

PEMBUATAN APLIKASI ANTIVIRUS INSAV MENGGUNAKAN METODE *BLIND SEARCH*

Heny Pratiwi ¹⁾, Andy Yushika Rangan ²⁾ M. Randi Irawan ³⁾

^{1,2,3}Teknik Informatika, STMIK Widya Cipta Dharma

^{1,2,3}Jl. Prof. M. Yamin No. 25 Samarinda Kalimantan Timur 75123, Telp: (0541) 736071, Fax: (0541) 203492

E-mail: heny_pratiwi@ymail.com¹⁾, yu5h1k4@gmail.com²⁾, m.randi.ira@gmail.com³⁾

ABSTRAK

Tujuan pada penelitian pembuatan aplikasi antivirus menggunakan metode *Blind Search* adalah untuk memberikan pengetahuan dan informasi tentang antivirus kepada *user* sehingga dapat diketahui dari jenis, penyebab, dan solusi dari virus.

Pembuatan Aplikasi Antivirus Insav menggunakan metode *Blind Search*, merupakan aplikasi yang dibuat untuk membantu dalam mendeteksi virus komputer dengan menggunakan bantuan metode *Blind Search* jenis *Depth First Search*.

Hasil dari penelitian ini adalah dibuatnya aplikasi antivirus dengan menggunakan metode *Blind Search*. *User* dapat memilih *folder*, kemudian aplikasi akan mencari virus dengan metode *Blind Search*. Setelah itu aplikasi akan menampilkan hasil virus yang terdeteksi dan menghapusnya.

Kata Kunci : *Aplikasi, Antivirus, Blind Search*

1. PENDAHULUAN

Semakin berkembangnya dunia teknologi, terutama dibidang *software*. Banyak *software* (aplikasi) yang bermunculan yang dapat kita *download* diinternet (*google*). Saat mendownload *software* (aplikasi), *game* dan *e-book*, sering kali tidak bisa dibuka dikarenakan terinfeksi virus atau *worm*. Selain *software* (aplikasi), *game* dan *e-book*, virus juga sering kali merusak *file* seperti dokumen *MS Office*, *folder*, gambar dan bahkan virus dapat merusak system operasi *windows*. Saat komputer terinfeksi virus, maka akan memperlambat proses kerja komputer. Dengan adanya sebuah, maka dapat membantu para pengguna komputer untuk mengamankan datanya agar tidak terinfeksi virus.

Antivirus saat ini, untuk melakukan proses *scan* menggunakan waktu relatif lebih lama sekitar 20 menit bahkan lebih dan berkali-kali, saat pemindaian antivirus biasanya menghapus *file* yang terkena virus dan terkadang menganggap *file* ber-ekstensi *.EXE* sebagai virus terutama pada *file crack*. Untuk *update* dan saat penginstalan memakan waktu cukup lama. Antivirus saat ini harganya cukup relatif mahal dan untuk versi *trial* hanya 30 hari.

Oleh karena itu, dibuat aplikasi antivirus Insav dengan menggunakan metode *Blind Search* jenis DFS (*Depth First Search*) karena kemudahannya dalam mendeteksi virus, dan juga memori yang digunakan dalam proses pencarian ini relatif lebih kecil jika

dibandingkan dengan BFS (*Breadth First Search*). Dengan menggunakan sebuah aplikasi antivirus Insav dapat memudahkan dan meminimalisir waktu yang ada dalam mendeteksi virus serta menghapusnya. Untuk harga relatif lebih murah dan proses *update* serta instal memakan waktu lebih cepat.

Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah dalam penelitian ini adalah “Bagaimana Membangun Aplikasi Antivirus Insav Menggunakan Metode *Blind Search* ? ”.

Batasan Masalah

Batasan masalah pada Aplikasi Anti virus menggunakan metode *Blind Search* jenis *Depth First Search* adalah sebagai berikut :

1. Metode yang digunakan dalam pembuatan Aplikasi Antivirus Insav ini adalah *Blind Search* jenis *Depth First Search*.
2. Keluaran dari Aplikasi ini adalah laporan hasil pemindaian virus yang ada.
3. Untuk menu *update* masih dalam pengembangan, terlebih antivirus insav masih dalam versi *beta*.
4. Dalam proses pemindaian antivirus Insav ini hanya bisa memilih *folder* yang akan di *scan*.

2. RUANG LINGKUP PENELITIAN

Kajian teoritik

Menurut Dhanta (2009), aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk membuat tugas-tugas tertentu, misalnya *Microsoft Word, Microsoft Excel*.

2.2.1 Aplikasi

Menurut Dhanta (2009), aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk membuat tugas-tugas tertentu, misalnya *Microsoft Word, Microsoft Excel*.

2.2.1 Antivirus

Menurut Hirin (2015), antivirus adalah program yang dibuat untuk mendeteksi *file* didalam suatu *drive* apakah terkena virus atau tidak. Tujuan dari antivirus adalah untuk mencapai target atau aksi tertentu yang harus dilakukan. Dalam pembuatan antivirus terdiri dari 3 tahapan, berikut adalah tahapannya :

1. Tahapan pertama dimulai dengan mencari sampel virus atau *worm*, sampel bisa diperoleh dari *file* yang dicurigai sebuah virus atau *worm* berdasarkan perilakunya, dan sampel bisa juga diperoleh dari beberapa *website* tertentu.
2. Antivirus melakukan *scanning* pada semua *file* yang terdapat pada sistem operasi komputer.
3. Berdasarkan *list* virus atau *worm* yang diperoleh, dilakukan pengambilan keputusan untuk menghapus *file* tersebut atau mengkarantinanya.

2.2.2 Virus

Menurut Warsito (2011), virus komputer didefinisikan sebagai suatu aplikasi yang dapat mereplikasikan diri dalam suatu jaringan komputer.

2.2.3 Worm

Menurut Achmad (2007), *Worm* atau yang sering disebut virus adalah program yang dapat mereplikasi dirinya dengan menggunakan media komputer. Sifatnya destruktif terhadap *disk* dan memori yang menyebabkan kerusakan pada sistem dan memperlambat kinerja komputer dalam mengaplikasikan sebuah program.

2.2.3.1 Macam-macam Virus

Berikut ini 5 macam virus komputer yang masih aktif dan terus menyebar hingga saat ini :

1. Virus *Boot Sector*

Virus *Boot Sector* adalah sebuah virus yang menginfeksi bagian pertama (*first sector*) sebuah *drive*, seperti *floppy disk* atau *hard drive*. *Boot sector* virus juga dapat menginfeksi MBR. Virus komputer pertama kali adalah *Brain*, sebuah *boot sector* virus

yang menggunakan teknik menghilangkan diri untuk menghindari deteksi antivirus.

2. Virus *File*

Virus yang memanfaatkan *file* yang dapat dijalankan atau dieksekusi secara langsung. Biasanya *file* *.EXE atau *.COM. tapi bisa juga menginfeksi *file* *.SYS, *.DRV, *.BIN, *.OVL, dan *.OVY. Jenis virus ini dapat berpindah dari satu media ke semua jenis media penyimpanan dan menyebar dalam sebuah jaringan.

3. Virus *Direct Action*

Tujuan utama dari virus ini adalah untuk meniru dan bertindak ketika dijalankan. Ketika kondisi terpenuhi, virus akan beraksi dan menginfeksi *file* dalam *directory* atau *folder* yang ditentukan oleh *Path file* AUTOEXEC.BAT. *File batch* selalu terletak di direktori *root hard disk* dan melakukan operasi tertentu ketika komputer *boot*.

4. *Multi Partition Virus*

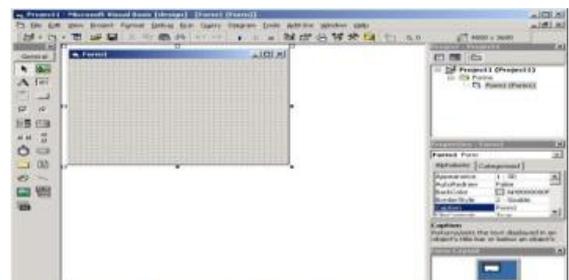
Virus ini menyebar dalam berbagai macam cara, tergantung sistem operasi yang terpasang pada *file* tertentu.

5. *Polymorphic Virus* menyandikan diri dengan cara yang berbeda (menggunakan algoritma yang berbeda dan kunci enkripsi) setiap kali mereka menginfeksi sistem.

2.2.4 Tinjauan *Microsoft Visual Basic 6.0*

Subari dan Yustanto (2008), *Visual Basic* selain disebut sebagai bahasa pemrograman (*Language Program*), juga sering disebut sebagai sarana (*Tool*) untuk menghasilkan program-program aplikasi berbasis *windows*.

Setelah *Visual Basic* dijalankan, akan muncul sebuah layar. Layar ini adalah lingkungan pengembangan aplikasi *Visual Basic* yang nantinya akan digunakan untuk membuat program-program aplikasi dengan *Visual Basic*.



Gambar 2.1 Lingkungan *Visual Basic 6.0*

Sumber : Subari dan Yustanto (2008)

Layar *Visual Basic* adalah suatu lingkungan besar yang terdiri dari beberapa bagian-bagian kecil yang kesemuanya memiliki sifat.

Visual Basic IDE (Interface Development Environment) ialah tampilan antarmuka program dengan pengguna yang sudah bersifat GUI (*Graphical User Interface*), menyajikan banyak kemudahan bagi para *programmer* untuk membuat aplikasi.

2.2.4.1 Cara kerja virus

Antivirus Insav ini memfokuskan pada virus seperti *salicy* dan *boot sector*. Kebanyakan virus lokal menggunakan teknik-teknik sederhana dalam *Social Engineering*. *Social Engineering* adalah teknik pendekatan khusus untuk mengelabui *user* agar menjalankan dirinya karena pada dasarnya *user* yang terkena virus disebabkan oleh kesalahannya sendiri. Berikut teknik *Social Engineering* yang biasa digunakan oleh virus lokal:

1. Menggunakan *folder icon* dan mengambil nama *folder* asli sebagai nama baru penggandaannya. Dengan begitu, *user* akan tertipu dan menganggap virus tersebut sebagai sebuah *folder* karena *folder* asli diset menggunakan atribut *hidden* agar tidak tampak secara *default*. Akan tetapi, jika *user* yang menggunakan *setting non-standar*, maka *Windows* pasti akan merasa janggal dengan teknik *Social Engineering* tersebut. Teknik itu banyak digunakan oleh *VM local*, contoh virusnya adalah *brontok*, *tati*, *zulnick*, dan lain-lain.
2. Menggunakan *notepad icon* dan mengambil nama-nama yang “asing” bagi *user* sebagai nama penggandaan dirinya (duplikasi). Dengan begitu, *user* akan tertipu oleh Teknik *Social Engineering*, contohnya adalah *vcybe*, *parents*, *netsky*, dan lain-lain.
3. Menggunakan *Word Standar Icon* dan memakai nama-nama unik sebagai duplikasi dirinya agar *user* tertipu dan menganggap *file* duplikasinya merupakan dokumen *Ms.Word*.

2.2.7 Metode Pengembangan Sistem

2.2.7.1 *Waterfall Model*

Menurut Rosa dan Shalahuddin (2011), Model SDLC air terjun (*Waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup

klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar *Waterfall Model* :

Gambar 2.11 Ilustrasi model *Waterfall*

Sumber : Rosa dan Shalahuddin (2011)

1. Analisis kebutuhan perangkat lunak
Proses pengumpulan kebutuhan dilakukan secara intensif untuk spesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini diperlukan untuk didokumentasikan.
2. Desain
Desain perangkat lunak adalah proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.
3. Pembuatan kode program
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. Pengujian
Fokus pada perangkat lunak dari segi logika dan fungsional untuk memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan

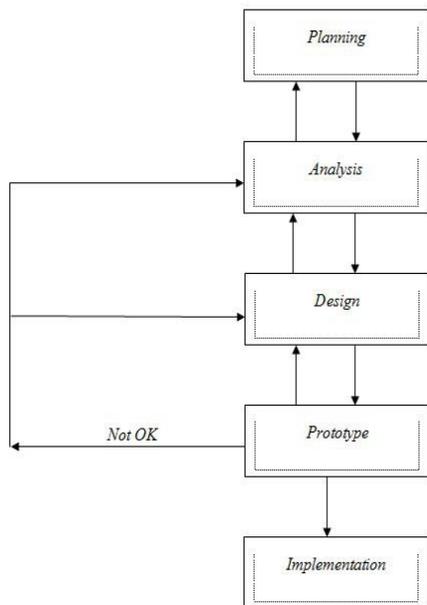
(error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.

2.2.7.2 Model Prototipe

Menurut Rosa dan Shalahuddin (2011), Model *prototipe* dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Berikut adalah gambar dari model *prototipe*:



Gambar 2.12 *Prototype Model*

(Sumber : Turban, 2011, *Decision Support System and Intelligent System*)

Secara umum, *prototype model* terdiri dari beberapa tahapan, yaitu :

1. *Planning*

Pada tahap *planning*, pembuat merencanakan atau membuat perencanaan mengenai apa yang ingin dibuat dan apa saja kebutuhan yang diinginkan oleh *user*.

2. *Analysis*

Proses ini merupakan proses yang fokus pada pembangunan aplikasi diantaranya struktur data, antarmuka dan prosedur pengkodean. Tahap ini juga dapat memperbaiki, jika terdapat kesalahan dalam aplikasi atau tidak sesuai dengan yang diinginkan oleh pelanggan.

3. *Design*

Proses ini merupakan tahap penerjemahan dari keperluan atau data yang telah dianalisis ke dalam bentuk yang mudah dimengerti oleh pemakai (*user*).

4. *Prototype*

Proses pembuatan model dari perangkat lunak yang akan dibuat atau dikerjakan sehingga pemakai dapat mengetahui hasil yang akan didapat. *Prototype* banyak digunakan pada pemakai yang masih baru mengenal pemakaian komputer. Tujuannya adalah untuk mendefinisikan perangkat lunak yang akan dihasilkan tanpa menyertakan rincian masukan data, proses, dan keluaran yang diperlukan.

5. *Implementation*

Implementasi. Diterapkan.

3. BAHAN DAN METODE

3.1 *Blind Search*

Sistem pakar adalah sistem yang memindahkan pengetahuan manusia ke komputer yang dibuat untuk menyelesaikan masalah seperti seorang pakar.

3.2 Konsep Dasar *Blind Search*

Menurut Suyanto (2014), hal terpenting dalam pencarian adalah suatu proses mencari solusi dari suatu permasalahan melalui sekumpulan ruang keadaan (*state space*). Ruang keadaan adalah suatu ruang yang berisi semua keadaan yang mungkin. Adapun secara umum, untuk mengukur metode keakuratan metode pencarian, terdapat empat kriteria. Keempat kriteria tersebut yaitu :

A. **Completeness** : Apakah metode tersebut menjamin penemuan solusi jika solusinya memang ada ?

B. **Time complexity** : Berapa lama waktu yang diperlukan ?

- C. **Space complexity** : Berapa banyak memori yang diperlukan ?
- D. **Optimality** : Apakah metode tersebut menjamin menemukan solusi yang terbaik jika terdapat solusi yang berbeda ?

3.3 Virus

Menurut Warsito (2011), virus komputer didefinisikan sebagai suatu aplikasi yang dapat mereplikasikan diri dalam suatu jaringan.

3.4 Depth First Search (DFS) Metode Blind Search

Pencarian dilakukan pada suatu simpul dalam setiap level dari yang paling kiri. Jika pada level yang terdalam solusi belum ditemukan, maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori. Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada level selanjutnya. Demikian seterusnya sampai ditemukannya solusi. Di bawah ini merupakan langkah-langkah proses pencarian menggunakan metode *Blind Search*, yaitu :

1. Tidak ada informasi awal yang digunakan dalam proses pencarian.
2. Semua node pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level selanjutnya, kemudian berpindah pada level selanjutnya hingga solusi ditemukan.

3.5 Metode Pengembangan Sistem

Dalam membuat aplikasi Antivirus Insav menggunakan metode pengembangan sistem *Prototyping*. Tahap-tahap dalam metode pengembangan sistem *Prototyping* adalah sebagai berikut.

1. Pengumpulan Kebutuhan, pada tahap ini penggunaan pengembang bersama-sama mendefinisikan alur aplikasi, mengidentifikasi semua kebutuhan, dan garis besar aplikasi yang akan dibuat.
2. Membangun *Prototyping*, yaitu dengan membuat perancangan sementara yang berfokus pada peyajian kepada pengguna (misalnya dengan membuat *design* tampilan aplikasi).
3. Evaluasi *Prototyping* ini dilakukan oleh pengguna apakah *Prototyping* sudah dibangun sesuai dengan keinginan pengguna. Jika sudah sesuai maka langkah 4 akan diambil, jika tidak maka *Prototyping* direvisi dengan mengulangi langkah 1, 2, 3.
4. Mengkodekan sistem, tahap ini *Prototyping* aplikasi yang sudah di sepakati, diterjemahkan ke dalam bahasa pemrograman yang sesuai yaitu *Visual Basic 6.0*.
5. Menguji sistem, yaitu setelah aplikasi sudah menjadi suatu perangkat lunak yang siap pakai, harus dites terlebih dahulu sebelum disebarkan

secara luas. Pengujian ini dilakukan dengan *Beta Testing* dan *White Box*.

6. Evaluasi Sistem, yaitu pengguna mengevaluasi apakah Aplikasi yang sudah jadi, sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan. Namun jika tidak, maka diulangi langkah 4 dan 5.
7. Mengkodekan sistem, yaitu aplikasi yang telah diuji dan diterima pengguna siap untuk digunakan dan disebarkan (di distribusikan) secara luas.

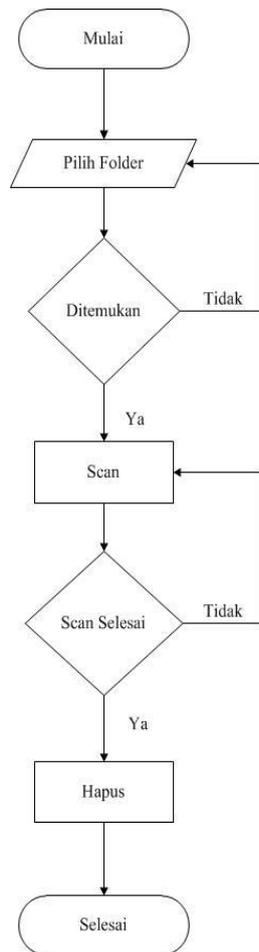
2. White Box

White Box Testing merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak.

Tabel 3.1 Rancangan Pengujian *White-box*

No	Nama Prosedur	Hasil Yang Diharapkan	Hasil Pengujian
1	Proses <i>scanning</i>	<i>Script</i>	Dapat memindai virus-virus yang ada.
2	Proses <i>view</i>	<i>Script</i>	Dapat menampilkan virus-virus yang ada.
3	Proses <i>hapus</i>	<i>Script</i>	Dapat menghapus virus yang ada.

3.6 Flowchart Program



Gambar 4.1 Flowchart Program

Perancangan Aplikasi Antivirus Insav ini menggunakan Diagram Alur (*Flowchart*) dapat menunjukkan secara jelas arus pengendalian algoritma. Yakni bagaimana aplikasi berjalan mulai awal sampai akhir. Berikut langkah-langkah diagram alur yang dibuat, yaitu “mulai” kemudian masuk ke menu utama, selanjutnya pada tampilan menu klik “pilih *folder*” jika ditemukan maka akan mulai proses pemindaian, jika belum ditemukan “pilih *folder*”. Jika virus ditemukan, maka akan dihapus, dan jika tidak maka proses akan selesai.

4. IMPLEMENTASI

4.1 Form Input Data Pasien

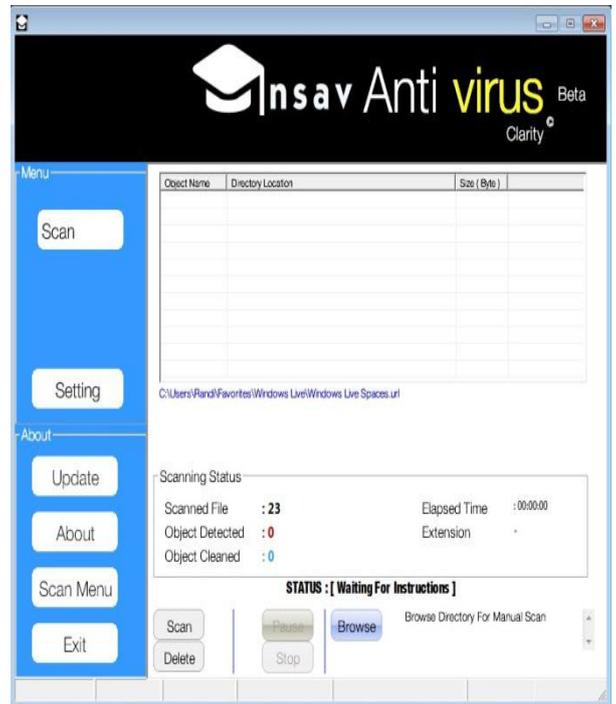
Form ini tampil saat pertama kali program dijalankan.



Gambar 5.1 Form Loading

4.2 Form Menu Utama

Form ini berfungsi untuk menu utama, terdapat lima tombol yang terdiri dari *scan*, *delete*, *pause*, *stop*, dan *browse*.



Gambar 5.2 Form Menu Utama

4.3 Form Hasil Deteksi

Form ini adalah hasil dari proses diagnosa penyakit diabetes mellitus.

Object Name	Directory Location	Size (Byte)
<input checked="" type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,138 KB
<input checked="" type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,148 KB
<input checked="" type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,146 KB
<input checked="" type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,148 KB
<input checked="" type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,141 KB
<input type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,147 KB
<input type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,175 KB
<input type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,167 KB
<input type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,158 KB
<input type="checkbox"/> W32.SALIT...	K:\Dari Randy's File\KKP\Uknov\khusus KKP rational rose Rati...	5,137 KB

Gambar 5.3 Form Hasil Deteksi Virus

5. KESIMPULAN

1. Berdasarkan hasil pengujian *beta testing* dan *white box testing*, pembuatan aplikasi antivirus

Insav dapat digunakan dengan baik pada sistem operasi *windows*, implementasi pada aplikasi sangat baik, menu yang terdapat pada aplikasi dapat mudah dipahami oleh pengguna.

6. SARAN

Adapun tindak lanjut yang diharapkan berdasarkan penelitian ini adalah

1. Untuk pengembangan Aplikasi ini, diharapkan pada penelitian selanjutnya agar ditambahkan fitur lebih banyak lagi dan dilengkapi kesalahan dalam pemindaian virus.

2. Penambahan fitur-fitur lengkap, seperti menu *scan USB*, *Quick scan*, dan *Full scan*.

3. Aplikasi ini hanya berbasis *Windows*, diharapkan ada pihak yang mampu mengembangkan menjadi berbasis *Android* atau *iOS*.

Demikian kesimpulan dan saran semoga bermanfaat bagi semua pihak.

6. DAFTAR PUSTAKA

- Achmad, Darmal., 2007. *Secret Of Underground Coding – Uncensored*. Jakarta Barat : JASAKOM.
- Akbar, Aidil., 2014. *Perancangan Perangkat Lunak Antivirus dengan Metode Scanning Cyclic Redudancy Checksum-32*. PADANG : Universitas Andalas.
- Dhanta, 2009. *Pengantar Ilmu Komputer*. Surabaya : INDAH.
- Koswara, Eko., 2011. *Visual Basic Beginner Guide*. Yogyakarta : Andi.
- Hirin., 2015. *Sehari Menjadi Programmer Antivirus Menggunakan Visual Basic 6.0*. Yogyakarta : Andi.
- Pressman. 2010. *Rekayasa Perangkat Lunak : Pendekatan Praktisi*. Yogyakarta : Andi.
- Rosa dan Shalahuddin, 2011, *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Bandung : Modula.
- Sutrisno, Sony., 2011. *Software Anti Virus dengan Metode Pencarian Header File dan Size Of Code dan Address Of Entry Point*.
- Subari dan Yustanto., 2008. *Pemrograman Dasar Microsoft Visual Basic 6.0*, Surabaya : PrestasiPustaka.
- Suyanto., 2014. *Artificial Intelligence*. Penerbit: Informatika. Yogyakarta : Andi.
- Warsito, Sigit., 2011. *Aneka Virus Komputer*. Penerbit: Informatika. Yogyakarta : Andi.