BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisa Sistem yang Sedang Berjalan

Pada sistem yang lama proses pendeteksi virus pada komputer sudah menggunakan aplikasi dengan teknologi yang paling canggih. Tetapi ada juga yang tidak menggunakan aplikasi apapun sebagai penangkal masuknya virus pada komputer.Sehingga ketika komputer yang sudah terinfeksi dengan virus, para pengguna tidak mengetahui apakah komputernya sudah terinfeksi virus atau tidak.Informasi jenis-jenis virus yang sering menginfeksi komputer kurang dipahami oleh sebagian masyarakat luas.Tentu hal ini sangat memprihatinkan mengingat saat ini komputer menjadi alat yang sangat dibutuhkan.

Sistem yang sedang berjalan juga membuat banyak sering frustasi dan kehilangan control terhadap komputernya sendiri. Hal ini terjadi dikarenakan pengguna tidak mempunyai atau memiliki solusi ketika komputer yang sedang digunakannya sedang terifeksi virus. Karena ketidaktahuan atau kurangnya pemahaman akan virus tersebut, pengguna tersebut frustasi. Sehingga jalan pintas yang sering diambil sebagai solusi dari permasalahan tersebut adalah dengan menginstall ulang ataupun menjual dan membeli komputer baru.

Selain hal tersebut juga, pengguna pada sistem yang lama lebih ditekankan sebagai pengguna saja tanpa adanya interaksi antara aplikasi dengan pengguna.Dan ketika aplikasi yang digunakan menemukan file yang dicurigai sebagai virus, maka ada banyak pengguna yang langsung mengiyakan tanpa

mendiagnosa terlebih dahulu apakah file tersebut memang benar-benar sudah terifeksi virus.

III.1.1. *Input*

Bentuk inputan pada sistem yang sedang berjalan sangat beragam.Hal ini tergantung dari aplikasi yang digunakan.Ada aplikasi yang mengizinkan penggunanya untuk mengupload contoh virus, memeriksa file yang mencurigakan sebagai virus dan bahkan ada juga yang tidak menyediakan inputan bagi penggunanya.

III.1.2. Proses

Pada sistem yang sedang berjalan, proses sangat berbeda-beda, tergantung aplikasi yang digunakan dan pengguna sistem itu sendiri.Sebagian pengguna ada yang sudah menggunakan aplikasi anti virus dan ada juga pengguna yang memang tidak menggunakan aplikasi anti virus apapun sebagai pelindung komputer. Tetapi secara umum, proses yang terjadi pada sistem yang sedang berjalan adalah sebagai berikut :

1. Pengguna akan mengaktifkan aplikasi antivirus.
2. Aplikasi antivirus akan langsung melakukan pencarian terhadap virus atau file yang dicurigai sebagai virus.
3. Aplikasi antivirus akan menampilkan dafar file yang dicurigai sebagai virus dari hasil *scanning* yang telah dilakukan.
4. File yang sudah terinfeksi langsung dihapus dari komputer.
5. Selesai.

III.1.3. *Output*

Hasil akhir yang diperoleh dari sistem yang sedang berjalan adalah daftar nama-nama file yang dicurigai sebagai virus ataupun file yang sudah terinfeksi oleh virus.

III.2. Evaluasi Sistem yang Berjalan

Sesuai dengan apa yang telah penulis jabarkan pada sub bab sebelumnya, bahwa sistem yang lama masih memiliki kelemahan. Kelemahan ini diharapkan dapat diatasi oleh sistem yang akan dirancang nantinya. Adapun beberapa kelemahan yang penulis temukan selama melakukan penelitian adalah sebagai berikut :

1. Tidak ada interaksi yang terjadi antara aplikasi yang disedang digunakan dengan penggunanya.
2. Kurangnya informasi mengenai virus yang telah menginfeksi komputer pengguna.
3. Solusi dan pencegahan yang diberikan oleh aplikasi mutlak untuk dilakukan saat itu juga.
4. Tidak menyediakan informasi mengenai gejala-gejala dari virus yang sedang menginfeksi komputer.

III.3. Desain Sistem

Kelemahan sistem yang sedang berjalan perlu dipikirkan dan mencari solusi terbaik. Kelemahan ini dapat diperkecil dengan merancang suatu sistem yang dapat menutupi kelemahan tersebut. Dalam hal ini penulis akan mendesain

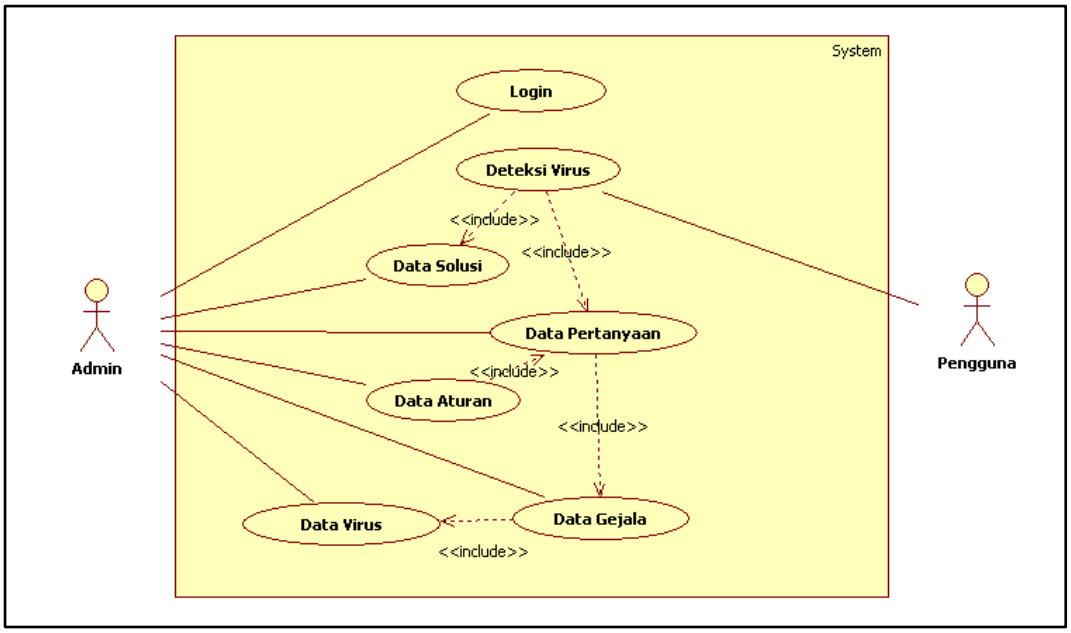
dan memberikan gambaran yang jelas mengenai rancang bangun sistem yang akan diusulkan.Pada tahap ini perlu membatasi rancang bangun sistem yang diusulkan agar lebih mudah dalam memahami sistem nantinya.Tahap ini terdapat dua bagian yakni, disain sistem secara global dan disain sistem secara detail.

III.3.1. Desain Sistem Secara Global

Perancangan sistem secara global akan menjelaskan gambaran umum sistem serta model sistem yang akan diusulkan. Karena sistem yang diusulkan akan menghasilkan sebuah perangkat lunak yang berorientasi objek, maka perlu melakukan pemodelan sistem berdasarkan objek-objek yang digunakan. Dalam pemodelan ini penulis menggunakan *Unfied Modeling Languange* (UML). Pada tahap pemodelan ataupun disain sistem secara global, penulis akan merancang sistem berdasarkan kebutuhan sistem yang akan diusulkan, seperti pembuata *use case diagram*, *sequence diagram* dan *class diagram.*

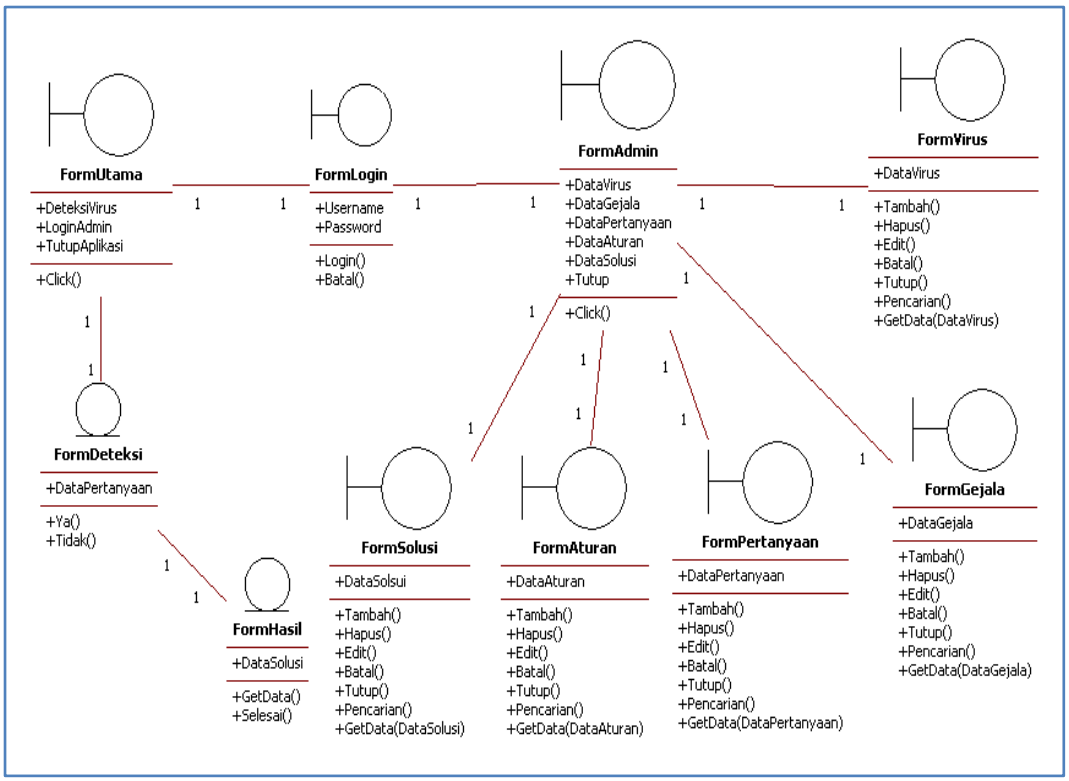
III.3.1.1. *Use Case Diagram*

Use case diagram digunakan untuk memahami bagaimana interaksi pengguna sistem dengan sistem yang dipakai secara keseluruhan. Pada *use case diagram* ini juga akan menjelaskan kegiatan apa saja yang dapat dilakukan oleh pengguna sistem dan batasan dalam mengakses sistem. Adapun *use case diagram* pada sistem yang diusulkan dapat dilihat pada gambar berikut ini :



Gambar III.1. *Use Case Diagram* Sistem Pakar Deteksi Virus

III.3.1.2. *Class Diagram*

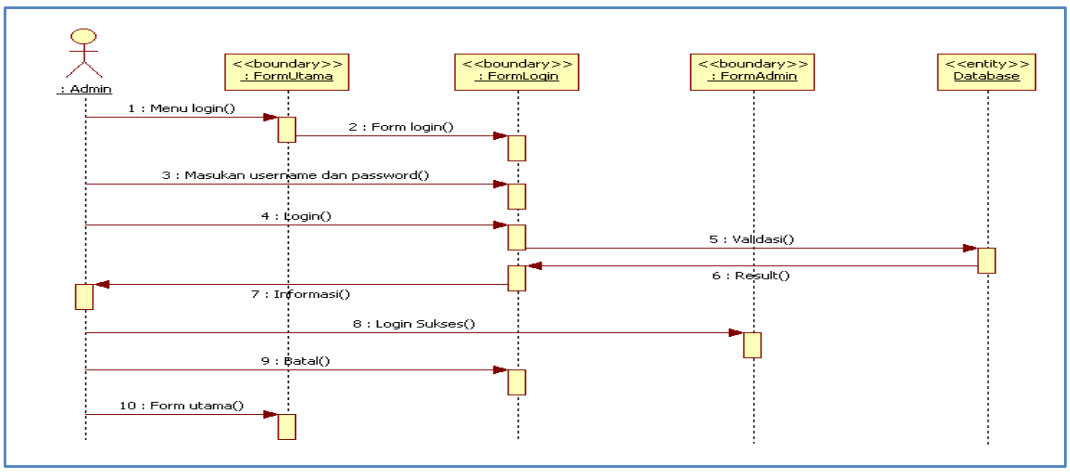


Gambar III.2. *Class Diagram* Sistem Pakar Deteksi Virus

III.3.1.3. *Sequence Diagram*

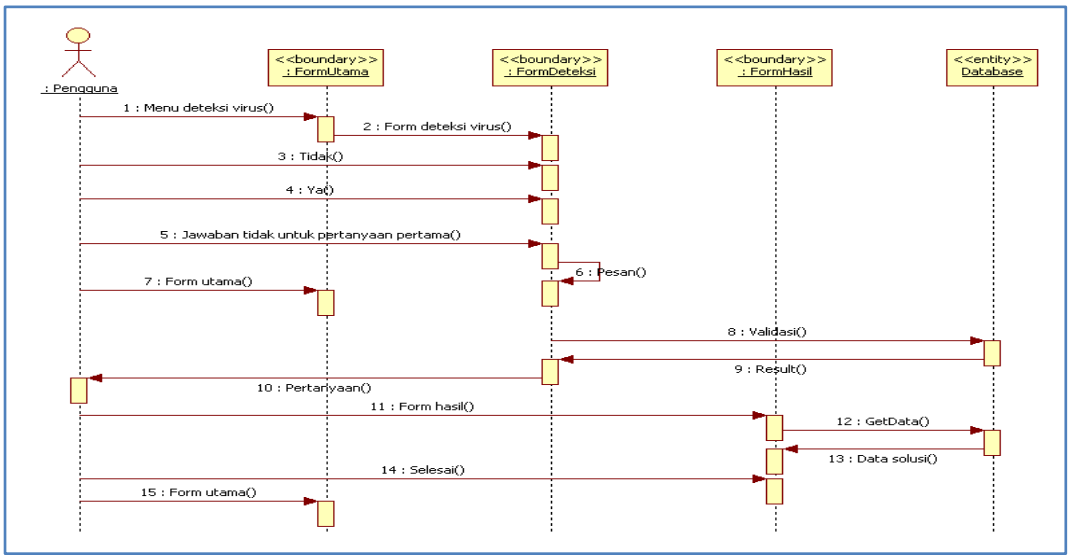
*Sequence Diagram*menggambarkan interaksi antara sejumlah *object* dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* serta interaksi antar object yang terjadi pada titik tertentu dalam eksekusi sistem yang diusulkan.

1. *Sequence Diagram Login*



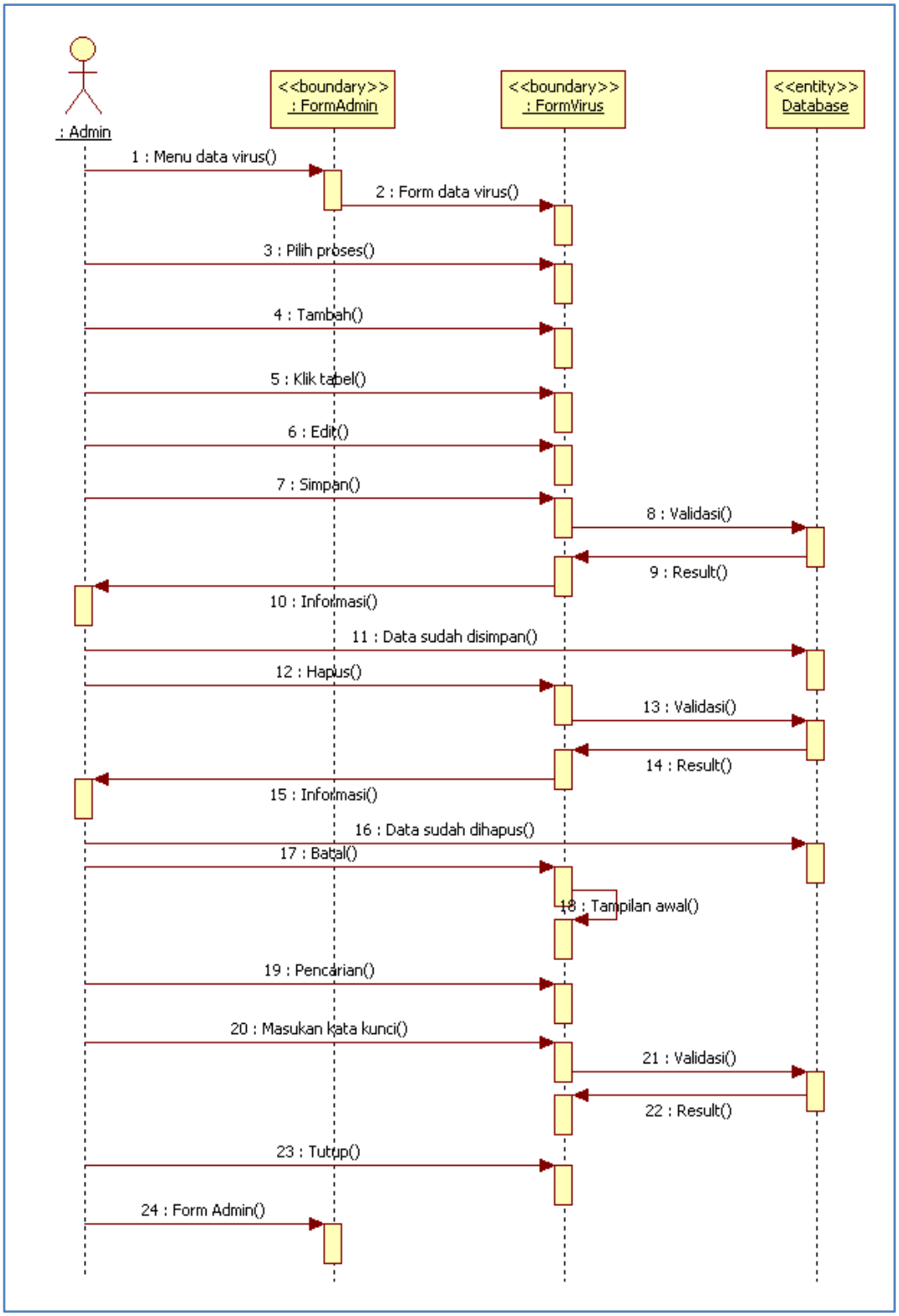
Gambar III.3. *Sequence Diagram Login*

2. *Sequence Diagram* Deteksi Virus



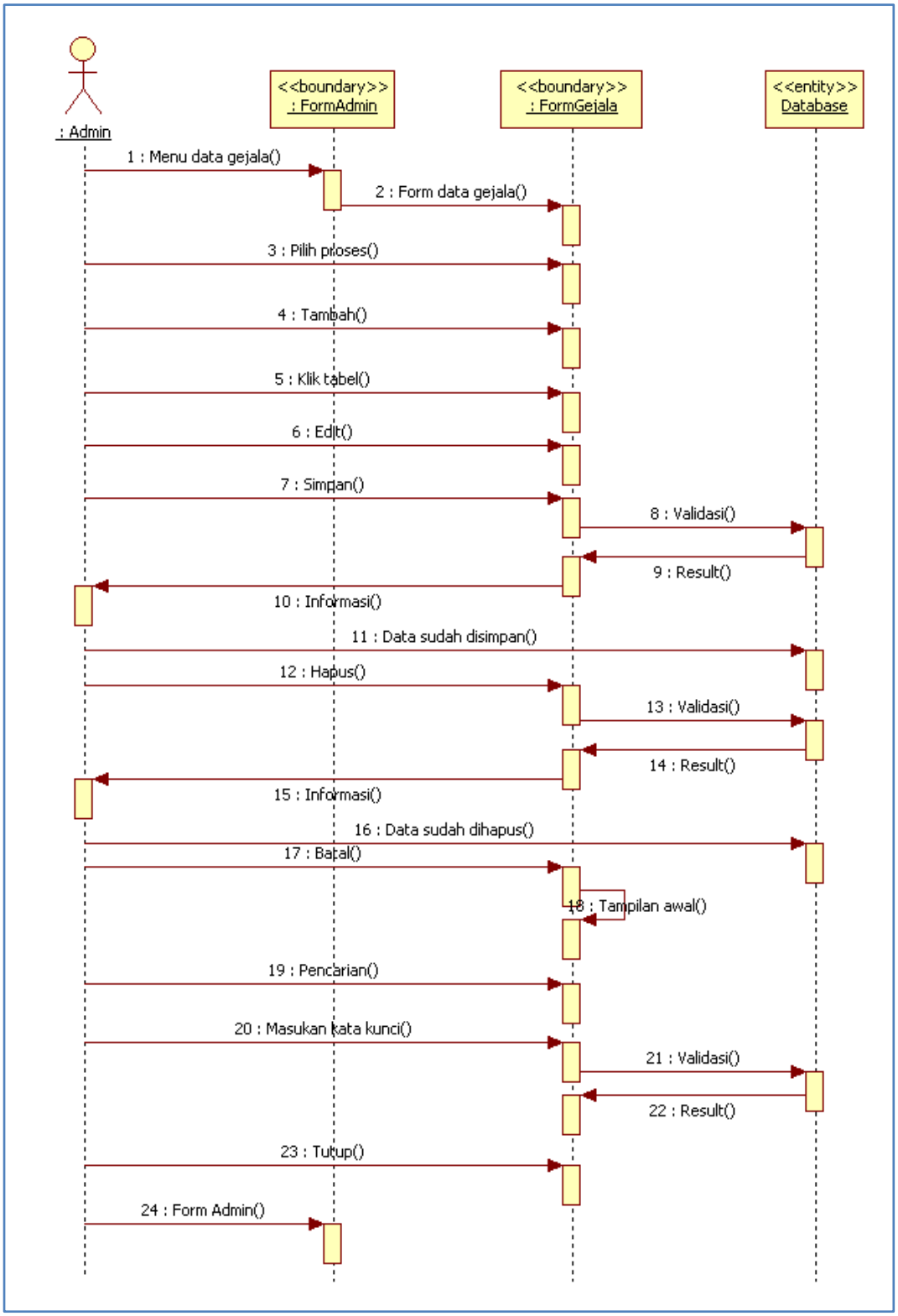
Gambar III.4. *Sequence Diagram* Deteksi Virus

3. *Sequence Diagram* Data Virus



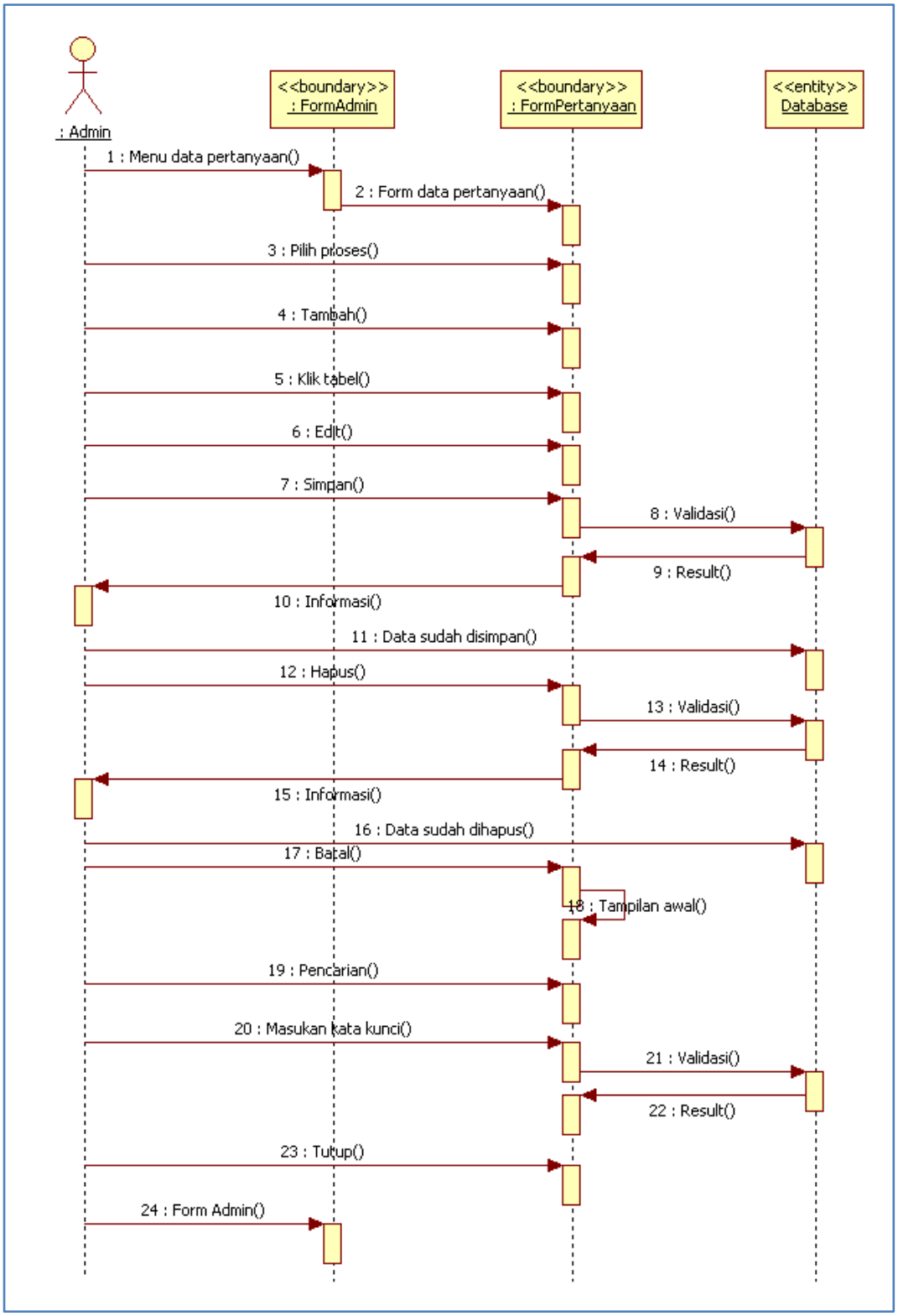
Gambar III.5. *Sequence Diagram* Data Virus

4. *Sequence Diagram* Data Gejala



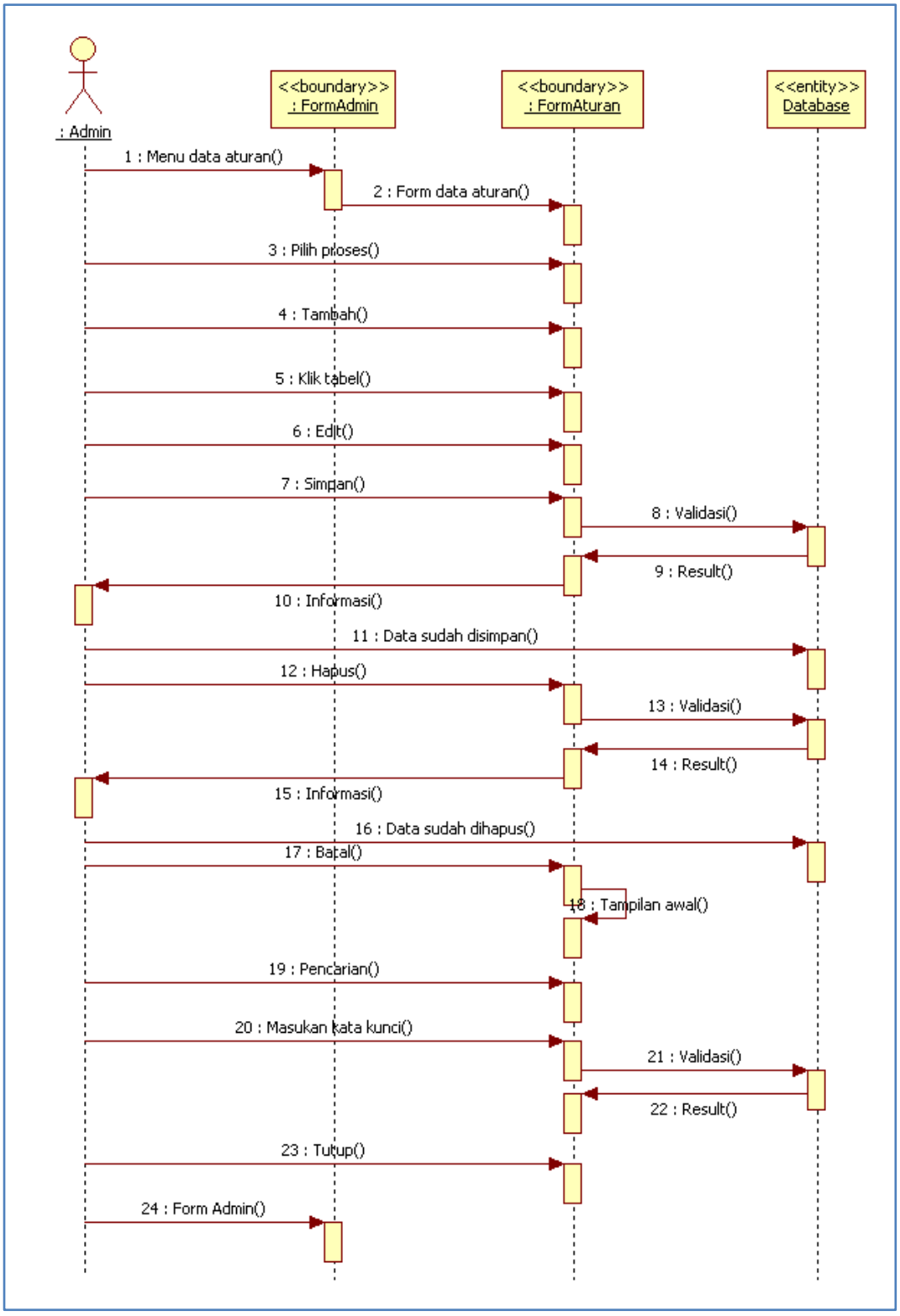
Gambar III.6. *Sequence Diagram* Data Gejala

5. *Sequence Diagram* Data Pertanyaan



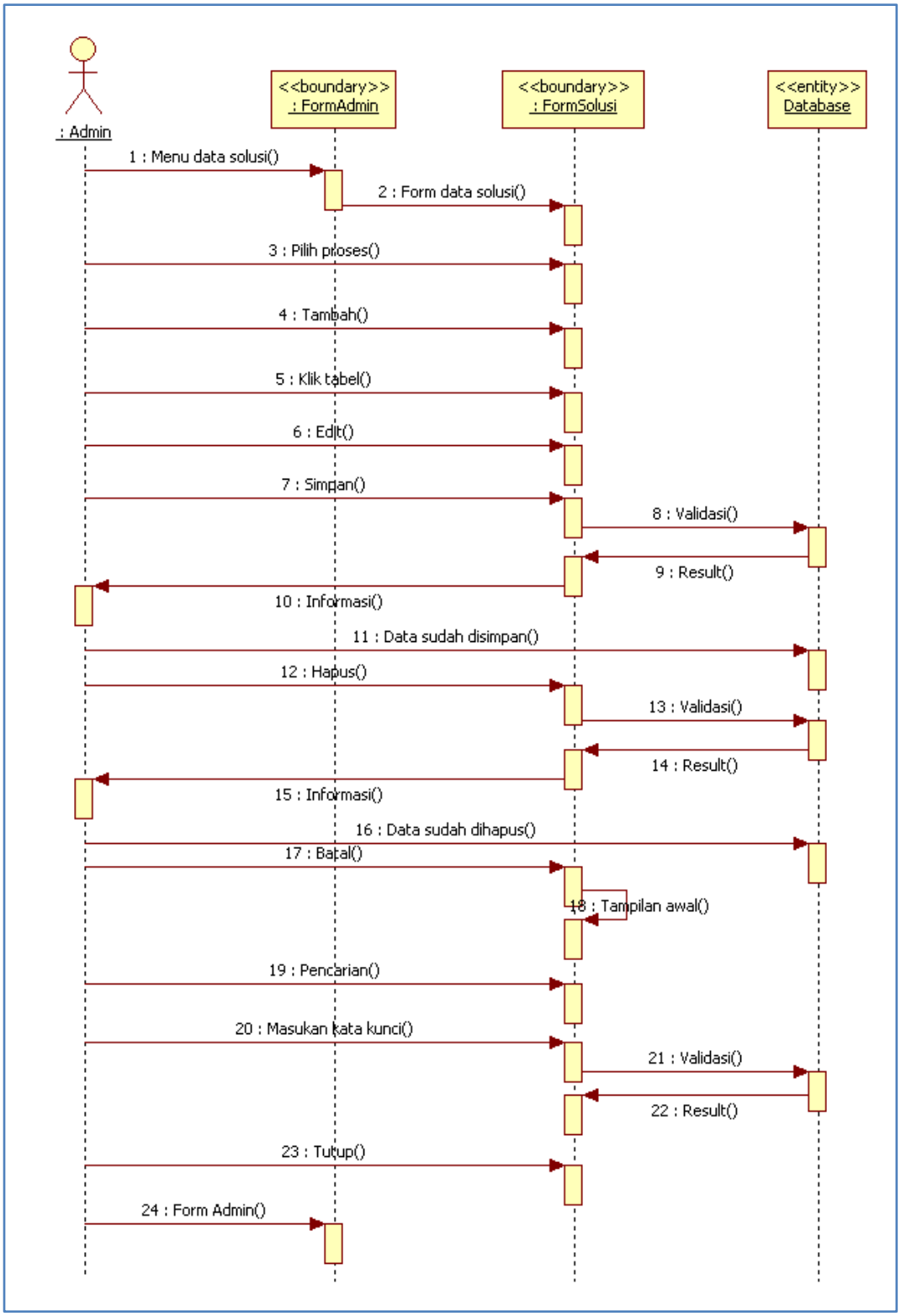
Gambar III.7. *Sequence Diagram* Data Pertanyaan

6. *Sequence Diagram* Data Aturan



Gambar III.8. *Sequence Diagram* Data Aturan

7. *Sequence Diagram* Data Solusi



Gambar III.9. *Sequence Diagram* Data Solusi

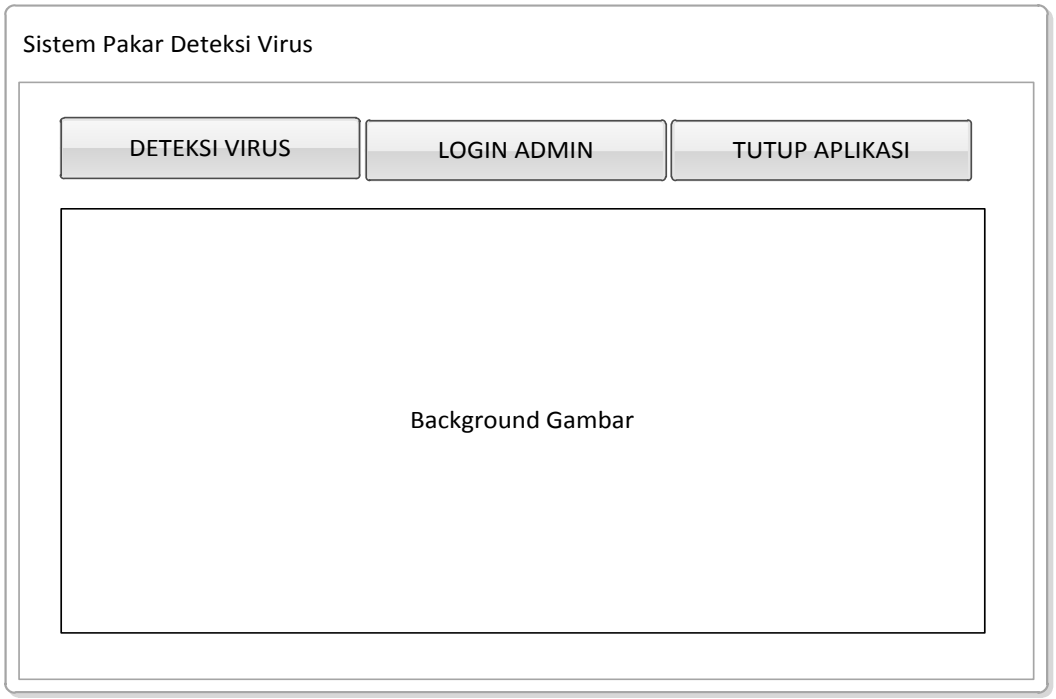
III.3.2. Desain Sistem Secara Detail

Sistem global sebagaimana telah dijelaskan di atas tidak dapat menggambarkan secara keseluruhan proses yang terjadi dalam sistem, sehingga dibutuhkan disain sistem secara detail yang dapat menjelaskan alur proses yang terjadi di dalam sistem tersebut. Adapun disain sistem secara detail yang diusulkan akan dijelaskan satu persatu berikut ini.

III.3.2.1. Desain *Output*

Bentuk desain *output* pada sistem yang diusulkan adalah dalam bentuk beberapa *form.Form-form* ini diberi nama sesuai kebutuhan dan fungsi dari *form* tersebut. Adapun bentuk desain *output* pada sistem yang diusulkan adalah sebagai berikut :

1. Desain *OutputForm* Utama



Gambar III.10. Desain *Output Form* Utama

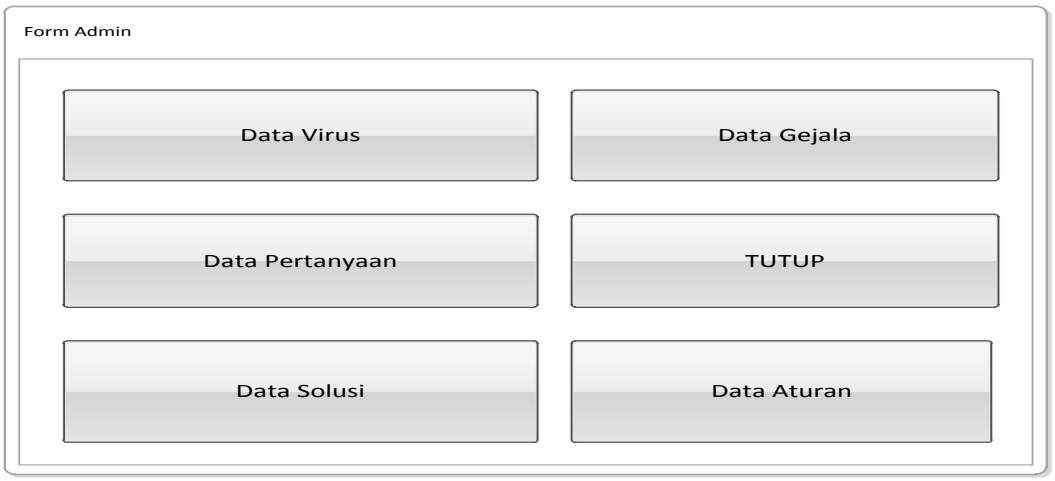
2. Desain *OutputForm* Deteksi Virus

Gambar III.11. Desain *Output Form* Deteksi Virus

3. Desain *OutputForm* Hasil

Gambar III.12. Desain *Output Form* Hasil

4. Desain *OutputFormAdmin*

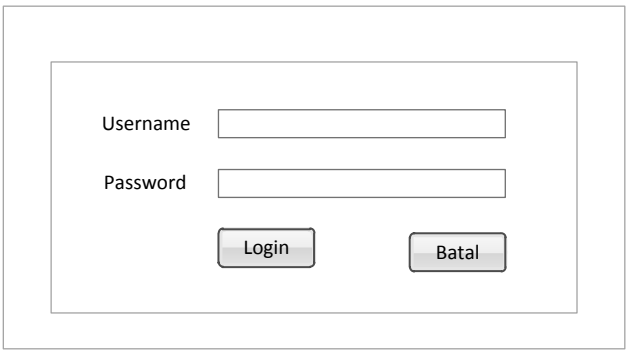


Gambar III.13. Desain *Output FormAdmin*

III.3.2.2. Desain *Input*

Menyesuaikan dengan kebutuhan dari sistem yang diusulkanl, maka diperlukan suatu *form* yang berfungsi untuk menerima masukan data dari pengguna.Mengingat sistem yang diusulkan hanya menyediakan dua tipe pengguna yang berbeda hak akses, maka dalam hal ini yang bertindak sebagai penginput data adalah *Admin*. Berikut ini adalah desain *input* pada sistem yang diusulkan :

1. Desain *Input Form Login*



Gambar III.14. Desain *Input Form Login*

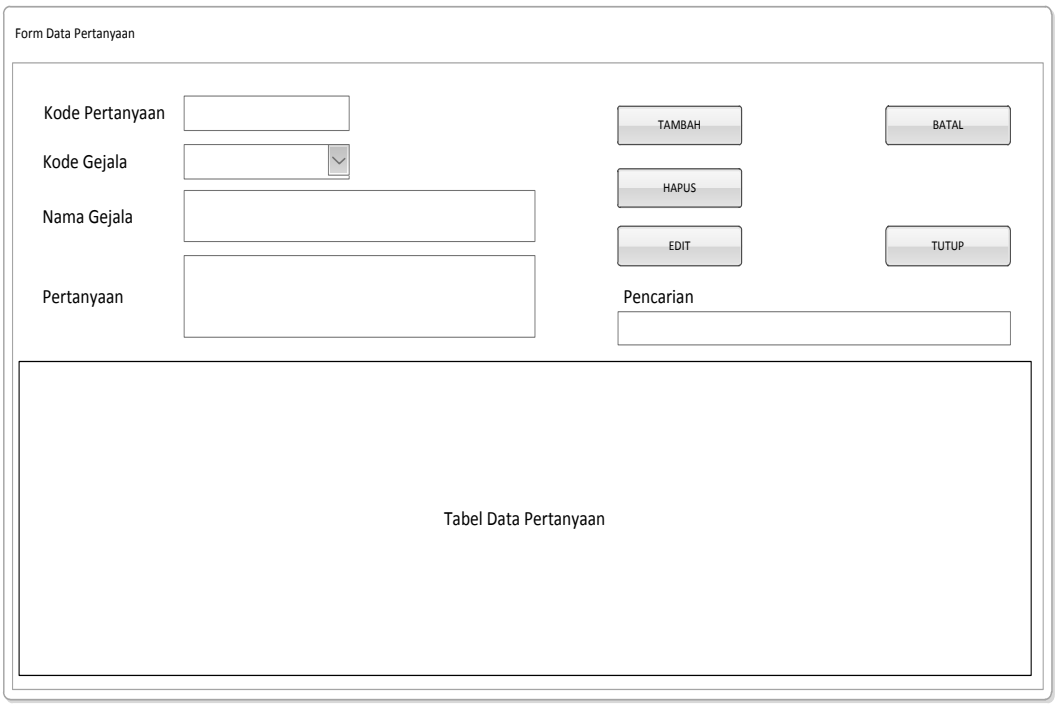
2. Desain *Input Form* Data Virus

Gambar III.15. Desain *Input Form* Data Virus

3. Desain *Input Form* Data Gejala

Gambar III.16. Desain *Input Form* Data Gejala

4. Desain *Input Form* Data Pertanyaan



Gambar III.17. Desain *Input Form* Data Pertanyaan

5. Desain *Input Form* Data Aturan

Gambar III.18. Desain *Input Form* Data Aturan

6. Desain *Input Form* Data Solusi

Gambar III.19. Desain *Input Form* Data Solusi

III.3.2.3. Desain *Database*

Database atau basis data merupakan elemen terpenting dalam perancanan sebuah aplikasi, karena baik buruknya aplikasi yang akan dibangun sangat bergantung pada baik buruknya proses perancangan *database* yang telah dilakukan. Dalam mendesain *database* pada sistem yang diusulkan, penulis membahas mengenai kamus data, normalisasi, desain tabel dan *entity relationship diagram* (ERD) dengan nama*database* db_sipedevi

III.3.2.3.1. Kamus Data

Kamus data merupakan deskripsi formal mengenai seluruh elemen yang tercakup dalam suatu *database*. Pada tahapan perancangan elemen - elemen pada kamus data akan menjadi bahan untuk menyusun basis data. Berikut ini adalah kamus data pada sistem yang diusulkan.

Virus = Kode Virus+, Nama Virus+, Nama Ilmiah+, {Jenis Virus}+, Ukuran Virus+

Gejala = Kode Gejala+, {Kode Virus}+, Nama Gejala+

Pertanyaan = Kode Pertanyaan+, Kode Gejala+, Pertanyaan+

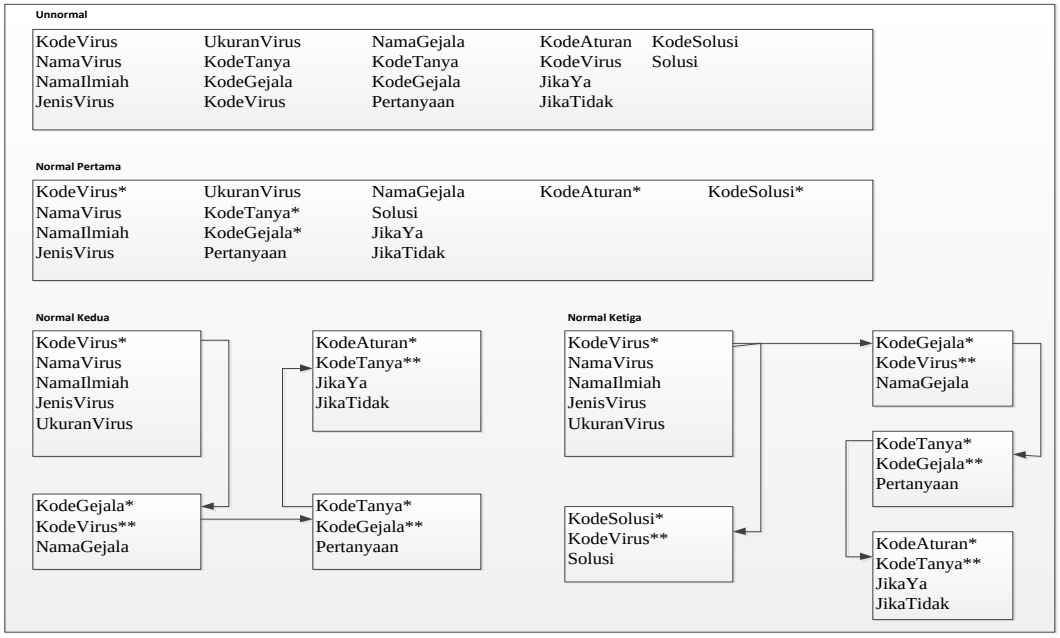
Aturan = Kode Aturan+, {Kode Pertanyaan}+, {Jika Ya}+, {Jika Tidak}+

Solusi = Kode Solusi+, {Kode Virus}+, Solusi+

Admin = *User Admin+, Password Admin+*

III.3.2.3.2. Normalisasi

Mengingat salah salah satu kelebihan pada sistem yang diusulkan adalah tidak adanya redudansi terhadap data yang digunakan, maka pada sistem yang diusulkan menggunakan teknik normalisasi untuk menormalkan data-data yang digunakan pada sistem. Adapun normalisasi pada sistem yang diusulkan, yaitu :



Gambar III.20. Normalisasi Sistem Pakar Deteksi Virus

III.3.2.3.3. Desain Tabel

Tabel merupakan komponen utama pendukung *database*. Tabel juga merupakan pertemuan antara baris dan kolom yang memuat suatu data atribut. Tabel merupakan sumber data bagi setiap aplikasi *database* seperti aplikasi yang akan dirancang dalam penelitian ini. Tabel juga sangat membantu penulis dalam pengelompokan permaslahan yang diangkat pada sistem yang diusulkan. Adapun tabel-tabel data yang dirancang untuk digunakan dalam sistem yang diusulkan adalah sebagai berikut :

1. Tabel Data *Admin*

Tabel ini berfungsi untuk menampung data pengguna, dalam hal ini data *admin* pada sistem yang diusulkan.

Nama *Database* : db_sipedevi

Nama Tabel : tbl_admin

Primary Key : UserAdmin

Foreign Key : -

Tabel III.1. Struktur Tabel *Admin*

Nama Kolom	Tipe Data	Ukuran	Keterangan
UserAdmin	*Varchar*	15	*Username*
PassAdmin	*Varchar*	15	*Password*

2. Tabel Data Virus

Tabel ini berfungsi untuk menampung data-data virus pada sistem yang diusulkan.

Nama *Database* : db_sipedevi

Nama Tabel : tbl_virus

Primary Key : KodeVirus

Foreign Key : -

Tabel III.2. Struktur Tabel Virus

Nama Kolom	Tipe Data	Ukuran	Keterangan
KodeVirus	*Varchar*	10	Kode virus
NamaVirus	*Varchar*	150	Nama virus
NamaIlmiah	*Varchar*	150	Nama ilmiah virus
JenisVirus	*Varchar*	20	Jenis virus

3. Tabel Data Gejala

Tabel ini berfungsi untuk menampung data-data gejala virus pada sistem yang diusulkan.

Nama *Database* : db_sipedevi

Nama Tabel : tbl_gejala

Primary Key : KodeGejala

Foreign Key : KodeVirus

Tabel III.3. Struktur Tabel Gejala

Nama Kolom	Tipe Data	Ukuran	Keterangan
KodeGejala	*Varchar*	10	Kode gejala
KodeVirus	*Varchar*	10	Kode virus
NamaGejala	*Text*	-	Nama gejala

4. Tabel Data Pertanyaan

Tabel ini berfungsi untuk menampung data-data pertanyaan pada sistem yang diusulkan.

Nama *Database* : db_sipedevi

Nama Tabel : tbl_pertanyaan

Primary Key : KodeTanya

Foreign Key : KodeGejala

Tabel III.4. Struktur Tabel Pertanyaan

Nama Kolom	Tipe Data	Ukuran	Keterangan
KodeTanya	*Varchar*	10	Kode pertanyaan
KodeGejala	*Varchar*	10	Kode gejala
Pertanyaan	*Text*	-	Pertanyaan

5. Tabel Data Aturan

Tabel ini berfungsi untuk menampung data-data aturan pada sistem yang diusulkan.

Nama *Database* : db_sipedevi

Nama Tabel : tbl_aturan

Primary Key : KodeAturan

Foreign Key : KodeTanya

Tabel III.5. Struktur Tabel Aturan

Nama Kolom	Tipe Data	Ukuran	Keterangan
KodeAturan	*Varchar*	10	Kode aturan
KodeTanya	*Varchar*	10	Kode pertanyaan
JikaYa	*Varchar*	10	Jika iya
JikaTidak	*Varchar*	10	Jika tidak

6. Tabel Data Solusi

Tabel ini berfungsi untuk menampung data-data solusi dari permasalahan pada sistem yang diusulkan.

Nama *Database* : db_sipedevi

Nama Tabel : tbl_solusi

Primary Key : KodeSolusi

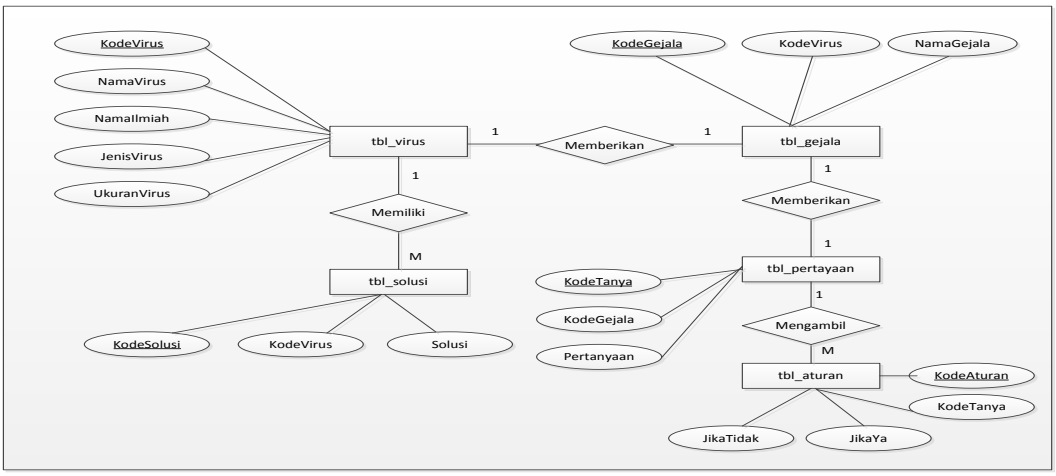
Foreign Key : KodeVirus

Tabel III.6. Struktur Tabel Solusi

Nama Kolom	Tipe Data	Ukuran	Keterangan
KodeSolusi	*Varchar*	10	Kode solusi
KodeVirus	*Varchar*	10	Kode virus
Solusi	*Text*	-	Solusi

III.3.2.3.4. *Entity Relationship Diagram*

Entity Relationship Diagram memberikan gambaran dari keterkaitan dan kesinambungan data pada setiap tabel dalam suatu *database*. Adapun *entity relationship diagram* yang digunakan pada sistem yang diusulkan adalah sebagai berikut :

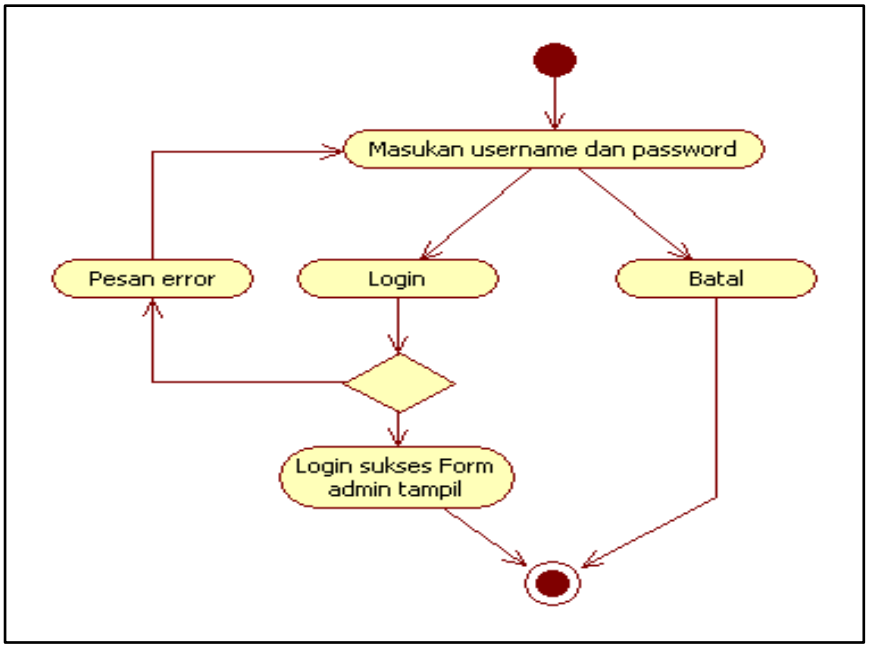


Gambar III.21. *Entity Relationship Diagram* Sistem Pakar Deteksi Virus

III.3.2.4. Logika Program

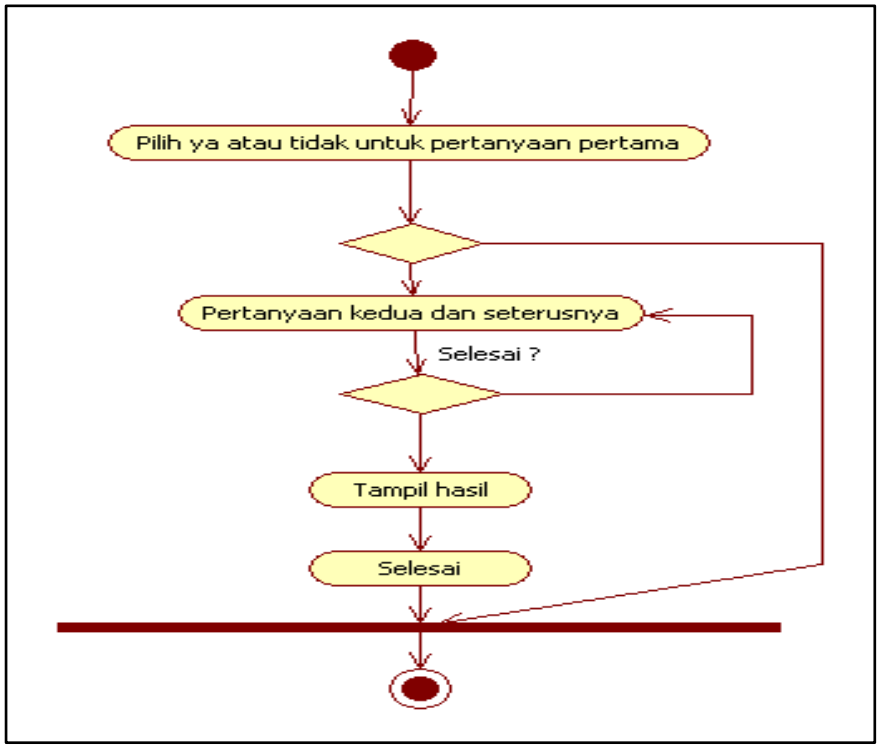
*Activity diagram*menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.Adapun *activity diagram* tersebut adalah sebagai berikut :

1. *Activity Diagram Login*



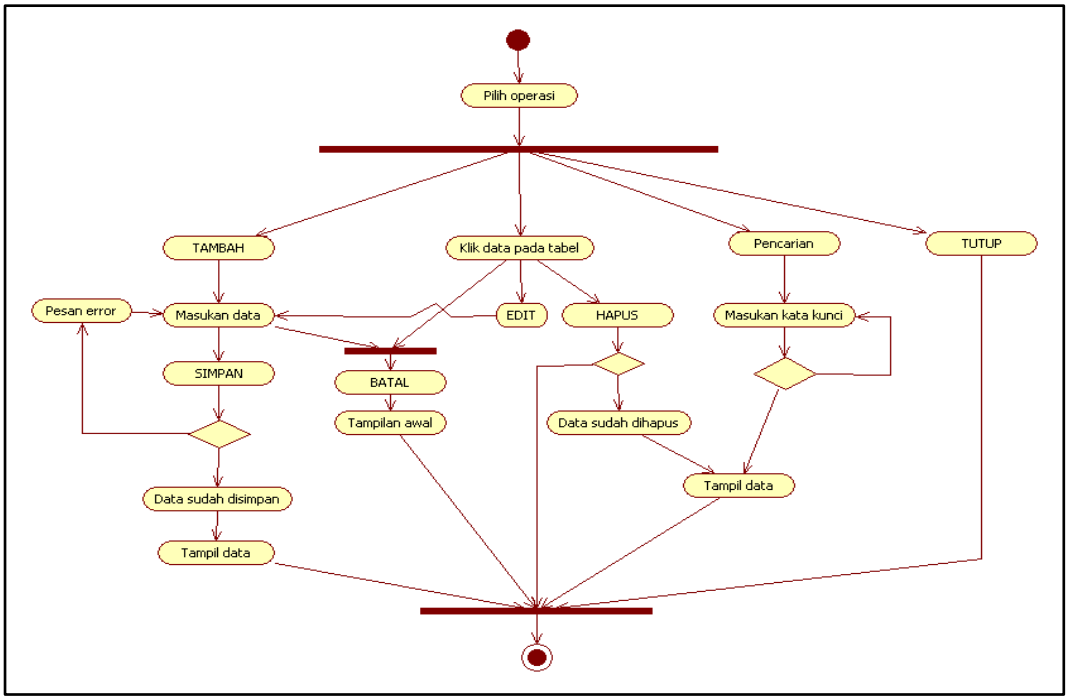
Gambar III.22. *Activity Diagram Login*

2. *Activity Diagram* Deteksi Virus



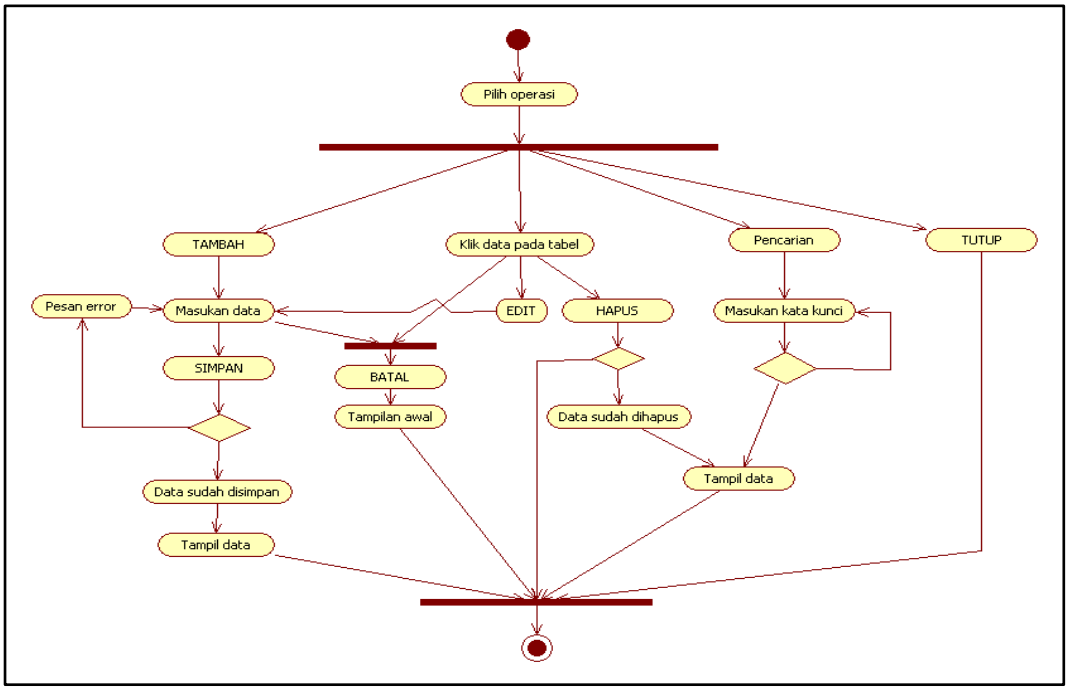
Gambar III.23. *Activity Diagram* Deteksi Virus

3. *Activity Diagram* Data Virus



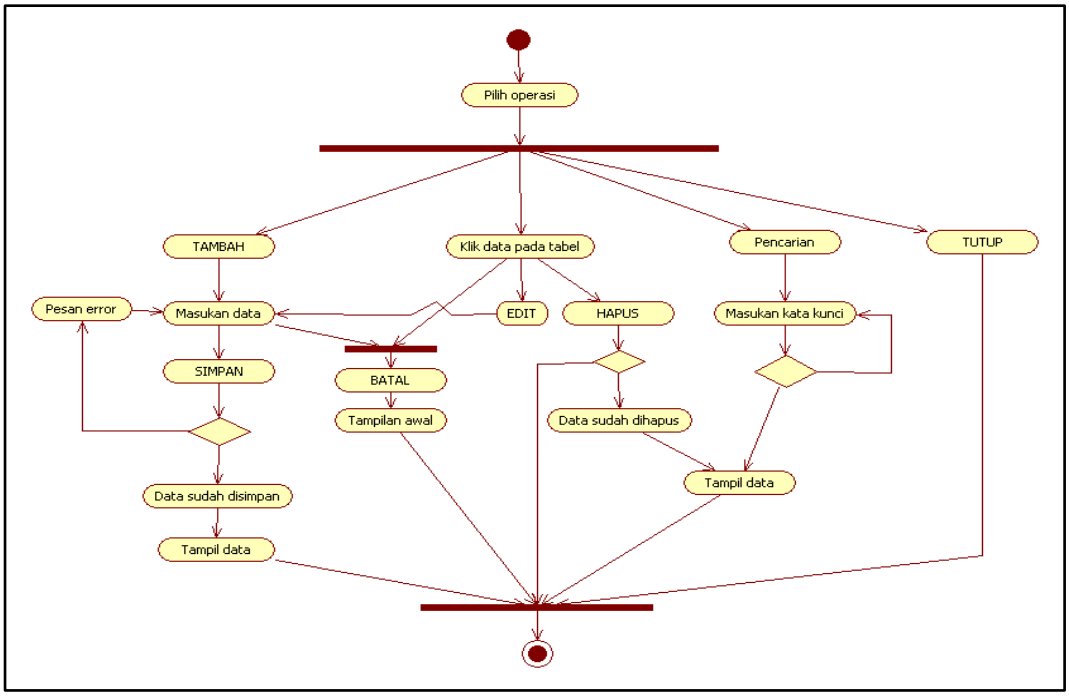
Gambar III.24. *Activity Diagram* Data Virus

4. *Activity Diagram* Data Gejala



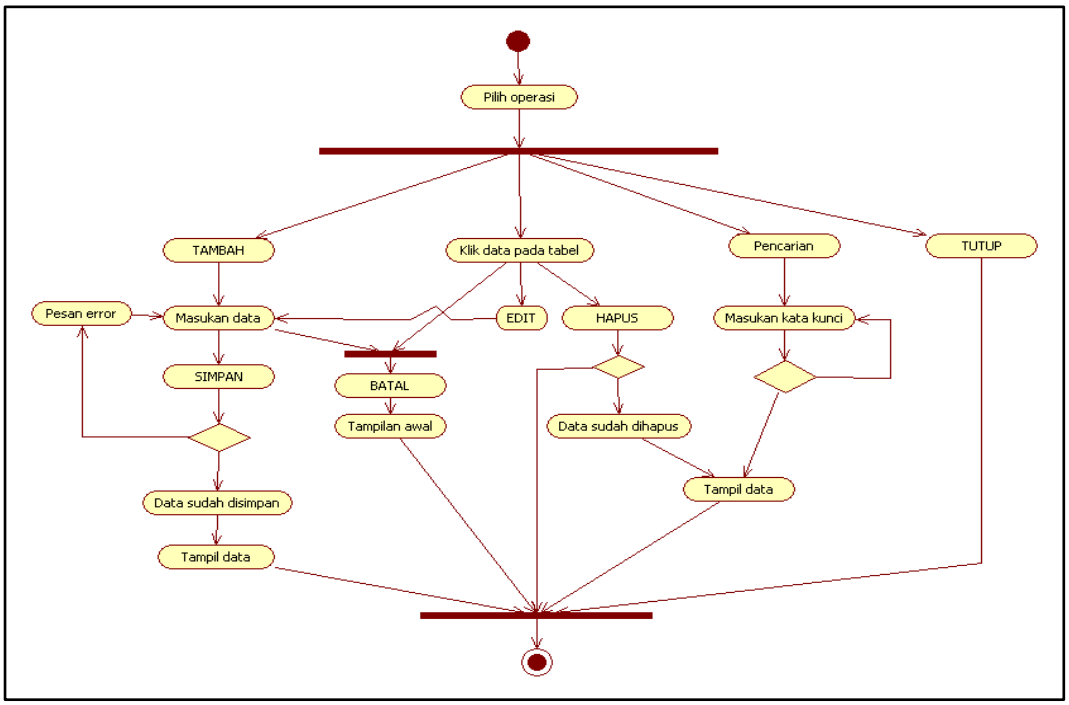
Gambar III.25. *Activity Diagram* Data Gejala

5. *Activity Diagram* Data Pertanyaan



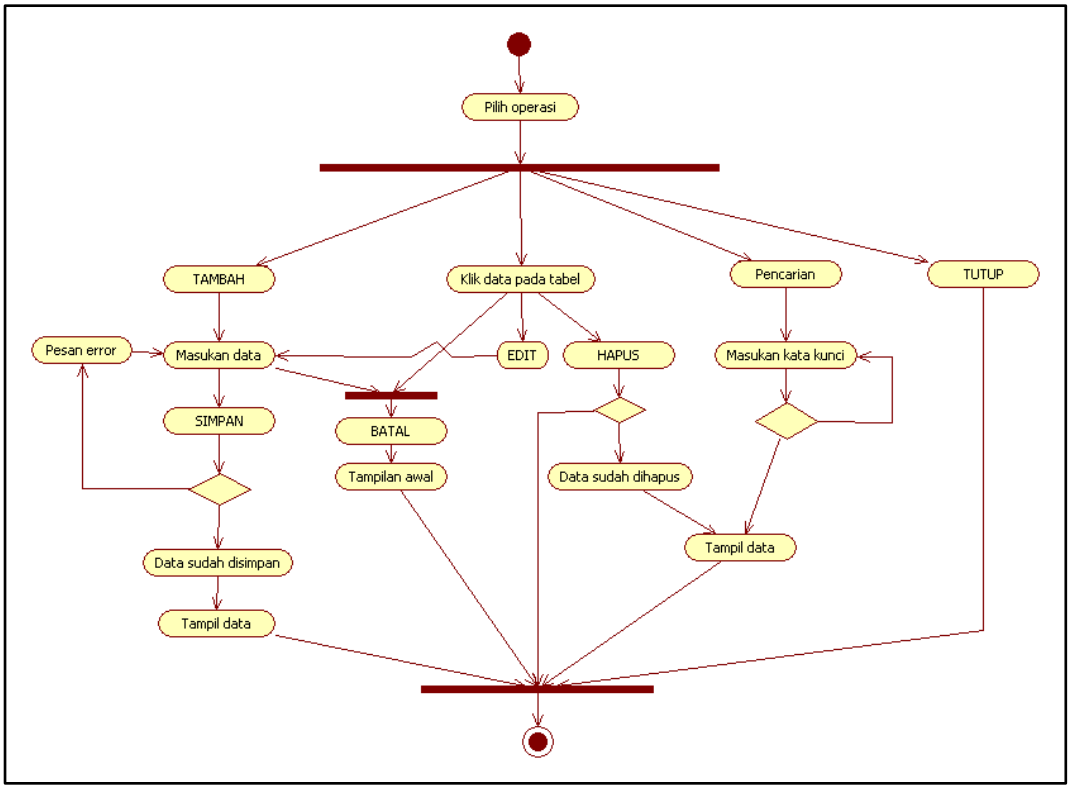
Gambar III.26. *Activity Diagram* Data Pertanyaan

6. *Activity Diagram* Data Aturan



Gambar III.27. *Activity Diagram* Data Aturan

7. *Activity Diagram* Data Solusi



Gambar III.28. *Activity Diagram* Data Solusi