculkan path yang sedang di scanning. Setelah pembacaan di-lakukan terhadap file yang berada pada lstScan, file-file tersebut akan dicek dari extensinya dahulu (dalam penelitian ini extensi File Virus dibatasi haya 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. Contoh sequence diagram pada gambar 10, merupakan *MD5 checksum.* Untuk sequence diagram implementasi Heuristics Arrs digambarkan pada gambar 11.

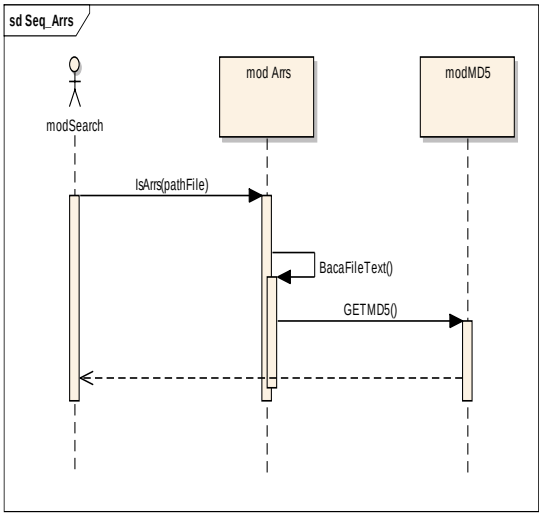
Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361



Gambar 9. State Diagram : Add Path Scanning



Gambar 10. Sequence Diagram : MD5

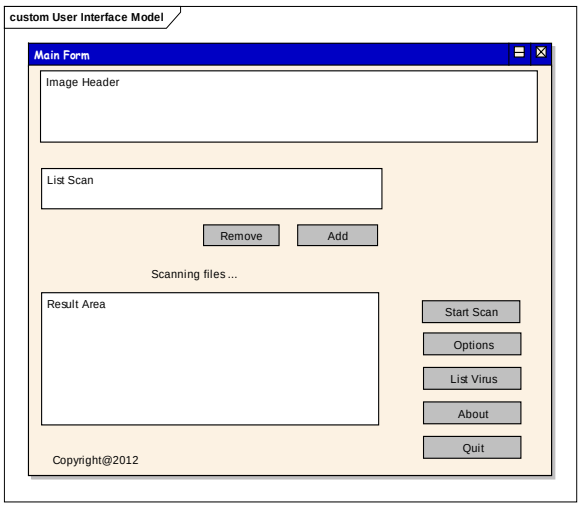


Gambar 11. Sequence Diagram : Heuristik ArrS

3.4.4. Perancangan Antarmuka

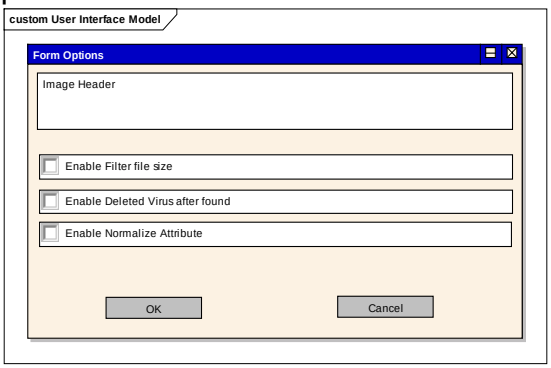
Pada gambar 12, merupakan tampilan layar diatas adalah rancangan aplikasi untuk Main Form Antivirus. Form ini berisi semua Fungsi yang dimiliki antivirus yang dirancang. *User* dapat menambah patfile yang akan di *Scan* dengan menekan tombol **Add** dan atau menghapus *Path File* tersebut dengan menekan tombol **Remove**. Hasilnya akan ditampuilkan pada **List Scan**. 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.

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361



Gambar 12. Perancangan Antar Muka : Main Form

Sedangkan pada gambar 13, 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.

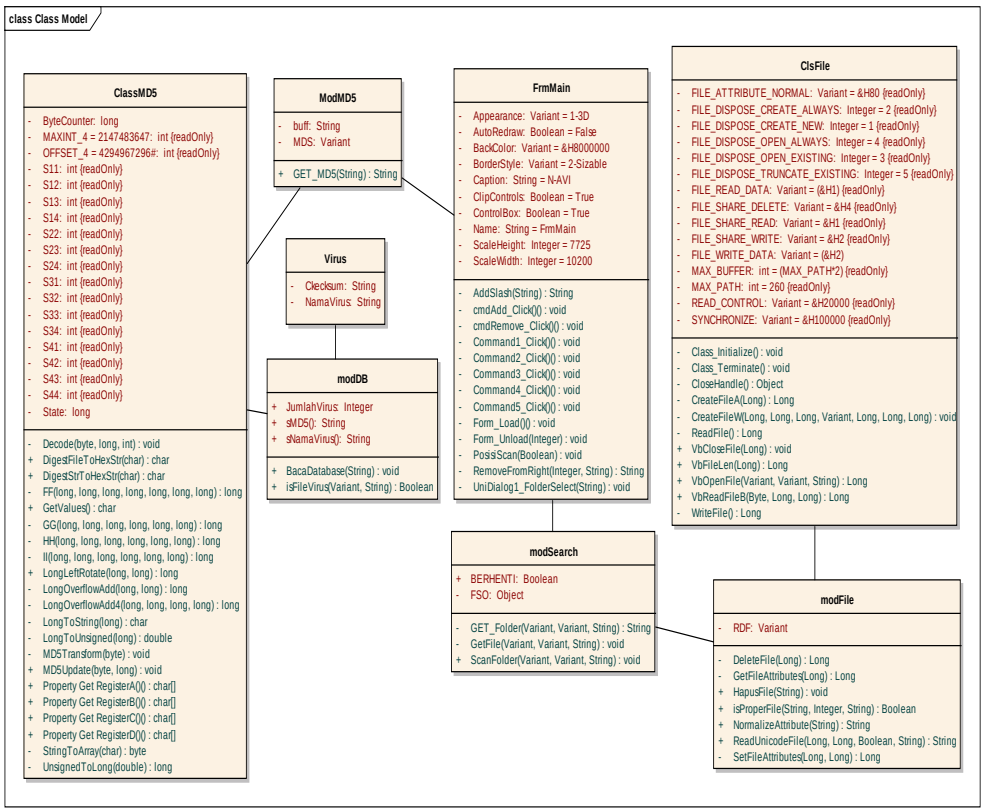


Gambar 13. Perancangan Antar Muka : Options Form

3.5. Class Diagram

Class diagram digunakan untuk menggambarkan hubungan antara objek yang dikerjakan. Pada gambar 13, merupakan class diagram untuk keseluruhan program antivirus.

Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361



Gambar 13. Class Diagram Antivirus

4. Implementasi dan Pengujian

4.1. Implementasi

Berikut adalah implementasi proses berdasarkan modul fungsi pada Antivirus yang dirancang :

Tabel 1. Implementasi Proses

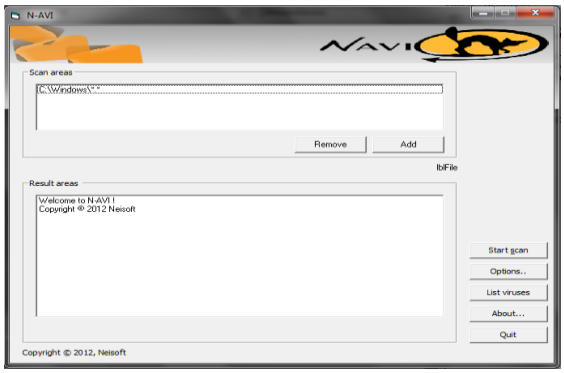
No.	Modul	Fungsi	Kode (VB)
1	Mod MD5	GET_ MD5(s Path)	Buff = ReadUnicodeFile(FileName, False, 1, 5000) Buff = Left(Buff, 5000) MD5.MD5Init MD5.DigestStrToH exStr Buff GET_MD5 = MD5.GetValues Set MD5 = Nothing Exit Function

2	Mod Arrs	IsArrs (pathFi le)	If IsFileX(PathFile) = True Then
			nmFile = Mid(PathFile, 4)
			strDrv = Left(PathFile, 3)
			If Dir(strDrv & "Autorun.Inf", vbNormal Or vbHidden Or vbSystem) <> "" Then isData = OpenFileInTeks(str Drv & "Autorun.Inf")
			If InStr(UCase(isDat a), UCase(nmFile)) > 0 Then
			IsArrs = True
			Else
			IsArrs = False
			End If
			Else
			Exit Function

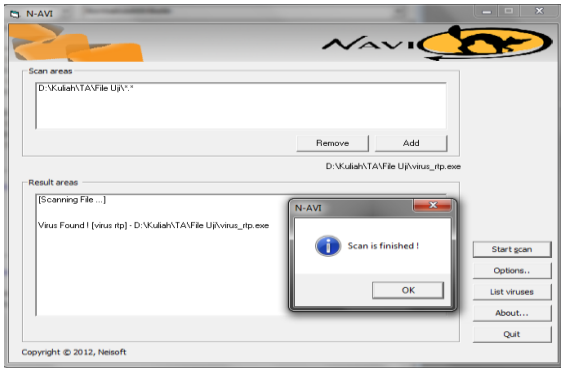
Jurnal Teknologi dan Informatika
Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN)
Volume : 8, No : 2 – November 2012, page 40 - 53
ISSN : 1412 - 9361

			End If End If Exit Function

Hasil dari implementasi rancang-an antar muka, dapat dilihat pada gambar 14 dan 15. Gambar ter-sebut merupakan eksekusi antar muka utama dan memulai pen-carian virus.



Gambar 14. Implementasi Antar Muka : Main Form



Gambar 15. Implementasi Antar Muka : Proses Scaning dan Identifikasi virus

4.2. Pengujian Perangkat Lunak

Pada tabel 2, merupakan pengu-jian terhadap modul program. Pengujian ini hanya dilakukan dengan model black-box, yaitu pengujian yang dilakukan hanya terhadap input dan output yang telah dirancang. Kesesuaian output terhadap rancangan akan menjadi-kan jaminan kesesuaian dengan penelitian yang dilakukan.

Tabel 2. Pengujian tiap modul program

No	Modul	Input	Output	KET
1	AddPath	Path	Path di List Scan	Berhasil
2	Scanning	Path di List Scan	Checksum File	Berhasil
3	Find Virus File	Checksum File dari Path di List Scan	File Virus terdeteksi	Berhasil
4	ModMD5	File	Checksum	Berhasil
5	MODArr S	File	Autorun terdeteksi	Berhasil
6	Filter File Size	File	File dgn parameter	Berhasil
7	Delete Virus File	File Virus Terdeteksi	File Virus Terhapus	Berhasil

5. Simpulan

Berdasarkan hasil yang didapat dalam penelitian ini, maka diper-oleh beberapa kesimpulan sebagai berikut :

1. Metode MD5 dapat digunakan dalam proses identifikasi virus file yag berextensi EXE, DLL, VBS, VMX, DB, COM, SCR dan BAT dengan membandingkan checksum yang didapat dengan checksum di database Virus.
2. Heuristik Arrs dapat digunakan sebagai pendekatan untuk identifikasi virus file dengan memanfaatkan file Autorun.inf.
3. Antivirus yang dibuat dapat melakukan fungsi standar sebuah

6. Saran

Pengembangan dilakukan agar antivirus yag dirancang dapat men-jadi lebih baik lagi, pengembangan tersebut dapat berupa : 1) Pe-nambahan fungsi update secara online; 2) Pengembangan Antivirus untuk sistem operasi yang lain; 3) pengembangan Antivirus untuk

Mobile; 4) penambahan fungsi perlindungan Realtime yang dapat memproteksi PC dan Antivirus itu sendiri

7. Referensi

1. Hirin A. M. (2010),” Cara Praktis Membuat Antivirus Komputer”, Jakarta : Mediakita.
2. Hirin, A. M. (2008), “Sehari Menjadi Programer Antivirus Menggunakan VB 6.0 “, Yogyakarta : Andi.
3. Nugraha, M. Pasca. (2010). ”Perbandingan Algoritma MD4 dan MD5 Serta Implementasinya dalam Kehidupan Sehari-hari”, Makalah IT, Institut Teknologi Bandung.
4. Pressman, Roger. (2001), “Software Engineering : A Practitioner's Approach -Fifth Edition-“, New York – USA : McGraw-Hill.
5. Resha, Ahlul Faradish. (2007), ”Membuat Virus dan Antivirus”, Yogyakarta : Gava Media.
6. Szor, Peter. (2005), “The Art Of Computer Vius Research And Defense”, Addison Wesley Professional.

Bibliografi

Rita Rahmawati,**ST** alumni Jurusan Teknik Informatika ST-INTEN dan lulus pada tahun 2013

Agus Nursikuwagus,MT., dosen dan peneliti di Jurusan Teknik Iinformatika ST-INTEN. Lulus Sarjana Teknik Jurusan Teknik Informatika ST-INTEN tahun 1999. Kemudian melanjutkan studi di Program Magister Informatika Institut Teknologi Bandung (ITB) dan lulus tahun 2005. Sekarang, adalah dosen Kopertis Wilayah IV dpk ST.INTEN