BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas secara singkat mengenai analisa masalah dan perancangan sistem yang meliputi perancangan fitur aplikasi, perancangan model sistem, perancangan *user interface*, mengetahui kelemahan dan kelebihan aplikasi.

3.1 Analisis Masalah

Kemajuan dibidang teknologi semakin pesat terutama bidang informasi dan multimedia, salah satunya teknologi komunikasi internet. Didalam media komunikasi dunia maya atau sering disebut dengan internet sering terjadi penyebaran berbagai macam virus yang dapat menyebabkan kerusakan pada sistem komputer yang terinfeksi. Salah satunya adalah *worm* dan berbagai jenisnya yang dapat merusak sistem komputer yang terinfeksi. Biasanya *worm* ini berasal dari jika komputer terkoneksi dengan internet dengan keamanan sistem yang rendah maka *worm* dapat masuk dengan mudahnya, atau *worm* dapat masuk melalui media penyimpanan seperti sekarang ini yaitu *flash disk*. Cara *worm* masuk adalah ketika kita menyimpan *file* pada saat terkoneksi ke internet sedangkan sistem keamanan komputer tidak terjaga maka *worm* atau internet *worm* dapat masuk ke dalam media penyimpanan tersebut.

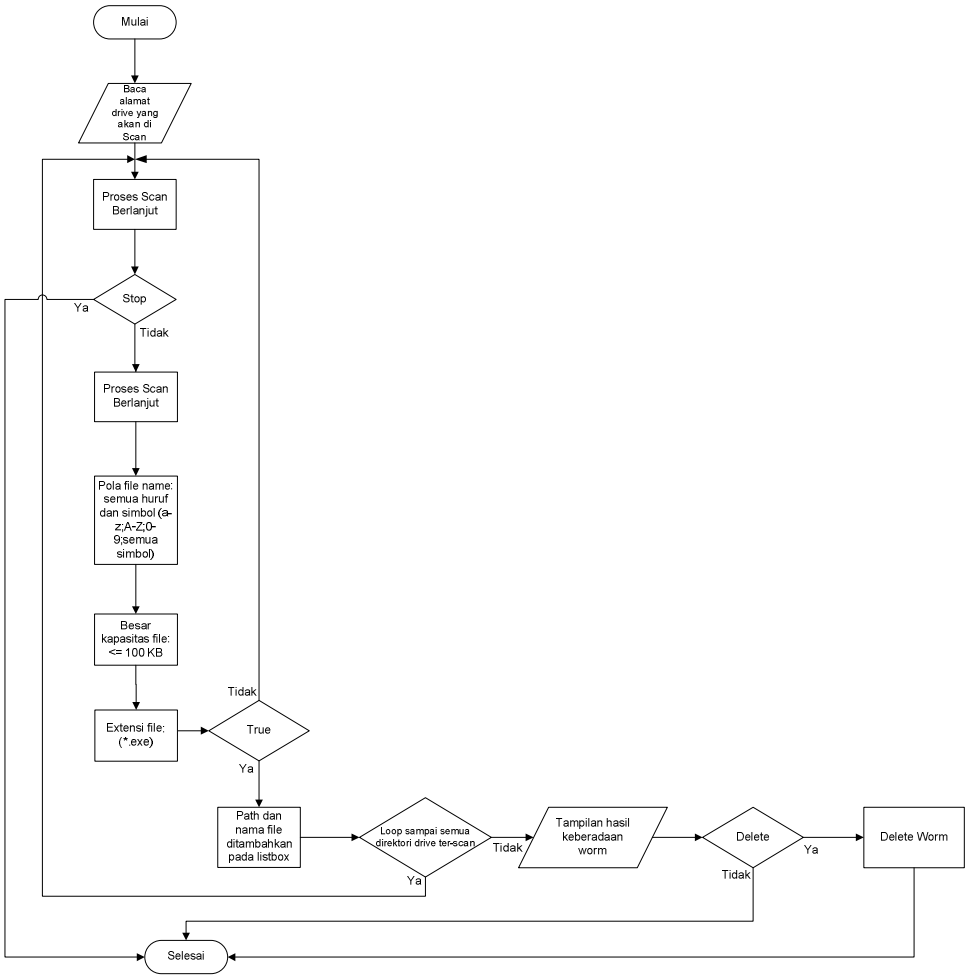
Worm yang tersebar di internet banyak sekali jenis dan berbagai macam nama sebagai identitas dari *worm* tersebut, sehingga untuk mengetahui ada tidaknya suatu *worm* pada sistem komputer diperlukan adanya pendeteksian untuk

mengetahui keberadaan dari *worm* tersebut dan nama yang dipakai dari *worm* sebagai identitas dirinya.

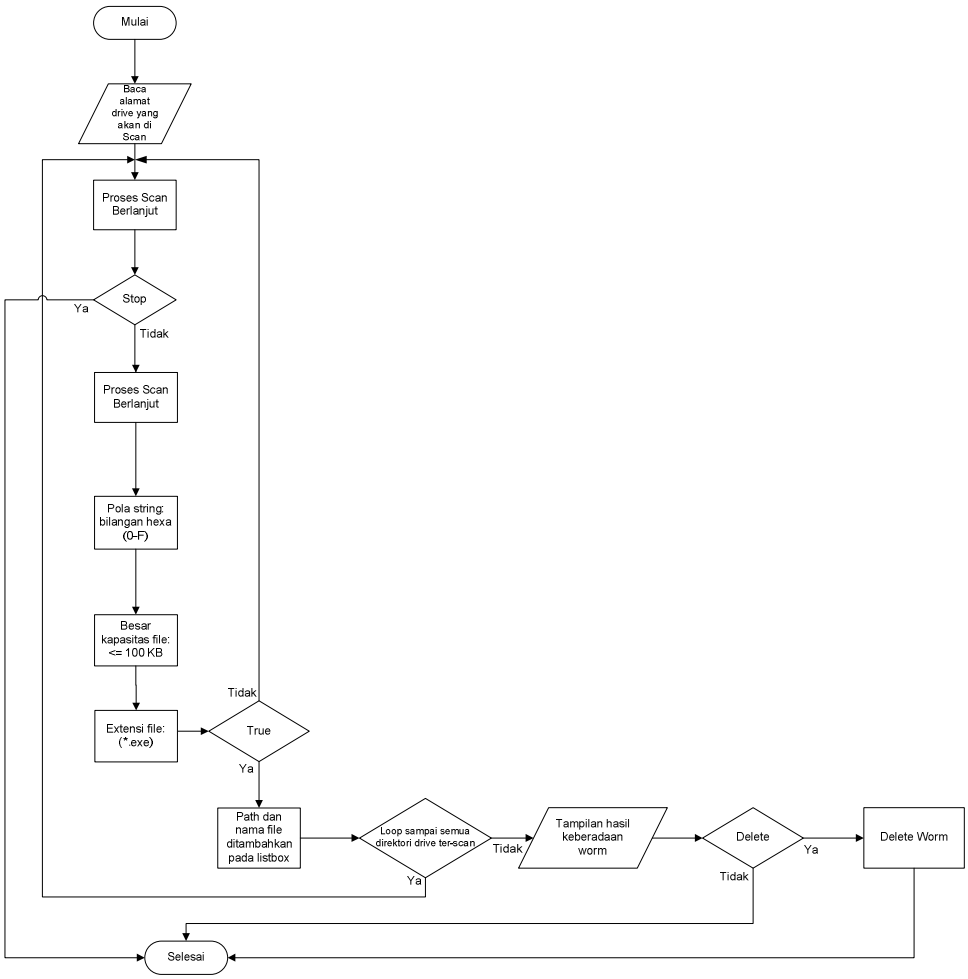
Untuk mengatasi masalah tersebut, diperlukan adanya suatu aplikasi yang dapat mendeteksi keberadaan *worm* dan nama yang dipakai sebagai identitas dari *worm* tersebut.

3.2 Langkah-Langkah Penyelesaian Masalah

Berikut adalah diagram alir yang menggambarkan langkah-langkah penyelesaian masalah tersebut.



Gambar 3.1 Proses Scanning Dengan Metode *File* Name Scanning



Gambar 3.2 Proses Scanning Dengan Metode String Scanning

Berikut akan dijelaskan mengenai gambar flow chart di atas.

Metode *File Name Scanning*

1. Program akan melakukan *scan* terhadap *drive* yang kita masukan sebagai *input drive*, sehingga program scan dapat melakukan *scanning* terhadap *drive* tertentu.

2. Selama melakukan *scan user* dapat menghentikan proses tersebut kapan saja, yang divalidasi oleh komputer dengan suatu pertanyaan apakah Anda yakin akan menghentikan atau tidak.
3. Proses scan berjalan.
4. Proses akan menyeleksi file dengan pola nama tertentu.
5. Proses akan menyeleksi besar file dengan kapasitas <= 100 KB.
6. Proses akan menyeleksi file yang hanya ber-extensi .exe.
7. Kemudian setelah proses *scan* selesai akan ada *result* mengenai keberadaan suatu *worm* pada sistem komputer.
8. Selanjutnya program mengijinkan untuk melakukan penghapusan terhadap *worm* yang terdeteksi dari sistem komputer.

Metode *String Scanning*

1. Program akan melakukan *scan* terhadap *drive* yang kita masukan sebagai *input drive*, sehingga program scan dapat melakukan *scanning* terhadap *drive-drive* tertentu.
2. Selama melakukan *scan user* dapat menghentikan proses tersebut kapan saja, yang divalidasi oleh komputer dengan suatu pertanyaan apakah Anda yakin akan menghentikan atau tidak.
3. Proses scan berjalan.
4. Proses akan menyeleksi file dengan pola nama tertentu.
5. Proses akan menyeleksi besar file dengan kapasitas <= 100 KB.
6. Proses akan menyeleksi file yang hanya ber-extensi .exe.

7. Kemudian setelah proses *scan* selesai akan ada *result* mengenai keberadaan suatu *worm* pada sistem komputer.
8. Selanjutnya program mengijinkan untuk melakukan penghapusan terhadap *worm* yang terdeteksi dari sistem komputer.

3.3 Analisis Metode

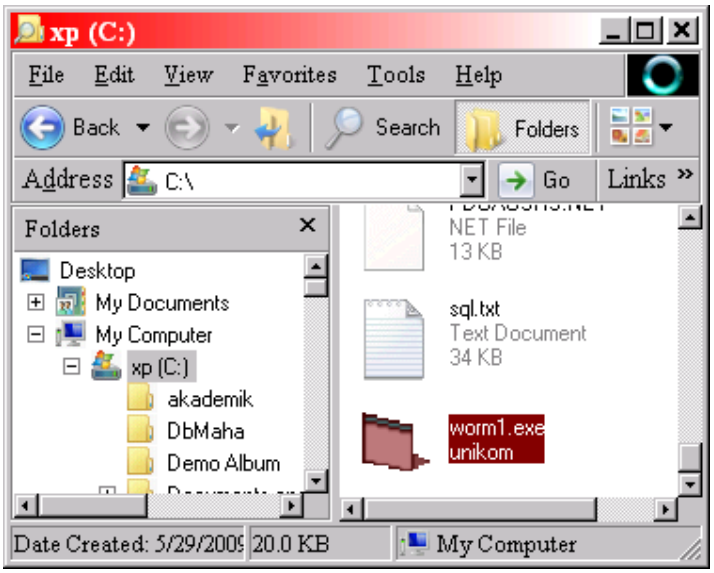
Berikut akan dibahas mengenai metode-metode yang dipergunakan untuk melakukan pendeteksian *worm* pada komputer.

3.3.1 Analisis Metode *File* Name Scanning

File Name Scanning adalah metode pencarian yang dilakukan dengan melakukan pencocokan nama *file* dengan data signature. Data signature bisa berupa nama *file* lengkap maupun berupa suatu pola yang diperoleh dari nama *file worm* tersebut.

Algoritma:

1. Jika pengguna mengklik tombol *scan* maka program melakukan pemeriksaan terhadap semua *file executable* pada *drive* dan direktori yang ditentukan oleh pengguna. Proses *scanning* dapat sewaktu-waktu dihentikan pengguna dengan mengklik tombol stop.
2. Jika nama *file* yang terperiksa memiliki nama *file* atau pola nama yang sama pada data *signature* dan memiliki besar file yang telah ditentukan, maka path dan nama *file* tersebut akan ditambahkan pada sebuah *ListBox*.
3. Pengguna dapat menghapus *file* yang ada pada *file ListBox* dengan mengklik ganda nama *file* tersebut pada *ListBox* atau menyeleksi *file-file* yang akan dihapus kemudian mengklik tombol *delete*.



Gambar 3.3 Sample Nama *File* yang diambil

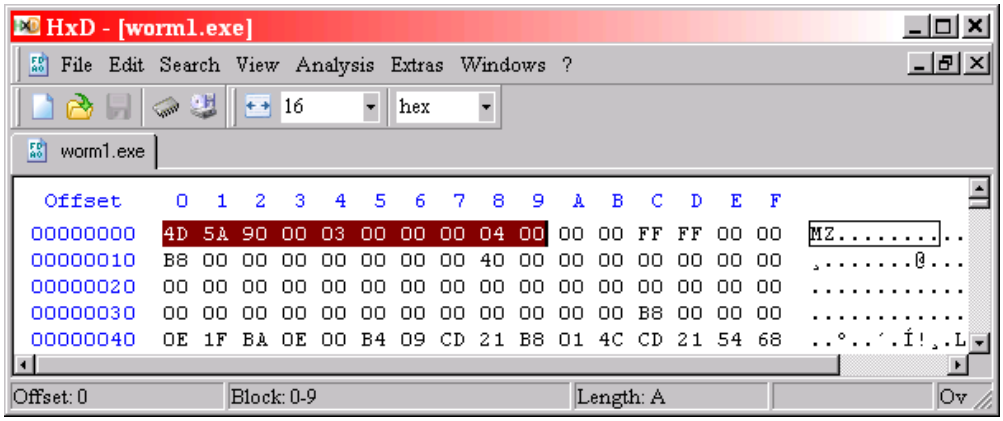
3.3.2 Analisis Metode String Scanning

String Scanning adalah metode pencarian pencarian yang dilakukan dengan melakukan pencocokan suatu kumpulan karakter pada data signature dengan isi *file*.

Algoritma:

1. Jika pengguna mengklik tombol *scan* maka program melakukan pemeriksaan terhadap semua *file executable* pada *drive* dan direktori yang ditentukan oleh pengguna. Proses *scanning* dapat sewaktu-waktu dihentikan pengguna dengan mengklik tombol stop.
2. Jika *file* yang terperiksa memiliki string yang sama pada data signature dan memiliki besar file yang telah ditentukan, maka path dan nama *file* tersebut akan ditambahkan pada sebuah *ListBox*.

3. Pengguna dapat menghapus *file* yang ada pada *ListBox* dengan mengklik ganda nama *file* tersebut pada *ListBox* atau menyeleksi *file-file* yang akan dihapus kemudian mengklik tombol *delete*.



Gambar 3.4 Sample String yang diambil

3.4 Analisis Karateristik *Worm*

Analisis karateristik *worm* ini dilakukan untuk mengetahui bagaimana karateristik suatu *worm* dari teknik penginfeksiannya, penyerangannya. Penyebaran *worm* cepat karena *design engineering* seperti virus. *Worm* berdiri sendiri (karena tidak memiliki *file* induk/*host* program) berbeda dengan virus.

A. Karateristik *Worm* Dilihat Dari Penyebarannya

1. *Worm* ada yang menyebar via *email*
2. *Worm* melakukan penyebarannya menggunakan *instant message* dengan mengirimkan *link* yang mengarah pada *web site* yang terinfeksi.
3. *Worm* menyebar dan aktif melalui komputer yang terkoneksi ke jaringan.
4. *Worm* ada yang menargetkan penyebarannya pada *channel chat* dengan mengirimkan *link*.

5. *Worm* menyebar dengan meng-*copy* dirinya pada *shared folder*.

B. Karateristik *Worm* Dilihat Dari Teknik Penyerangannya

1. *Worm* mempunyai kemampuan penyerangan secara spesifik, pertama melakukan proses pengintaian atau tidak, dengan bertujuan mendapatkan akses masuk dan mengusai sistem.
2. *Worm* Melakukan Penyerangan Pada Sistem Target

   a. Scanning

      Worm melakukan scanning secara acak pada IP *Address* yang dibuat menggunakan *random* IP *address generator*, dikenal sebagai *hit list scanning*. Proses pertama yang dilakukan adalah melakukan pemeriksaan terhadap status sistem target. Jika terget ditemukan sedang dalam kondisi menyala, proses selanjutnya adalah mendeteksi *servive* yang tersedia pada *remote*. *Worm* modern dapat menggunakan metode *scanning* permutasi, berdasarkan perubahan urutan angka.

   b. Multimode dan Dual Mode

      Worm jenis multimode akan secara otomaits melakukan scanning pada banyak *vulnerability holes* secara sekaligus pada setiap titik target, sementara *worm* jenis dual mode akan menyelesaiakan proses *scanning* sebuah *vulnerability hole* pada sejumlah target dan akan mengulanginya lagi pada *vulnerability hole* kedua.

c. Stealth Defense

Worm mempunyai kemampuan menyembunyikan dan melindungi diri dari antivirus yang umum. Hal ini menyebakan *worm* akan erus bercokol meskipun pemeriksaan antivirus pada sistem menyatakan bahwa mesin tersebut benar-benar bersih atau jika antivirus menemukan sebuah *worm*, antivirus maupun sistem yang menjadi korban tidak mampu membersihkan *worm* tersebut. *Worm* yang telah berhasil menginfeksi *file-file* penting yang dibuthkan oleh sistem untuk berjalan, menyebabkan proses removal *worm* akan menjadi sangat sulit.

d. *Worm* meng-*copy* pada pusat jaringan

e. Meng-eksploit kelemahan dari sistem operasi untuk mengambil alih komputer atau jaringan.

f. Mengambil alih jaringan publik.

g. Menggunakan/menumpang *malware* lain yang bertindak sebagai pembawa dari *worm*.

3.5 Analisis Pengenalan File Worm

Analisis pengenalan ini bertujuan untuk mengetahui apa saja yang menjadi dasar atau parameter dalam pendeteksian suatu file, dalam hal ini *file worm*.

3.5.1 File Name Scanning

Dasar yang digunakan atau aturan yang digunakan oleh *fiile name scanning* sebagai berikut:

Nama File (name file)

(A-Z;a-z;0-1) → kangen

Besar kapasitas file

(A-Z;a-z;0-1) → kangen (20kb)

Extensi file

(A-Z;a-z;0-1) → kangen.exe (20kb)

Table 3.1 Aturan file name scanning

File Name Scanning			
Worm	**Nama**	**Kapasitas**	**Extensi**
Kangen	Kangen	28 KB	*.exe
Pesin	Pesin	28 KB	*.exe
Gita	Gita	60 KB	*.exe

3.5.2 String Scanning

Dasar yang digunakan atau aturan yang digunakan oleh *string scanning* sebagai berikut:

String (hexa number)

(0-F) → kangen

Besar kapasitas file

(0-F) → kangen (20kb)

Extensi file

(0-F) → kangen.exe (20kb)

Table 3.2 Aturan string scanning

String Scanning			
Worm	**Hexa number**	**Kapasitas**	**Extensi**
Kangen	C0007700C000B7B78000	28 KB	*.exe
Pesin	F83F8301FC7FFFFFFC7F	28 KB	*.exe
Gita	BCACACFF837D7DFF6666	60 KB	*.exe

3.6 Analisis Kerugian

Worm dapat menyebabkan kerugian-kerugian, salah satunya adalah sebagai berikut.

3.6.1 Spionasme dan Pengumpulan Data

Worm dapat dilakukan sebagai alat untuk melakukan spionasi dan pengumpulan data dengan mencari keyword tertentu seperti nomor kartu kredit atau informasi penting lainnya pada dokumen-dokumen yang tersimpan pada host yang sudah menjadi korban.

3.6.2 Pengerusakan Data

Ada banyak virus dan worm yang melakukan pengerusakan data seperti Chernobyl dan Klez, yang memiliki perintah-perintah penghapusan data. Karena worm dapat menyebar dengan cepat, mereka dapat mulai menghapus atau memanipulasi data dari awal proses infeksi.

3.6.3 Pengerusakan Hardware

Walaupun sebagian besar BIOS memiliki kemampuan untuk mencegah proses reflashing, beberapa worm memiliki routine yang mampu melakukan pengerusakan terhadap BIOS jenis tertentu.

3.7 Analisis Keakuratan Metode

Worm dari segi bahasanya berarti cacing, worm merupakan salah satu anggota keluarga *malware* yang biasanya banyak orang salah kaprah menyamakan worm dengan virus. Dari segi desain, worm hampir mirip dengan virus, namun tidak sama dengan virus. Perbedaannya terletak pada cara bereproduksi (*spread action*), dimana worm bereproduksi tidak dengan cara infeksi tetapi dengan cara menggandakan diri (*self copy*) dan biasanya tidak bersifat polymorphic (berubah-ubah), sehingga mudah dideteksi oleh antivirus maupun removal (khususnya *removal* yang menggunakan teknik *File Name Scanning* dan String Scanning sebagai *engine detection*-nya), sehingga pembersihan worm relatif lebih mudah dibandingkan dengan virus. Kebanyakan dan hampir semua virus lokal adalah dari jenis ini. Contohnya adalah brontok, decoil, kangen dll.

3.8 Analisis Kecepatan

Analisis kecepatan antara kedua metode yang digunakan, akan didapat dari hasil uji coba pada aplikasi yang akan dibahas pada bab 4.

3.9 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan bagian penting agar program aplikasi dapat berjalan dengan baik. Program aplikasi pendeteksian *worm* membutuhkan spesifikasi perangkat keras, perangkat lunak dan spesifikasi *user* (pengguna) pendukung dengan spesifikasi sebagai berikut:

3.9.1 Analisis Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan dan menguji sistem ini adalah sebagai berikut:

1. Prosesor : Intel Core 2 Duo 1.66 Ghz
2. Memori : 1 Gb
3. Harddisk : 80 Gb
4. VGA : Intel
5. Monitor : 15” 1024 x 800 pixel
6. Keyboard & Mouse

3.9.2 Analisis Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membuat serta untuk mengimplementasikan program aplikasi pendeteksian *worm* ini adalah sebagai berikut:

1. Sistem Operasi Windows Xp Professional SP2
2. Ms. Visio 2003
3. Ms. Word 2007
4. Visual Basic 6.0

3.9.3 Analisis Kebutuhan Pengguna

Pengguna dari aplikasi ini adalah orang yang sudah berpengalaman dalam mengaplikasikan komputer atau yang telah mengikuti *training* aplikasi ini.

3.10 Analisis Basis Data

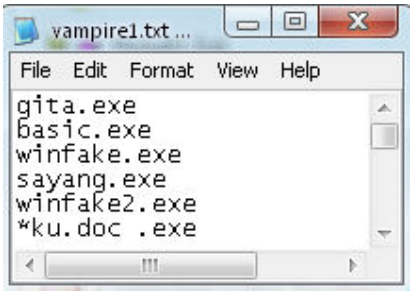
Basis data dalam aplikasi ini hanya berupa *file* yang menyimpan beberapa *field* yang berkaitan dengan *worm*, seperti kerusakan, penyebaran, perilaku dan solusi. Tentu kesemuanya itu adalah dari *worm* yang sudah atau telah diketahui.

Kemudian mengingat kedua metode yang digunakan adalah metode yang didasarkan pada keberadaan worm yang telah diketahui dari sisi namanya dan

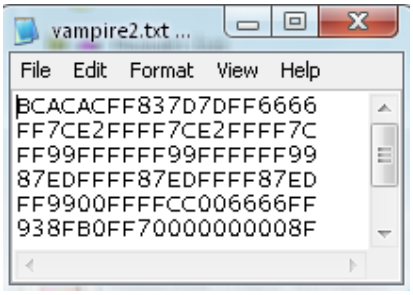
string-nya. Oleh karena itu, di dalam program ditambah suatu *database* berupa *file* teks yang di dalamnya dipergunakan untuk mendaftarkan *worm-worm* yang sudah diketahui. Sehingga program akan mengacu ke file teks tersebut untuk melihat *worm* apa saja yang dapat dikenali oleh program aplikasi.

Dari hasil analisis, didapat data yang akan dipakai dalam proses pembangunan program aplikasi.

Kemudian untuk dua file teks dapat dilihat pada gambar di bawah ini.



Gambar 3.5 File Teks Untuk *Name*



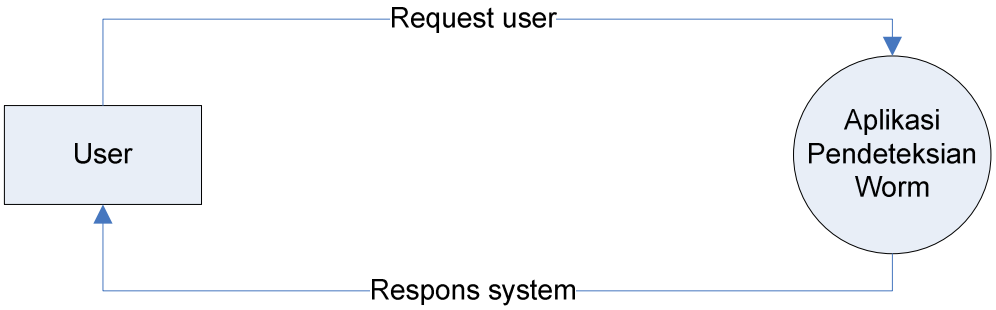
Gambar 3.6 File Teks Untuk String

3.11 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk memberikan gambaran aliran data yang ada pada program aplikasi yang akan dibangun.

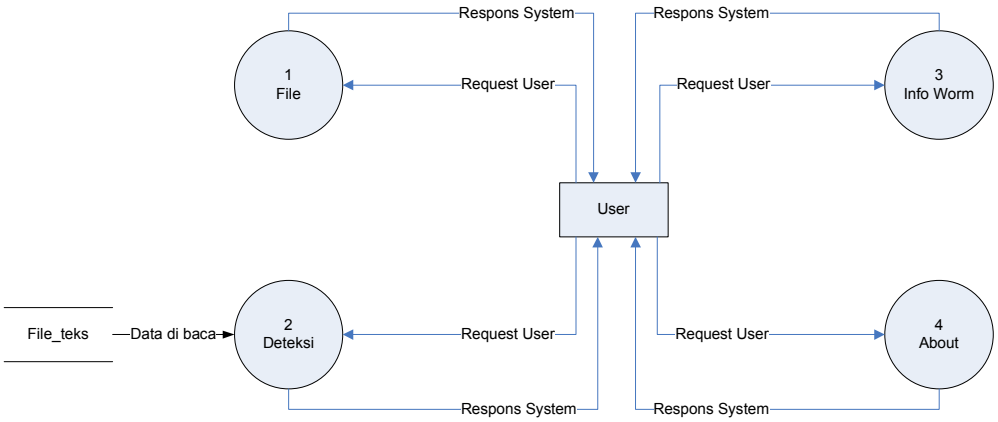
3.11.1 Diagram Konteks

Diagram konteks dibentuk memiliki tujuan untuk menggambarkan hubungan keseluruhan dari sistem dengan entitas-entitas yang ada. Diagram konteks adalah diagram tingkat atas yang merupakan diagram global dari sistem informasi yang menggambarkan aliran-aliran data dari entitas-entitas yang masuk dan yang keluar dari sistem.



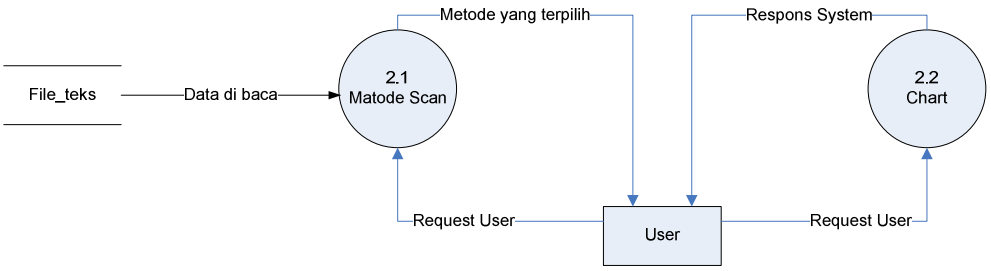
Gambar 3.7 Diagram Konteks Aplikasi Pendeteksian *Worm*

3.11.1.1 DFD Level 0



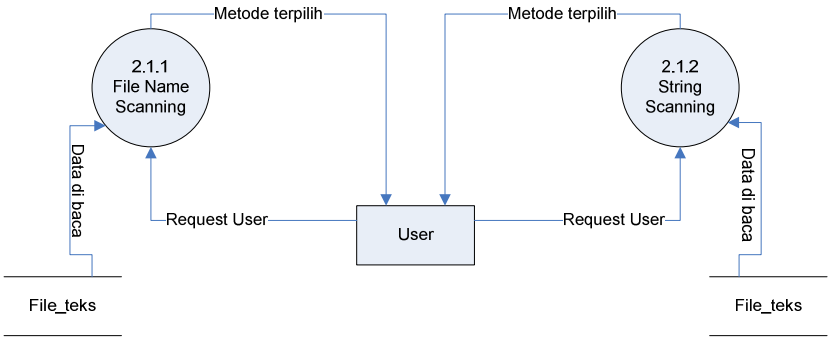
Gambar 3.8 DFD Level 0

3.11.1.2 DFD Level 1 Proses 2 (Pemilihan Menu Deteksi)



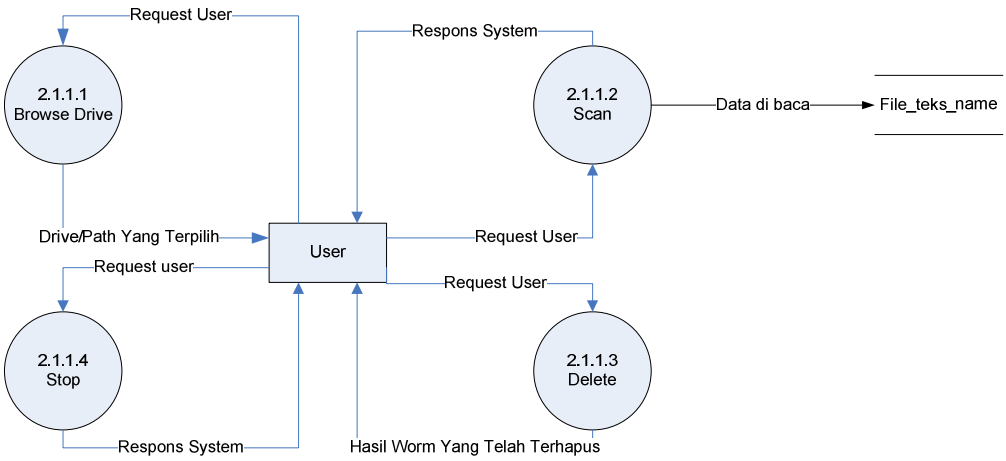
Gambar 3.9 DFD Level 1 Pemilihan Menu Deteksi

3.11.1.3 DFD Level 2 Proses 2.1 (Metode Scan)



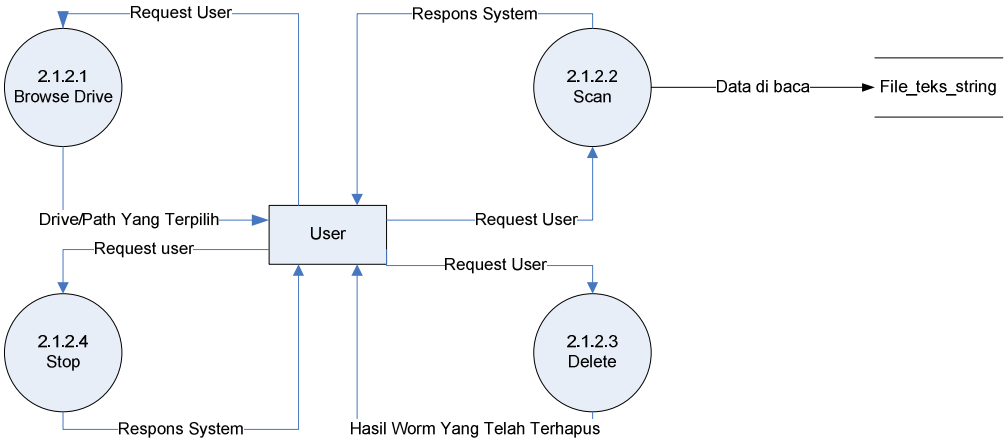
Gambar 3.10 DFD Level 2 Pemilihan Menu Metode Scan

3.11.1.4 DFD Level 3 Proses 2.1.1 (File Name Scanning)



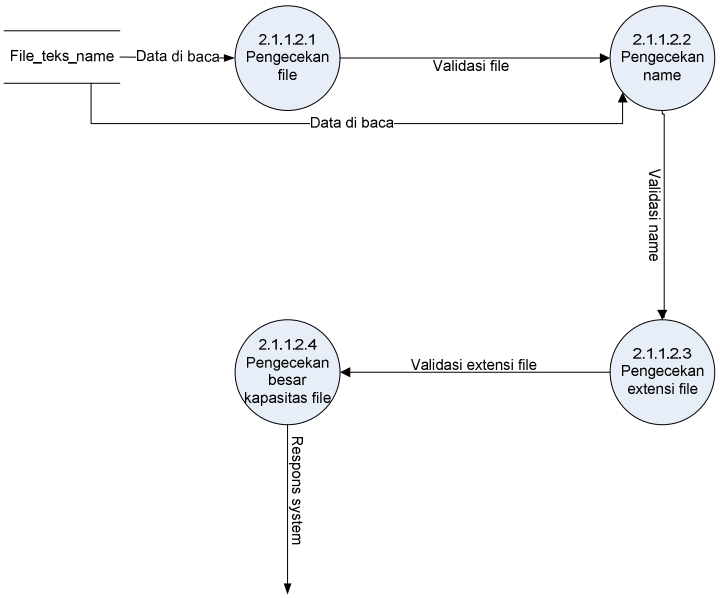
Gambar 3.11 DFD Level 3 Pemilihan Menu File Name Scanning

3.11.1.5 DFD Level 3 Proses 2.1.2 (String Scanning)



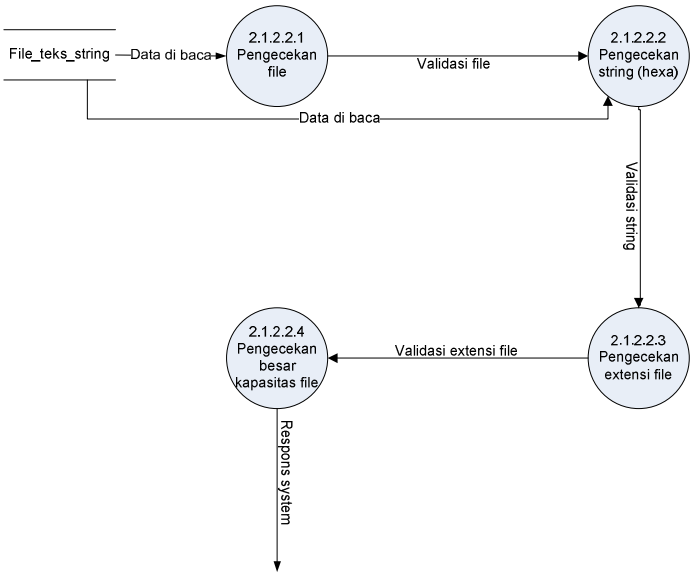
Gambar 3.12 DFD Level 3 Pemilihan Menu String Scanning

3.11.1.6 DFD Level 4 Proses 2.1.1.2 (Scan)



Gambar 3.13 DFD Level 4 Pemilihan Tombol Scan

3.11.1.7 DFD Level 4 Proses 2.1.2.2 (Scan)



Gambar 3.14 DFD Level 4 Pemilihan Tombol Scan

3.12 Spesifikasi Proses

Spesifikasi proses digunakan untuk menggambarkan semua proses model aliran yang nampak pada tingkat akhir penyaringan. Dengan memberikan sebuah PSPEC untuk mengiringi masing-masing gelembung didalam model aliran, akan menciptakan sebuah spesifikasi mini yang dapat berfungsi sebagai penuntun bagi desain komponen program yang akan mengimplementasikan program.

Table 3.3 Spesifikasi Proses DFD Level 0

NO	PROSES	KETERANGAN
1	No. Proses	1
	Nama Proses	*File*
	Source (sumber)	User
	Input	Klik menu *file*
	Output	Menampilkan menu *file*
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu *file* aplikasi} if *file* dipilih then tampil menu *file* End
2	No. Proses	2
	Nama Proses	Deteksi
	Source (sumber)	User
	Input	Klik menu metode
	Output	Menampilkan menu metode
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu metode aplikasi} If deteksi dipilih then tampil menu scan End
3	No. Proses	3
	Nama Proses	Info Worm
	Source (sumber)	User
	Input	Klik menu window
	Output	Menampilkan menu window
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu window aplikasi}

		If info worm dipilih then tampil menu info worm End
4	No. Proses	4
	Nama Proses	About
	Source (sumber)	User
	Input	Klik menu about
	Output	Menampilkan jendela about
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu about aplikasi} If about dipilih then tampil jendela about End

Table 3.4 Spesifikasi Proses DFD Level 1 Proses 2 (Pemilihan Menu Deteksi)

NO	PROSES	KETERANGAN
1	No. Proses	2.1
	Nama Proses	Metode Scan
	Source (sumber)	User
	Input	Pilih metode
	Output	Menmpilkan antarmuka metode
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu metode} If metode dipilih then tampil jendela metode End
2	No. Proses	2.2
	Nama Proses	Chart
	Source (sumber)	User
	Input	Pilih chart
	Output	Menampilkan antarmuka chart
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu chart} If chart dipilih then tampil jendela chart End

Table 3.5 Spesifikasi Proses DFD Level 2 Proses 2.1 (Pemilihan Menu Metode Scan)

NO	PROSES	KETERANGAN
1	No. Proses	2.1.1
	Nama Proses	File Name Scanning
	Source (sumber)	User
	Input	Pilih metode
	Output	Menmpilkan antarmuka metode
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu metode} If metode dipilih then tampil jendela metode End
2	No. Proses	2.1.2
	Nama Proses	String Scanning
	Source (sumber)	User
	Input	Pilih chart
	Output	Menampilkan antarmuka chart
	Destination (tujuan)	User
	Logika Proses	Begin {user dapat memilih menu chart} If chart dipilih then tampil jendela chart End

Table 3.6 Spesifikasi Proses DFD Level 3 Proses 2.1.1 (Pemilihan Menu File Name Scanning)

NO	PROSES	KETERANGAN
1	No. Proses	2.1.1.1
	Nama Proses	Browse Drive
	Source (sumber)	User
	Input	Klik tombol browse
	Output	Tampilan drive harddisk
	Destination (tujuan)	User

	Logika Proses	Begin
		Alamat drive dimasukan
		If (Alamat drive sesuai)
		{ Alamat berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else

		{Alamat gagal dimasukan} End
2	No. Proses	2.1.1.2
	Nama Proses	Scan
	Source (sumber)	User
	Input	Klik tombol scan
	Output	Proses scan berlangsung
	Destination (tujuan)	User

	Logika Proses	Begin
		scan dijalankan
		If (alamat drive sesuai)
		{ scan berhasil dijalankan
		Then
		lanjut ke proses berikutnya}
		Else
		{scan gagal dijalankan}
		End

3	No. Proses	2.1.1.3
	Nama Proses	Delete
	Source (sumber)	User
	Input	Klik tombol delete
	Output	*Worm* terhapus
	Destination (tujuan)	User

	Logika Proses	Begin
		User menekan tombol delete
		If (data sesuai)
		{ Data berhasil dihapus
		Then
		lanjut ke proses berikutnya}
		Else
		{Data gagal dihapus}
		End

4	No. Proses	2.1.1.4
	Nama Proses	Stop Scan
	Source (sumber)	User
	Input	Klik tombol stop
	Output	Proses scan berhenti
	Destination (tujuan)	User

	Logika Proses	Begin
		User menekan tombol
		If (proses sukses)
		{Stop berhasil dijalankan

		Then lanjut ke proses berikutnya} Else {Stop gagal dijalankan} End

Table 3.7 Spesifikasi Proses DFD Level 3 Proses 2.1.2 (Pemilihan Menu String Scanning)

NO	PROSES	KETERANGAN
1	No. Proses	2.1.2.1
	Nama Proses	Browse Drive
	Source (sumber)	User
	Input	Klik tombol browse
	Output	Tampilan drive harddisk
	Destination (tujuan)	User

	Logika Proses	Begin
		Alamat drive dimasukan
		If (Alamat drive sesuai)
		{ Alamat berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{Alamat gagal dimasukan}
		End

2	No. Proses	2.1.2.2
	Nama Proses	Scan
	Source (sumber)	User
	Input	Klik tombol scan
	Output	Proses scan berlangsung
	Destination (tujuan)	User

	Logika Proses	Begin
		scan dijalankan
		If (alamat drive sesuai)
		{ scan berhasil dijalankan
		Then
		lanjut ke proses berikutnya}
		Else
		{scan gagal dijalankan}
		End

3	No. Proses	2.1.2.3

	Nama Proses	Delete
	Source (sumber)	User
	Input	Klik tombol delete
	Output	*Worm* terhapus
	Destination (tujuan)	User

	Logika Proses	Begin
		User menekan tombol delete
		If (data sesuai)
		{ Data berhasil dihapus
		Then
		lanjut ke proses berikutnya}
		Else
		{Data gagal dihapus}
		End

4	No. Proses	2.1.2.4
	Nama Proses	Stop Scan
	Source (sumber)	User
	Input	Klik tombol stop
	Output	Proses scan berhenti
	Destination (tujuan)	User

	Logika Proses	Begin
		User menekan tombol
		If (proses sukses)
		{Stop berhasil dijalankan
		Then
		lanjut ke proses berikutnya}
		Else
		{Stop gagal dijalankan}
		End

Table 3.8 Spesifikasi Proses DFD Level 4 Proses 2.1.1.2 (Scan)

NO	PROSES	KETERANGAN
1	No. Proses	2.1.1.2.1
	Nama Proses	Pengecekan file name
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (file sesuai)
		{ file berhasil dimasukan

		Then lanjut ke proses berikutnya} Else {file gagal ditemukan} End
2	No. Proses	2.1.1.2.2
	Nama Proses	Pengecekan pola name
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (pola sesuai)
		{ file berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{file gagal ditemukan}
		End

3	No. Proses	2.1.1.2.3
	Nama Proses	Pengecekan extensi file
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (extensi sesuai)
		{ file berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{file gagal ditemukan}
		End

4	No. Proses	2.1.1.2.4
	Nama Proses	Pengecekan besar kapasitas file
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (besar kapasitas file sesuai)
		{ file berhasil dimasukan

		Then lanjut ke proses berikutnya} Else {file gagal ditemukan} End

Table 3.9 Spesifikasi Proses DFD Level 4 Proses 2.1.2.2 (Scan)

NO	PROSES	KETERANGAN
1	No. Proses	2.1.2.2.1
	Nama Proses	Pengecekan file name
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (file sesuai)
		{ file berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{file gagal ditemukan}
		End

2	No. Proses	2.1.2.2.2
	Nama Proses	Pengecekan pola name
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (pola sesuai)
		{ file berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{file gagal ditemukan}
		End

3	No. Proses	2.1.2.2.3
	Nama Proses	Pengecekan extensi file
	Source (sumber)	User
	Input	File teks

	Output	Terdeteksinya file
	Destination (tujuan)	User

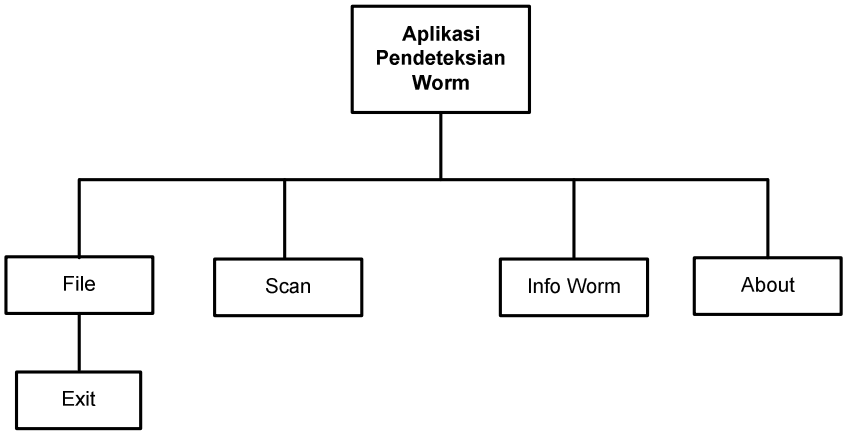
	Logika Proses	Begin
		If (extensi sesuai)
		{ file berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{file gagal ditemukan}
		End

4	No. Proses	2.1.2.2.4
	Nama Proses	Pengecekan besar kapasitas file
	Source (sumber)	User
	Input	File teks
	Output	Terdeteksinya file
	Destination (tujuan)	User

	Logika Proses	Begin
		If (besar kapasitas file sesuai)
		{ file berhasil dimasukan
		Then
		lanjut ke proses berikutnya}
		Else
		{file gagal ditemukan}
		End

3.13 Perancangan Struktur Menu

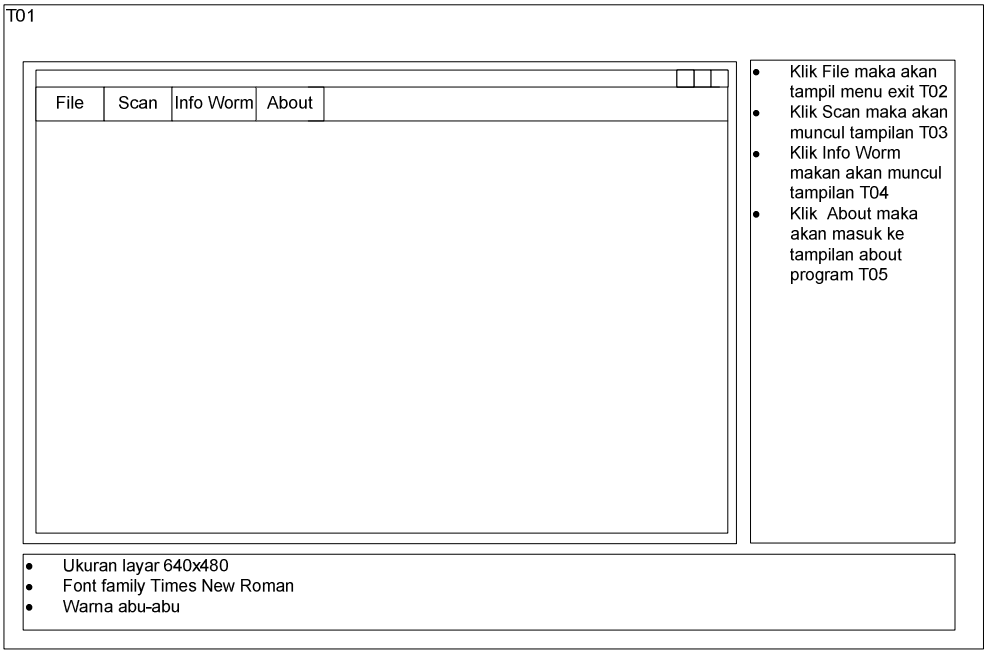
Dalam pembangunan aplikasi diperlukan sebuah struktur menu untuk menunjukan keterkaitan dan tingkatan antara menu utama dan sub menu, sehingga dapat lebih mudah dimengerti. Berikut ini adalah struktur menu dari aplikasi yang akan dibangun :



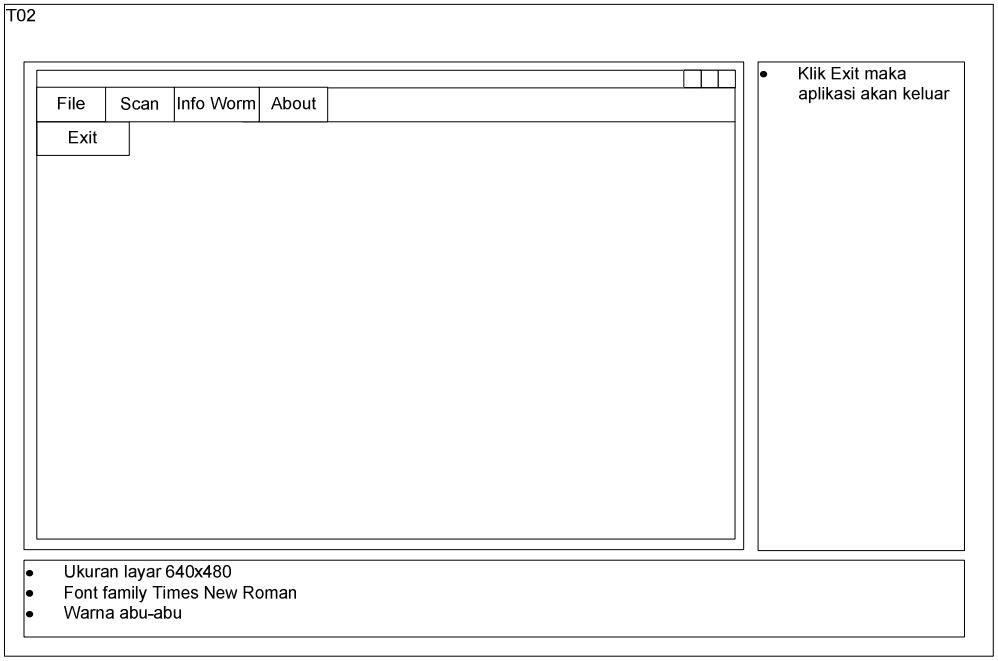
Gambar 3.15 Struktur Menu Aplikasi

3.14 Perancangan Antarmuka

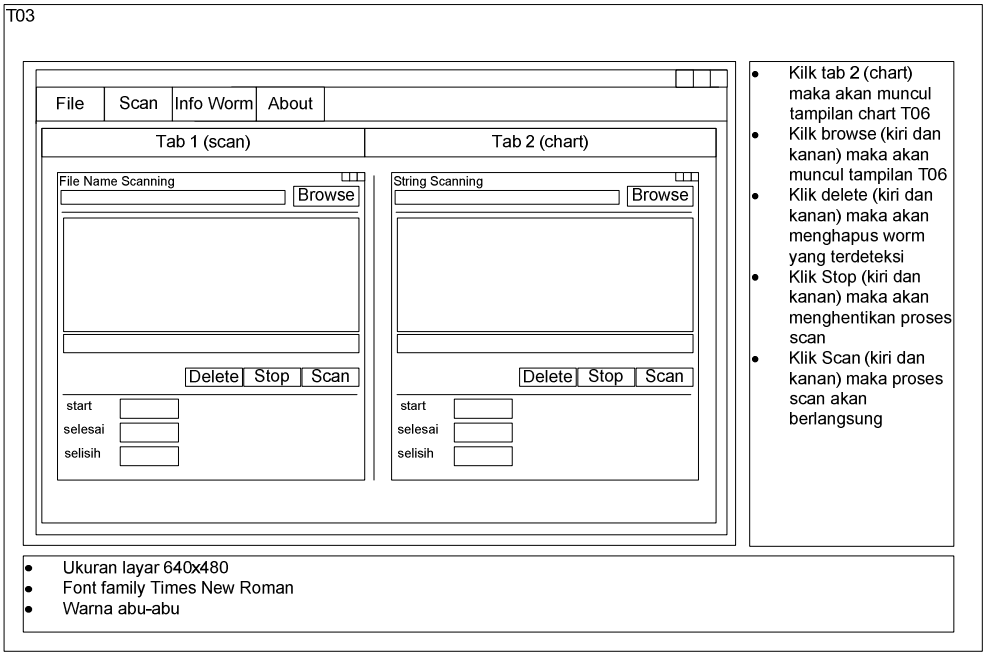
Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut akan disajikan perancangan antarmuka dari aplikasi yang akan dibangun.



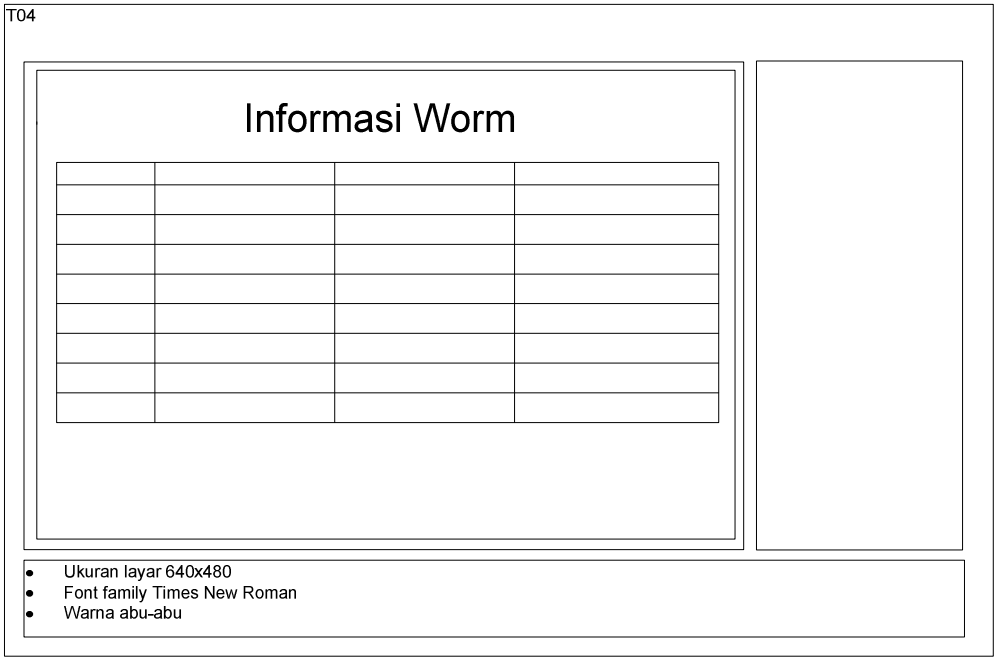
Gambar 3.16 Tampilan Menu Utama



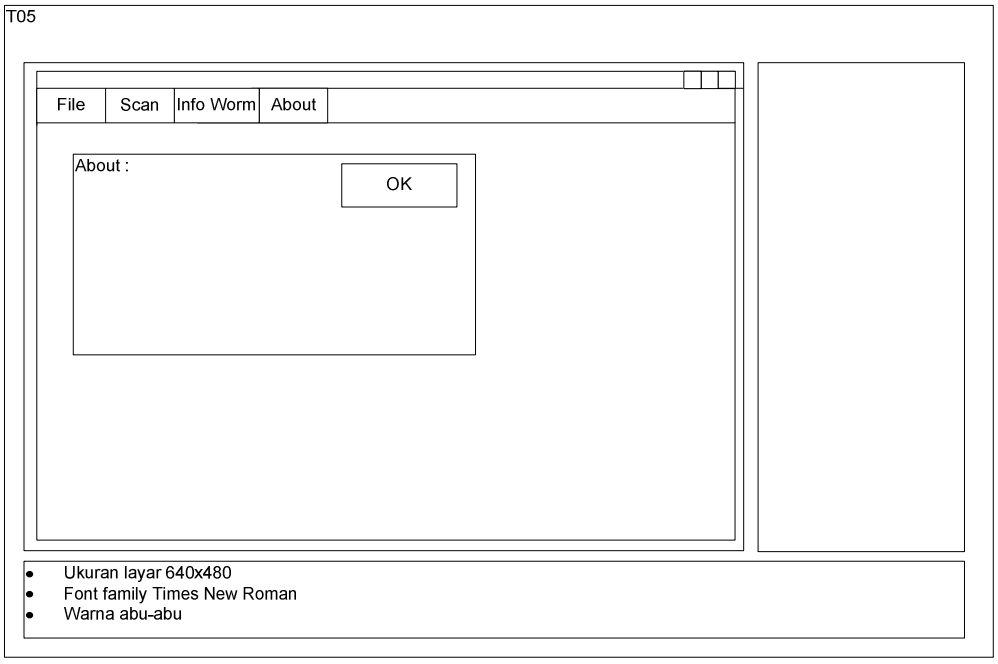
Gambar 3.17 Tampilan Menu *File*



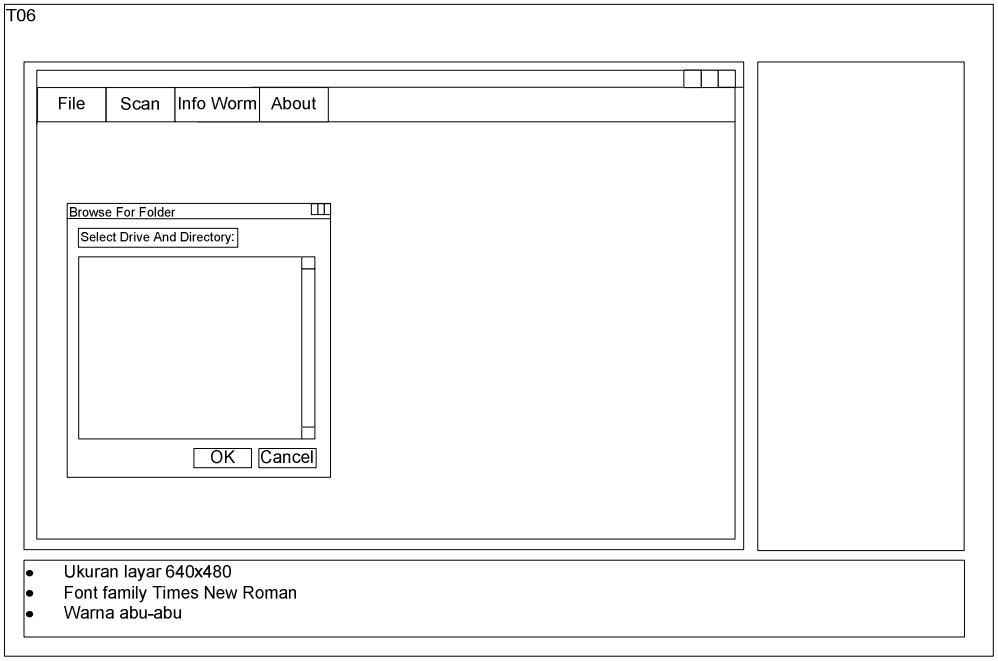
Gambar 3.18 Tampilan Menu *Scan*



Gambar 3.19 Tampilan Menu *Info Worm*



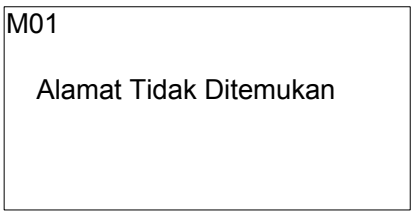
Gambar 3.20 Tampilan Menu *About*



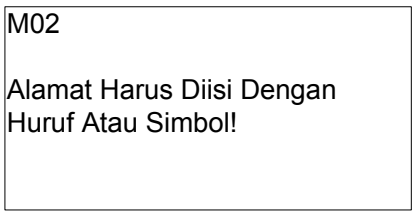
Gambar 3.21 Tampilan Menu *Browse*

3.15 Perancangan Pesan

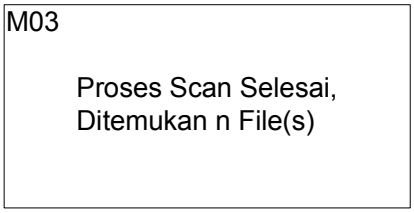
Dibawah ini adalah perancangan pesan yang ada pada aplikasi Pendeteksian *Worm*.



Gambar 2.22 Tampilan Pesan M01



Gambar 3.23 Tampilan Pesan M02



Gambar 3.24 Tampilan Pesan M03

3.16 Jaringan Semantik

Jaringan semantik merupakan alat efektif untuk merepresentasikan pemetaan data, yang bertujuan mencegah terjadinya duplikasi data.

Untuk penjelasan dari jaringan semantik yang ada dibawah ini akan diuraikan sebagai berikut :

T01 : Menu Utama

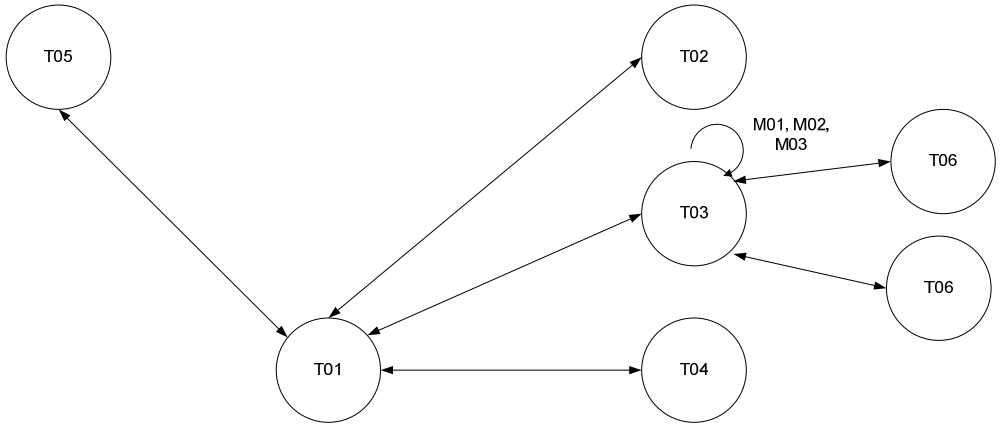
T02 : Menu *File*

T03 : Menu Scan

T04 : Menu Info Worm

T05 : About Program

T06 : Tampilan Browse



Gambar 3.25 Jaringan Semantik