ISSN : 2339-210X

IMPLEMENTASI METODE HEURISTIC ANTIVIRUS DALAM MENDETEKSI MALICIOUS EXECUTABLES

Iksan Hasari Nasution (0811109)

Mahasiswa Program Studi Teknik Informatika STMIK Budi Darma Medan
Jl. Sisingamangaraja No. 338 Simpang Limun Medan
http : // www.stmik-budidarma.ac.id // Email : iksanhasari@gmail.com

ABSTRAK

Antivirus adalah sebuah jenis perangkat lunak yang digunakan untuk mengamankan, mendeteksi, dan menghapus virus komputer dari sistem komputer. Antivirus disebut juga Virus Protection Software. Aplikasi ini dapat menentukan apakah sebuah sistem komputer telah terinfeksi dengan sebuah virus atau tidak. Umumnya, perangkat lunak ini berjalan di latar belakang (background) dan melakukan pemindaian terhadap semua berkas yang diakses (dibuka, dimodifikasi, atau ketika disimpan).

Penelitian ini bertujuan untuk mengetahui tingkah laku dari virus dan worm. mengetahui bagian-bagian apa saja dari sebuah sistem operasi (windows) yang di serang oleh virus dan worm. Membuat sebuah anti virus sendiri.

Metode lain yang dapat dipakai user adalah metode heuristik. Pada metode ini program akan menganggap suatu file adalah virus jika file tersebut mempunyai sifat seperti sifat virus (misalnya merubah nilai registry dan memasuki program start up system).

Kata Kunci : Antivirus, Metode Heuristic Antivirus Dalam Mendeteksi Malicious Executables.

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Seiring dengan pesatnya perkembangan penggunaan komputer sebagai alat bantu manusia di berbagai bidang kehidupan, semakin besar pula jenis *software* yang digunakan.Virus komputer merupakan salah satu *software* komputer yang menjadi ancaman bagi keamanan sistem komputer. Virus komputer sebagai salah satu jenis infeksi elektronik, dapat menyebabkan kerusakan pada sistem komputer yang diserangnya. Para *user* yang komputernya diserang oleh virus merasa akan tidak nyaman terhadap keberadaan virus tersebut yang mungkin akan memperlambat kinerja atau bahkan menghilangkan beberapa fungsi dari komputer.

Antivirus adalah sebuah jenis perangkat lunak yang digunakan untuk mengamankan, mendeteksi, dan menghapus virus komputer dari sistem komputer. Antivirus disebut juga Virus *Protection Software*. Aplikasi ini dapat menentukan apakah sebuah sistem komputer telah terinfeksi dengan sebuah virus atau tidak. Umumnya, perangkat lunak ini berjalan di latar belakang *(background)* dan melakukan pemindaian terhadap semua berkas yang diakses (dibuka, dimodifikasi, atau ketika disimpan).

Metode yang dapat dipakai user sebagai metode pada proses *scanning* salah satunya adalah metode CRC32. Sesuai dengan fungsi utama dari fungsi hashing, CRC32 berfungsi untuk mengambil penanda dari sebuah file yang nantinya akan dipakai sebagai acuan untuk memeriksa apakah suatu file adalah file virus atau bukan.

Kecil sekali kemungkinan bahwa dua buah file mempunyai nilai CRC32 yang sama. Hal ini disebabkan perbedaan 1 bit saja pada file akan mengubah nilai CRC32 file tersebut. CRC32 hanya mengambil 32 bit dari sebuah file yang dijadikan sebagai penanda file tersebut. Hal ini berbeda dengan metode MD5 yang mengambil 128 bit dari file. Keuntungan memakai CRC32 adalah karena hanya terdiri dari 32 bit sehingga mempercepat proses scanning.

Pendeteksian dengan melihat cara bagaimana virus bekerja: Cara kerja antivirus seperti ini merupakan pendekatan yang baru yang dipinjam dari teknologi yang diterapkan dalam *Intrusion Detection System* (IDS). Cara ini sering disebut juga sebagai *Behavior-blocking detection*. Cara ini menggunakan *policy* (kebijakan) yang harus diterapkan untuk mendeteksi keberadaan sebuah virus. Jika ada kelakuan perangkat lunak yang "tidak wajar" menurut policy yang diterapkan metode *heuristic* dalam mendeteksi virus seperti virus Virus W32.Beagle.CO@m, Trojan.Lodear, Virus Sistem dengan program antivirus.

Masalah yang dihadapi selama ini adalah semakin pesatnya perkembangan virus mengharuskan user untuk mengunduh data base virus yang semakin besar ukurannya. Setelah *user* mengunduh data base antivirus, belum tentu virus yang menginfeksi komputer user terdapat dalam data base antivirus tersebut. Oleh karena itu, akan lebih efektif jika user hanya memeriksa rutinitas kerja dari sebuah file apakah cenderung memiliki kerja yang sama seperti layaknya sebuah virus atau tidak, jenis file yang memiliki rutinitas seperti virus biasanya berekstensi .exe atau biasa disebut dengan *executable, file exe* yang diduga sebuah virus disebut dengan malicious *executable* dan biasanya beberapa antivirus langsung membersihkan virus dari file exe tersebut.

1.2 Rumusan Masalah

Adapun rumusan masalah yang diangkat adalah sebagai berikut:

1. Bagaimana implementasi yang dibuat pada *heuristic* untuk antivirus yang dirancang dapat mendeteksi virus yang terdapat pada komputer?
2. Bagaimana menerapkan metode *heuristic* Antivirus dalam mendeteksi virus seperti virus Virus W32.Beagle.CO@m, Trojan.Lodear, Virus Sistem dengan program antivirus?
3. Bagaimana proses dari metode *heuristic* Antivirus tersebut dapat melakukan pemulihan atau berkas yang terdapat pada komputer?

1.3 Batasan Masalah

Untuk menghindari terjadinya penyimpangan–penyimpangan permasalahan, penulis membatasi permasalahan yang akan dibahas, antara lain:

1. Antivirus ini hanya dipergunakan sebagai bahan pembelajaran dan penelitian untuk mendeteksi dan menghapus virus pada sistem operasi *Windows*.
2. Menerapankan metode *heuristik* sebagai pengenal hashing terhadap virus yang dikenali.
3. Menggunakan bahasa pemrograman visual basic 6.0 sebagai *software* pembuat aplikasi.

1.4 Tujuan dan Manfaat Penelitian

Adapun tujuan Penelitian ini bertujuan sebagai berikut:

1. Membuat aplikasi yang bisa mendeteksi file apakah sebuah virus atau tidak.
2. Mengimplementasikan metode *heuristic* Antivirus sebagai pemrosesan file exe.
3. Untuk mengetahui berkas yang sudah dipulihkan dari file yang terkena virus, sehingga dapat dideteksi oleh antivirus.

Manfaat yang diperoleh adalah sebagai berikut :

User dapat menggunakan aplikasi ini untuk memeriksa komputer dari virus dengan adanya aplikasi ini pengguna mengetahui teknik *heuristic* Antivirus dalam mendeteksi virus.

2. LANDASAN TEORI

2.1 Sejarah Virus Komputer

Virus komputer pertama kalinya tercipta bersamaan dengan komputer. Pada tahun 1949 salah seorang pencipta komputer Johnvon Newman, yang menciptakan *Electronic Discrete Variable Automatic Computer (EDVAC)*, memaparkan suatu makalahnya yang berjudul *"Theory and Organization of Complicated Automata"*.

Dalam makalahnya dibahas kemungkinan program yang dapat menyebar dengan sendirinya.Perkembangan virus komputer selanjutnya terjadi di *AT&T Bell Laboratory* salah satu laboratorium komputer terbesar di dunia yang telah menghasilkan banyak hal, seperti bahasa C dan C++.1 Di laboratorium ini, sekitar tahun 1960-an, setiap waktu istirahat para peneliti membuat permainan dengan suatu program yang dapat memusnahkan kemampuan membetulkan dirinya dan balik menyerang kedudukan lawan. Selain itu, program permainan dapat memperbanyak dirinya secara otomatis.

Perang program ini disebut *Core War*, yaitu pemenangnya adalah pemilik program sisa terbanyak dalam selang waktu tertentu. Karena sadar akan bahaya program tersebut, terutama bila bocor keluar laboratorium tersebut, maka setiap selesai permainan, program tersebut selalu dimusnahkan. Sekitar tahun 1970-an , perusahaan *Xerox* memperkenalkan suatu program yang digunakan untuk membantu kelancaran kerja. Struktur programnya menyerupai virus, namun program ini adalah untuk memanfaatkan waktu semaksimal mungkin dan pada waktu yang bersamaan dua tugas dapat dilakukan. Pada tahun 1980-an, perang virus di dunia terbuka bermula atas pemaparan *Fred Cohen*, seorang peneliti dan asisten profesor di Universitas *Cincinati, Ohio*.

Dalam pemaparannya, *Fred* juga mendemonstrasikan sebuah program ciptaannya, yaitu suatu virus yang dapat menyebar secara cepat pada sejumlah komputer. Sementara virus berkembang, Indonesia juga mulai terkena wabah virus. Virus komputer ini pertama menyebar di Indonesia juga pada tahun 1988.

Virus yang begitu menggemparkan seluruh pemakai komputer di Indonesia, saat itu, adalah virus ©*Brain* yang dikenal dengan nama virus Pakistan.

2.2 Pengertian Virus Komputer

Virus komputer adalah sebuah program kecil yang bisa menggandakan dirinya sendiri dalam media penyimpanan suatu komputer. Virus juga mempu, baik secara langsung ataupun tak langsung menginfeksi, mengkopi maupun menyebarkan program *file* yang bisa dieksekusi maupun program yang ada di sektor dalam sebuah media penyimpanan *(Hardisk, Disket, CD-R)* , (Madcoms, 2011).

Virus juga bisa menginfeksi file yang tidak bisa dieksekusi *file* data dengan menggunakan *macros* (program sederhanayang biasanya digunakan untuk melakukan suatu perintah). Intinya adalah kemampuan untuk menempel dan menulari suatu program. Virus bukanlah suatu yang terjadi karenakecelakaan ataupun kelemahan perangkat komputer karna pada hakekatnya, semua virus merupakan hasil rancangan intelegensi manusia setelah melalui beberapa percobaan terlebih dahulu layaknya eksperimen –eksperimen ilmiah di dalam bidang-bidang lainnya.

Istilah virus komputer tak asing lagi bagi kalangan pengguna komputer saat ini. Padahal, sekitar 12 tahun yang lalu, istilah ini telah dikenal oleh masyarakat pengguna komputer. Baru pada tahun 1988, muncul artikel-artikel di media massa yang dengan gencar memberitakan mengenai

ancaman baru bagi para pemakai komputer yang kemudian dikenal dengan sebutan 'virus komputer'. Virus yang terdapat pada komputer hanyalah berupa program biasa, sebagaimana layaknya program-program lain (Yohanes Nugroho, 2005). Tetapi terdapat perbedaan yang sangat mendasar pada virus komputer dan program lainnya.Virus dibuat oleh seseorang dengan tujuan yang bermacam-macam, tetapi umumnya para pembuat virus hanyalah ingin mengejar popularitas dan juga hanya demi kesenangan semata. Tetapi apabila seseorang membuat virus dengan tujuan merusak maka tentu saja akan mengacaukan komputer yang ditularinya.

3. PEMBAHASAN

3.1 Analisa

Langkah pertama dari pembuatan program antivirus adalah memiliki *pattern* atau dari virus yang akan dikenali oleh program antivirus. *Pattern* bisa juga disebut pola atau susunan. Sebuah program antivirus tidak akan bermanfaat jika tidak didukung oleh virus *definition* yang lengkap, *definition file* adalah suatu kumpulan data dari *malicious code*. Jadi industri antivirus sangat tergantung dari dukungan sample virus yang dikirim kepada mereka.

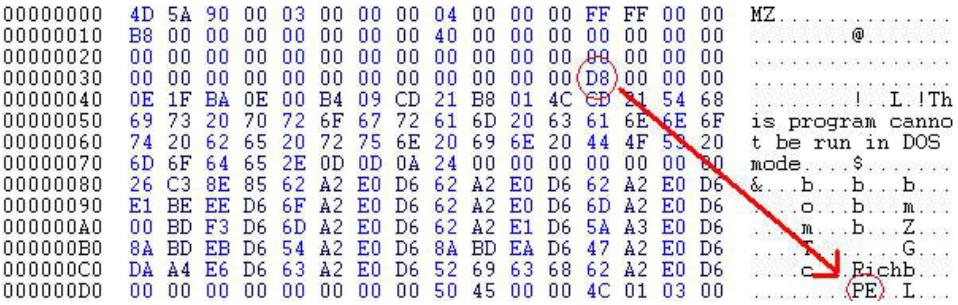
Walaupun dewasa ini telah dilakukan berbagai pendekatan *heuristic* dalam pendeteksian program-program virus, tetapi hal tersebut sering tidak efektif karena akan membuat sistem menjadi rewel dan sering memberikan *false* alarm dimana program-program utility tertentu dianggap sebagai virus, oleh karena itu pembuat virus telah memperbaiki teknik pengkodean *(polymorphism)* sehingga dapat memperdaya program-program antivirus.

Tentu saja hal ini membutuhkan penelitian dan analisa yang mendalam sehingga teknik pengumpulan *pattern* virus menjadi efektif untuk mendeteksi keberadaan process virus dimemori maupun *file* virus dimedia penyimpanan. Masalah lain adalah teknik penyiapan *pattern* virus akan mempengaruhi teknik pendeteksian yang tentu saja sangat menentukan *performance* dari antivirus yang dibuat, misalnya *pattern* virus telah berkembang menjadi 100 *pattern* dan jumlah *file* yang akan dideteksi adalah 1000 *files*, sehingga dilakukan perkalian menjadi 100000 kali proses pendeteksian, bagaimana kalau *pattern* berkembang menjadi 1000, dan terakhir adalah *resource* yang digunakan untuk penyimpanan *pattern* tersebut baik media disk maupun memori. Jadi teknik pembuatan *pattern* virus merupakan isu yang terpenting dari kesuksesan pendeteksian program antivirus dan perfomancenya, serta resource yang dihabiskan. Ini merupakan bagian yang paling penting dari suatu program antivirus. Karena bagaimana *pattern* virus tersebut dilakukan (mengambil pattern dari *file* virus), maka dengan cara sebaliknya program anti virus akan mendeteksi keberadaan virus *file* dan process virus dimemori (dengan membandingkan pattern virus dengan data *file* dan process).

Teknik pendekatan yang digunakan penulis merupakan hasil pengamatan dan penelitian terhadap beberapa program virus yang beredar di Indonesia, seperti varian Brontok dan *MyHeart.* Pendekatan ini cukup efektif untuk digunakan untuk mengenali keberadaan process virus di memori maupun di *file* dengan satu pendekatan yang sama.

3.1.1 Struktur file PE (Portable Executable)

Suatu *file* PE akan diawali dengan suatu DOS *Header* yang memiliki suatu e_magic number dalam hexa 4D 5A (MZ, lihat gambar dibawah), kemudian pada *offset* ke 60 atau hexa D8 akan terdapat suatu pointer *(e_lfanew)* yang menunjuk kelokasi dimana PE *file header*.



Gambar 1 : DOS Header dari suatu PE file

3.2 Analisa Metode Heuristic

Pencarian *Heuristik* adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namum dengan kemungkinan mengorbankan kelengkapan(*completeness*). Fungsi *heuristic* digunakan untuk mengevaluasi keadaan-keadaan problem individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan. Dan dalam penulisan skripsi ini dibuat dengan menggunakan jenis *heuristic Best Fist Search*.

Tahap analisa sistem dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain. Tahap analisa merupakan tahap yang kritis dan sangat penting, karena kesalahan didalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya.

Heuristic Mengubah graph awal menjadi suatu pohon dengan cari sebagai berikut :

1. Pilih induk dari pohon (misal f) pilih yang terhubung dengan sirkuit
2. Pilih titik-titik yang mnghasilkan jalur terpanjang dengan mengabaikan adanya sirkuit
3. (setelah mengambil f kita bisa beralih ke titik yang berada di kanannya dahulu atau di kirinya, missal hubungkan titik f-g-h-k-j)
4. Setelah mencapai ujung jalur, lakukan pengecekan dengan kembali ke titik sebelumnya. Mengecek apakah pada titik tersebut masih mempunyai cabang atau tidak, Jika ada cabang tuliskan titik tersebut sebagai cabang. Kemudian cek lagi cabang tersebut apakah bercabang lagi atau tidak.
5. (setelah mencapai ujung j, kembali lagi ke titik sebelumnya k, lakukan pengecekan, lakukan langkah ini hingga semua titik terdapat dalam satu pohon).

Dan Algoritma dari Heuristic adalah sebagai berikut ini :

1. Algoritma
2. Step 1: letakkan *root parent* di tempat yang paling atas
3. Step 2: Lakukan *looping* hingga titik terakhir
4. Step 3: letakkan titik yang terhubung dengan root parent pada satu garis lurus
5. Step 4: lakukan pengecekan, jika titik tidak memiliki child letakkan pada garis, pindah ke titik selanjutnya
6. Step 5: jika titik yang ditemui memiliki child hubungkan dengan titik sebelumnya.

Dalam prosedur ini kami mengambil asumsi bahwa setiap *node* telah merepresentasikan karakter yang diwakilinya, Perubahan dari status ke status (dari *node* ke anaknya) diasumsikan sebagai operasi string untuk membentuk kata. Kemudian kata tebakan harus lebih kecil dari kata awal.

1. Masukkan *node* awal ke *Queue*
2. Set Kedalaman dengan 0
3. Ambil kepala *Queue*
4. Bangkitkan semua anak
5. Jika anak memenuhi syarat *push* ke *Queue*
6. Jika memenuhi solusi *stop*, jika tidak ulangi langkah 3 sampai kedalaman *maks*
7. Cetak solusi dan perhitungan skor berdasarkan kedalaman

Fungsi pencarian :

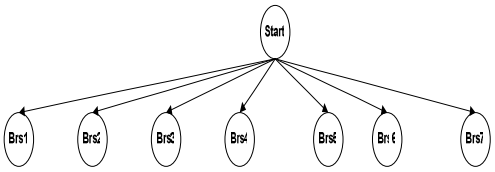
1. Bangkitkan anak Prosedur ini membangkitkan anak dari *Node* dengan elemen status yang tersisa
2. Fungsi Bagian *(input : String,String)* Fungsi ini memeriksa apakah *Node* tersebut memenuhi syarat atau merupakan bagian dari virus.

Contoh pencarian

File yang akan dicari Brontok dengan kode binary = "D8"

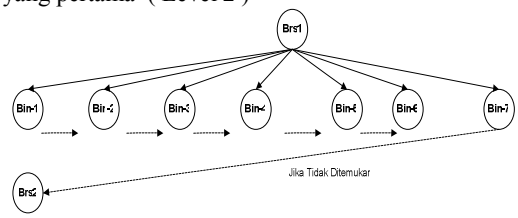
Maka tahap yang dilakukan adalah sebagai berikut :

a. Menset Semua Baris ke dalam sebuah *Queue* (Level-1)



Gambar 2 : Set Semua Baris dalam *Queue*

b. Cek file yang akan dicari mulai dari Baris Pertama yang pertama (Level 2)



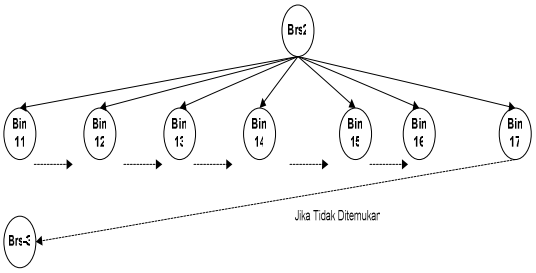
Gambar 3 : Proses Pencarian Level 2

Pada tahap level-2 maka dilakukan pencarian pada Baris "BRS1" dengan cara membandingkan apa yang dicari dengan file yang ada didalam Baris tersebut, dan prosesnya dilakukan seperti langkah berikut ini :

- Cek apakah "Bin-1" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-1" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke Vi selanjutnya
- Cek apakah "Bin-2" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-2" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-3" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-3" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-4" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-5" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-6" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-6" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-7" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-7" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke **Baris** yang lainnya karena

c. Cek file yang akan dicari mulai dari Baris Level -3

Majalah Ilmiah
Informasi dan Teknologi Ilmiah (INTI)
ISSN : 2339-210X

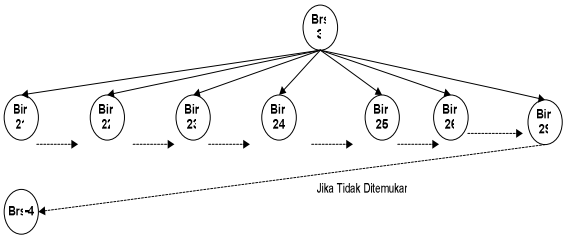


Gambar 4 : Proses Pencarian Level 3

Pada tahap level-3 maka dilakukan pencarian pada Baris "BRS-2" dengan cara membandingkan apa yang dicari dengan *file* yang ada didalam Baris tersebut,sehingga dapat memproses setiap file yang terdapat pada setiap barisnya dan prosesnya dilakukan seperti langkah berikut ini :

- Cek apakah "Bin-11" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-11" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-12" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-12" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-13" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-13" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-14" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-4" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-15" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-15" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-16" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-16" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-17" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-17" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke **Baris** "BRS-3"

d. Cek file yang akan dicari mulai dari Baris Level -4

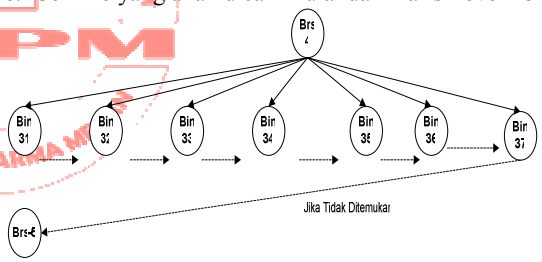


Gambar 5 : Proses Pencarian Level 4

Pada tahap level-4 maka dilakukan pencarian pada Baris "BRS-3" dengan cara membandingkan apa yang dicari dengan file yang ada didalam Baris tersebut, dan prosesnya dilakukan seperti langkah berikut ini :

- Cek apakah "Bin-21" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-21" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-22" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-22" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-23" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-23" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-24" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-24" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-25" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-25" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-26" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-26" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-27" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-27" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke **Baris** "BRS-4"

e. Cek file yang akan dicari mulai dari Baris Level - 5



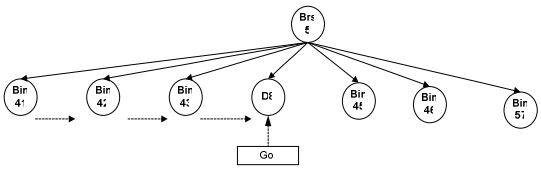
Gambar 6 : Proses Pencarian Level 5

Pada tahap level-5 maka dilakukan pencarian pada Baris "BRS-4" dengan cara membandingkan apa yang dicari dengan file yang ada didalam Baris tersebut, dan prosesnya dilakukan seperti langkah berikut ini :

- Cek apakah "Bin-31" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-31" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-32" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-32" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-33" = "Virus yang dicari" ?
  - Jika "YA" maka "Bin-33" adalah "GOL" atau data ditemukan
  - Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-34" = "Virus yang dicari" ?

- Jika "YA" maka "Bin-34" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-35" = "Virus yang dicari" ?
- Jika "YA" maka "Bin-35" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-36" = "Virus yang dicari" ?
- Jika "YA" maka "Bin-36" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-37" = "Virus yang dicari" ?
- Jika "YA" maka "Bin-37" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke **Baris** "BRS-5"

f. Cek file yang akan dicari mulai dari Baris Level - 6



Gambar 7 : Proses Pencarian Level 6

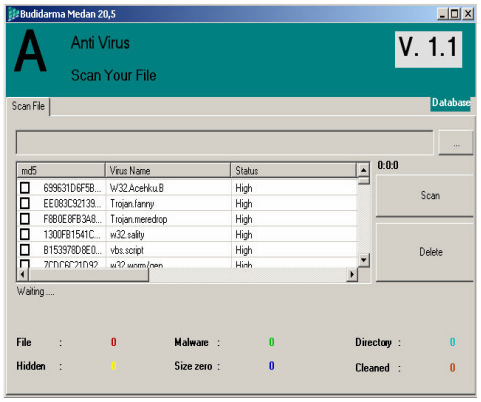
Pada tahap level-6 maka dilakukan pencarian pada Baris "BRS-5" dengan cara membandingkan apa yang dicari dengan file yang ada didalam Baris tersebut, dan prosesnya dilakukan seperti langkah berikut ini :

- Cek apakah "Bin-41" = "Virus yang dicari" ?
- Jika "YA" maka "Bin-41" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-42" = "Virus yang dicari" ?
- Jika "YA" maka "Bin-42" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-43" = "Virus yang dicari" ?
- Jika "YA" maka "Bin-43" adalah "GOL" atau data ditemukan
- Jika Tidak maka Lanjutkan ke data selanjutnya
- Cek apakah "Bin-44" = "Virus yang dicari" ?
- Jika "YA" maka "D8" adalah "GOL" ➔ GOL (Virus ditemukan)

4. IMPLEMENTASI

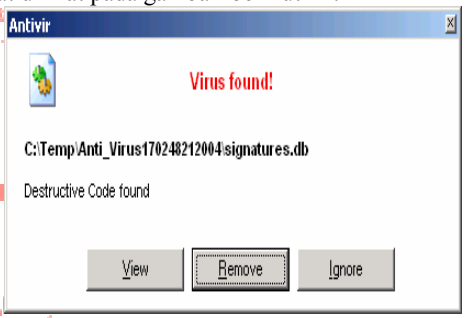
a. Tampilan Utama Sistem Virus

Tampilan utama dari sistem antivirus yang sudah dibangun dapat dilihat pada gambar berikut ini.



Gambar 8 : Form Utama

Berikut ini adalah tampilan form Virus Ditemukan dari sistem antivirus yang sudah dibangun dapat dilihat pada gambar berikut ini.



Gambar 9 : Form Virus Ditemukan

Setelah selesai dari pemrosesan Program Antivirus yang digunakan pada proses dari metode *heuristic* Antivirus di atas, berikut hasil dari pengujian yang dapat dilihat pada tabel 1 di bawah ini:

Tabel 1 Pengujian

NO	Virus Name	Status	Detected	Path	Status
1	w32.sality.aa	82432 kb	Hidden file	Not been cleaned	Infected file
2	W32.worm. Senna	278 kb	Hidden file	Not been cleaned	Infected file
3	Trojan.jorik	2535 kb	Hidden file	Not been cleaned	Infected file
4	Dos.Tiny	1326 kb	Hidden file	Not been cleaned	Infected file

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil teknik uji coba dari sistem anti virus ini dapat di ambil kesimpulan bahwa:

1. Implementasi yang dibuat pada metode *Heuristic* adalah dengan solusi pencarian *file* menggunakan metode pencararian melebar pertama *(Best-First Search).*

2. Teknik penggunaan *header file addres Ofentrypoint* dan *sizeofcode*, Sangat akurat dalam mengenal virus, walaupun virus telah merubah header filenya tapi datanya tetap sama
3. Metode *heuristic* yang digunakan oleh *software* antivirus ini adalah metode *heuristic statik* atau secara umum dinamakan metode *generate and test* dimana cara kerja metode *heuristic* ini adalah dengan membangkitkan/*generate* keseluruhan *string* yang ada pada *file* target terlebih dahulu, kemudian *software* antivirus akan membaca keseluruhan *string* dari *file* target dan mencocokkan dengan kumpulan *array* yang berisi *string-string* mencurigakan.

5.2 Saran

1. Anti virus ini juga perlu di tingkatkan sensitifitasnya, karena bila terdapat *section dummy* pada urutan *section* kedua atau tidak ada pada *section*, atau pembelokan pada *entrypointnya* maka data yang di teliti bisa saja salah, Fungsi ini juga masih rawan dari kesalahan analisa.
2. Maka dari itulah butuh penelitian lebih lanjut, dengan menambahkan algoritma baru fungsi ataupun prosedur, agar dapat mengembangkan dan memajukan kualitas aplikasi anti virus.

6. DAFTAR PUSTAKA

[1] Aat Shadewa, 2006, Rahasia Membuat Anti virus Menggunakan Visual Basic,Yogyakarta: Penerbit DSI Publishing
[2] Madcoms, 2011, Pengenalan Sistem Antivirus dan Pengamanan Data.
[3] Yohanes Nugroho, 2005, Mengetahui Cara kerja Virus dan Mengamankan Data.
[4] Http://www/Kaspersky Lab (2006), Viruslist.com – Network Worms, URL
[5] Jurnal Gordon, A., Lawrence et. al., (2009), CSI/FBI Computer Crime and Security Survey 2008, CSI Publication, Washington DC, http://www.GoCSI.com/, 1 November 2008.
[6] Jurnal Nazario, Jose, et. al., (2004), Defense and Detection Srategies Againts Internet Worms, Artech House inc., Norwood MA. Pietrek, Mat; Peering Inside the PE A tour of the PE: A Tour of the Win32 Portable Executable File format; MSDN;1994 Szor, Peter (2005),
[7] Jurnal The Art of Computer Virus Research and Defense, Addison Wesley Proffesional, New Jersey. Yohanes Nugroho (2005), Analisis Lengkap Virus Brontok, http://www.compactbyte.com /brontok/Analisis Lengkap Virus Brontok.html, 11 September 2006.