

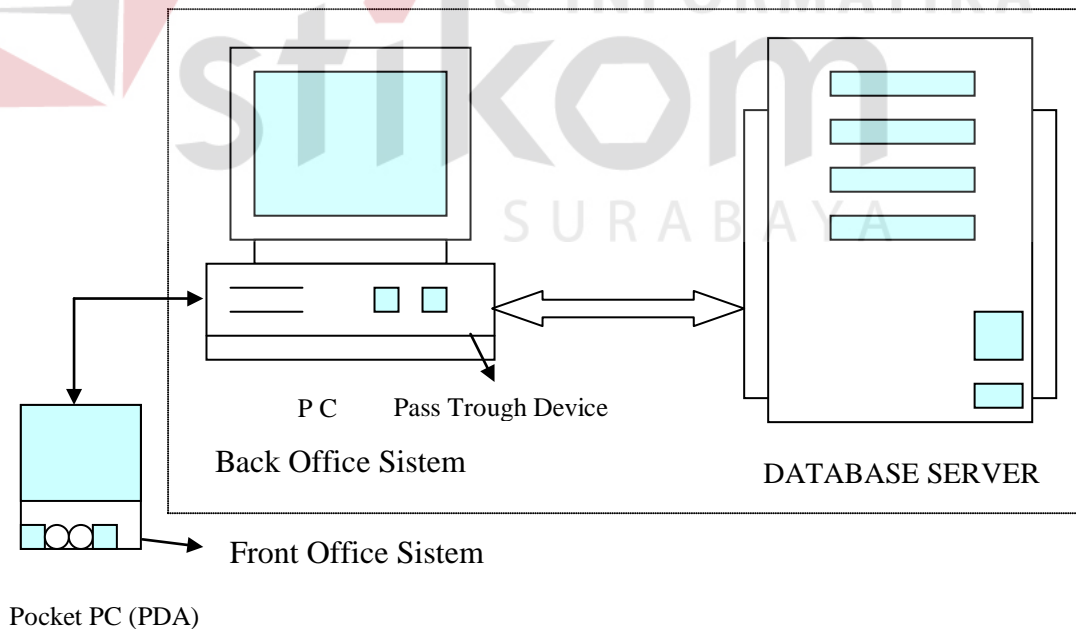
BAB III

PERANCANGAN SISTEM

3.1 Abstraksi Sistem

Sistem ini nantinya akan dibagi menjadi dua, yaitu *Front Office* dan *Back Office*. Sistem front office menggunakan *platform PDA* dan pada back office menggunakan *platform PC*. Adapun dilakukannya pembuatan dua sistem ini dilakukan mengingat keterbatasan yang dimiliki oleh PDA antara lain memori, tampilan, warna dan lain-lain. Semua perancangan, desain sistem, penulisan kode program dan database awal dilakukan pada *PC Development*, setelah dilakukan uji coba simulasi pada *emulator* maka aplikasi siap untuk *di-upload* ke PDA.

3.2 Bagan Sistem



Gambar 3.1 Bagan Sistem

Pada bagan sistem terlihat pada gambar 3.1 sistem terdiri dari *Front Office* Sistem (yang terdiri dari PDA), dan *Back Office* Sistem (yang terdiri dari sebuah PC dan sebuah sever). Garis putus-putus yang melingkupi PC dengan Server menunjukkan bahwa nantinya PC juga berperan sekaligus menjadi Server.

Proses pendistribusian data dari sumber data (*publisher*) ke penerima data (*subscriber*). Pada keadaan ini maka SQL Server akan berperan sebagai *publisher* dan SQL Server CE akan berperan sebagai *subscriber*. Konfigurasi ini memungkinkan *publisher* dan *subscriber* melakukan *update* pada masing-masing database, baik keduanya sedang terhubung ataupun tidak. Data-data yang *di-update* akan disinkronisasi ketika koneksi terjadi. Namun resiko terjadinya konflik pada sinkronisasi ini sangat mungkin terjadi. Konflik tersebut dapat diatasi dengan membuat beberapa table penghubung yang bertugas menangani database PDA dan PC.

3.3 Perancangan Sistem

3.3.1 Diagram Alir Sistem

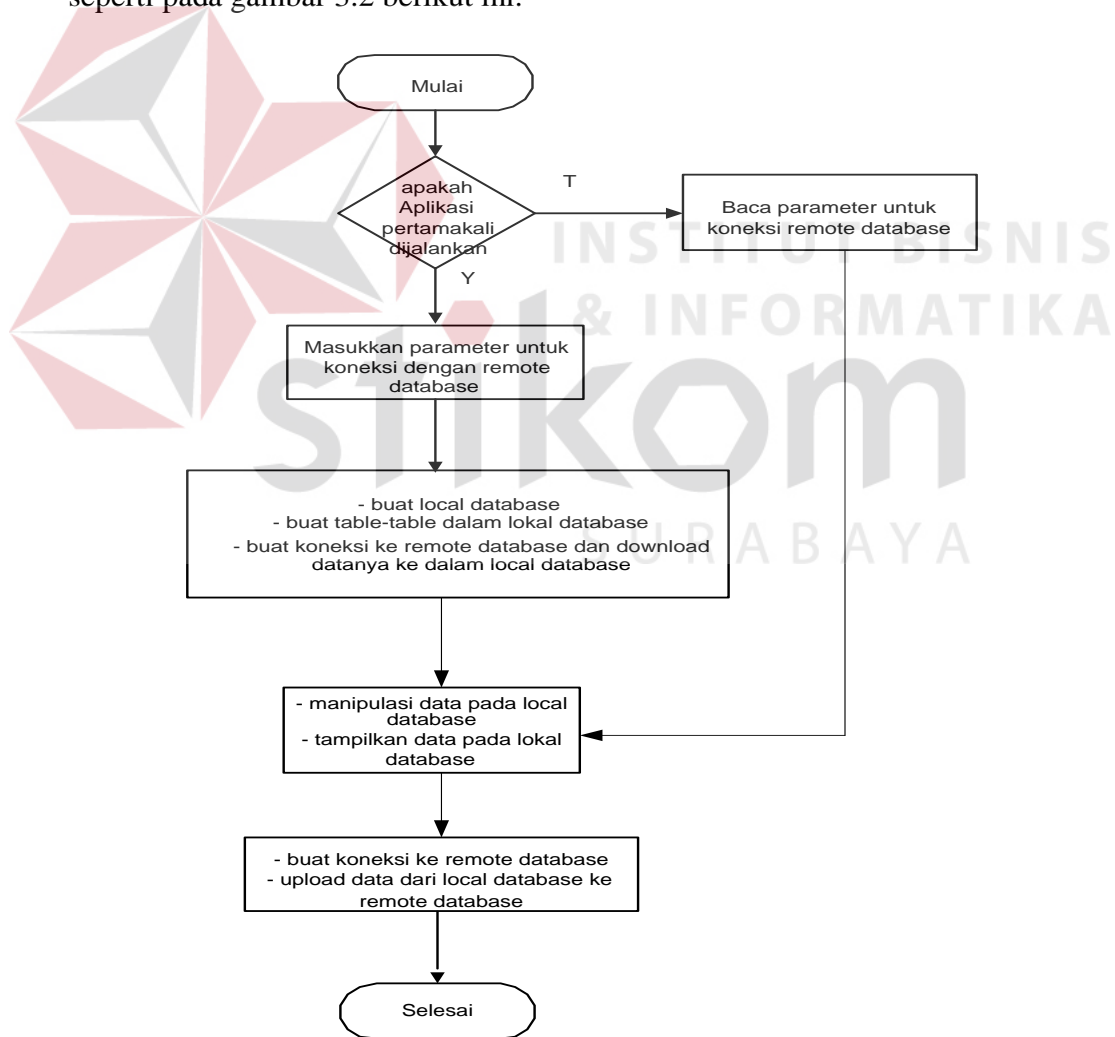
Diagram alir (*flowchart*) adalah diagram (*chart*) yang menunjukkan alir proses (*flow*) didalam program atau prosedur sistem secara logika dan menjelaskan secara rinci langkah-langkah dari proses program.

Setelah konsep database SQL Server CE dijelaskan pada bab sebelumnya, yaitu pada pada bab Konsep Database pada PocketPC, maka berikut ini akan menjelaskan mengenai teknik pembuatan aplikasi dengan akses database dan *sinkronisasi* menggunakan Remote Data Access (RDA).

Aplikasi yang akan dibuat adalah program yang akan melakukan proses-proses sebagai berikut:

1. Membuat local database (SQL Server CE) melalui kode program.
2. Men-*download* data dari remote database (SQL Server) dan menyimpan kedalam local database.
3. Memanipulasi isi *record*, menambah, mengedit dan menghapus record pada local database.
4. Melakukan *upload* data yang dimanipulasi pada local database ke remote database.

Dengan proses seperti diatas, maka sistem ini mempunyai diagram alir seperti pada gambar 3.2 berikut ini.



Gambar 3.2 Diagram Alir Sistem PDA

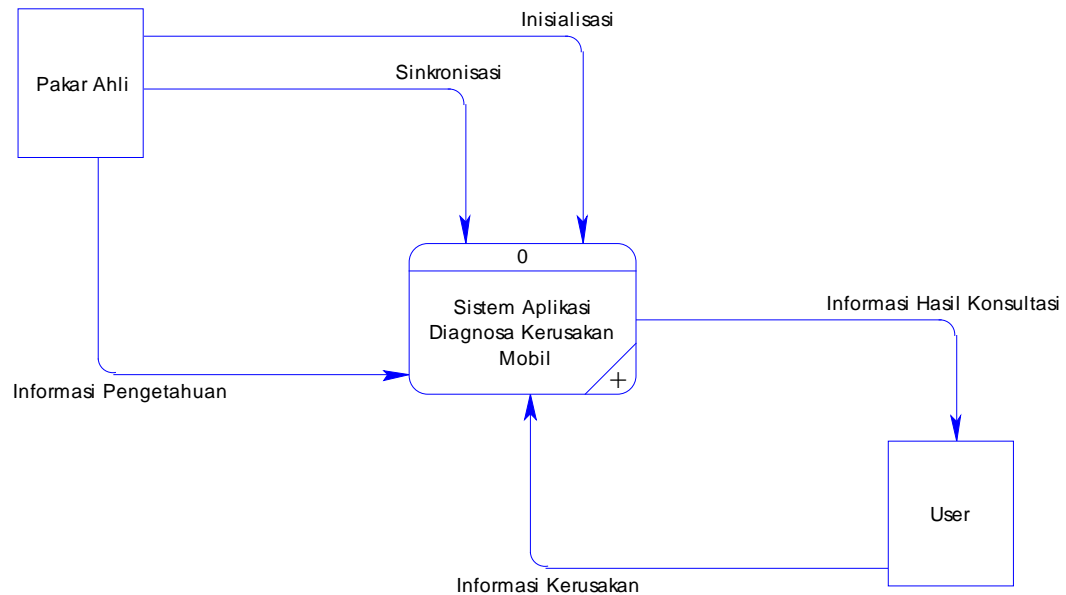
3.3.2 Data Flow Diagram

Data Flow Diagram (DFD) merupakan teknik grafik yang digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data sampai keluaran. Digunakannya DFD pada sistem ini dimaksudkan untuk mempermudah pemahaman aliran data yang terjadi saat sinkronisasi dari *download* ke local database sampai *upload* kembali ke server database.

A. DFD Level Kontex

DFD Level kontex disebut juga dengan *contex diagram* ini menggambarkan pandangan mengenai sistem secara keseluruhan. Dalam level ini hanya memuat aliran data antara external entity dan proses, sama sekali belum melibatkan data store.

Gambar 3.3 adalah gambar DFD level kontex dari sistem aplikasi diagnosa kerusakan mobil, dimana melihat dua entity yaitu entity pakar ahli untuk memberi informasi pengetahuan dan melakukan inisialisasi atau sinkronisasi, dan entity user yang nantinya sebagai pengguna dan melakukan konsultasi.

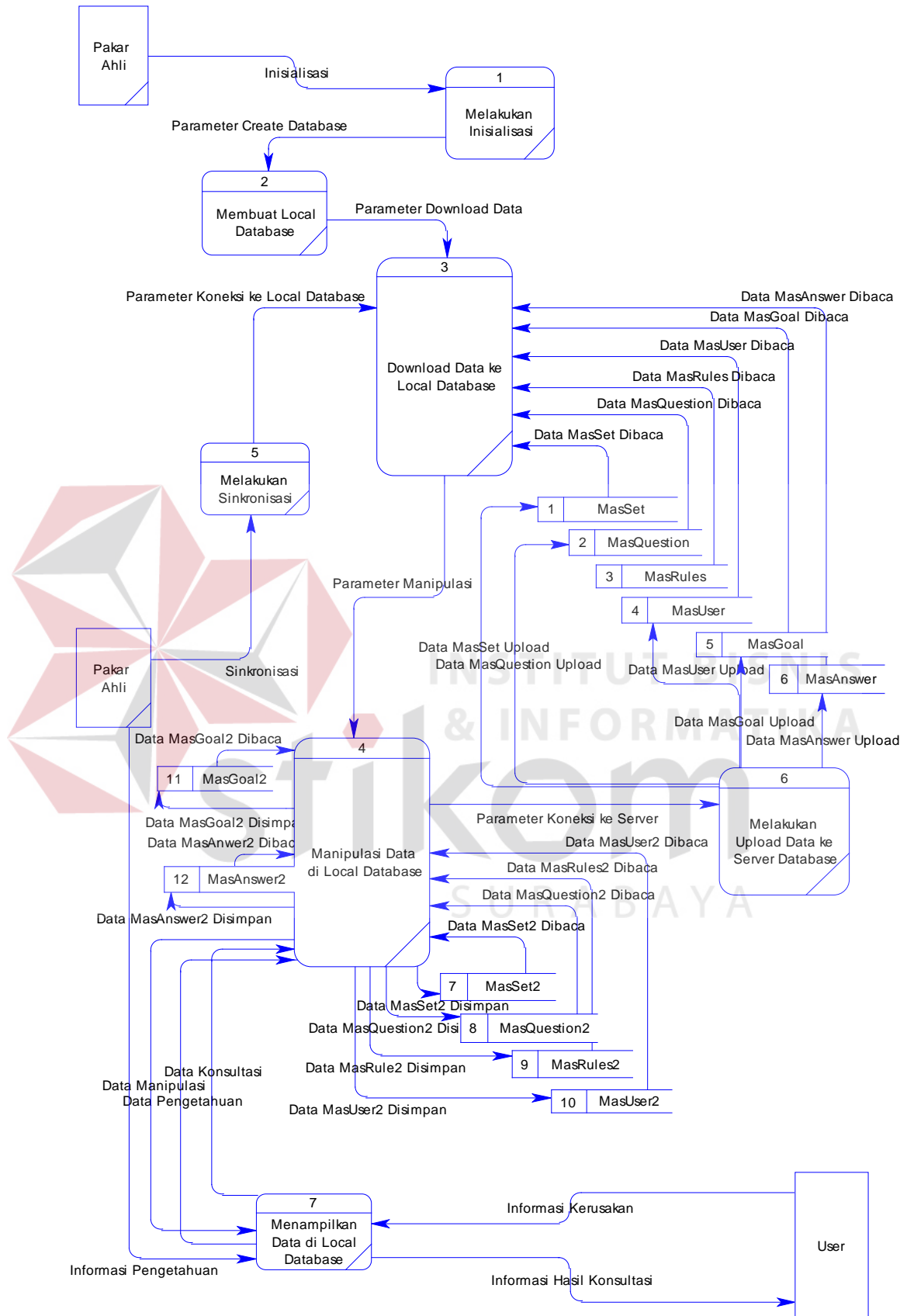


Gambar 3.3 Diagram Level Kontex

B. DFD Level 0

DFD Level 0 atau disebut juga dengan *overview diagram* dimulai dari pakar ahli melakukan inisialisasi, kemudian membuat local database dan tabel. Setelah database siap, selanjutnya dilakukan *download* data dari server database (PC) ke local database (PDA), maka data siap untuk dimanipulasi yaitu *update*, *insert*, *delete* serta siap ditampilkan ke layar.

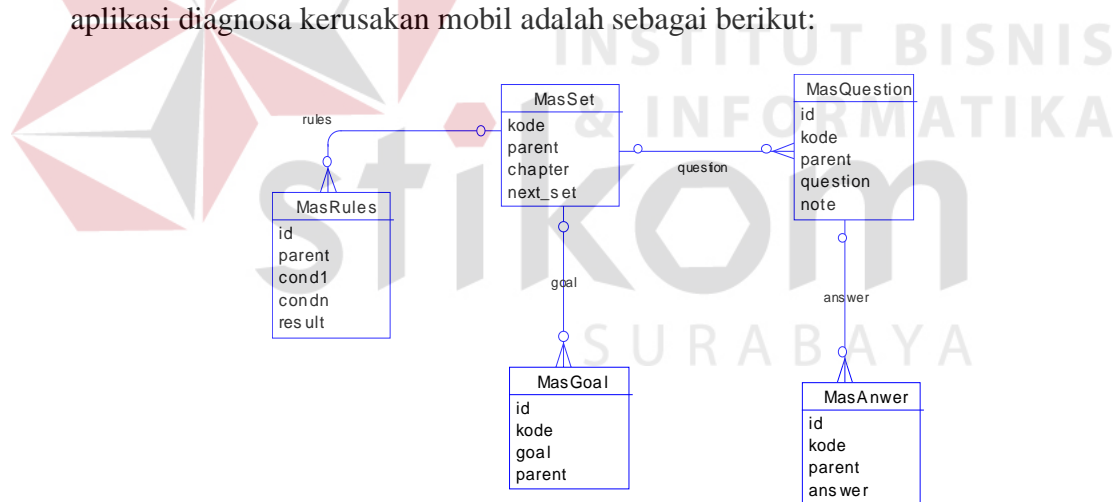
Data hasil manipulasi tersebut tetap berada di local database sampai dilakukan sinkronisasi. Jika dilakukan sinkronisasi, maka data hasil manipulasi tersebut akan di upload ke server database, sehingga data yang berada di server maupun local menjadi sama. Gambar 3.4 adalah gambar DFD Level 0 dari sistem aplikasi diagnosa kerusakan.



Gambar 3.4 DFD Level 0

3.3.3 Entity Relationship Diagram

ERD merupakan suatu desain sistem yang digunakan untuk merepresentasikan atau menentukan kebutuhan-kebutuhan untuk pemrosesan database. Pada tugas ini ERD digunakan untuk menggambarkan relasi antar tabel. Tabel-tabel yang ada nantinya menggunakan Microsoft SQL Server pada PC (yang juga berfungsi sebagai server) dan Microsoft SQL Server CE pada PDA. Adapun susunan dari kedua database tersebut adalah sama karena metode koneksi database tersebut menggunakan konsep *Remote Data Access (RDA)* yang menuntut kesamaan nama tipe field dan kolom dari *Remote Database* dan *Local Database*. Bentuk dari ERD yang akan digunakan untuk manipulasi data pada aplikasi diagnosa kerusakan mobil adalah sebagai berikut:



Gambar 3.5 ERD Diagnosa Kerusakan

3.3.4 Perancangan Struktur Database

Agar sistem dapat berjalan sebagaimana mestinya, maka dirancang struktur database dan struktur tabel yang nantinya digunakan untuk manipulasi data, yaitu *update*, *insert*, *delete*. Tabel-tabel yang terdapat didalam database tersebut yaitu

tabel MasSet untuk menyimpan parent atau set induk, tabel MasQuestion untuk menyimpan premise dari masing-masing parent dimana table MasQuestion ini juga akan menjadi pertanyaan untuk konsultasi. Untuk menyimpan goal atau tujuan dari masing-masing parent digunakan tabel MasGoal, sedangkan untuk menyimpan jawaban dari premise digunakan tabel MasAnswer. Tabel Rules untuk menyimpan kombinasi dari tabel-tabel tersebut diatas aturan atau konklusi dari permasalahan.

Dibawah ini merupakan rancangan rancangan seluruh tabel yang digunakan dalam sistem ini:

1. Nama : MasSet
- Primary Key : Id
- Foreign Key : -
- Fungsi : Untuk menyimpan set parent

Tabel 3.1 Tabel *MasSet*

NO	Nama Field	Tipe	Lebar	PK	FK		Keterangan
					Tabel	Kolom	
1	Id	Int	4	PK			Id Record
2	Kode	Int	4				Indeks Parent
3	Parent	Int	4				Kode Parent
4	Chapter	Text	50				Teks Parent
5	Next_set	Int	4				Urutan Set

2. Nama : MasQuestion
- Primary Key : Id
- Foreign Key : -
- Fungsi : Untuk menyimpan premise pertanyaan

Tabel 3.2 Tabel *MasQuestion*

NO	Nama Field	Tipe	Lebar	PK	FK		Keterangan
					Tabel	Kolom	
1	Id	Int	4	PK			Id Record
2	Kode	Int	4				Indeks Question
3	Parent	Int	4				Kode Parent
4	Question	Text	50				Teks Parent
5	Note	Text	1000				Teks Keterangan
6	Status	Int	4				Root

3. Nama : MasGoal
- Primary Key : Id
- Foreign Key : -
- Fungsi : Untuk menyimpan goal dari parent

Tabel 3.3 Tabel *MasGoal*

NO	Nama Field	Tipe	Lebar	PK	FK		Keterangan
					Tabel	Kolom	
1	Id	Int	4	PK			Id Record
2	Kode	Int	4				Indeks Goal
3	Goal	Text	50				Teks Goal
3	Repair	Text	1000				Teks Perbaikan
4	Parent	Int	4				Kode Parent
5	Next_set	Int	4				Ke Set Berikutnya

4. Nama : MasAnswer
- Primary Key : Id
- Foreign Key : -
- Fungsi : Untuk menyimpan jawaban dari pertanyaan

Tabel 3.4 Tabel *MasAnswer*

NO	Nama Field	Tipe	Lebar	PK	FK		Keterangan
					Tabel	Kolom	
1	Id	Int	4	PK			Id Record
2	Kode	Int	4				Urutan Pertanyaan
3	Parent	Int	4				Kode Parent
4	Answer	Text	20				Teks Pertanyaan

5. Nama : MasRules
- Primary Key : Id
- Foreign Key : -
- Fungsi : Untuk menyimpan kombinasi konklusi

Tabel 3.5 Tabel *MasRules*

NO	Nama Field	Type	Lebar	PK	FK		Keterangan
					Tabel	Kolom	
1	Id	Int	4	PK			Id Record
2	Parent	Int	4				Kode Parent
3	Cond1...n	Text	20				Kondisi 1 sampai n
4	Result	Text	20				Hasil Diagnosa

6. Nama : MasUser
- Primary Key : Id
- Foreign Key : -
- Fungsi : Untuk menyimpan user dan password admin

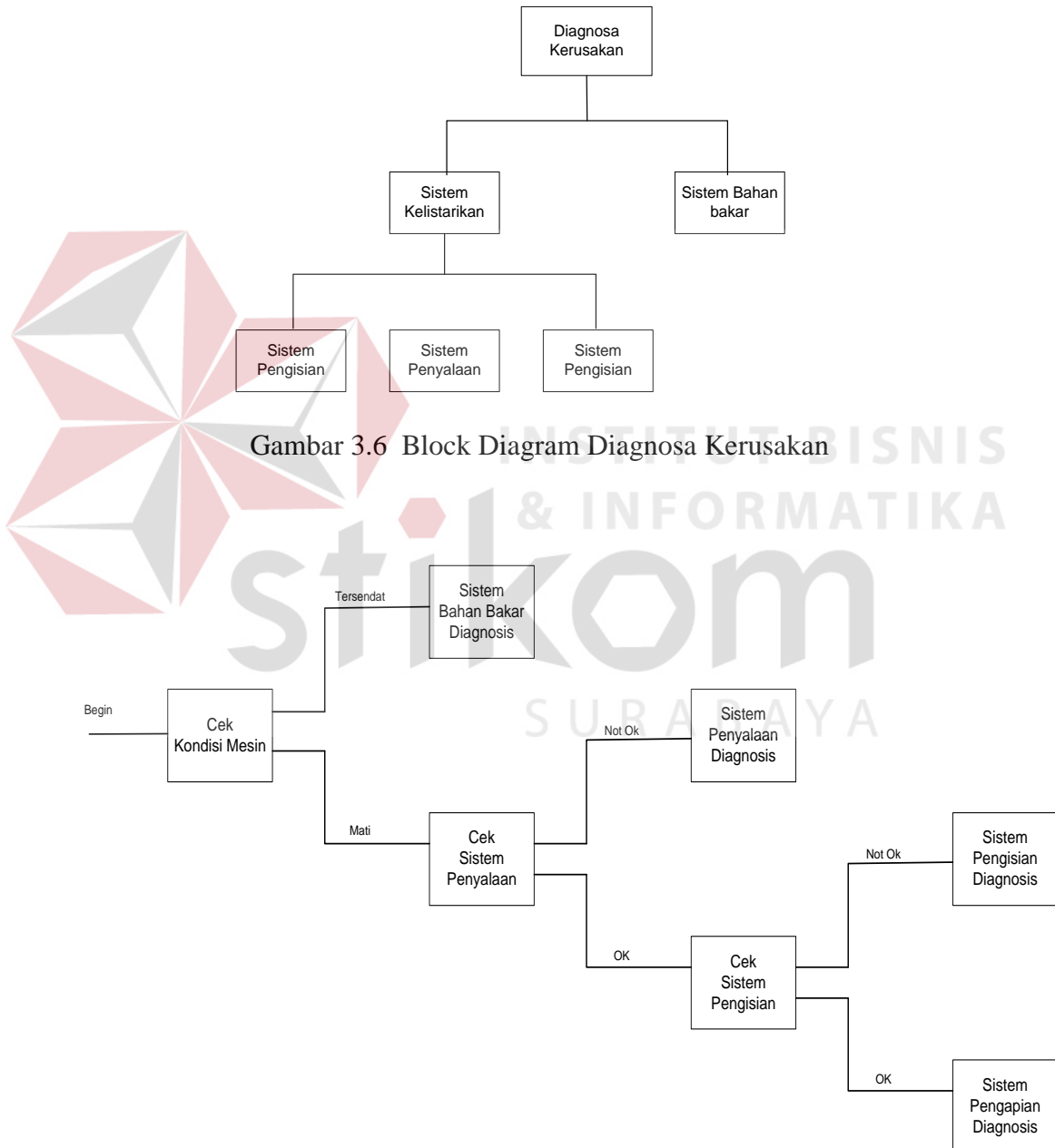
Tabel 3.6 Tabel *MasUser*

NO	Nama Field	Type	Lebar	PK	FK		Keterangan
					Tabel	Kolom	
1	Id	Int	4	PK			Id Record
2	Username	Text	20				Nama User
3	Password	Text	20				Password User

3.4 Perancangan Rule Base

3.4.1 Block Diagram

Block Diagram diperlukan untuk mengetahui urutan kerja sistem dalam menemukan suatu diagnosa kerusakan. Gambar 3.6 merupakan gambar block diagram diagnosa, sedangkan gambar 3.7 merupakan detil dari block diagram.

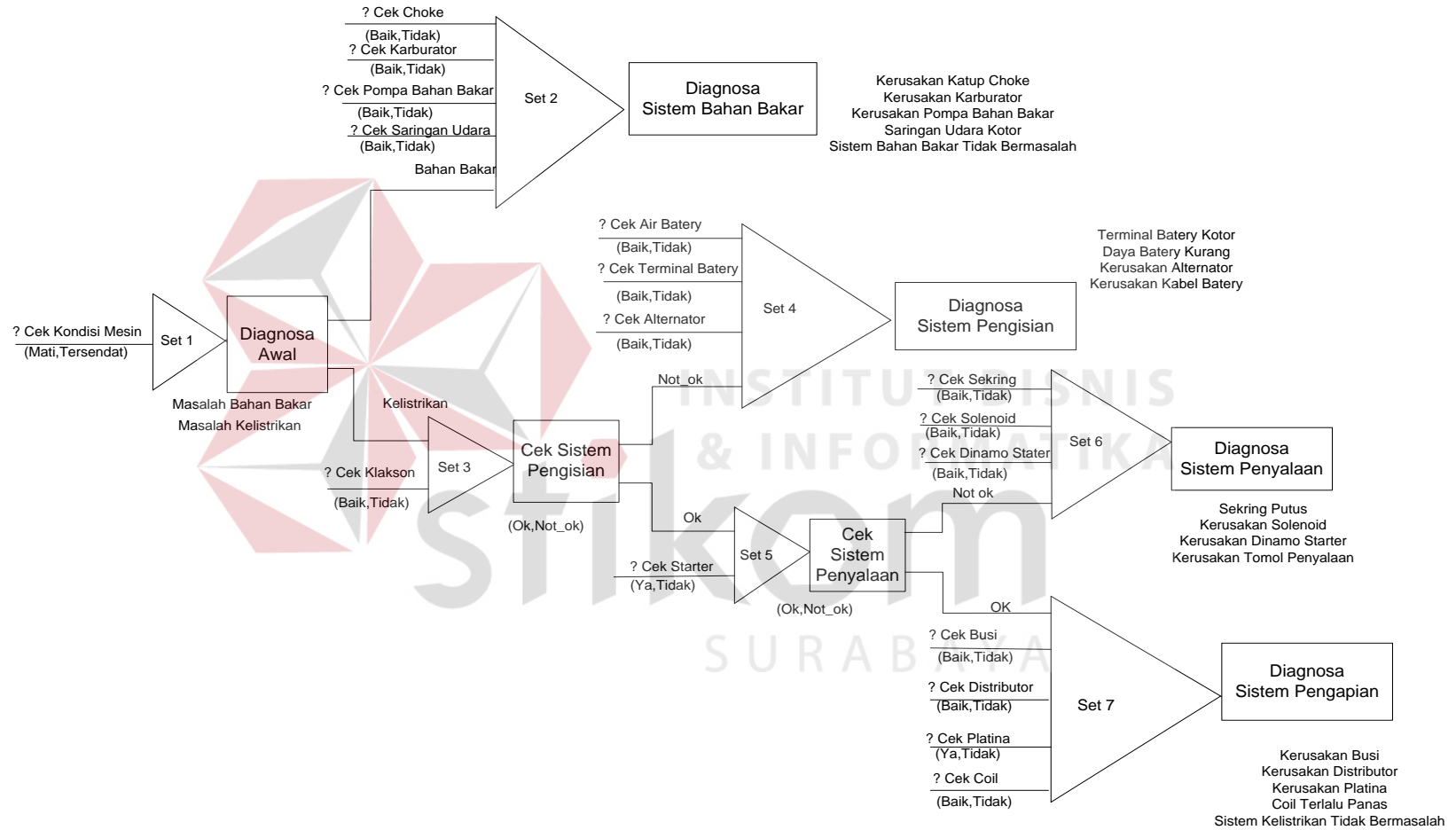


Gambar 3.7 Detail Block Diagram Diagnosa Kerusakan

3.4.2 Diagram Ketergantungan

Dari *block digram* seperti diatas, selanjutnya dapat dibuat diagram ketergantungan (*dependency diagram*) yang menunjukkan hubungan atau ketergantungan antara input jawaban, aturan-aturan (*rule*), nilai-nilai dan direkomendasikan untuk menjadi berbasis pengetahuan.[Dologite G.G,1993]. Gambar 3.8 berikut ini merupakan dependency diagram dari diagnosa kerusakan mobil.





Gambar 3.8 Dependency Diagram Diagnosa Kerusakan

3.4.3 Perancangan *Decision Table*

Decision Table dibuat untuk menunjukkan antar hubungan nilai-nilai pada hasil fase antara atau rekomendasi akhir *knowledge base system*. Pada Table 3.7 berikut menunjukkan salah satu contoh perancangan *decision table* untuk *rule set 4* yaitu parameter pengisian yang berdasarkan pada perancangan *dependency diagram*.

Tabel 3.7 Reduksi *Decision Table Rule Set 4*

Step 1 : *Plan*

Kondisi :	Air Batery Habis	(Ya/Tidak)	= 2
	Terminal Batery Kotor	(Ya/Tidak)	= 2
Baris : 2 x 2 = 4			

Step 2 : *Completed Decision Table*

Rule	Air Batery Habis	Terminal Batery Kotor	Kerusakan
A 1	Y	Y	Daya Batery Kurang
A 2	Y	T	Daya Batery Kurang
A 3	T	Y	Terminal Batery Kotor
A 4	T	T	Tdk Ada Kerusakan Sist Pengisian

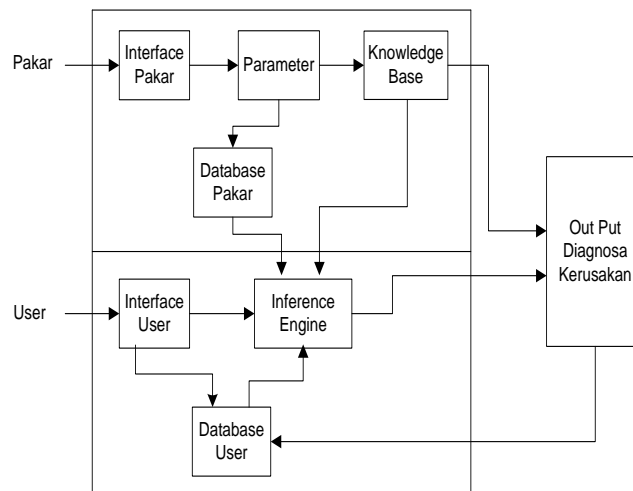
Step 3 : *Reduced Decision Table*

Rule	Air Batery Habis	Terminal Batery Kotor	Kerusakan
B 1	Y	-	Daya Batery Kurang
B 2	T	Y	Terminal Batery kotor
B 3	T	T	Tdk Ada Kerusakan Sist Pengisian

Dalam Tabel 3.8 rencana *decision table* adalah untuk rangkaian aturan akhir kondisi yang saling terkait, dimana yang masing-masing dapat memiliki sejumlah nilai berbeda. Air Batery Habis, kondisi pertama hanya memiliki dua nilai yaitu Ya atau Tidak. Begitu pula Terminal Batery Kotor, kondisi kedua juga memiliki dua nilai yaitu Ya atau Tidak.

3.4.4 Perancangan Desain Arsitektur.

Desain Arsitektur seperti terlihat pada Gambar 3.9 menggambarkan hubungan antara elemen-elemen utama.



Gambar 3.9 Desain Arsitektur

Berikut adalah keterangan dari desain arsitektur

1. *Interface Pakar* : media yang digunakan oleh pakar untuk mengembangkan sistem dan untuk berhubungan dengan sistem
2. *Parameter* : parameter koneksi untuk manipulasi *knowledge base* dan database pakar.
3. *Knowledge base* : basis pengetahuan yang terdiri dari fakta dan aturan.
4. *Database Pakar* : digunakan untuk mengembangkan basis pengetahuan apabila pakar ingin menambah, mengedit, atau menghapus rule.
5. *Interface User* : sebagai media oleh user untuk melihat dan berkonsultasi dengan sistem.
6. *Inference Engine* : mekanisme inferensi yang digunakan adalah *Backward Chaining* yaitu memunculkan salah satu *goal* kemudian dicari atau dibuktikan kondisi yang menyusun *goal* tersebut.

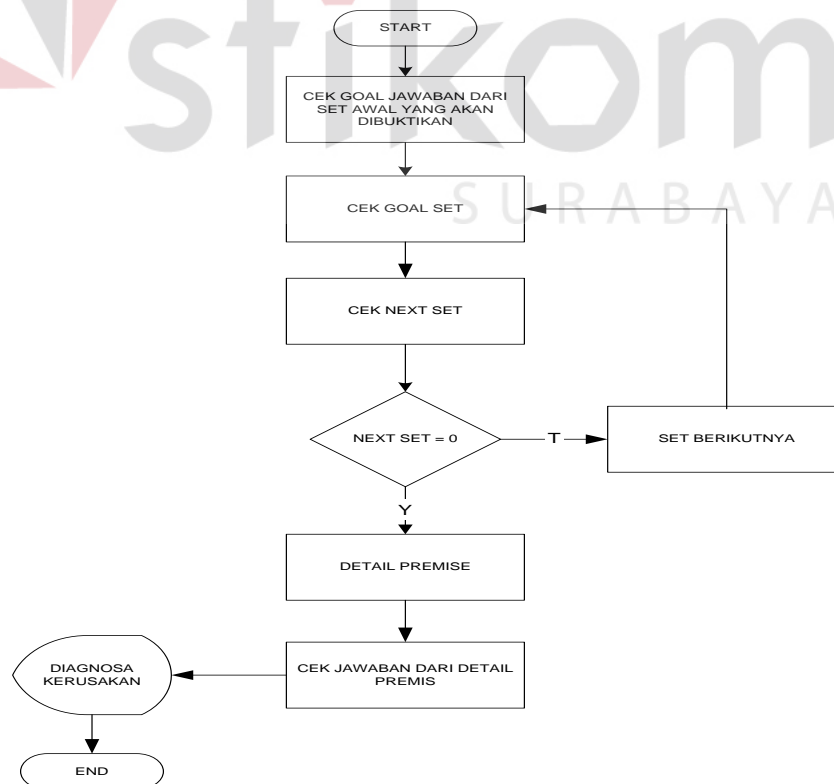
7. Output : output dari program akan menunjukkan hasil dari konsultasi, yaitu hasil diagnosa pembuktian yang dilakukan.

3.5 Perancangan Proses

Berdasar dari teori-teori yang telah dijelaskan diatas, dapat dilakukan perancangan untuk desain proses yang akan berjalas. Perancangan proses dalam sistem ini adalah sebagai berikut :

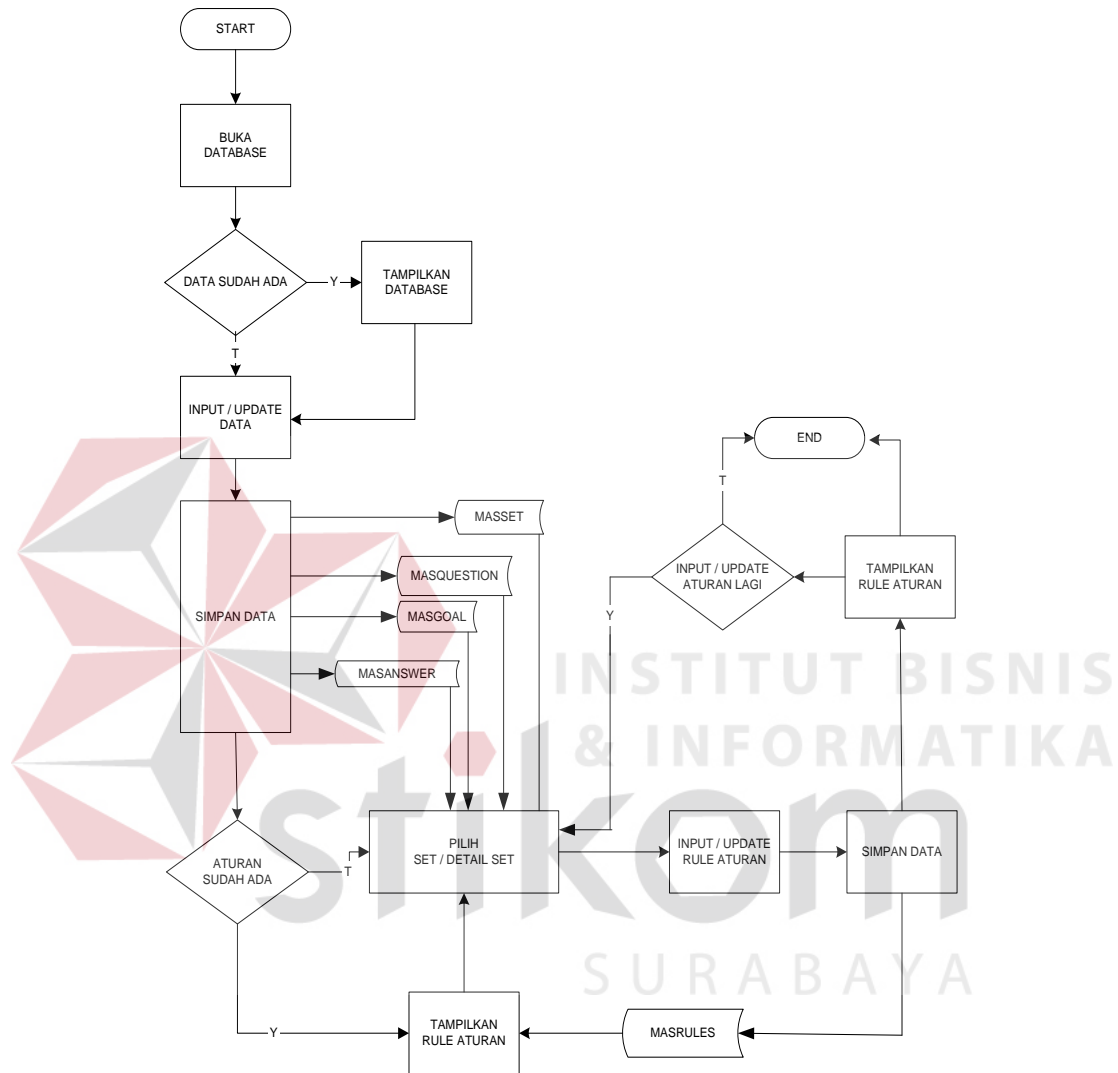
1. Diagram Alir untuk Proses *Inference Engine*.
2. Diagram Alir untuk Desain *Expert*.
3. Diagram Alir untuk Desain Konsultasi *User*.

Gambar 3.10 menjelaskan proses *inference engine* dari menggunakan metode *backward chaining* yaitu penelusuran data yang dimulai dari goal diagnosa awal yang akan dibuktikan kebenarannya.



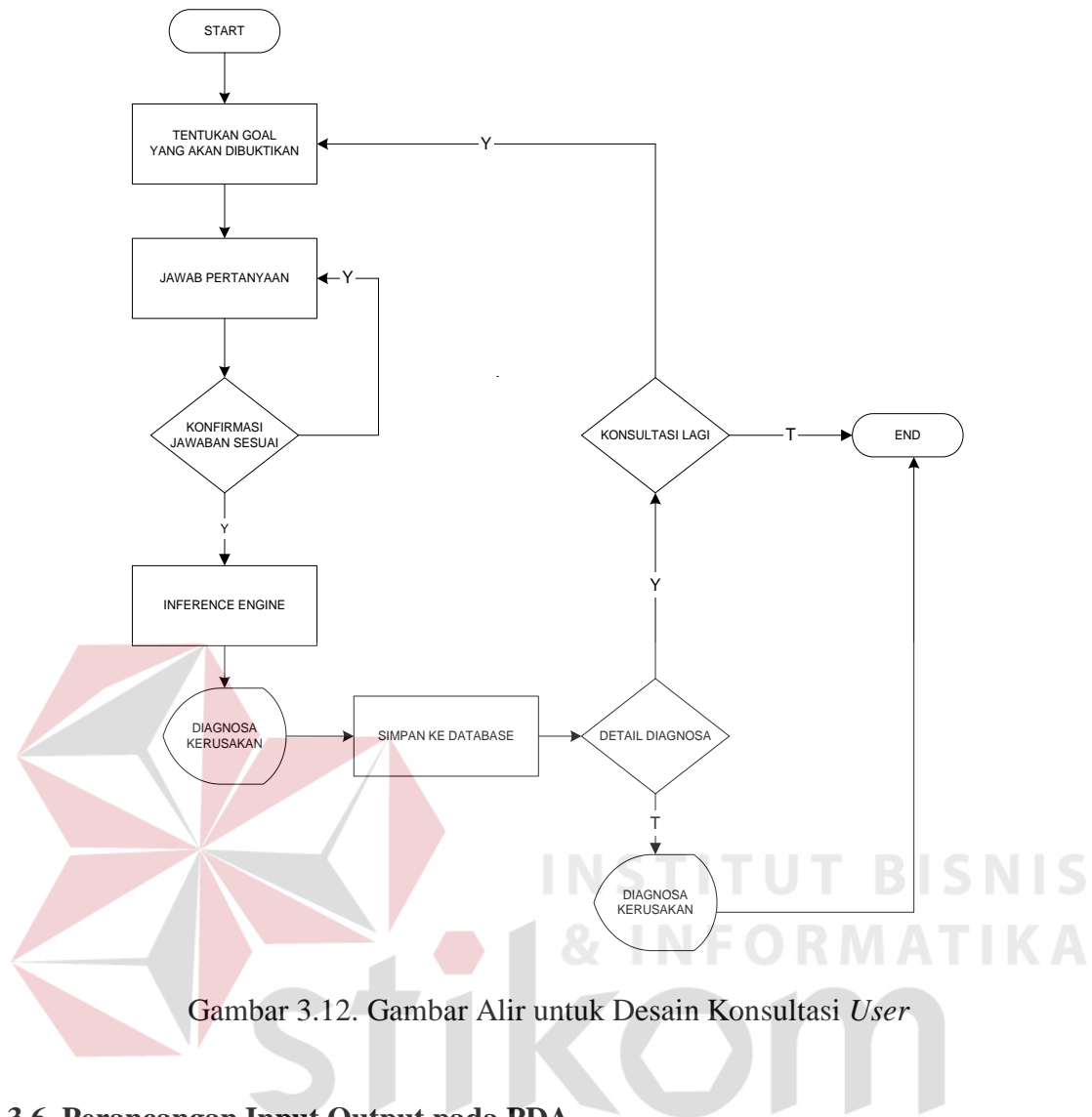
Gambar 3.10 Diagram Alir Proses *Inference Engine*

Kemudian disusun diagram alir untuk maintenance pakar untuk mendefinisikan aturan-aturan yang akan digunakan, seperti yang terlihat pada gambar 3.11 berikut ini



Gambar 3.11 Diagram Alir Proses Desain *Expert*

Sedangkan gambar 3.12 menjelaskan proses jalannya rule aturan yang berdasarkan pilihan jawaban dari user.



Gambar 3.12. Gambar Alir untuk Desain Konsultasi *User*

3.6 Perancangan Input Output pada PDA

Adapun perancangan input output front office ini yang nantinya akan diimplementasikan pada PDA adalah sebagai berikut:

3.6.1 Rancangan *Interface Login*

The screenshot shows a login window titled "Aplikasi Diagnosa Kerusakan Mobil". It contains two main sections: "Authorized Option" and "Owner Password".

Authorized Option: This section has two radio buttons: "Guest" and "Owner".

Owner Password: This section contains two text input fields: "Owner Name" and "Password". Below these fields are three buttons: "Setting Initial", "Login", and "Cancel".

At the bottom right of the window is an "Exit" button.

Gambar 3.13 *Interface Login*

Interfacae Login digunakan untuk verifikasi apakah pengguna merupakan admin atau bukan. User admin (*owner*) dapat melakukan *setting* keseluruhan atau *update*, jika bukan hanya bisa menggunakan aplikasi untuk user saja. Seperti yang terlihat pada gambar 3.13

3.6.2 Rancangan *Interface Maintenance Sets*

A. Rancangan *Interface Tree of Sets*

The screenshot shows a window titled "Maintenance Sets". It is divided into two main sections: "Tree of Sets" and "Control of Set".

Tree of Sets: This section contains a list of "Sample Node" entries, with a tree structure indicated by dots and lines.

Control of Set: This section contains four buttons: "Add", "Remove", "Edit", and "Refresh".

At the bottom of the window are three buttons: "TreeQuestions", "Detail Set", and "Close".

Gambar 3.14 Rancangan *Interface Tree of Sets*

Pada gambar 3.14 Rancangan *Interface Tree of Sets* digunakan untuk *maintenance set* inti dari pakar yang nantinya akan digunakan. Set tersebut digambarkan dengan menggunakan *control tree view*, untuk mempermudah gambaran mengenai aturan yang nantinya akan dibuat.

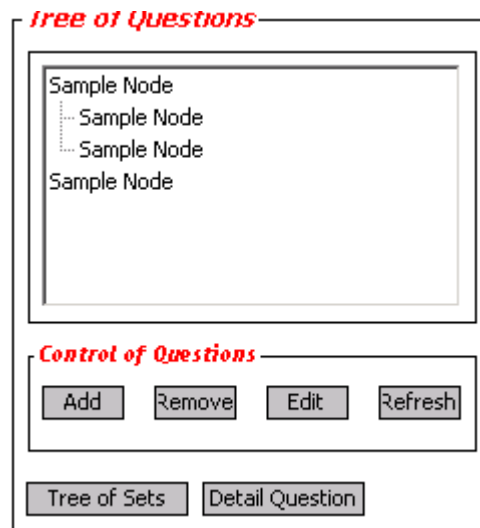
B. Rancangan *Interface Detail of Set*



Gambar 3.15 Rancangan *Interface Detail of Set*

Pada gambar 3.15 *detail set* merupakan detail dari set inti, dimana digunakan untuk menambah atau menghapus *goal* dari masing-masing rule inti. Goal tersebut merupakan hasil dari pembuktian diagnosa.

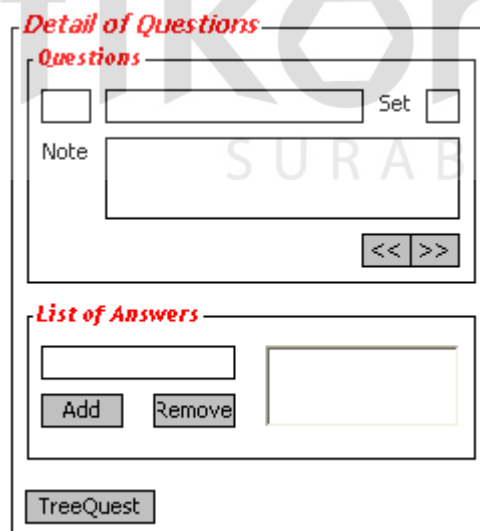
C. Rancangan *Interface Tree of Questions*



Gambar 3.16 Rancangan *Interface Tree of Questions*

Interface Tree of Questions digunakan menggambarkan *premise* (isi) dari set inti, yang nantinya akan digunakan sebagai pertanyaan untuk user. *Interface Tree of Questions* dapat dilihat pada gambar 3.16.

D. Rancangan *Interface Detail of Questions*



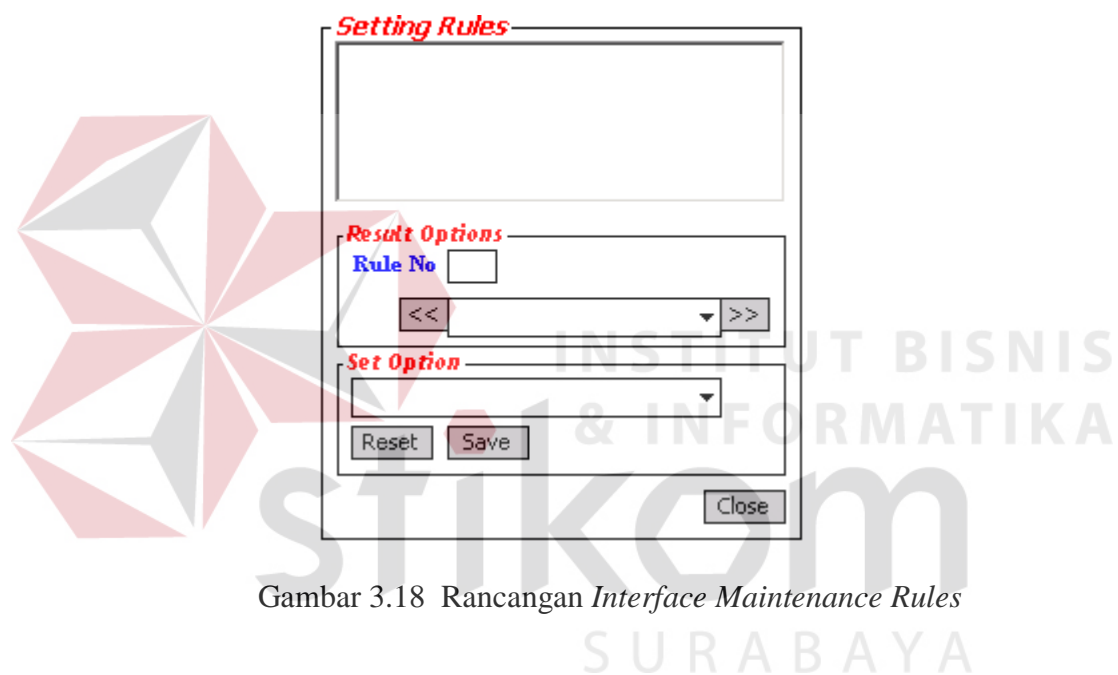
Gambar 3.17 Rancangan *Interface Detail of Questions*

Interface Detail Questions merupakan detail dari masing-masing *premise*, dimana terdapat *note* (catatan) yaitu untuk memberi keterangan kepada user mengenai

premise yang ditanyakan, juga terdapat *list of answers* yaitu untuk menambah atau menghapus jawaban dari premise. *Interface Detail Questions* dapat dilihat pada gambar 3.17.

3.6.3 Rancangan *Interface Maintenance Rules*

Pada *Interface Maintenance Rules* akan digunakan untuk membuat aturan-aturan tertentu, yang dijadikan dasar untuk menemukan diagnosa. Bentuk interfacenya seperti pada gambar 3.18



Gambar 3.18 Rancangan *Interface Maintenance Rules*

3.6.4 Rancangan *Interface Consulting*

Merupakan antarmuka yang menghubungkan user dengan pakar. Halaman ini berbentuk konsultasi, dimana pakar akan memberi pertanyaan dan harus dijawab oleh user. *Interface Consulting* dapat dilihat pada gambar 3.19.

Consulting

Question

Answer

List of Questions

Gambar 3.19 Rancangan *Interface Consulting*

