

BAB III

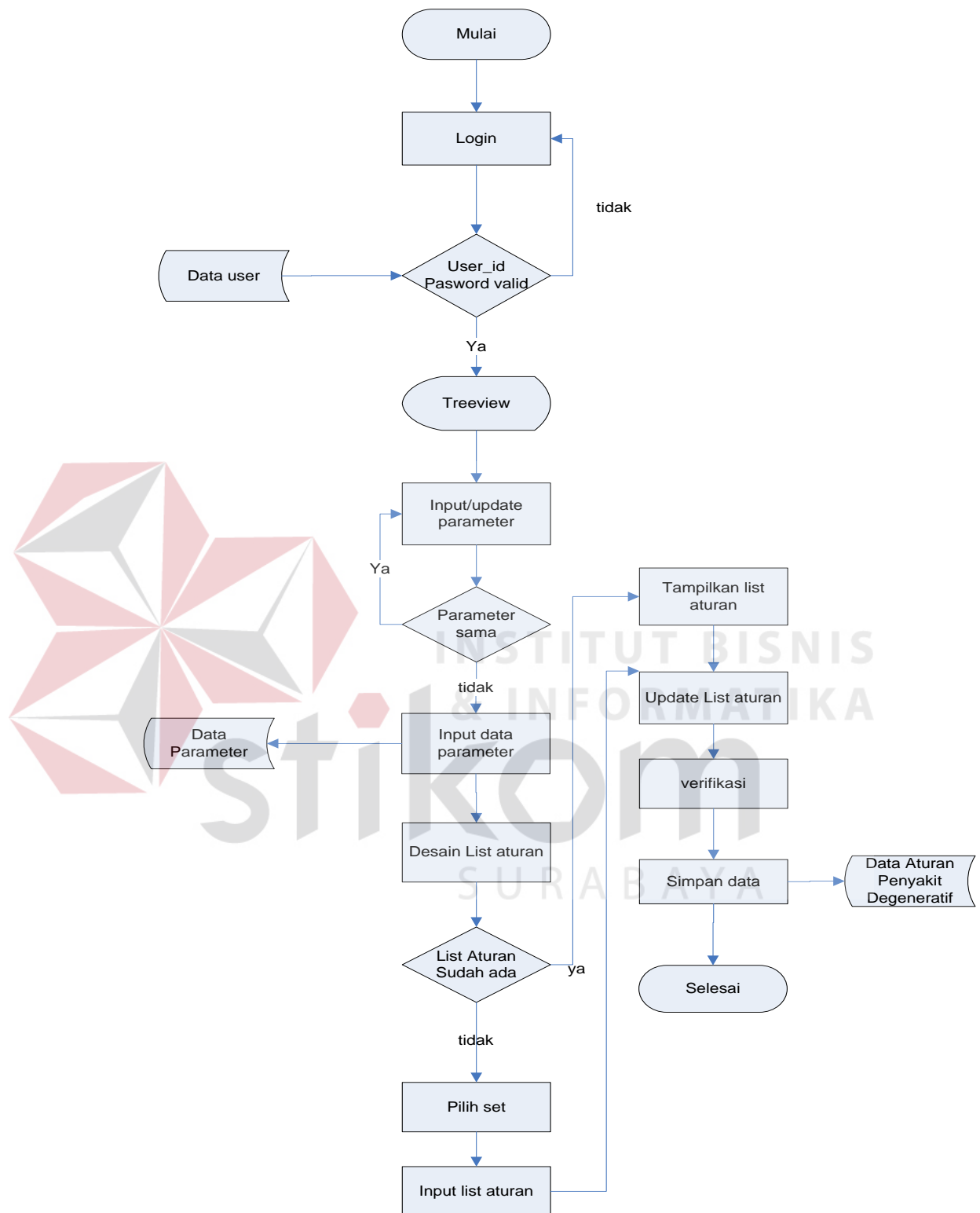
PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang langkah-langkah dalam perancangan sistem. Hal-hal yang akan dibahas di bab ini, antara lain perancangan *system flow* yang menunjukkan alur jalan dari sistem, desain arsitektur yang menunjukkan hubungan antar elemen, dan perancangan sistem. Perancangan sistem ini terdiri dari perancangan diagram blok, perancangan *dependency diagram*, perancangan *decision table*, perancangan reduksi tabel, proses verifikasi, dan perancangan *rule base*. Dalam bab ini juga dilengkapi dengan struktur tabel dan desain *input output*.

3.1 Perancangan System Flow

System flow merupakan gambaran suatu aliran data proses dan hubungan antara proses satu dengan yang lain dalam suatu sistem komputer dengan menggunakan simbol-simbol tertentu. Oleh karena itu, penganalisa dapat menginformasikan jalannya suatu sistem dan dapat memahami sistematika sistem dengan mudah.

System flow untuk admin memiliki fungsi untuk melakukan proses memasukkan data *user* serta memasukkan data parameter yang dibutuhkan dalam bentuk *treeview* untuk membuat list aturan. Selain itu, admin juga dapat merubah data parameter yang telah ada. *System flow* untuk admin dapat dilihat pada gambar 3.1.

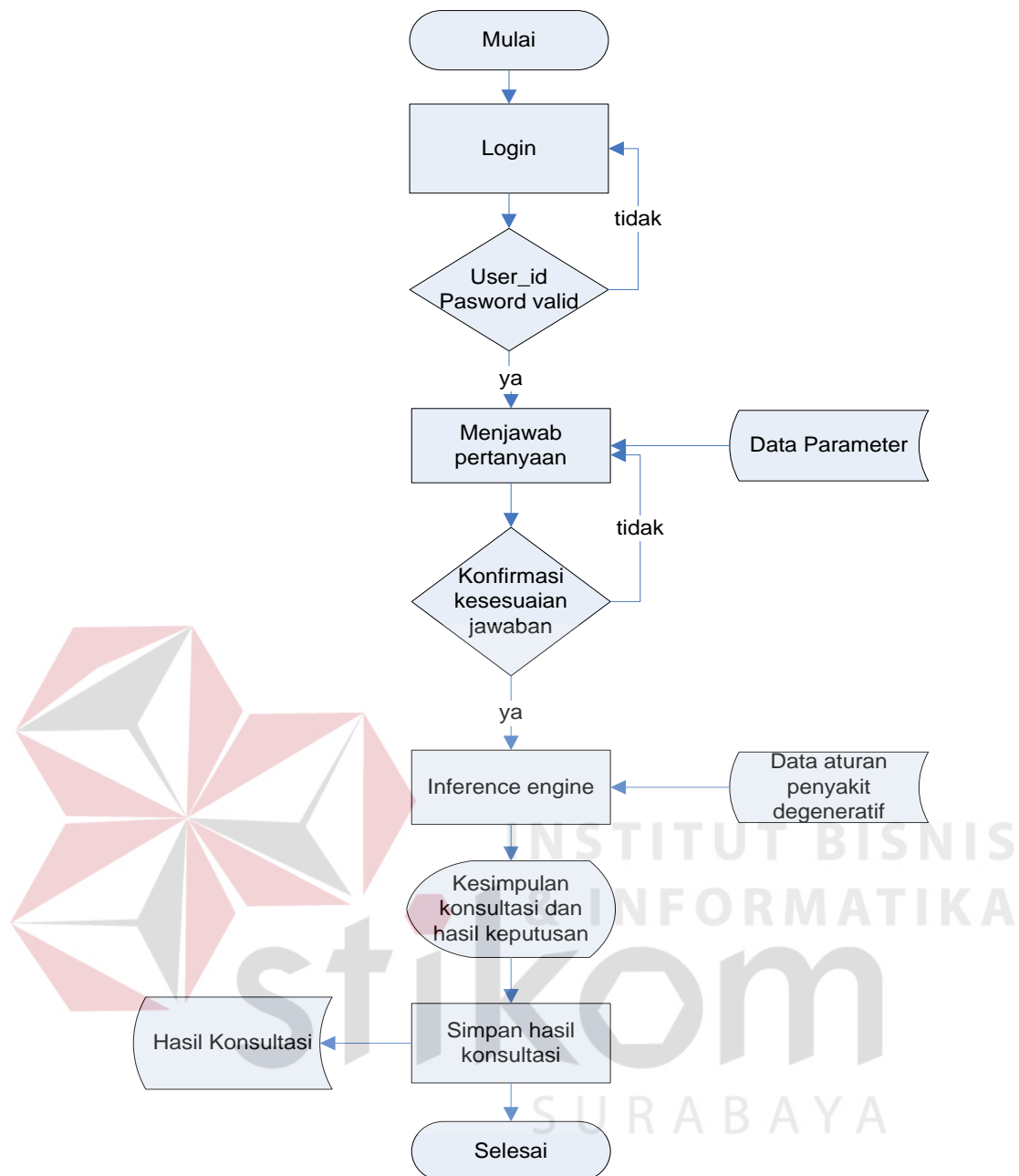


Gambar 3.1 *System flow* untuk admin

Alur kerja *system flow* untuk admin dimulai dari proses *login* kemudian dilakukan validasi *user_id* dan *password*. Setelah validasi valid, masuk ke menu *treeview* untuk melakukan penambahan dan perubahan parameter. Setelah parameter sudah dimasukkan, dilakukan proses desain list aturan, yang kemudian dilakukan proses perubahan list aturan. List aturan yang sudah valid kemudian akan melalui proses verifikasi dan hasil verifikasi akan disimpan pada tabel data aturan penyakit degeneratif.

System flow untuk *user* umum menjelaskan proses jalannya sistem pada saat *user* melakukan konsultasi. *System flow* untuk user umum dapat dilihat pada gambar 3.2.



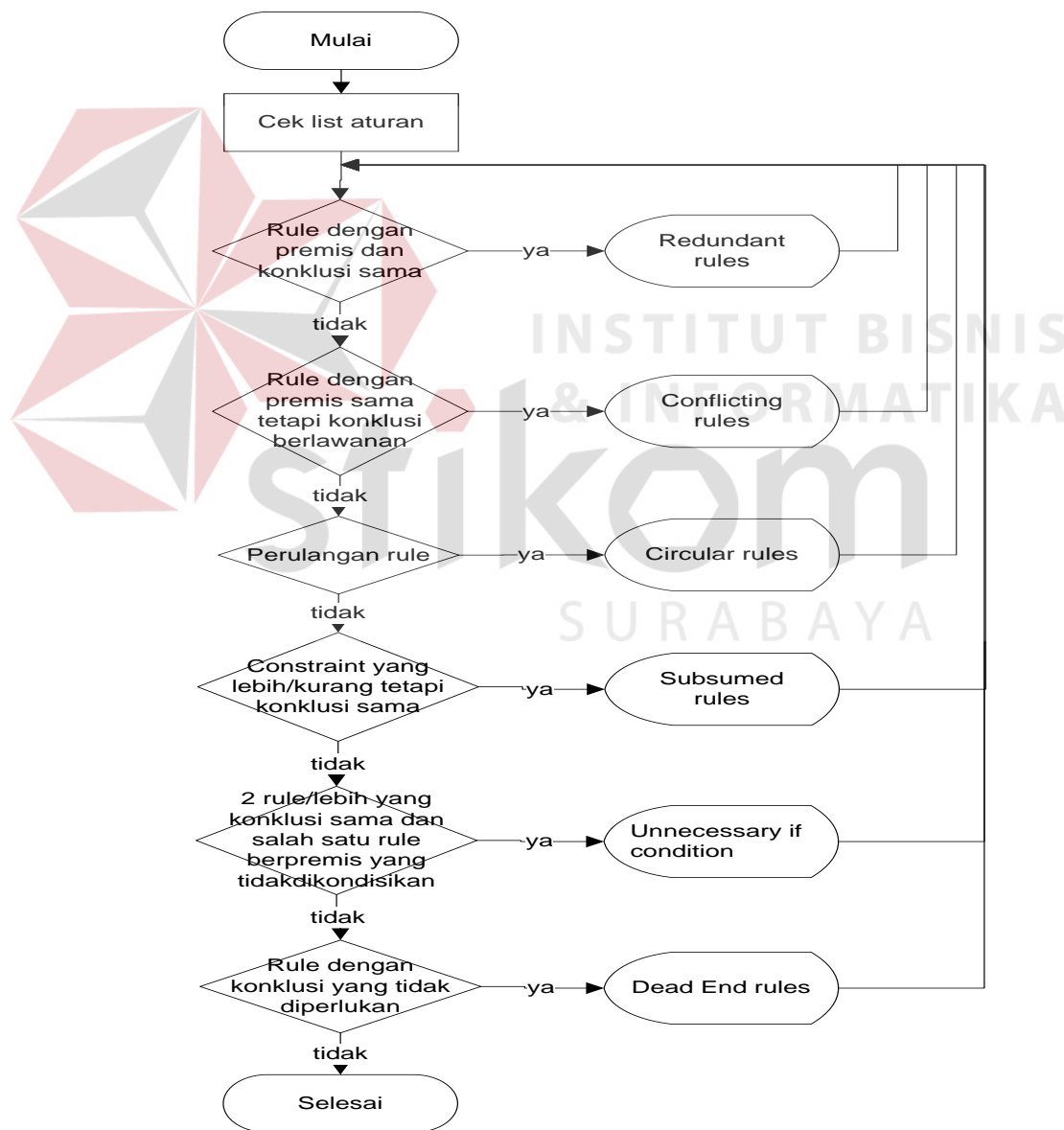


Gambar 3.2 *System flow* untuk *user* umum

Alur kerja *system flow* untuk *user* umum dimulai dari proses *login* kemudian dilakukan validasi *user_id* dan *password*. Setelah validasi valid, *user* umum menjawab pertanyaan yang diajukan dari sistem, yang setelah semua pertanyaan dijawab akan diminta melakukan konfirmasi jawaban. Kemudian masuk ke dalam proses *inference engine*. setelah proses *inference engine*

mendapatkan hasil, sistem akan menampilkan menghasilkan hasil yang berupa kesimpulan konsultasi dan hasil keputusan. Hasil konsultasi yang ditampilkan akan disimpan pada tabel hasil konsultasi.

System flow proses verifikasi menjelaskan proses jalannya verifikasi pada sistem. Proses verifikasi diperlukan untuk melakukan reduksi pada tabel keputusan agar tabel keputusan bebas dari kesalahan. *System flow* proses verifikasi dapat dilihat pada gambar 3.3.



Gambar 3.3 *System flow* proses verifikasi

System flow proses verifikasi untuk proses *redundant rules*, *conflicting rules*, *circular rules*, *subsumed rules*, *unnecessary if condition*, dan *dead end rules*. Proses diawali dengan pengecekan *list* aturan.

Pengecekan pertama adalah pengecekan untuk *redundant rules*, apakah pada suatu *rule* terdapat premis dan konklusi yang sama. Jika ada, maka akan diperiksa dan ditampilkan pada *display redundant rules* dan proses akan dilanjutkan untuk pengecekan selanjutnya.

Pengecekan kedua adalah pengecekan untuk *conflicting rules*, apakah pada suatu *rule* terdapat premis yang sama tetapi konklusi berlawanan. Jika ada, maka akan diperiksa dan ditampilkan pada *display conflicting rules* dan proses akan dilanjutkan untuk pengecekan selanjutnya.

Pengecekan ketiga adalah pengecekan untuk *circular rules*, apakah pada suatu *rule* terdapat premis yang merupakan konklusi dari *rule* lain. Jika ada, maka akan diperiksa dan ditampilkan pada *display circular rules* dan proses akan dilanjutkan untuk pengecekan selanjutnya.

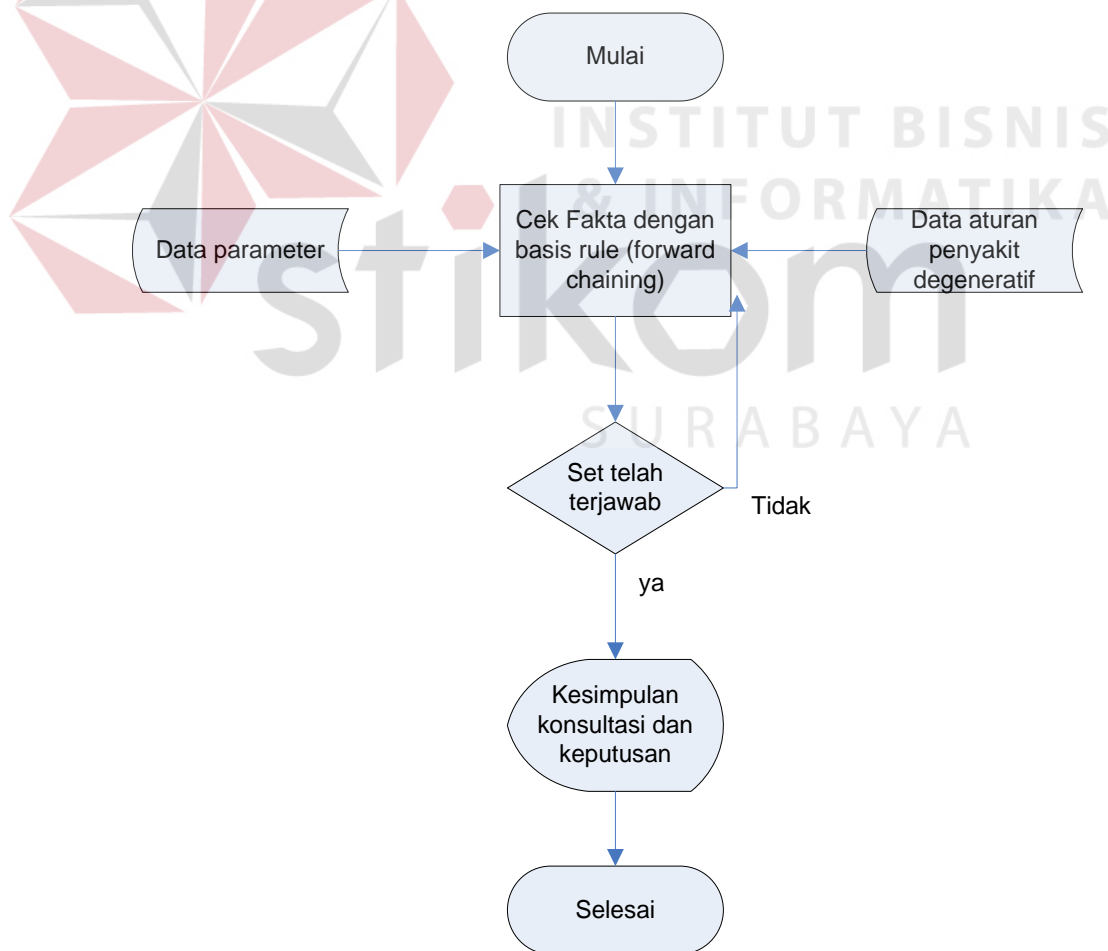
Pengecekan keempat adalah pengecekan untuk *subsumed rules*, apakah pada suatu *rule* terdapat *constraint* yang lebih atau kurang tetapi mempunyai konklusi yang sama. Jika ada, maka akan diperiksa dan ditampilkan pada *display subsumed rules* dan proses akan dilanjutkan untuk pengecekan selanjutnya.

Pengecekan kelima adalah pengecekan untuk *unnecessary if condition*, apakah pada 2 *rule* atau lebih terdapat konklusi yang sama tetapi salah satu dari *rule* tersebut mempunyai premis yang tidak perlu dikondisikan dalam *rule* karena tidak mempunyai pengaruh apapun. Jika ada, maka akan diperiksa dan

ditampilkan pada *display unnecessary if condition* dan proses akan dilanjutkan untuk pengecekan selanjutnya.

Pengecekan keenam adalah pengecekan untuk *dead end rules*, apakah pada suatu *rule* terdapat konklusi yang tidak diperlukan oleh *rule-rule* lainnya. Jika ada, maka akan diperiksa dan ditampilkan pada *display dead end rules* dan proses telah selesai.

System flow proses *inference engine* menjelaskan proses penelusuran untuk menentukan jawaban yang tepat. Pada sistem berbasis aturan ini akan digunakan pencarian arah maju (*forward chaining*). *System flow* proses *inference engine* dapat dilihat pada gambar 3.4.

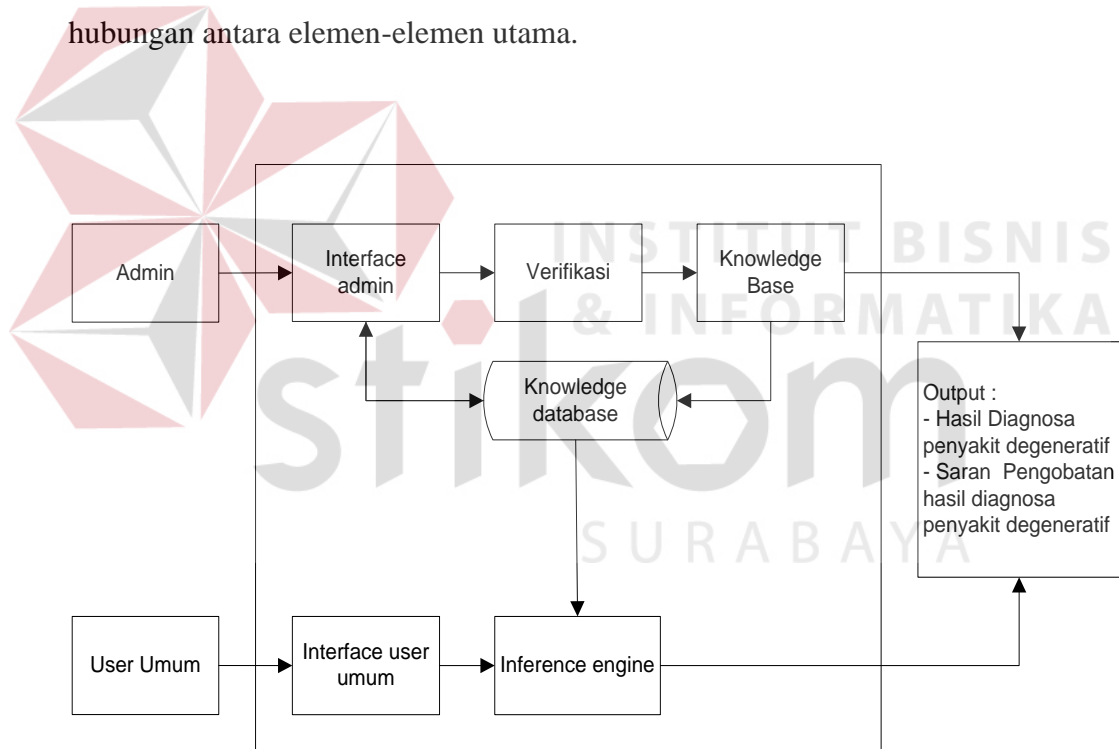


Gambar 3.4 *System flow* Proses *Inference Engine*

System flow proses *inference engine* akan menerima respon data yang diterima dari *user* umum berupa jawaban dari pertanyaan-pertanyaan sistem, kemudian melakukan proses terhadap basis pengetahuan yang dimiliki dengan masukan data dari tabel data aturan penyakit degeneratif dan tabel data parameter. Setelah proses *inference engine* menemukan jawaban yang sesuai, akan ditampilkan hasil yang berupa kesimpulan konsultasi dan hasil keputusan.

3.2 Desain Arsitektur

Desain arsitektur seperti terlihat pada gambar 3.5 menggambarkan hubungan antara elemen-elemen utama.



Gambar 3.5 Desain Arsitektur Menentukan Jenis Penyakit Degeneratif

Penjelasan dari desain arsitektur untuk menentukan jenis penyakit degeneratif adalah sebagai berikut:

1. *Interface admin*, yaitu suatu media bagi admin untuk menginputkan parameter *rule*. *Rule* yang diinputkan adalah gejala-gejala penyakit degeneratif untuk menghasilkan *knowledge base*.
2. Verifikasi, yaitu suatu proses validasi *rule* dengan menggunakan teori verifikasi anantara lain:
 - a. *Redundant rules* adalah suatu kondisi jika 2 *rule* atau lebih mempunyai *premise* dan *conclusion* yang sama.
 - b. *conflicting rules* adalah suatu kondisi ketika 2 *rule* atau lebih mempunyai *premise* yang sama tetapi *conclusion* yang berbeda.
 - c. *circular rules* adalah suatu kondisi dimana suatu *premise* dari salah satu *rule* merupakan *conclusion* dari *rule* yang lain, atau kebalikannya.
 - d. *subsumed rules* adalah suatu kondisi jika *rule* tersebut mempunyai *constraint* yang lebih atau kurang tetapi mempunyai *conclusion* yang sama.
 - e. *unnecessary if condition* adalah suatu kondisi ketika 2 *rule* atau lebih mempunyai *conclusion* yang sama tetapi salah satu dari *rule* tersebut mempunyai *premise* yang tidak perlu dikondisikan dalam *rule* karena tidak mempunyai pengaruh apapun.
 - f. *dead end rules* adalah suatu kondisi suatu *rule* yang *conclusion*nya tidak diperlukan oleh *rule-rule* lainnya.
3. *Knowledge database*, yaitu suatu media penyimpanan data yang digunakan untuk admin yang berfungsi menyimpan data basis pengetahuan, menyimpan hasil pengembangan basis pengetahuan, dan maintenance data admin dan

untuk user umum yang berfungsi untuk menyimpan data-data yang berkaitan dengan user umum.

4. *Knowledge base*, yaitu kumpulan dari fakta dan aturan tentang permasalahan-permasalahan yang telah didesain oleh admin yang diperoleh sistem selama proses berlangsung.
5. *Interface user umum*, yaitu suatu media bagi user umum untuk berinteraksi dengan sistem dengan menginputkan fakta-fakta untuk mendapatkan suatu kesimpulan.
6. *Inference engine*, yaitu mekanisme inferensi yang digunakan untuk mensimulasikan strategi penyelesaian masalah. Mekanisme yang digunakan adalah *forward chaining*, yaitu penelusuran dari data-data yang ada untuk mencapai suatu kesimpulan.
7. *Output*, yaitu hasil yang didapatkan dari sistem yang bisa menunjukkan jawaban dari fakta-fakta yang telah diinputkan. *Output* yang dihasilkan adalah sebagai berikut:
 - a. Hasil diagnosa penyakit degeneratif
 - b. Saran pengobatan hasil diagnosa penyakit degeneratif

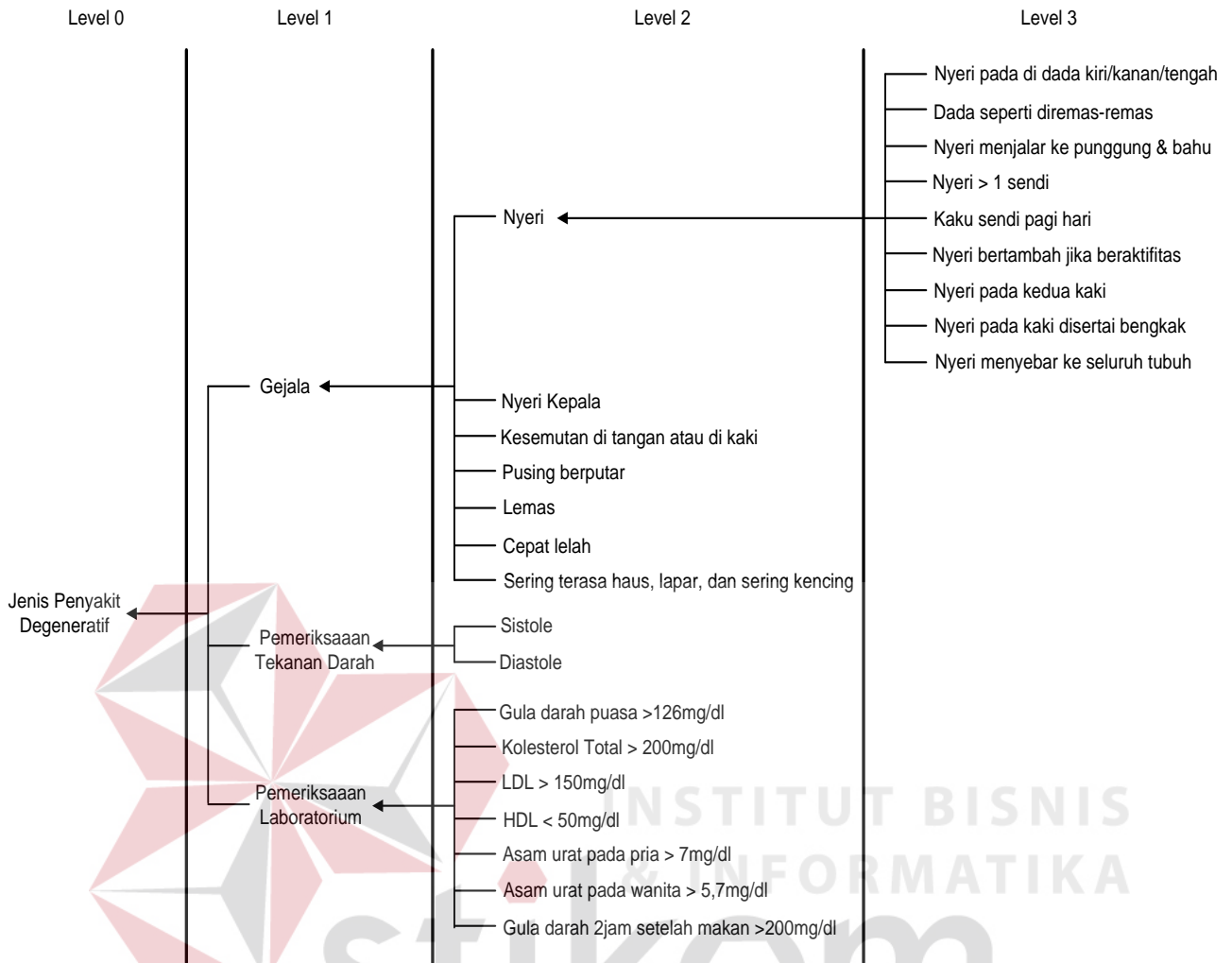
3.3 Perancangan Sistem

3.3.1 Perancangan Diagram Blok

Diagram blok diperlukan untuk mengetahui urutan-urutan kerja sistem dalam mencari suatu keputusan. Dalam sistem berbasis aturan ini terdapat satu buah diagram blok yaitu diagram blok menentukan jenis penyakit degeneratif. Diagram blok ini terdiri dari empat level yaitu level 0, level 1, level 2, dan level 3. Level 3 berisi tentang gejala-gejala nyeri. Level 2 berisi tentang gejala-gejala

penyakit degeneratif secara khusus, selain itu terdapat parameter nyeri. Parameter nyeri terdiri dari sub parameter nyeri dada, nyeri sendi, nyeri seluruh badan dan bukan nyeri. Level 1 terdapat tiga parameter yaitu gejala, pemeriksaan tekanan darah, dan pemeriksaan laboratorium. Parameter gejala terdiri dari sub parameter gejala jantung, gejala diabetes melitus, gejala hipertensi, gejala osteoarthritis, gejala dislipidemia, gejala hiperurisemia, dan bukan gejala. Parameter pemeriksaan tekanan darah terdiri dari sub parameter darah tinggi dan darah normal. Parameter pemeriksaan laboratorium terdiri dari sub parameter gula darah, profil lipid, asam urat, normal. Diagram blok untuk menentukan jenis penyakit degeneratif dapat dilihat pada gambar 3.6.

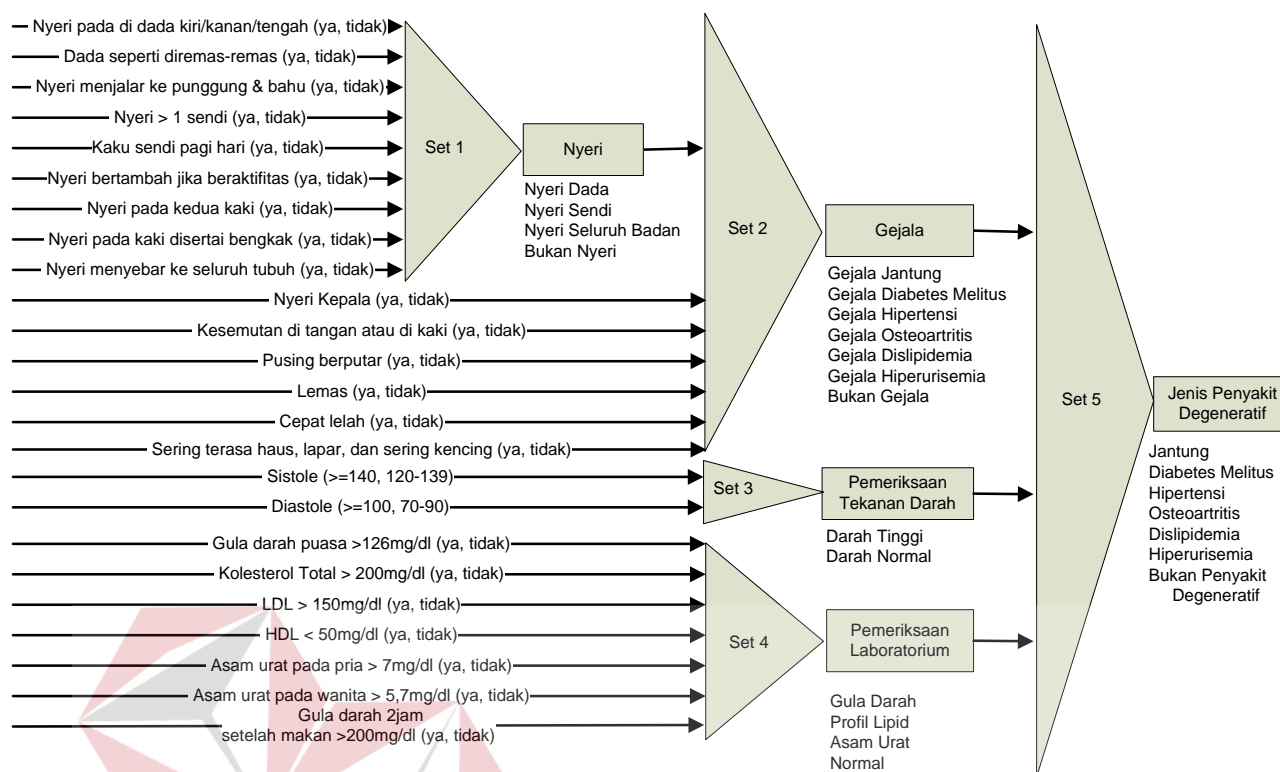




Gambar 3.6 Diagram Blok Menentukan Jenis Penyakit Degeneratif

3.3.2 Perancangan Dependency Diagram

Dependency diagram menunjukkan hubungan atau ketergantungan antara inputan jawaban, aturan-aturan (*rules*), nilai-nilai dan direkomendasikan untuk *knowledge base system*. *Dependency diagram* menentukan jenis penyakit degeneratif dapat dilihat pada gambar 3.7.



Gambar 3.7 *Dependency Diagram* Menentukan Jenis Penyakit Degeneratif

Pada *rule set 5* merupakan *rule set* yang memberikan konklusi akhir dari konklusi yang dihasilkan oleh *rule set 2*, *rule set 3*, dan *rule set 4*. Konklusi akhir terdiri dari jenis-jenis penyakit degeneratif. *Rule set 4* merupakan *rule set* yang memberikan konklusi sementara berupa pemeriksaan laboratorium yaitu gula darah, profil lipid, asam urat, dan normal. *Rule set 3* merupakan *rule set* yang memberikan konklusi sementara berupa pemeriksaan tekanan darah yaitu darah tinggi dan darah normal. *Rule set 2* merupakan *rule set* yang memberikan konklusi sementara berupa gejala. Gejala yang dihasilkan merupakan gejala penyakit jantung, diabetes melitus, hipertensi, osteoarthritis, dislipidemia, hiperurisemia, dan bukan gejala apapun. *Rule set 1* merupakan *rule set* yang memberikan konklusi sementara berupa nyeri dada, nyeri sendi, nyeri seluruh badan, dan bukan nyeri kepada *rule set 2*.

3.3.3 Perancangan Decision Table

Decision table dibuat untuk menunjukkan hubungan antara nilai-nilai pada hasil fase antara atau rekomendasi akhir *knowledge base system*. Pada tabel 3.2 menunjukkan salah satu contoh perancangan *decision table* untuk *rule set 3* yaitu pemeriksaan laboratorium. *Decision table* berikut dibuat berdasarkan perancangan *dependency diagram* menentukan jenis penyakit degeneratif.

Tabel 3.1 *Decision Table* Langkah 1

Langkah 1 : *Plan*

Kondisi	: Sistole (≥ 140 , 120 - 139)	= 2
	Diastole (≥ 100 , 70 - 90)	= 2
Baris	$2 \times 2 = 4$	

Tabel 3.2 *Decision Table* Langkah 2

Langkah 2 : *Complete Decision Table*

Rule	Sistole	Diastole	Kesimpulan
A1	≥ 140	≥ 100	Darah Tinggi
A2	≥ 140	70 - 90	Darah Tinggi
A3	120 - 139	≥ 100	Darah Tinggi
A4	121 - 139	70 - 90	Normal

Pada langkah 1 ditentukan jumlah baris maksimal yang akan terdapat pada *decision table* dengan cara mengkalikan jumlah jawaban tiap-tiap parameter pada *rule set 3*. Kemudian pada langkah 2 tiap-tiap jawaban parameter dikombinasikan sejumlah baris yang telah ditentukan pada langkah 1.

3.3.4 Perancangan Reduksi Tabel

Pada sistem ini proses perancangan reduksi tabel untuk setiap *decision table* dilakukan secara manual. Perancangan reduksi berdasarkan *decision table* pada tabel 3.4 menghasilkan parameter seperti pada tabel 3.5.

Tabel 3.3 *Reduced Decision Table* Langkah 1

Langkah 1 : *Plan*

Kondisi	: Sistole (≥ 140 , 120 - 139)	= 2
	Diastole (≥ 100 , 70 - 90)	= 2
Baris	$2 \times 2 = 4$	

Tabel 3.4 *Reduced Decision Table* Langkah 2

Langkah 2 : *Complete Decision Table*

Rule	Sistole	Diastole	Kesimpulan
A1	≥ 140	≥ 100	Darah Tinggi
A2	≥ 140	70 - 90	Darah Tinggi
A3	120 - 139	≥ 100	Darah Tinggi
A4	121 - 139	70 - 90	Normal

Tabel 3.5 *Reduced Decision Table* Langkah 3

Langkah 3 : *Reduced Decision Table*

Rule	Sistole	Diastole	Kesimpulan
B1	≥ 140	-	Darah Tinggi
B2	120 - 139	≥ 100	Darah Tinggi
B3	121 - 139	70 - 90	Normal

Pada langkah 1 ditentukan jumlah baris maksimal yang akan terdapat pada *decision table* dengan cara mengkalikan jumlah jawaban tiap-tiap parameter pada *rule set* 3. Kemudian pada langkah 2 tiap-tiap jawaban parameter dikombinasikan sejumlah baris yang telah ditentukan pada langkah 1. Selanjutnya pada langkah 3 tabel 3.4 dilakukan reduksi untuk tiap-tiap parameter. Reduksi tabel dimulai dengan mengambil 2 kolom dari paling kanan, kemudian tiap-tiap baris kolom sebelah kanan dari parameter kolom kiri tadi dilihat kesimpulannya, jika tidak sama maka menjadi rule baru, akan tetapi jika sama baris yang sama tadi dihilangkan dan ditambah kolom baru ke 2 kolom tadi yaitu kolom berasal dari sebelah kiri 2 kolom tadi, kemudian diulang lagi proses perbandingan diatas hingga tinggal satu kolom dan satu baris yang kemudian menjadi rule baru.

3.3.5 Proses Verifikasi

Proses verifikasi dijalankan ketika ada penambahan atau perubahan pada rule, karena rule tersebut sudah ada pada sistem. Sedangkan tujuan verifikasi adalah untuk memastikan adanya kecocokan antara sistem dengan apa yang sistem kerjakan dan juga untuk memastikan bahwa sistem itu terbebas dari error.

Berikut ini adalah yang harus dicek dalam suatu basis pengetahuan:

1. *Redundant rules*

Dikatakan *redundant rules* jika 2 rule atau lebih mempunyai *premise* dan *conclusion* yang sama.

Contoh :

Rule 1 : If gula darah puasa > 126 mg/dl and gula darah 2 jam setelah makan > 200 mg/dl

Then pemeriksaan laboratorium gula darah

Rule 2 : If gula darah 2 jam setelah makan > 200 mg/dl and darah puasa > 126 mg/dl

Then pemeriksaan laboratorium gula darah

2. *Conflicting rules*

Conflicting rules terjadi ketika 2 *rule* atau lebih mempunyai *premise* yang sama tetapi *conclusion* yang berbeda.

Contoh :

Rule 1 : If kolesterol total >200 mg/dl and LDL > 150 mg/dl

Then pemeriksaan laboratorium profil lipid

Rule 2 : If kolesterol total >200 mg/dl and LDL > 150 mg/dl

Then pemeriksaan laboratorium gula darah

3. *Subsumed rules*

Suatu keadaan dapat dikatakan *subsumed rules* jika *rule* tersebut mempunyai *constraint* yang lebih atau kurang tetapi mempunyai *conclusion* yang sama.

Contoh :

Rule 1 : If kolesterol total >200 mg/dl and LDL > 150 mg/dl

Then pemeriksaan laboratorium profil lipid

Rule 2 : If kolesterol total >200 mg/dl

Then pemeriksaan laboratorium profil lipid

4. *Circular rules*

Circular rules ialah suatu keadaan dimana terjadinya proses perulangan dari suatu *rule*. Ini dikarenakan suatu *premise* dari salah satu *rule* merupakan *conclusion* dari *rule* yang lain, atau sebaliknya.

Contoh :

Rule 1 : If kolesterol total >200 mg/dl

Then pemeriksaan laboratorium profil lipid

Rule 2 : If pemeriksaan laboratorium profil lipid

Then X kolesterol total >200 mg/dl

5. *Unnecessary IF condition*

Unnecessary IF terjadi ketika 2 *rule* atau lebih mempunyai *conclusion* yang sama tetapi salah satu dari *rule* tersebut mempunyai *premise* yang tidak perlu dikondisikan dalam *rule* karena tidak mempunyai pengaruh apapun.

Contoh :

Rule 1 : If kolesterol total >200 mg/dl and LDL > 150 mg/dl

Then pemeriksaan laboratorium profil lipid

Rule 2 : If kolesterol total >200 mg/dl and LDL > 250 mg/dl

Then pemeriksaan laboratorium profil lipid

6. *Dead-end rules*

Dead-end rules adalah suatu *rule* yang *conclusionnya* tidak diperlukan oleh *rule-rule* lainnya.

Contoh :

Rule 1 : If bukan gejala penyakit degeneratif

Then bukan penyakit degeneratif

3.3.6 Perancangan *Rule Base*

Pengembangan *rule base* telah digambarkan dalam bentuk diagram blok yang kemudian akan diimplementasikan dalam bentuk aturan-aturan (*rules*), yaitu struktur berbasis pengetahuan. *Rule base* yang dihasilkan oleh sistem akan disimpan dalam tabel *rule*.

Suatu rule terdiri dari dua bagian utama, yaitu *IF* yang sering disebut premis atau kondisi dan bagian *THEN* yang sering disebut konklusi atau kesimpulan. Berikut ini merupakan contoh dari struktur basis pengetahuan yang sebelumnya telah dirancang menggunakan diagram blok.

Rule 1 : If Sistol \geq 140

Then Darah Tinggi

Rule 2 : If Sistol = 120-139 *and* diastole \geq 100

Then Darah Tinggi

Rule 3 : If Sistol = 120-139 *and* diastole = 70-90

Then Normal

3.4 Struktur Tabel

Struktur tabel merupakan penjabaran dan penjelasan tabel tersebut, mulai dari fungsi masing-masing tabel sampai masing-masing *field* yang ada dalam tabel. Adapun struktur tabel adalah sebagai berikut:

1. Nama : Login
 Primary key :
 Foreign key : -
 Fungsi : Digunakan menyimpan data Login

Tabel 3.6 Login

Field	Tipe Data	Constraint	Keterangan
username	<i>Text(50)</i>	PK	id user
password	<i>Text(50)</i>	Not Null	<i>password user</i>
Tipe	<i>Text(50)</i>	Not Null	tipe user yaitu admin atau user

2. Nama : Pengguna
- Primary key : id_pengguna
- Foreign key : -
- Fungsi : digunakan untuk menyimpan data-data pengguna

Tabel 3.7 Pengguna

Field	Type Data	Constraint	Keterangan
id_pengguna	<i>Number</i>	PK	id pengguna berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi id_pengguna
nama	<i>Text(50)</i>	Not Null	nama pengguna
tanggal_lahir	<i>Date</i>	Not Null	tanggal lahir pengguna
jenis_kelamin	<i>Text(1)</i>	Not Null	jenis kelamin pengguna
alamat	<i>Text(50)</i>	Not Null	alamat pengguna
telepon	<i>Text(15)</i>	Not Null	telepon pengguna
kota	<i>Text(50)</i>	Not Null	kota domisili pengguna

3. Nama : Solusi
- Primary key : id_solusi
- Foreign key : id_value dan id_parameter
- Fungsi : Digunakan menyimpan solusi dari setiap kesimpulan akhir

Tabel 3.8 Solusi

Field	Type Data	Constraint	Keterangan
id_solusi	<i>Number</i>	PK	id solusi berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi id_solusi
id_value	<i>Number</i>	FK	id value berasal dari tabel PossibleValue
id_parameter		FK	id parameter berasal dari tabel PossibleValue
isi_solusi	Memo	Not Null	isi solusi

4. Nama : PossibleValue
- Primary key : id_value dan id_parameter
- Foreign key : -
- Fungsi : Digunakan menyimpan data jawaban pertanyaan dari masing-masing parameter

Tabel 3.9 Possible Value

Field	Tipe Data	Constraint	Keterangan
id_value	Number	PK	id value berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi id_value
id_parameter	Number	PK	id parameter yang berasal dari tabel Parameter
nama_value	Text(50)	Not Null	nama value

5. Nama : Parameter
- Primary key : id_parameter
- Foreign key : -
- Fungsi : Digunakan menyimpan data parameter berdasarkan *Tree View* yang telah dibuat.

Tabel 3.10 Parameter

Field	Tipe Data	Constraint	Keterangan
id_parameter	Number	PK	id parameter berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi id_parameter
nama_parameter	Text(50)	Not Null	nama parameter di dalam <i>tree view</i>
path	Text(50)	Not Null	path parameter di dalam <i>tree view</i>
Idx	Number	Not Null	index parameter di dalam <i>tree view</i>
pertanyaan	Memo	Not Null	pertanyaan yang akan ditampilkan kepada <i>user</i>

6. Nama : Rule
- Primary key : id_rule
- Foreign key : set
- Fungsi : digunakan untuk menyimpan *rule* hasil *generate*

Tabel 3.11 Rule

Field	Type Data	Constraint	Keterangan
id_rule	<i>Number</i>	PK	id rule berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi id rule
Set	<i>Number</i>	FK	berupa <i>number</i> yang menunjukkan posisi <i>rule</i> dalam <i>dependency diagram</i>

7. Nama : DetailRule
- Primary key : id_detail, id_parameter dan id_value
- Foreign key : -
- Fungsi : Digunakan menyimpan detail dari setiap *rule*

Tabel 3.12 Detail Rule

Field	Type Data	Constraint	Keterangan
id_rule	<i>Number</i>	PK	id rule berasal dari tabel Rule
id_value	<i>Number</i>	PK	id value berasal dari tabel PossibleValue
Id_parameter	<i>Number</i>	PK	id parameter berasal dari tabel PossibleValue
is_conclusion	<i>Yes/No</i>		keterangan yang menunjukkan apakah sebagai <i>conclusion</i> atau <i>premise</i>

8. Nama : Konsultasi
- Primary key : id_konsultasi
- Foreign key : -
- Fungsi : Digunakan menyimpan data hasil konsultasi

Tabel 3.13 Konsultasi

Field	Type Data	Constraint	Keterangan
id_konsultasi	Number	PK	id konsultasi berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi id konsultasi
jawaban	Text(100)	Not Null	jawaban dari pertanyaan
kesimpulan	Text(100)	Not Null	kesimpulan konsultasi
kesimpulan akhir	Text(100)	Not Null	kesimpulan akhir konsultasi

3.5 Desain Input Output

Bagian ini menjelaskan tentang desain *input output* dari aplikasi sistem berbasis aturan menentukan penyakit degeneratif. *Form-form* yang dirancang meliputi konsep interaksi manusia dan komputer dimana seorang *user* hanya dengan melihat form, dapat mengerti langkah apa yang dilakukan selanjutnya.

3.5.1 Desain Form Login

Desain *form login* berfungsi untuk masuk ke dalam sistem. Pada desain *form login* ini terdapat level *user* yang digunakan untuk menentukan siapa yang melakukan *login*, apakah admin atau *user* umum. *user* umum hanya dapat melakukan konsultasi, sedangkan admin dapat melakukan *maintenance* data, *input* atau *update* treeview, melakukan proses verifikasi *rule*, membuat pertanyaan dan mengisi kesimpulan. Desain *form login* dapat dilihat pada gambar 3.8.

Gambar 3.8 Desain *Form* Login

Fungsi-fungsi obyek yang ada pada desain *form login* dapat dilihat pada tabel 3.14.

Tabel 3.14 Fungsi Obyek Desain *Form* Login

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field</i> (<i>Username/Password</i>)	<i>TextBox</i>	Digunakan untuk mengisi data <i>username</i> , dan <i>password</i> sesuai dengan data yang ada dalam <i>database</i> .
2	<i>Login</i>	<i>Button</i>	Digunakan sebagai autentikasi dan otorisasi agar dapat masuk dan menggunakan sistem.
3	Batal	<i>Button</i>	Digunakan untuk membersihkan <i>field</i>
4	Keluar	<i>Button</i>	Digunakan untuk keluar dari sistem

3.5.2 Desain *Form* Utama

Desain *form* utama digunakan sebagai penghubung antara *form* yang satu dengan *form* yang lainnya. Berdasarkan hak aksesnya, desain *form* utama dibagi menjadi dua yaitu, *form* utama admin dan *form* utama *user* umum. Desain *form* utama admin dapat dapat mengakses seluruh menu yang ada, sedangkan untuk *user* umum hanya dapat mengakses menu konsultasi. Desain *form* utama dapat dilihat pada gambar 3.9.

Menu	
Home Log Out	
Master	
Pengguna Pertanyaan Saran Pengobatan	
Transaksi	
Treeview Varifikasi Rule	
Laporan	
Data Pertanyaan Data Rule	
Konsultasi	
Konsultasi Histori Konsultasi	

Gambar 3.9 Desain *Form* Utama

Fungsi-fungsi obyek dalam desain *form* utama dapat dilihat pada tabel 3.15.

Tabel 3.15 Fungsi Obyek Desain *Form* Utama

No	Nama obyek	Tipe Obyek	Fungsi
1	Menu	<i>Link Label</i>	Digunakan untuk memanggil <i>form</i> anak

3.5.3 Desain *Form* Master Pengguna

Desain *form* master pengguna digunakan untuk melakukan *maintenance* data-data pengguna. Kode pengguna akan di *generate* langsung oleh sistem sehingga *user* tidak perlu melakukan *input* manual. Master pengguna terbagi atas dua bagian yaitu *form* data pengguna dan *form* tambah data pengguna. *Form* data pengguna digunakan untuk proses mencari data pengguna, hapus data pengguna, dan ubah data pengguna. Desain *form* data pengguna dapat dilihat pada gambar 3.10. *Form* tambah data pengguna digunakan untuk menambah data pengguna. Desain *form* tambah data pengguna dapat dilihat pada gambar 3.11.

Data Pengguna

Cari Berdasarkan

Kata Kunci

Gambar 3.10 Desain *Form* Data Pengguna

Fungsi-fungsi obyek dalam desain *form* data pengguna dapat dilihat pada tabel 3.16.

Tabel 3.16 Fungsi Obyek Desain *Form* Data Pengguna

No	Nama Obyek	Tipe Obyek	Fungsi
1	Cari berdasarkan	<i>Combobox</i>	Digunakan untuk memilih kategori pencarian data
2	Kata kunci	<i>Textbox</i>	Digunakan untuk memasukkan kata kunci pencarian data
3	Cari	<i>Button</i>	Digunakan untuk mencari data
4	Simpan	<i>Button</i>	Digunakan untuk menyimpan data pengguna
5	Hapus	<i>Button</i>	Digunakan untuk menghapus data
6	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>

Tambah Data Pengguna

Kode

Nama

Tgl Lahir

Usia

Jenis Kelamin Laki-laki Perempuan

Alamat

Telepon

Kota

Gambar 3.11 Desain *Form* Tambah Data Pengguna

Fungsi-fungsi obyek dalam desain *form* tambah data pengguna dapat dilihat pada tabel 3.17.

Tabel 3.17 Fungsi Obyek Desain *Form* Tambah Data Pengguna

No	Nama Obyek	Tipe Obyek	Fungsi
1	Kode	<i>Textbox</i>	Digunakan untuk data kode pengguna
2	Nama	<i>Textbox</i>	Digunakan untuk memasukkan data nama pengguna
4	Tgl Lahir	<i>Textbox</i>	Digunakan untuk memasukkan data tanggal lahir pengguna
5	Usia	<i>Textbox</i>	Digunakan untuk memasukkan data usia pengguna
6	Jenis Kelamin	<i>Radiobutton</i>	Digunakan untuk memilih jenis kelamin pengguna
7	Alamat	<i>Textbox</i>	Digunakan untuk memasukkan data alamat pengguna

8	Telepon	<i>Textbox</i>	Digunakan untuk memasukkan data telepon pengguna
9	Kota	<i>Textbox</i>	Digunakan untuk memasukkan data kota pengguna
10	Simpan	<i>Button</i>	Digunakan untuk menyimpan data pengguna
11	Batal	<i>Button</i>	Digunakan untuk membersihkan <i>field</i> apabila <i>user</i> batal menyimpan
12	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>

3.5.4 Desain Form Master Pertanyaan

Desain *form* master pertanyaan digunakan untuk melakukan *maintenance* terhadap pertanyaan tiap-tiap parameter. *Form* master pertanyaan digunakan untuk menambah dan mengubah pertanyaan untuk parameter yang ada. *Form* master pertanyaan dapat dilihat pada gambar 3.12.

Maintenance Pertanyaan

Parameter	<input type="text" value="Varchar(50)"/>
Pertanyaan	<input type="text" value="Varchar(50)"/>
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>	

Gambar 3.12 Desain *Form* Master Pertanyaan

Fungsi-fungsi obyek dalam desain *form* master pertanyaan dapat dilihat pada tabel 3.18.

Tabel 3.18 Fungsi Obyek Desain *Form* Master Pertanyaan

No	Nama Obyek	Tipe Obyek	Fungsi
1	Parameter	<i>Textbox</i>	Digunakan untuk menampilkan parameter yang dipilih
2	Pertanyaan	<i>Textbox</i>	Digunakan untuk memasukkan pertanyaan untuk parameter yang dipilih
3	Simpan	<i>Button</i>	Digunakan untuk menyimpan data pertanyaan
4	Batal	<i>Button</i>	Digunakan untuk Membatalkan penyimpanan data pertanyaan
5	Isi	<i>Button</i>	Digunakan untuk menambah data pertanyaan baru
6	Ubah	<i>Button</i>	Digunakan untuk mengubah data pertanyaan untuk parameter
7	Keluar	<i>Button</i>	Digunakan untuk keluar <i>form</i>

3.5.5 Desain Form Master Saran Pengobatan

Desain *form* master saran pengobatan digunakan untuk melakukan *maintenance* terhadap kesimpulan dan saran pengobatan. *Form* master saran pengobatan digunakan untuk menambah dan mengubah saran pengobatan untuk tipa-tiapa kesimpulan yang ada. *Form* master saran pengobatan dapat dilihat pada gambar 3.13.

Gambar 3.13 Desain *Form* Master Saran Pengobatan

Fungsi-fungsi obyek dalam desain *form* master saran pengobatan dapat dilihat pada tabel 3.19.

Tabel 3.19 Fungsi Obyek Desain *Form* Master Saran Pengobatan

No	Nama Obyek	Tipe Obyek	Fungsi
1	Kesimpulan	<i>Textbox</i>	Digunakan untuk menampilkan Kesimpulan yang dipilih
2	Saran Pengobatan	<i>Textbox</i>	Digunakan untuk memasukkan saran pengobatan untuk kesimpulan yang dipilih
3	Simpan	<i>Button</i>	Digunakan untuk menyimpan data saran pengobatan
4	Batal	<i>Button</i>	Digunakan untuk Membatalkan penyimpanan data saran pengobatan
5	Isi	<i>Button</i>	Digunakan untuk manambah data saran pengobatan baru

6	Ubah	<i>Button</i>	Digunakan untuk mengubah data saran pengobatan untuk kesimpulan
7	Keluar	<i>Button</i>	Digunakan untuk keluar <i>form</i>

3.5.6 Desain Form Treeview

Desain *form treeview* digunakan untuk membuat *treeview* yang merupakan langkah awal dalam membuat *rule*. Desain *form treeview* dapat dilihat pada gambar 3.14.

Gambar 3.14 Desain *Form Treeview*

Fungsi-fungsi obyek dalam desain *form treeview* dapat dilihat pada tabel 3.20.

Tabel 3.20 Fungsi Obyek Desain *Form Treeview*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Nama node	<i>Textbox</i>	Digunakan untuk menginputkan nama node
2	Level node	<i>Textbox</i>	Digunakan untuk memberikan informasi berupa level node
3	Indeks node	<i>Textbox</i>	Digunakan untuk memberikan informasi berupa indeks node
4	Nama parent	<i>Textbox</i>	Digunakan untuk memberikan informasi berupa nama parent
5	Set rule	<i>Radiobutton</i>	Digunakan untuk memilih kondisi set rule
6	Pertanyaan	<i>Textbox</i>	Digunakan untuk menginputkan pertanyaan
7	Pilihan jawaban	<i>Textbox</i>	Digunakan untuk menginputkan pilihan jawaban
8	Jumlah value	<i>numeric updown</i>	Digunakan untuk menginputkan pilihan jumlah value
9	Pilihan value	<i>Textbox</i>	Digunakan untuk menginputkan isi value
10	Tambah node	<i>Button</i>	Digunakan untuk menambah node
11	Ubah node	<i>Button</i>	Digunakan untuk mengubah node
12	Hapus node	<i>Button</i>	Digunakan untuk menghapus node
13	Batal	<i>Button</i>	Digunakan untuk membersihkan field
14	Simpan	<i>Button</i>	Digunakan untuk menyimpan data treeview dan data node
15	Ubah	<i>Button</i>	Digunakan untuk mengubah data treeview dan data node
16	Keluar	<i>Button</i>	Digunakan untuk keluar dari form
17	Treeview	<i>Treeview</i>	Untuk menampilkan treeview

3.5.7 Desain Form Verifikasi Rule

Desain *form* verifikasi *rule* digunakan untuk proses verifikasi *rule*. Langkah pertama yang harus dilakukan untuk melakukan verifikasi *rule* adalah memasukkan set *rule* untuk verifikasi. Setelah set *rule* dipilih, kemudian set *rule* beserta jawabannya tampil pada detail *rule*. Setelah menekan tombol ok, sistem menampilkan set *rule* dan jawabannya yang ada pada detail *rule* untuk ditampilkan pada *listbox*. User harus memilih kesimpulan dari *rule* yang akan *digenerate* dan diverifikasi. Setelah kesimpulan dipilih, sistem akan melakukan *generate rule* dan hasil *generate* harus diverifikasi terlebih dahulu untuk menghindari terjadinya kesalahan pada *rule*. Desain *form* verifikasi *rule* dapat dilihat pada gambar 3.15.

Gambar 3.15 Desain *Form* Verifikasi Rule

Fungsi-fungsi obyek dalam desain *form* verifikasi *rule* dapat dilihat pada tabel 3.21.

Tabel 3.21 Fungsi Obyek Desain *Form* Verifikasi *Rule*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Set rule	<i>combobox</i>	Digunakan untuk memilih set <i>rule</i>
2	Parameter	<i>combobox</i>	Digunakan untuk memilih parameter dari set <i>rule</i> yang telah dipilih
3	Jawaban	<i>combobox</i>	Digunakan untuk memilih jawaban dari parameter yang telah dipilih
4	Nama set	<i>combobox</i>	Digunakan untuk memilih nama set
5	Value set	<i>combobox</i>	Digunakan untuk memilih <i>value</i> dari nama set yang dipilih
6	Listbox kesimpulan	<i>Listbox</i>	Sebagai tempat untuk menampilkan set <i>rule</i> yang dipilih beserta jawabannya
7	Kesimpulan	<i>combobox</i>	Digunakan untuk memilih kesimpulan set <i>rule</i>
8	Ok	<i>Button</i>	Digunakan untuk menampilkan set <i>rule</i> yang dipilih beserta jawabannya pada <i>listbox</i>
9	Generate Rule	<i>Button</i>	Digunakan untuk menampilkan/melakukan <i>generate rule</i> pada <i>listview</i>
10	Batal	<i>Button</i>	Digunakan untuk membersihkan data di <i>listbox</i>
11	Simpan	<i>Button</i>	Digunakan untuk menyimpan <i>rule</i> yang telah diverifikasi
12	Verifikasi	<i>Button</i>	Digunakan untuk memverifikasi <i>rule</i> berdasarkan teori verifikasi
13	Rule	<i>Listview</i>	Sebagai tempat untuk menampilkan <i>rule</i> yang telah <i>digenerate</i>
14	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>

3.5.8 Desain Form Konsultasi

Desain *form* konsultasi berisi pertanyaan-pertanyaan yang diajukan kepada *user*. Pilihan jawaban yang ditampilkan sistem merupakan jawaban yang ada dalam *database*. Desain *form* konsultasi dapat dilihat pada gambar 3.16.

Gambar 3.16 Desain *Form* Konsultasi

Fungsi-fungsi obyek dalam desain *form* konsultasi dapat dilihat pada tabel 3.22.

Tabel 3.22 Fungsi Obyek Desain *Form* Konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Id konsultasi	<i>Label</i>	Digunakan untuk menampilkan id konsultasi
2	Kode pengguna	<i>Label</i>	Digunakan untuk menampilkan kode pengguna
3	Nama	<i>Label</i>	Digunakan untuk menampilkan nama pengguna
4	Usia	<i>Label</i>	Digunakan untuk menampilkan usia pengguna
7	Pertanyaan	<i>Label</i>	Sebagai tempat menampung data pertanyaan dari <i>database</i>

8	Jawaban	<i>radiobutton</i>	Sebagai tempat menampung data jawaban dari <i>database</i>
9	Lanjut	<i>Button</i>	Digunakan untuk menuju pertanyaan dan jawaban selanjutnya
10	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>
11	Lihat Hasil konsultasi	<i>Button</i>	Digunakan untuk ke <i>form</i> hasil konsultasi pengguna

3.5.9 Desain Form Hasil Konsultasi Pengguna

Desain *form* hasil konsultasi digunakan untuk menampilkan data hasil konsultasi pengguna. *Form* ini berisi jawaban-jawaban hasil konsultasi, kesimpulan beserta saran pengobatannya. Desain *form* hasil konsultasi dapat dilihat pada gambar 3.17.

The image shows a web form titled "Hasil Konsultasi". It includes the following elements:

- Input fields for "Id konsultasi" (with label "Idkonsultasi"), "Nama pengguna" (with label "namaPegguna"), and "Kode pengguna" (with label "kodePegguna").
- Three large text input areas labeled "Jawaban", "Kesimpulan", and "Saran Pengobatan".
- Two buttons at the bottom right: "Simpan" and "Keluar".

Gambar 3.17 Desain *Form* Hasil Konsultasi

Fungsi-fungsi obyek dalam desain *form* hasil konsultasi dapat dilihat pada tabel 3.23.

Tabel 3.23 Fungsi Obyek Desain *Form* Hasil Konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Id Konsultasi	<i>Label</i>	Digunakan untuk menampilkan id konsultasi
2	Nama pengguna	<i>Label</i>	Digunakan untuk menampilkan nama pengguna
3	Kode pengguna	<i>Label</i>	Digunakan untuk menginputkan kode pengguna
4	Jawaban	<i>Listbox</i>	Digunakan untuk menampilkan jawaban-jawaban hasil konsultasi pengguna
5	Kesimpulan	<i>Listbox</i>	Digunakan untuk menampilkan kesimpulan dari fakta-fakta yang diberikan <i>user</i>
6	Saran pengobatan	<i>Listbox</i>	Digunakan untuk menampilkan saran pengobatan
7	Simpan	<i>Button</i>	Digunakan untuk menyimpan hasil konsultasi
8	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>

3.5.10 Desain Form Laporan History Konsultasi Pengguna

Desain *form* laporan *history* konsultasi pengguna digunakan untuk menampilkan data *history* konsultasi pengguna yang tersimpan di *database* dalam bentuk *crystal report*. Gambar desain *form* laporan *history* konsultasi pengguna dapat dilihat pada gambar 3.18.

Gambar 3.18 Desain *Form* Laporan *History* Konsultasi Pengguna

Fungsi-fungsi obyek dalam desain *form* laporan *history* konsultasi pengguna dapat dilihat pada tabel 3.24.

Tabel 3.24 Fungsi Obyek Desain *Form* Laporan *History* Konsultasi Pengguna

No	Nama Obyek	Tipe Obyek	Fungsi
1	Crystal report history pengguna	<i>crystal report</i>	Digunakan untuk menampilkan data history konsultasi pengguna pada crytal report
2	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>

3.5.11 Desain Form Laporan Pertanyaan

Desain *form* laporan pertanyaan digunakan untuk menampilkan data pertanyaan yang tersimpan di *database*. Desain *form* laporan pertanyaan dapat dilihat pada gambar 3.19.

The image shows a web form with a title bar that says "Laporan Data Pertanyaan". In the top right corner of the form, there is a button labeled "Keluar". Below the button is a large, empty rectangular box with a thin border, which is labeled "Crystal Report Data Pertanyaan" in the center. The background of the page has a watermark for "stikom SURABAYA" and "INSTITUT BISNIS & INFORMATIKA".

Gambar 3.19 Desain *Form* Laporan Pertanyaan

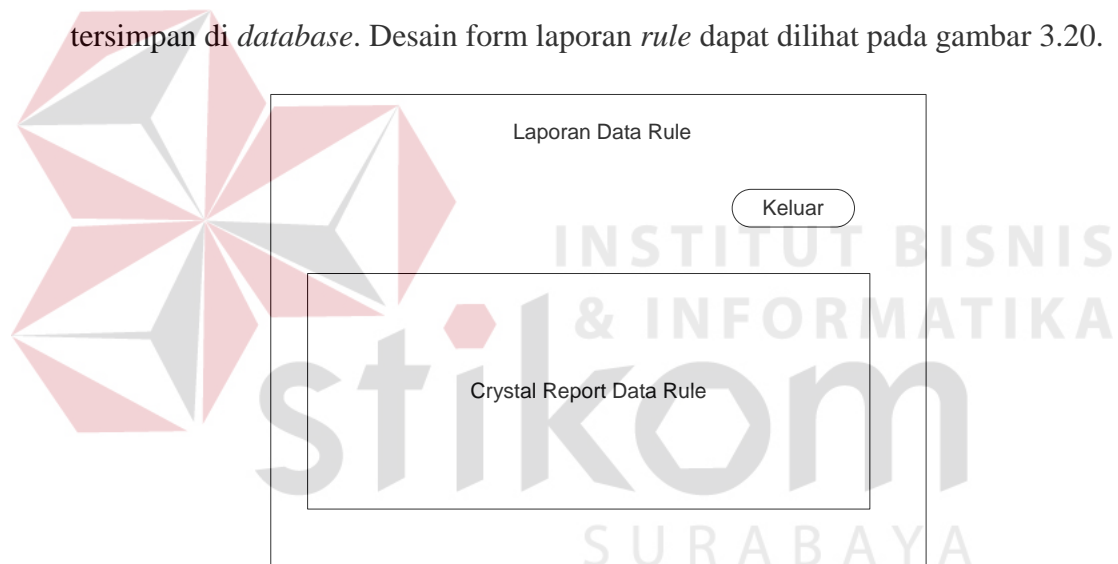
Fungsi-fungsi obyek dalam desain *form* laporan pertanyaan dapat dilihat pada tabel 3.25.

Tabel 3.25 Fungsi Obyek Desain *Form* Laporan Pertanyaan

No	Nama Obyek	Tipe Obyek	Fungsi
1	Crystal report pertanyaan	<i>crystal report</i>	Digunakan untuk menampilkan data pertanyaan pada crystal report
2	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>

3.5.12 Desain Form Laporan Rule

Desain form laporan *rule* digunakan untuk menampilkan data *rule* yang tersimpan di *database*. Desain form laporan *rule* dapat dilihat pada gambar 3.20.

Gambar 3.20 Desain *Form* Laporan *Rule*

Fungsi-fungsi obyek dalam desain *form* laporan *rule* dapat dilihat pada tabel 3.26.

Tabel 3.26 Fungsi Obyek Desain *Form* Laporan *rule*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Crystal report rule	<i>crystal report</i>	Digunakan untuk menampilkan data <i>rule</i> pada crystal report
2	Keluar	<i>Button</i>	Digunakan untuk keluar dari <i>form</i>