

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang analisis sistem dan langkah-langkah dalam perancangan sistem. Hal-hal yang dibahas dalam bab ini adalah: analisis system yang berupa identifikasi masalah dan analisis masalah, perancangan sistem dan perancangan evaluasi sistem yang berupa desain uji coba *form*.

3.1 Analisis Sistem

3.1.1 Identifikasi Masalah

Menurut Kasubdit Tanaman Teh dan Kopi Direktorat Budidaya Tanaman Rempah dan Penyegar, Direktorat Jenderal Perkebunan Departemen Pertanian, Nyoman Sudarsana, tingkat produktivitas kopi di Indonesia masih rendah, yakni rata-rata sebesar 700 Kg/ha/tahun, atau baru mencapai 60% dari potensi produktivitasnya. Rendahnya tingkat produktivitas dan produksi kopi karena 96% diusahakan oleh perkebunan rakyat. Bila dibandingkan dengan Negara produsen utama kopi dunia lainnya tingkat produktivitas kopinya rata-rata di atas 1000 Kg/hektar/tahun. Penyebabnya adalah karena rendahnya kualitas kopi yang sebagian besar dihasilkan oleh perkebunan rakyat.

Salah satu faktor yang menyebabkan rendahnya kualitas kopi di Indonesia adalah karena perkebunan kopi di Indonesia selalu diancam oleh hama dan penyakit tumbuhan. Pada tahun 1885 perkembangan perkebunan kopi di Indonesia berhenti akibat penyakit karat daun dan antara tahun 1986 dan 1990 produksi kopi merosot menjadi 25% dari semula, salah satu penyebabnya adalah akibat terserang hama dan penyakit (Sukamto, 1998).

Kurangnya informasi yang diketahui oleh pihak perkebunan kopi tentang jenis penyakit yang menyerang tanaman kopi, menyebabkan banyak tanaman kopi yang tidak tertangani dengan benar. Hal ini mengakibatkan banyak tanaman kopi yang seharusnya bisa terselamatkan menjadi mati atau menjadikan kualitas kopi tersebut menurun. Jika hal ini dibiarkan terus-menerus, maka akan berimbas pada tingkat produktifitas tanaman kopi tersebut. Timbul masalah bagaimana supaya pihak perkebunan kopi dapat mengetahui jenis penyakit yang menyerang tanaman kopi mereka dan mengambil tindakan serta melakukan pencegahan yang tepat tanpa adanya seorang ahli.

Berdasarkan permasalahan di atas, pihak perkebunan kopi membutuhkan sebuah alat bantu yang dapat memberikan informasi mengenai penyakit yang menyerang tanaman kopi dan memberikan solusi untuk menangani penyakit tersebut. Alat bantu tersebut dapat berupa sebuah aplikasi sistem pakar, dengan harapan aplikasi sistem pakar ini dapat membantu memberikan kesimpulan berupa hasil diagnosis jenis penyakit yang menyerang tanaman kopi dan membantu memberikan solusi untuk mengambil sebuah tindakan yang tepat dalam menangani tanaman kopi yang terserang penyakit.

3.1.2 Analisis Masalah

Dalam pembuatan aplikasi sistem pakar untuk mendiagnosis penyakit pada tanaman kopi, diperlukan langkah awal, yaitu menganalisis masalah. Pada bagian ini, akan diuraikan kebutuhan dasar sistem dalam membuat aplikasi sistem pakar ini agar dapat membentuk suatu sistem berbasis aturan yang memenuhi persyaratannya.

Langkah pertama yang harus dilakukan adalah membuat *dependency diagram*. Dalam aplikasi sistem pakar ini, *dependency diagram* dibuat atau direpresentasikan dalam bentuk *tree view*. *Dependency diagram* berguna untuk menggambarkan susunan parameter yang ada. Di dalam sebuah *dependency diagram* minimal terdapat dua parameter (setiap parameter tidak boleh memiliki nama yang sama) dan sebuah parameter minimal memiliki dua *possible value* yang berbeda. Parameter yang memiliki cabang dibawahnya secara otomatis disebut sebagai set. Pengguna harus memasukkan sebuah pertanyaan pada setiap parameter yang bukan set (parameter yang tidak memiliki cabang dibawahnya). Pertanyaan ini nantinya akan ditampilkan kepada pengguna yang melakukan konsultasi untuk memperoleh hasil diagnosis penyakit pada tanaman kopi.

Setelah pembuatan *dependency diagram* selesai dilakukan, dilanjutkan dengan mengisi *decision table*. Setiap set dalam *dependency diagram* memiliki *decision table* tersendiri. *Decision table* pada aplikasi ini akan dibuat otomatis oleh sistem berdasarkan set yang telah dipilih oleh pengguna. Jumlah baris pada setiap *decision table* diperoleh dengan cara mengalikan jumlah *possible value* dari setiap parameter yang berada dalam satu set yang dipilih pengguna (lihat Tabel 2.1, Step 1).

Isi dari *decision table* kemudian oleh sistem dibangkitkan menjadi beberapa *rule*. Karena proses kombinasi *possible value* pada *decision table* dilakukan oleh sistem, maka tidak mungkin terjadi *redundant rules*, *subsumed rules* dan *circular rules*. Setiap baris pada *decision table* hanya bisa diisi dengan satu *conclusion* atau kesimpulan, sehingga tidak mungkin terjadi *conflicting rules*. Kesalahan yang mungkin terjadi adalah *unnecessary if condition*. Namun hal ini

dapat diatasi dengan adanya proses *reduction decision table* yang ada pada setiap pengisian *decision table*. Proses *reduction* ini secara otomatis mencari dan menghapus *premise* atau kondisi yang tidak perlu dikondisikan. Jadi *rules* hasil *generate* dari aplikasi ini akan bebas dari kesalahan yang ada.

Setelah proses membuat *rule*, dilanjutkan dengan mengisi solusi atau penjelasan dari setiap kesimpulan akhir yang mungkin. Kesimpulan akhir ini diambil dari *possible value* yang dimiliki oleh parameter paling atas pada *tree view* atau parameter yang paling ujung dari sebuah *dependency diagram*. Dan pada aplikasi sistem pakar ini, kesimpulan akhirnya adalah jenis penyakit tanaman kopi dan penjelasan yang diisi adalah penyebab dari penyakit tersebut, gejala-gejalanya dan cara pengendaliannya.

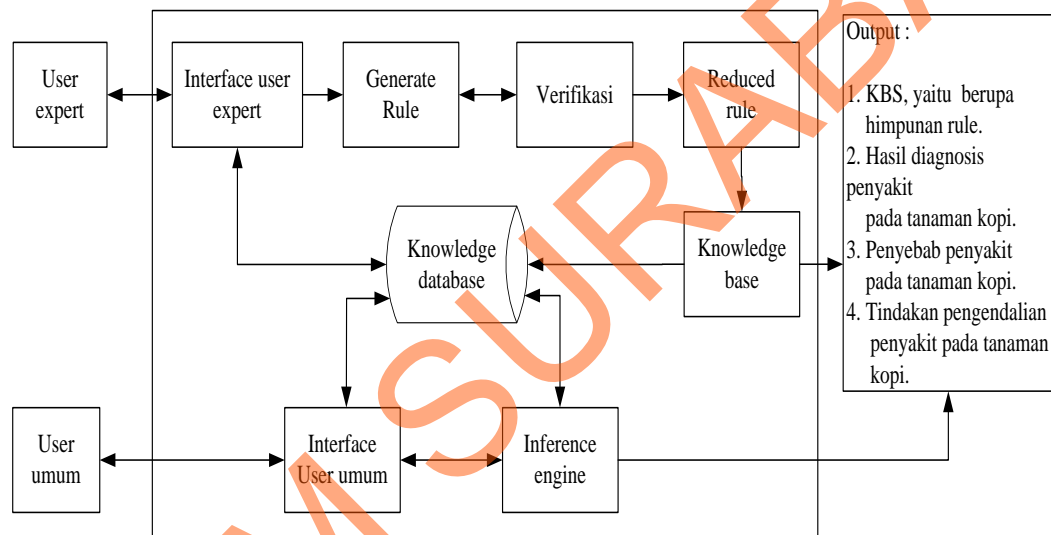
Proses selanjutnya adalah pengguna dapat melakukan konsultasi untuk mendapatkan hasil akhir atau output dari aplikasi sistem pakar ini. Untuk melakukan konsultasi, pengguna harus memilih terlebih dahulu bagian dari tanaman kopi yang tampak gejalanya, misalnya Daun, maka sistem akan menampilkan secara otomatis semua gejala-gejala yang terlihat pada bagian daun. Selanjutnya adalah proses diagnosis penyakit untuk menampilkan hasil akhir, yaitu hasil diagnosis penyakit, sesuai dengan gejala yang telah dipilih.

Aplikasi sistem pakar untuk mendiagnosis penyakit pada tanaman kopi ini berbasis desktop dan memiliki dua pengguna, yaitu *user expert* dan *user umum*. *User expert* dalam aplikasi sistem pakar ini adalah seorang pakar atau memiliki kompetensi dibidang perkebunan, khususnya perkebunan kopi.

3.2 Perancangan Sistem

3.2.1 Desain arsitektur

Desain arsitektur aplikasi sistem pakar ini terdiri dari dua pengguna, yaitu *user expert* dan *user umum*. Desain arsitektur dari sisi *user expert* tidak dapat dipisahkan dari sisi *user umum*, karena semua elemen–elemen yang ada saling berhubungan. Desain arsitektur aplikasi sistem pakar ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain Arsitektur Diagnosis Penyakit Pada Tanaman Kopi

Penjelasan dari Gambar 3.1 adalah sebagai berikut:

1. *Interface user expert*: suatu media yang digunakan oleh *user expert* (pakar) untuk memasukkan parameter *rules*. Parameter yang dimasukan adalah tentang jenis penyakit, gejala-gejala penyakit, penyebab penyakit dan tindakan pengendalian penyakit pada tanaman kopi untuk menghasilkan basis pengetahuan (*knowledge base*).
2. *Generate rule*: proses ini dilakukan untuk membangkitkan *rules*.

3. Verifikasi: yaitu melakukan proses verifikasi pada *rules* yang dibangun dan dimasukan ke dalam sistem dengan tujuan untuk mendapatkan *rules* yang benar sesuai dengan teori verifikasi.
4. Reduced Rule: *rules* yang telah diverifikasi.
5. *Knowledge base*: merupakan kumpulan pengetahuan dan *rules* yang diperoleh sistem selama proses berlangsung.
6. *Knowledge database*: digunakan untuk mengembangkan *knowledge base* apabila ada perubahan pada *rules* yang ada.
7. *Interface user* umum: digunakan oleh *user* umum untuk berinteraksi dengan sistem, dengan memasukkan fakta-fakta untuk mendapatkan suatu kesimpulan.
8. *Inference engine*: mekanisme inferensi yang digunakan adalah metode runut maju (*forward chaining*). Metode ini akan memproses setiap masukan dari *user* umum untuk mendapatkan suatu diagnosis penyakit pada tanaman kopi dan memberikan solusi dalam pengendaliannya sesuai dengan *knowledge base* yang telah dibuat oleh *user expert*.
9. *Output*: hasil yang didapat dari sistem yang dapat menunjukkan jawaban dari fakta-fakta yang telah dimasukan. *Output* yang dihasilkan adalah himpunan *rule*, hasil diagnosis penyakit, penyebab penyakit dan tindakan pengendalian penyakit pada tanaman kopi.

3.2.2 Perancangan Sistem Pakar

Dalam melakukan perancangan sistem pakar ini, ada beberapa tahap yang harus dilakukan agar aplikasi sistem pakar yang dibuat dapat berfungsi sesuai dengan yang diharapkan. Adapun tahap-tahap dalam perancangan sistem

pakar adalah perancangan *block diagram*, *dependency diagram*, *decision table*, dan *reduced decision table*.

A Perancangan *Block Diagram*

Block diagram diperlukan untuk mengetahui urutan kerja sistem dalam mencari suatu keputusan. Dalam aplikasi sistem pakar ini terdapat sebuah *block diagram*, yaitu *block diagram* untuk mendiagnosis penyakit pada tanaman kopi.

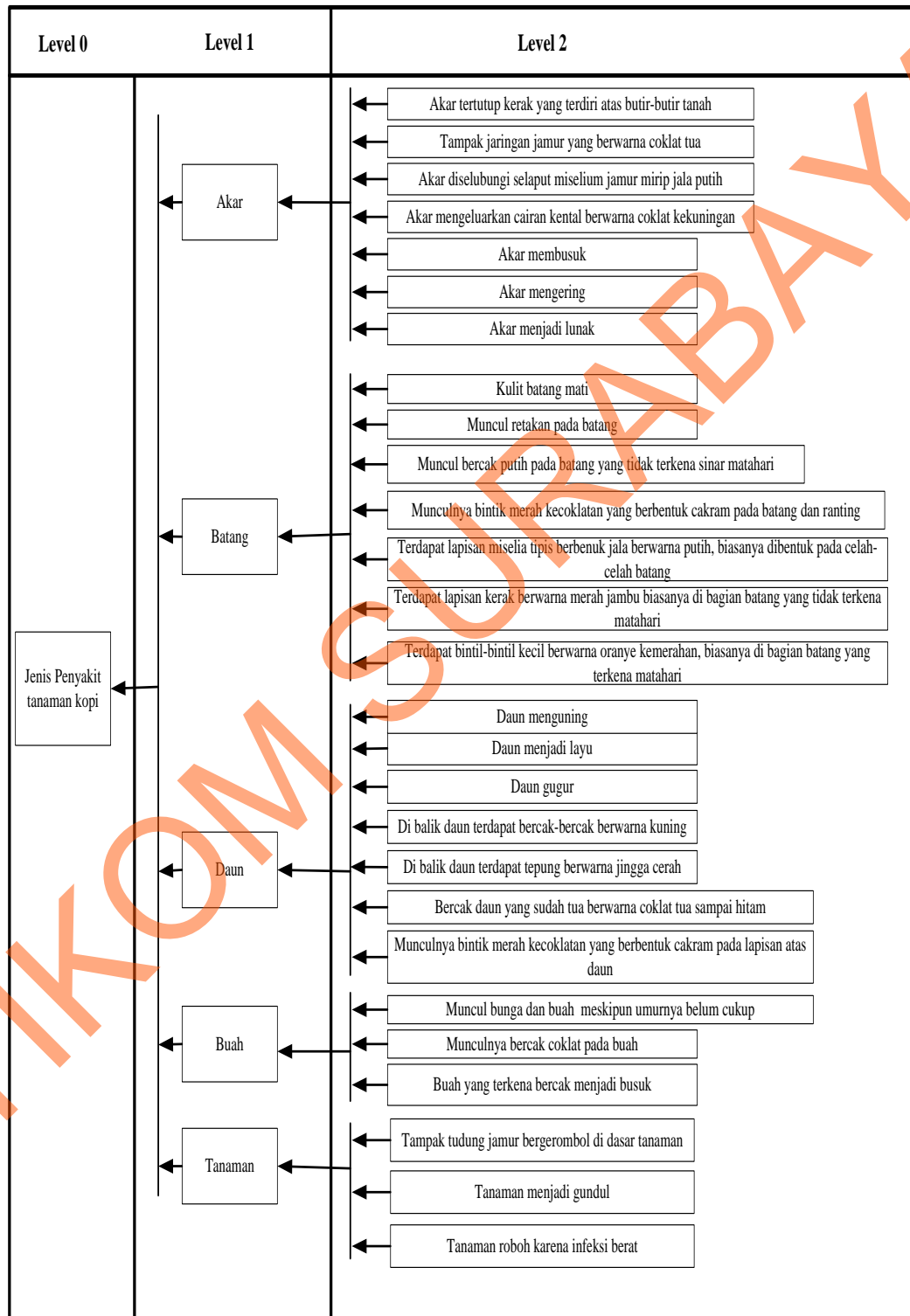
Block diagram ini terdiri dari tiga level, yaitu level 0, level 1, dan level 2. Level 0 berisi hasil akhir berupa jenis penyakit yang menyerang tanaman kopi. Pada level 1 terdapat 5 parameter, yaitu akar, batang, daun, buah, dan tanaman. Terakhir pada level 2 dijelaskan sub parameter dari masing-masing parameter berupa gejala-gejala penyakit yang menyerang tanaman kopi. *Block diagram* untuk diagnosis penyakit pada tanaman kopi dapat dilihat pada Gambar 3.2.

B Perancangan *Dependency Diagram*

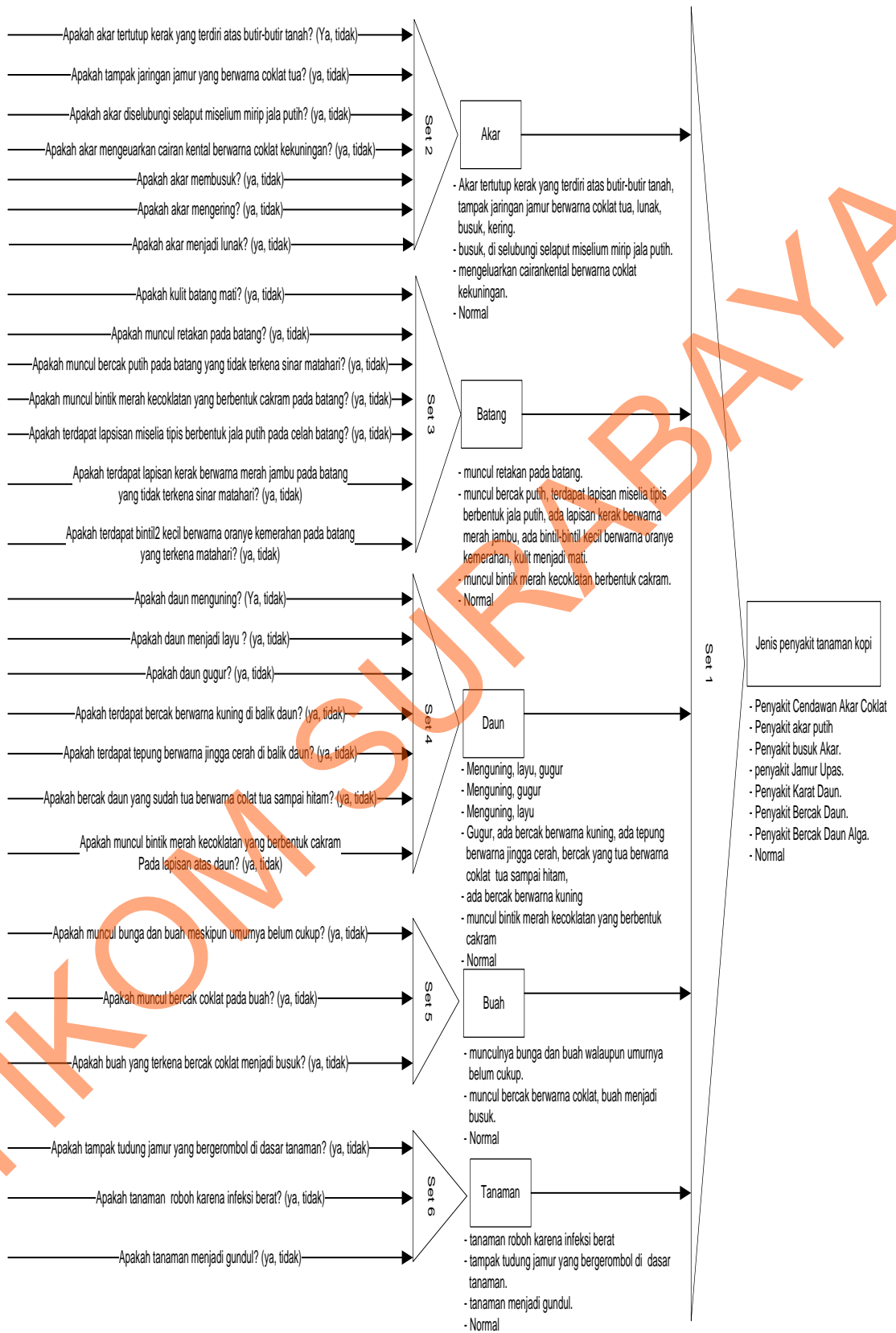
Dependency diagram menunjukkan hubungan atau ketergantungan antara inputan pertanyaan, *rules*, nilai dan rekomendasi yang dibuat oleh *prototype Knowledge Based System (KBS)*. *Dependency diagram* untuk aplikasi sistem pakar ini dapat dilihat pada Gambar 3.3.

Dependency diagram pada Gambar 3.3 memiliki enam set. Set 1 yang merupakan konklusi akhir yang merupakan jenis penyakit dari tanaman kopi. Jenis penyakit ini adalah penyakit cendawan akar coklat, penyakit akar putih, penyakit busuk akar, penyakit jamur upas, penyakit karat daun, penyakit bercak daun dan penyakit bercak daun alga. Set 2 merupakan set dari parameter akar, set 3 merupakan set dari parameter batang, set 4 merupakan set dari parameter daun,

set 5 merupakan set dari parameter buah dan set 6 merupakan set dari parameter tanaman.



Gambar 3.2 Block Diagram Diagnosis Penyakit Pada Tanaman Kopi



Gambar 3.3 Dependency Diagram Diagnosis Penyakit Pada Tanaman Kopi

C Perancangan *Decision Table*

Decision table diperlukan untuk menunjukkan hubungan timbal balik antara nilai-nilai pada hasil fase atau rekomendasi akhir *knowledge based system* (KBS). Tabel 3.1 menunjukkan salah satu perancangan *decision table* aplikasi sistem pakar untuk mendiagnosis penyakit pada tanaman kopi dengan *set* 5, yaitu *set* parameter buah berdasarkan pada perancangan *dependency diagram*.

Tabel 3.1 adalah rangkaian aturan akhir yang terkait dengan tiga kondisi. Masing-masing kondisi dapat memiliki sejumlah nilai yang berbeda. Munculnya bunga dan buah meskipun umurnya belum cukup adalah kondisi pertama yang memiliki dua nilai, yaitu ya dan tidak. Munculnya bercak coklat pada buah adalah kondisi kedua yang memiliki dua nilai, yaitu ya dan tidak. Buah yang terkena bercak menjadi busuk adalah kondisi ketiga yang memiliki dua nilai, yaitu ya dan tidak.

D Perancangan *Reduced Decision Table*

Reduced decision table adalah pembuatan *table* yang nilai-nilainya didapat dari mereduksi *decision table*. Setelah didapatkan nilai dari *decision table*, nilai tersebut akan direduksi untuk mendapatkan nilai dari kondisi terakhir. Pada sistem ini perancangan *reduced decision table* untuk setiap *decision table* dilakukan secara manual. Perancangan *reduced decision table* berdasarkan *decision table* pada Tabel 3.1 menghasilkan parameter seperti yang dapat dilihat pada Tabel 3.2.

Tabel 3.1 *Decision Table Rule Set 5*

Step 1: Plan

Kondisi	Munculnya bunga dan buah meskipun umurnya belum cukup (ya, tidak)	2
	Munculnya bercak coklat pada buah (ya, tidak)	2
	Buah yang terkena bercak menjadi busuk (ya, tidak)	2
Baris	$2 \times 2 \times 2 = 8$	

Step 2: Completed decision table

No	berbuah meskipun umurnya belum cukup (A)	Munculnya bercak coklat pada buah (B)	Buah yang terkena bercak menjadi busuk (C)	Buah
1	Ya	Ya	Ya	Tidak teridentifikasi
2	Ya	Ya	Tidak	Tidak teridentifikasi
3	Ya	Tidak	Ya	Tidak teridentifikasi
4	Ya	Tidak	Tidak	A
5	Tidak	Ya	Ya	B, C
6	Tidak	Ya	Tidak	Tidak teridentifikasi
7	Tidak	Tidak	Ya	Tidak teridentifikasi
8	Tidak	Tidak	Tidak	Normal

Tabel 3.2 *Reduced Decision Table Rule Set 5*

No	berbuah meskipun umurnya belum cukup (A)	Munculnya bercak coklat pada buah (B)	Buah yang terkena bercak menjadi busuk (C)	Buah
1	Ya	Tidak	Tidak	A
2	Tidak	Ya	Ya	B, C
3	Tidak	Tidak	Tidak	Normal

3.2.3 Perancangan Diagram Alir

Diagram alir merupakan gambaran suatu data proses dan hubungan antara proses satu dengan yang lainnya dalam suatu sistem komputer. Perancangan diagram alir dalam aplikasi sistem pakar ini ada 4, yaitu: diagram alir sistem untuk admin, diagram alir sistem untuk pakar, diagram alir sistem untuk *user* dan diagram alir sistem untuk proses *inference engine*.

A Diagram Alir Sistem Untuk Admin

Diagram alir sistem untuk admin ini menggambarkan proses-proses yang dilakukan oleh seorang admin. Proses untuk maintenance data *user* pakar dan membantu pakar dalam memasukkan parameter-parameter yang sesuai dengan permasalahan, seperti gejala penyakit dan solusinya. Sumber data yang diperoleh dalam pengisian parameter dan solusinya diambil dari beberapa buku (telah dibahas dalam batasan masalah pada bab I) dan juga informasi dari pakar. Diagram alir sistem untuk admin ini dapat dilihat pada Gambar 3.4.

B Diagram Alir Sistem Untuk Pakar

Diagram alir sistem untuk pakar ini menggambarkan proses-proses yang dilakukan oleh pakar. Proses untuk memasukkan setiap parameter yang sesuai dengan permasalahan, seperti gejala-gejala penyakit, jenis-jenis penyakit dan cara pengendaliannya dan membuat *rule*. Sumber data yang diperoleh dalam pengisian parameter dan solusinya diambil dari beberapa buku dan juga pengetahuan yang dimiliki oleh pakar itu sendiri. Diagram alir sistem untuk pakar dapat dilihat pada Gambar 3.5.

C Diagram Alir Sistem Untuk *User*

Diagram alir sistem untuk *user* ini menggambarkan jalannya sistem pada desain *user* dalam melakukan konsultasi sampai mendapatkan hasil diagnosis penyakit dan tindakan pengendalian pada tanaman kopi. Diagram alir untuk *user* dapat dilihat pada Gambar 3.6.

D Diagram Alir Untuk Proses *Inference Engine*

Diagram alir sistem untuk proses *inference engine* menggambarkan proses penelusuran untuk menentukan kesimpulan yang tepat. *Inference engine* akan menerima respon data yang berasal dari jawaban *user*, kemudian melakukan proses terhadap basis pengetahuan (*knowledge base*) yang sesuai. Metode yang digunakan adalah runut maju (*forward chaining*). Diagram alir sistem untuk proses *inference engine* dapat dilihat pada Gambar 3.7.

3.2.4 Perancangan *Rule Base*

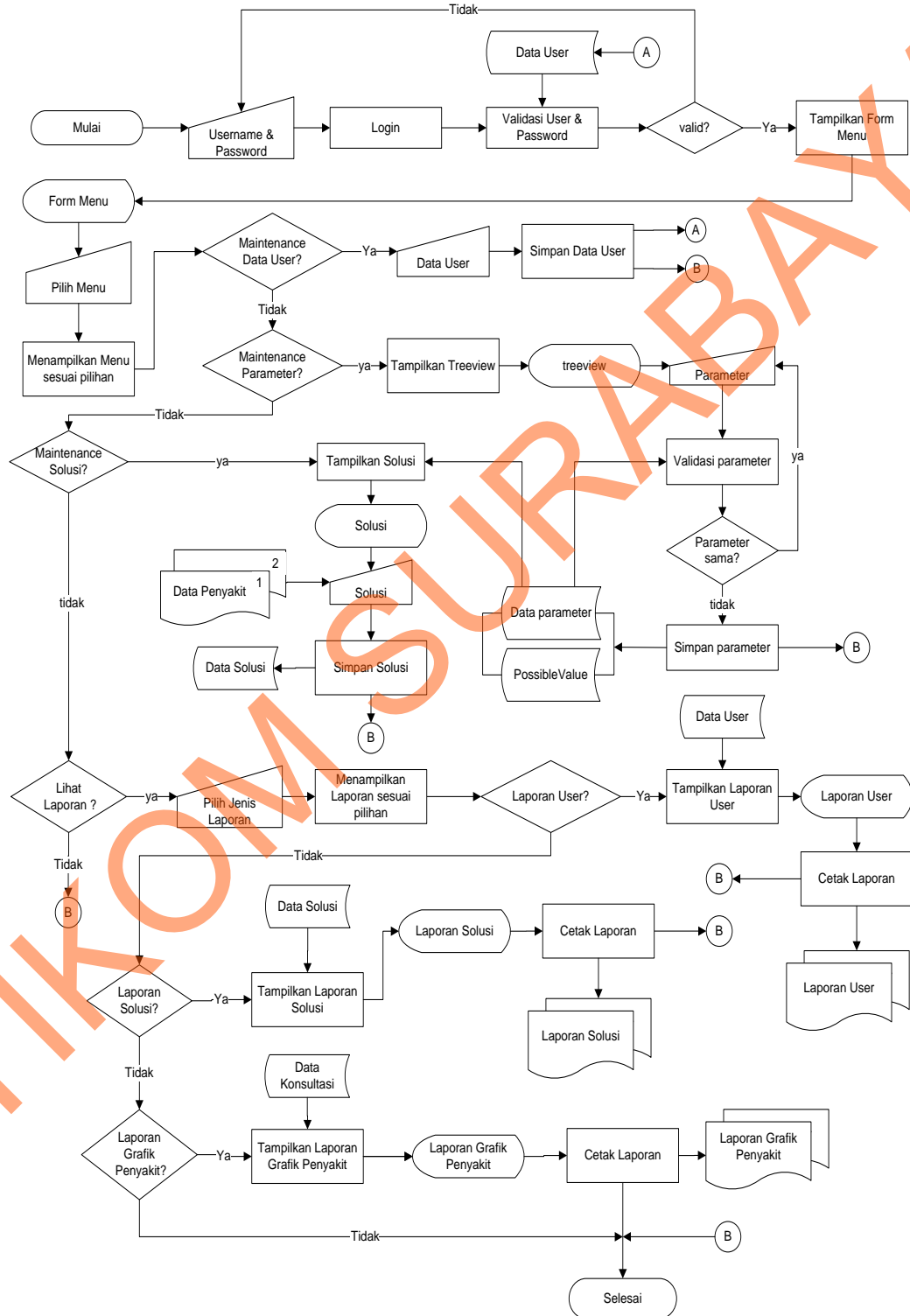
Pengembangan *rule base* telah direpresentasikan dalam bentuk *block diagram* yang kemudian diimplementasikan dalam bentuk *rules*, yaitu struktur berbasis pengetahuan. *Rule* pada dasarnya terdiri dari dua bagian pokok, yaitu bagian *IF* yang sering disebut premis atau kondisi dan bagian *THEN* yang sering disebut konklusi atau kesimpulan. Berikut ini merupakan contoh dari struktur basis pengetahuan yang sebelumnya telah dirancang mempergunakan *block diagram*.

Contoh: *Rule 1: IF* Daun = menguning, layu

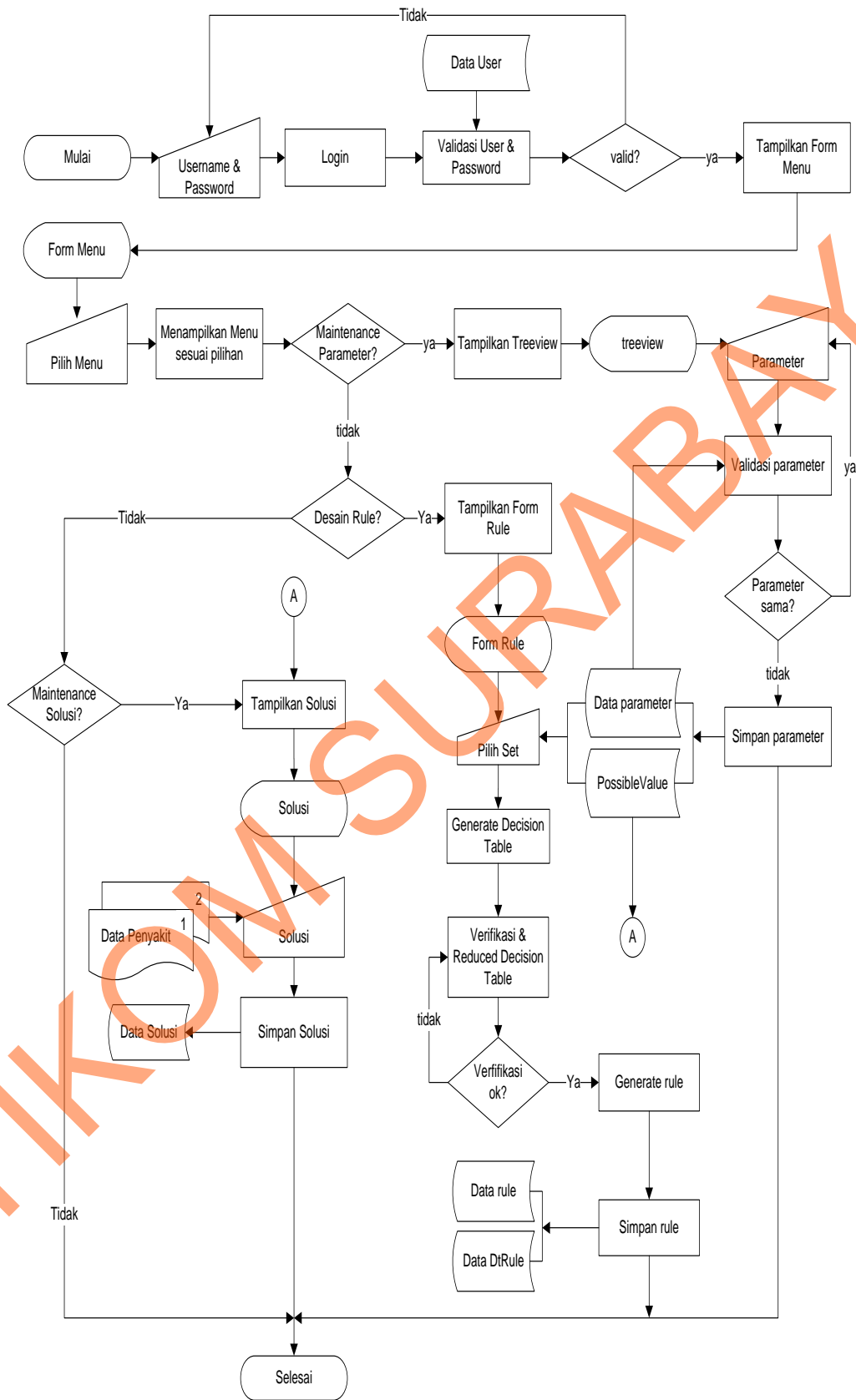
AND Batang = muncul retakan

AND Akar = mengeluarkan cairan kental coklat kekuningan

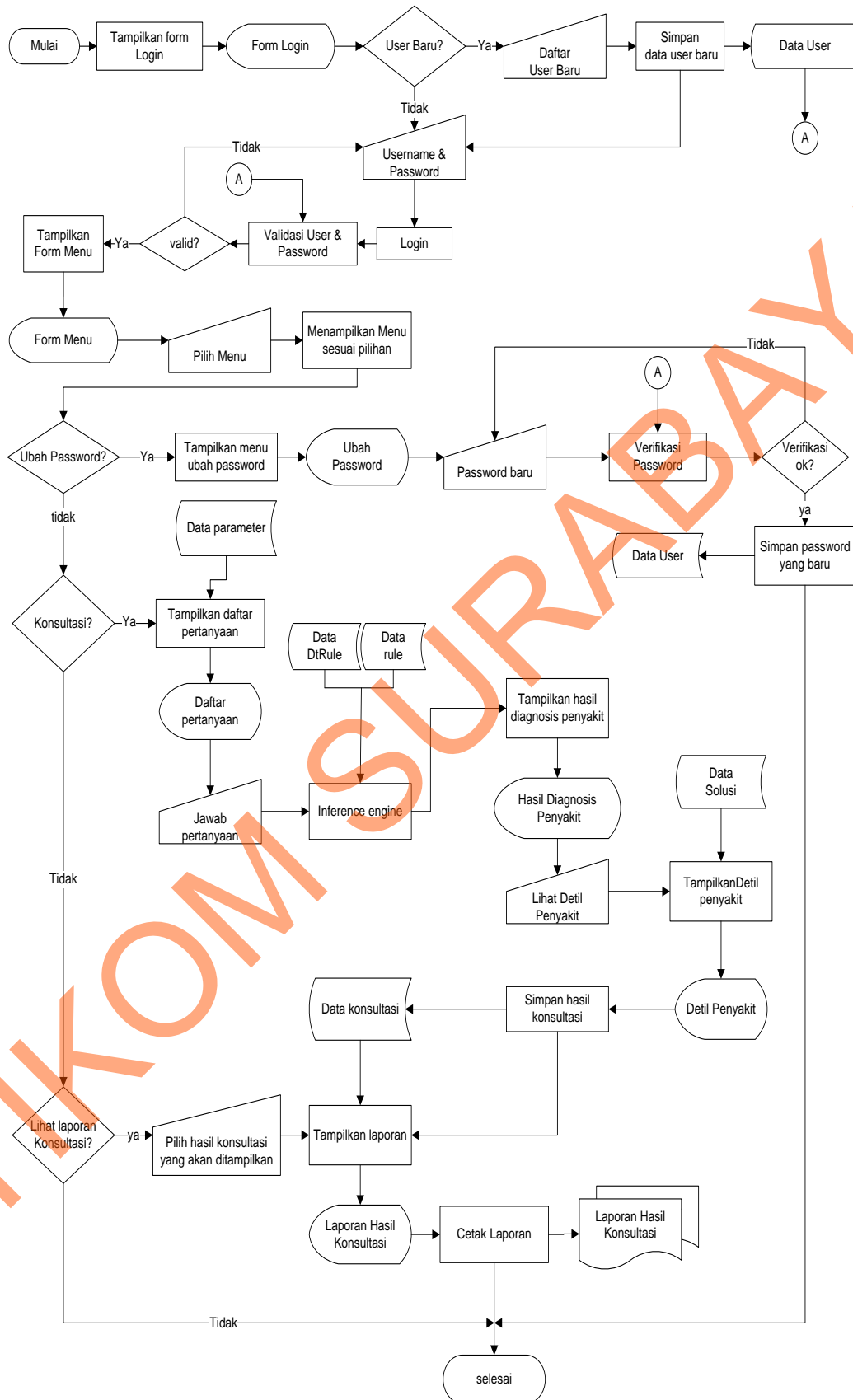
AND Tanaman = tampak tudung jamur bergerombol di dasar tanaman
THEN Jenis Penyakit Kopi = Busuk akar



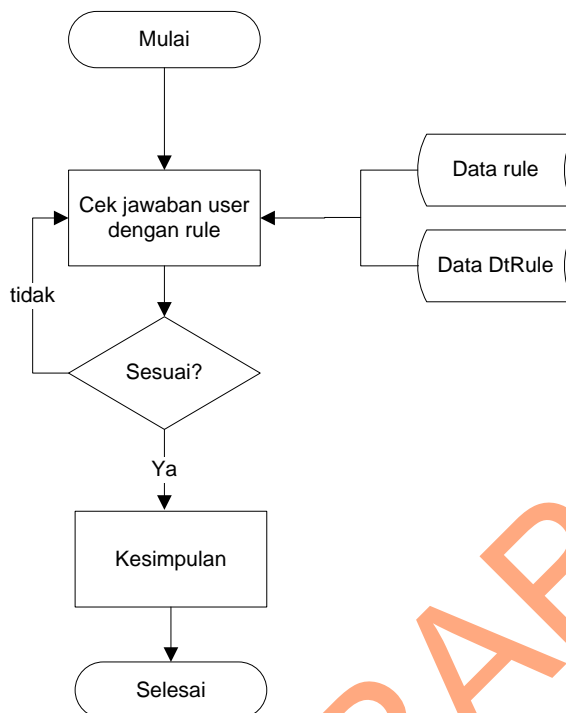
Gambar 3.4 Diagram Alir Sistem Untuk Admin



Gambar 3.5 Diagram Alir Sistem Untuk Pakar



Gambar 3.6 Diagram Alir Sistem Untuk User



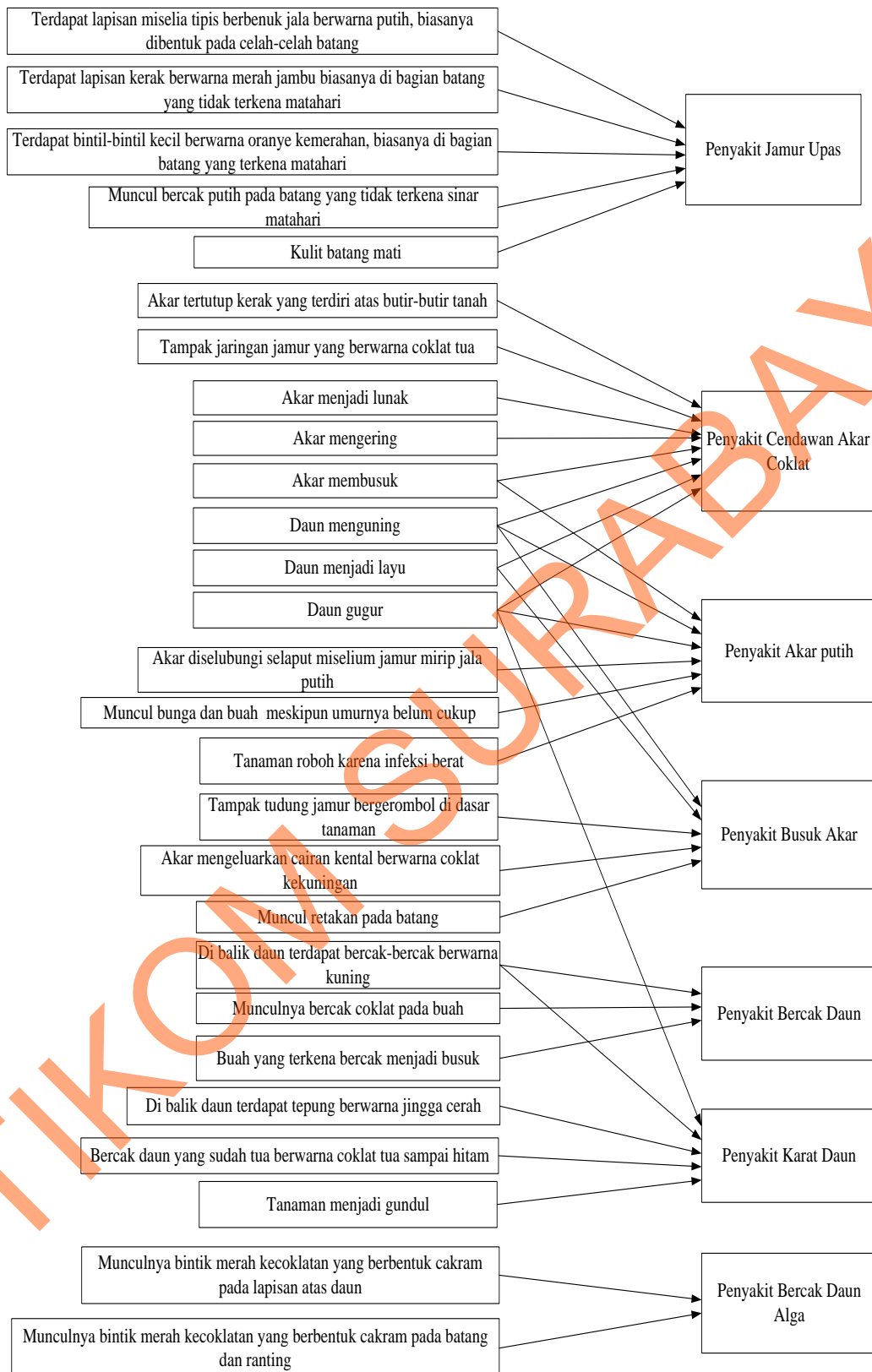
Gambar 3.7 Diagram Alir Sistem Untuk Proses *Inference Engine*

3.2.5 Rancangan Proses *Forward Chaining*

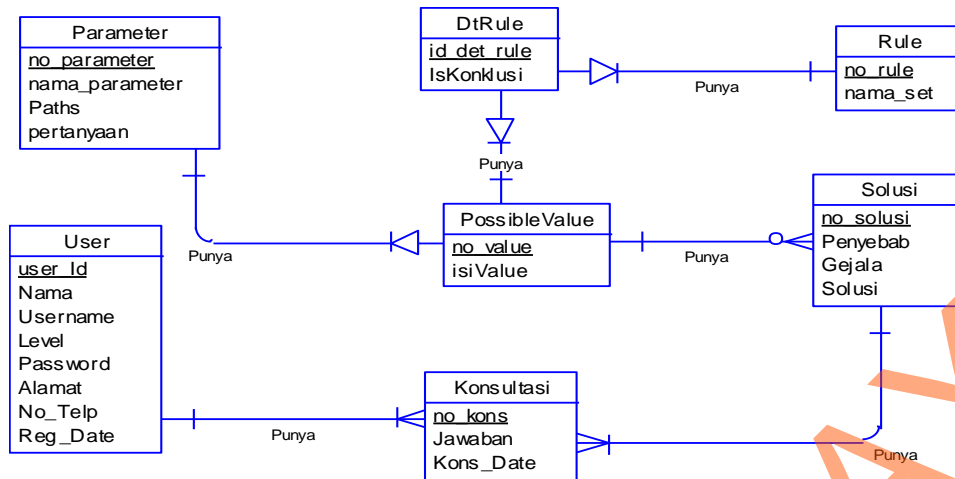
Rancangan proses *forward chaining* ini menggambarkan alur dari setiap gejala-gejala yang ada menuju kepada kesimpulan akhir berupa jenis penyakit dari tanaman kopi. Rancangan proses *forward chaining* ini dapat dilihat pada Gambar 3.8.

3.2.6 *Entity Relationship Diagram (ERD)*

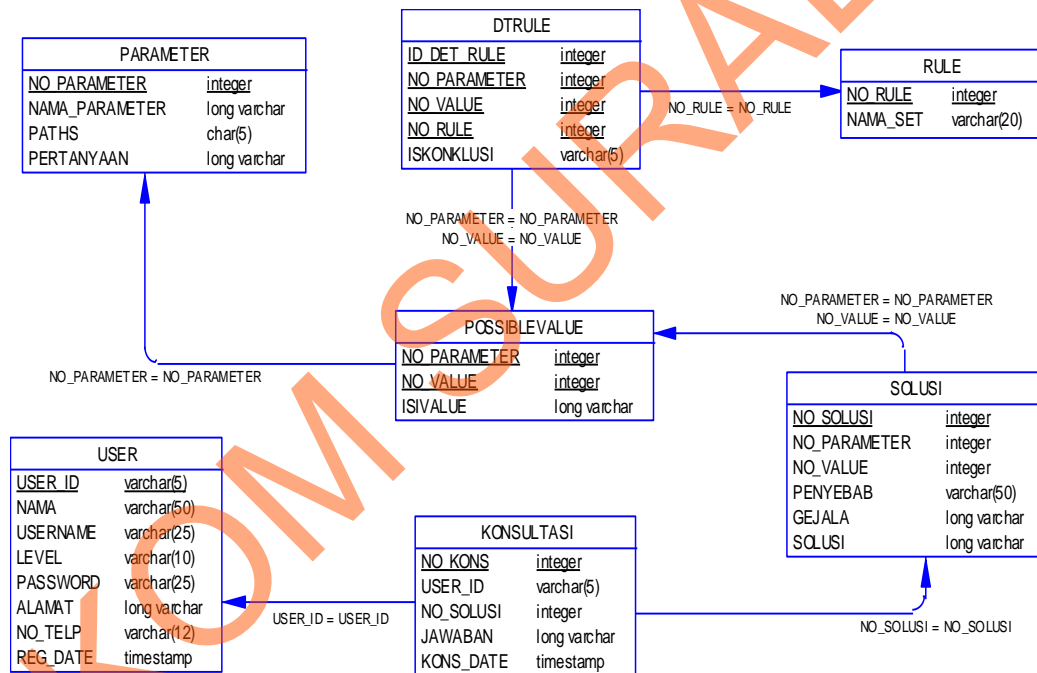
ERD digunakan untuk menggambarkan hubungan tabel-tabel yang digunakan dalam sistem. Dalam laporan ini ERD disajikan dalam bentuk *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)*. Model CDM dari aplikasi ini dapat dilihat pada Gambar 3.9 sedangkan model PDM yang merupakan hasil generate dari CDM dapat dilihat pada Gambar 3.10.



Gambar 3.8 Rancangan *Forward Chaining* Aplikasi Sistem Pakar Untuk Diagnosis Penyakit Pada Tanaman Kopi



Gambar 3.9 CDM Aplikasi Sistem Pakar Untuk Diagnosis Penyakit Kopi



Gambar 3.10 PDM Aplikasi Sistem Pakar Untuk Diagnosis Penyakit Kopi

3.2.7 Struktur Tabel

Struktur tabel merupakan penjelasan dan penjabaran dari suatu *database*. Dalam struktur tabel dijelaskan fungsi dari masing-masing tabel dan *field* yang ada, juga tipe dari setiap *field* beserta dengan konstrainnya. Struktur tabel dari aplikasi sistem pakar ini adalah sebagai berikut:

1. Nama : Parameter
- Primary Key : no_parameter
- Foreign Key : -
- Fungsi : Menyimpan data gejala-gejala penyakit tanaman kopi

Tabel 3.3 Parameter

No	Nama Field	Type Data	Length	Key	Keterangan
1	no_parameter	Integer	-	PK	Nomor parameter
2	Nama_parameter	Varchar	Max	-	Nama dari parameter
3	Paths	Char	5	-	Paths
4	Pertanyaan	Varchar	Max	-	Pertanyaan yang akan ditampilkan ke User

2. Nama : PossibleValue
- Primary Key : no_value
- Foreign Key : no_parameter *reference* Parameter (no_parameter)
- Fungsi : Menyimpan data *possible value* dari masing-masing parameter

Tabel 3.4 PossibleValue

No	Nama Field	Type Data	Length	Key	Keterangan
1	no_value	Integer	-	PK	Nomor value
2	no_parameter	Integer	-	FK	Nomor parameter
3	IsiValue	Varchar	Max	-	Nama value

3. Nama : Rule
- Primary Key : no_rule
- Foreign Key : -
- Fungsi : Menyimpan *Rule*

Tabel 3.5 Rule

No	Nama Field	Tipe Data	Length	Key	Keterangan
1	no_rule	Integer	-	PK	Nomor <i>rule</i>
2	Nama_Set	Varchar	20	-	Nama set

4. Nama : DtRule
- Primary Key : id_det_rule
- Foreign Key : no_rule *reference* Rule (no_rule), no_parameter *reference* Parameter (no_parameter) dan no_value *reference* PossibleValue (no_value)
- Fungsi : Menyimpan detil rule

Tabel 3.6 DtRule

No	Nama Field	Tipe Data	Length	Key	Keterangan
1	Id_det_rule	Integer	-	PK	ID detil rule
2	no_rule	Integer	-	FK	Nomor rule
3	no_parameter	Integer	-	FK	Nomor parameter
4	no_value	Integer	-	FK	Nomor Value
5	is_konklusi	Varchar	5	-	Keterangan yang menunjukkan apakah sebagai konklusi atau premis

5. Nama : Solusi
- Primary Key : no_solusi
- Foreign Key : no_parameter *reference* Parameter (no_parameter) dan no_value *reference* PossibleValue (no_value)
- Fungsi : Menyimpan solusi dari kesimpulan akhir

Tabel 3.7 Solusi

No	Nama Field	Tipe Data	Length	Key	Keterangan
1	no_solusi	Integer	-	PK	Nomor solusi
2	no_parameter	Integer	-	FK	Nomor parameter
3	no_value	Integer	-	FK	Nomor value
4	Penyebab	Varchar	50	-	Penyebab Penyakit
5	Gejala	Varchar	Max	-	Gejala Penyakit
6	Solusi	Varchar	Max	-	Isi solusi

6. Nama : Konsultasi

Primary Key : no_kons

Foreign Key : user_id *reference* User (user_id) dan no_solusi *reference*
Solusi (no_solusi)

Fungsi : Menyimpan hasil konsultasi

Tabel 3.8 Konsultasi

No	Nama Field	Tipe Data	Length	Key	Keterangan
1	No_kons	Integer	-	PK	Nomor konsultasi
2	user_id	Integer	-	FK	Id Pengguna
3	no_solusi	Integer	-	FK	Nomor solusi
4	Jawaban	Varchar	Max	-	Hasil jawaban pengguna
5	Kons_date	DateTime	-	-	Tanggal konsultasi

7. Nama : User

Primary Key : user_id

Foreign Key : -

Fungsi : Menyimpan data pengguna aplikasi

Tabel 3.9 User

No	Nama Field	Tipe Data	Length	Key	Keterangan
1	User_id	Varchar	5	PK	Id pengguna
2	Nama	Varchar	50	-	Nama pengguna
3	Username	Varchar	25	-	Username pengguna
4	Level	Varchar	10	-	Level pengguna
5	Password	Varchar	25	-	Password pengguna
6	Alamat	Varchar	Max	-	Alamat pengguna
7	No_Telp	Varchar	12	-	No telp pengguna
8	Reg_date	Datetime			Tanggal registrasi

3.2.8 Desain Input Output

Desain *input output* yang dibuat berfungsi untuk memudahkan *user* pakar dan *user* umum dalam menggunakan sistem. *Forms* yang dirancang meliputi konsep interaksi manusia dan komputer dimana seorang *user* hanya dengan melihat *form* dapat mengerti langkah yang akan dilakukan selanjutnya.

1. Desain *Form* Utama

Desain *form* Utama adalah *form* pertama saat pengguna pertama kali menjalankan aplikasi sistem pakar ini. *Form* ini berisi dua *button*, yaitu *button* *user* pakar dan *user* umum. *Form* selanjutnya dapat diakses oleh *user* sesuai dengan *button* yang dipilih. Desain *form* Utama dapat dilihat pada Gambar 3.10 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.11.

Aplikasi Sistem Pakar Untuk Mendiagnosis Penyakit Pada Tanaman Kopi

User Pakar

User Umum

Gambar 3.11 Desain *Form* Utama

Tabel 3.10 Fungsi Obyek Desain *Form* Utama

No	Nama Obyek	Tipe Obyek	Fungsi
1	User Pakar	Button	Menampilkan <i>form Login</i> user pakar
2	User Umum	Button	Menampilkan <i>form Login</i> pada user umum

2. Desain *Form Login* User Pakar

Desain *form Login* ini digunakan oleh *user* pakar untuk masuk ke dalam sistem. Desain *form Login* ini memiliki dua level, yaitu *user* admin dan pakar. Desain *form Login* dapat dilihat pada Gambar 3.12 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.11.

The image shows a login form for a 'User Pakar'. It consists of a rectangular box containing the following elements:

- A label 'Username' followed by a colon and a text input field.
- A label 'Password' followed by a colon and a text input field.
- Two buttons at the bottom: 'OK' and 'Batal'.

Gambar 3.12 Desain *Form Login* User PakarTabel 3.11 Fungsi Obyek Desain *Form Login* User Pakar

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field username</i>	<i>TextBox</i>	Memasukkan data <i>username</i>
2	<i>Field password</i>	<i>TextBox</i>	Memasukkan data <i>password</i>
3	OK	Button	Sebagai autentifikasi dan autorisasi untuk menampilkan <i>form</i> berikutnya.
4	Batal	Button	Membatalkan <i>user</i> pakar untuk masuk ke <i>form</i> selanjutnya dan menampilkan <i>form</i> Utama

3. Desain *Form Login User Umum*

Desain *form Login* ini digunakan oleh *user umum* untuk masuk ke dalam sistem. *User* yang memiliki *ID* bisa langsung mengisi *form* ini, jika tidak maka *user* harus mendaftar terlebih dahulu. Desain *form Login* ini dapat dilihat pada Gambar 3.13 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.12.

The image shows a login form with the following elements:

- A label "Username" followed by a text input field.
- A label "Password" followed by a text input field.
- Two buttons: "OK" and "Batal".
- A link labeled "Klik disini untuk buat user baru" located below the buttons.

Gambar 3.13 Desain *Form Login User Umum*

Tabel 3.12 Fungsi Obyek Desain *Form Login User Umum*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field username</i>	<i>TextBox</i>	Memasukkan data <i>username</i>
2	<i>Field password</i>	<i>TextBox</i>	Memasukkan data <i>password</i>
3	<i>OK</i>	<i>Button</i>	Sebagai autentifikasi dan otorisasi untuk menampilkan <i>form</i> berikutnya.
4	Batal	<i>Button</i>	Membatalkan <i>user</i> pakar untuk masuk ke <i>form</i> selanjutnya dan menampilkan <i>form</i> Utama
5	<u>Klik disini untuk membuat <i>user</i> baru</u>	<i>LinkLabel</i>	Menampilkan <i>form</i> Daftar <i>User</i> untuk membuat <i>user</i> baru

4. Desain *Form* Daftar *User*

Desain *form* ini digunakan untuk membuat data *user umum* yang baru pertama kali mengakses aplikasi sistem pakar ini. Desain *form* ini dapat

dilihat pada Gambar 3.14 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.13.

...: Daftar User ...:

Nama

Username

Password

Alamat

No. Telp

Gambar 3.14 Desain *Form* Daftar *User*

Tabel 3.13 Fungsi Obyek Desain *Form* Daftar *User*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field Nama</i>	<i>TextBox</i>	Memasukkan nama <i>user</i>
2	<i>Field Username</i>	<i>TextBox</i>	Memasukan <i>username</i> yang nantinya akan digunakan saat melakukan <i>login</i>
3	<i>Field password</i>	<i>TextBox</i>	Memasukkan data <i>password</i>
4	<i>Field Alamat</i>	<i>TextBox</i>	Memasukkan data alamat
5	<i>Field No.Telp</i>	<i>TextBox</i>	Memasukkan data nomor telepon
6	<i>OK</i>	<i>Button</i>	Menyimpan data <i>user</i> yang telah diisi dan menampilkan <i>form Menu</i>
7	<i>Batal</i>	<i>Button</i>	Membatalkan transaksi dan kembali ke <i>form Utama</i>

5. Desain *Form Menu*

Desain *form Menu* ini memiliki delapan *menu*, yaitu *menu* Ubah Password, Konsultasi, Data User, Parameter, Desain Rule, Solusi, Laporan dan Keluar.

Menu Laporan memiliki lima sub *menu*, yaitu Laporan User, Solusi, Rule, Hasil Konsultasi dan Grafik Penyakit.

Form Menu ini ditampilkan setelah *user* melakukan proses *Login*. *Menu* yang dapat diakses oleh *user* umum adalah: Ubah Password, Konsultasi, Laporan dengan sub *menu* Hasil Konsultasi dan *menu* Keluar. *User* yang *login* sebagai admin dapat mengakses beberapa *menu*, yaitu: *menu* Data User, Parameter Solusi dan juga *menu* Laporan dengan sub *menu* User dan Solusi. Terakhir *user* yang *login* sebagai pakar dapat mengakses beberapa *menu*, yaitu: *menu* Konsultasi, Parameter, Desain *Rule*, Solusi dan juga *menu* Laporan dengan sub *menu* Laporan Hasil konsultasi. Desain *form Menu* ini dapat dilihat pada Gambar 3.15 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.14.

Ubah Password	Konsultasi	User	Parameter	Desain Rue	Solusi	Laporan		Keluar
Gambar								
Selamat Datang						Tanggal & Jam Sistem		

Gambar 3.15 Desain *Form Menu*

Tabel 3.14 Fungsi Obyek Desain *Form Menu*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Ubah Password	<i>MenuStrip</i>	Menampilkan <i>form</i> Ubah Password
2	Konsultasi	<i>MenuStrip</i>	Menampilkan <i>form</i> Konsultasi
3	User	<i>MenuStrip</i>	Menampilkan <i>form</i> Data User
4	Parameter	<i>MenuStrip</i>	Menampilkan <i>form</i> Treeview
5	Desain Rule	<i>MenuStrip</i>	Menampilkan <i>form</i> Rule
6	Solusi	<i>MenuStrip</i>	Menampilkan <i>form</i> Solusi
7	Laporan	<i>MenuStrip</i>	Menampilkan Sub <i>Menu</i> User, <i>Solusi</i> dan Hasil Konsultasi
8	Keluar	<i>MenuStrip</i>	Keluar dari <i>form</i> Menu

6. Desain *Form* Ubah *Password*

Desain *form* ini hanya dapat diakses oleh *user* umum untuk mengubah *password*. Desain *form* Ubah *Password* ini dapat dilihat pada Gambar 3.16 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.15.

Gambar 3.16 Desain *Form* Ubah *Password*Tabel 3.15 Fungsi Obyek Desain *Form* Ubah *Password*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field password</i> Lama	<i>TextBox</i>	Memasukkan data <i>password</i> yang lama
2	<i>Field password</i> Baru	<i>TextBox</i>	Memasukkan data <i>password</i> yang baru
3	Ulangi <i>Password</i>	<i>TextBox</i>	Memverifikasi <i>Password</i> baru yang dibuat
4	OK	<i>Button</i>	Menyimpan hasil ubah <i>password</i>
5	Batal	<i>Button</i>	Membatalkan proses ubah <i>password</i>

7. Desain *Form Data User*

Desain *form Data User* berfungsi untuk memelihara data-data *user* pakar yang mengoperasikan sistem. *User ID* dihasilkan langsung oleh sistem sehingga *user* tidak perlu memasukkan *User ID* secara manual. Desain *form Data User* dapat dilihat pada Gambar 3.17 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.16.

Gambar 3.17 Desain *Form Data User*

Tabel 3.16 Fungsi Obyek Desain *Form Data User*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field Nama</i>	<i>TextBox</i>	Memasukkan nama pengguna
2	<i>Field Username</i>	<i>TextBox</i>	Memasukan username yang nantinya akan digunakan saat <i>login</i>
3	Level	<i>ComboBox</i>	Menentukan level pengguna dengan memilih level yang tersedia
4	<i>Field Password</i>	<i>TextBox</i>	Memasukkan <i>password</i> pengguna
5	<i>Field Alamat</i>	<i>TextBox</i>	Memasukkan alamat pengguna

Tabel 3.16 Fungsi Obyek Desain *Form Data User* (Lanjutan)

No	Nama Obyek	Tipe Obyek	Fungsi
6	<i>Field</i> No. Telp	<i>TextBox</i>	Memasukkan nomor telepon pengguna
7	Simpan	<i>Button</i>	Meyimpan data masukan
8	Ubah	<i>Button</i>	Mengubah data pengguna
9	Cari	<i>Button</i>	Menampilkan <i>form</i> Cari User untuk mencari data pengguna berdasarkan nama pengguna
10	Batal	<i>Button</i>	Membatalkan proses
11	Keluar	<i>Button</i>	Keluar dari <i>form</i> Data User dan menampilkan <i>form</i> Menu

8. Desain *Form* Cari User

Desain *form* ini akan tampil saat pengguna menekan *button* Cari. *Form* cari User ini digunakan untuk mencari data user yang akan diubah datanya. Pencarian data user ini bisa dengan mengisi nama pada *textbox* nama setelah itu menekan *enter*. Data akan langsung disortir sesuai dengan nama yang diisi setelah itu *double* klik pada tabel untuk memilih data. Data user ini juga bisa dipilih langsung dari tabelnya. Desain *form* ini dapat dilihat pada Gambar 3.18 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.17

... Cari User ...

Nama

Data user			

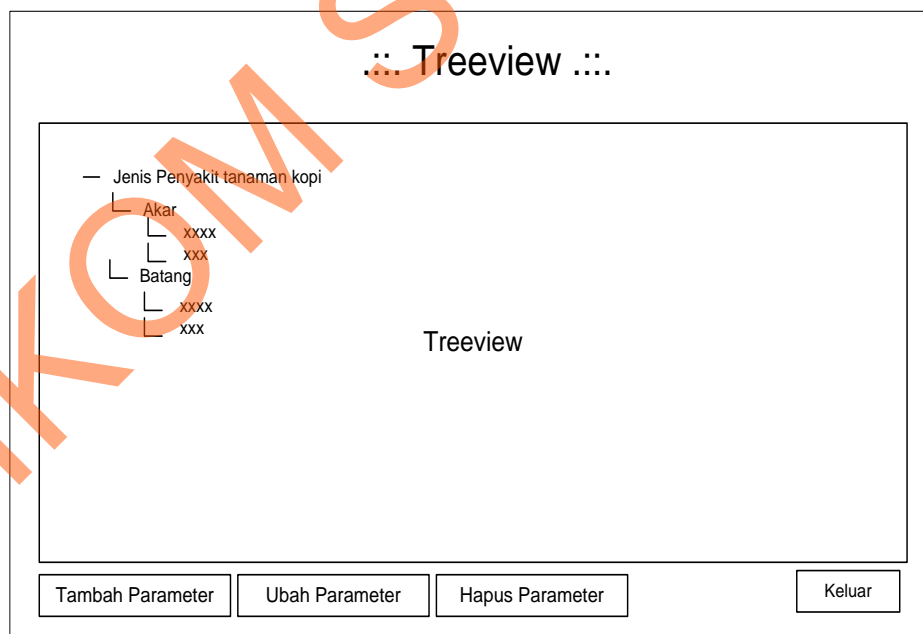
Gambar 3.18 Desain *Form* Cari User

Tabel 3.17 Fungsi Obyek Desain *Form* Cari *User*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Field</i> Nama	<i>TextBox</i>	Mengisi nama yang akan dicari setelah itu menekan <i>enter</i>
2	Data <i>User</i>	<i>DataGridView</i>	Berisi data <i>User</i>
3	Lihat Laporan	<i>Button</i>	Menampilkan <i>form</i> Laporan <i>User</i>
4	Batal	<i>Button</i>	Membatalkan proses cari <i>User</i> dan kembali ke <i>form</i> Data <i>User</i>

9. Desain *Form Treeview*

Desain *form Treeview* ini digunakan untuk mendesain dan membuat diagram ketergantungan (*dependency diagram*) dalam bentuk *treeview* yang merupakan langkah awal dalam membuat *rule*. Desain *form Treeview* dapat dilihat pada Gambar 3.19 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.18.

Gambar 3.19 Desain *Form Treeview*

Tabel 3.18 Fungsi Obyek Desain *Form Treeview*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Treeview</i>	<i>Treeview</i>	Tampilan <i>Treeview</i>
2	Tambah Parameter	<i>Button</i>	Menampilkan <i>form</i> Parameter untuk menambah parameter baru
3	Ubah Parameter	<i>Button</i>	Menampilkan <i>form</i> Parameter untuk mengubah data parameter
4	Hapus Parameter	<i>Button</i>	Menghapus data parameter yang dipilih
5	Keluar	<i>Button</i>	Keluar dari <i>form Treeview</i>

10. Desain *Form* Parameter

Desain *form* ini ditampilkan setelah memilih *button* Tambah dan Ubah Parameter pada *form Treeview*. Desain *form* ini digunakan untuk menambah dan mengubah data parameter dan juga menambah dan mengubah *value* atau jawaban pada setiap parameter yang dibuat. Desain *form* Parameter ini dapat dilihat pada Gambar 3.20 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.19.

Gambar 3.20 Desain *Form* Parameter

Tabel 3.19 Fungsi Obyek Desain *Form* Parameter

No	Nama Obyek	Tipe Obyek	Fungsi
1	Field Nama Parameter	<i>TextBox</i>	Memasukkan nama parameter
2	Field Pertanyaan	<i>TextBox</i>	Memasukkan pertanyaan sesuai dengan nama parameter
3	Tambah Gambar	<i>Button</i>	Untuk memilih dan menampilkan gambar
4	Gambar	<i>PictureBox</i>	Tampilan Gambar yang telah dipilih
5	Tambah	<i>Button</i>	Untuk menyimpan dan menambah data parameter
6	Ubah	<i>Button</i>	Mengubah data parameter yang telah dipilih pada <i>form Treeview</i>
7	Field Jawaban	<i>Textbox</i>	Memasukkan jawaban sesuai dengan parameter yang dibuat
8	Option Jawaban	<i>ListBox</i>	Menampung setiap Jawaban.
9	<u>Batal edit</u>	<i>LinkLabel</i>	Untuk membatalkan proses edit Jawaban setelah memilih jawaban pada option jawaban
10	<input type="text" value="-"/>	<i>Button</i>	Untuk menghapus jawaban yang dipilih pada option jawaban
11	OK	<i>Button</i>	Menyimpan hasil masukkan dan kembali ke <i>form Treeview</i>
12	Batal	<i>Button</i>	Membatalkan proses yang dilakukan dan kembali ke <i>form Treeview</i>

11. Desain *Form Rule*

Desain *form Rule* ini digunakan untuk membuat *rule*. Sebelum menghasilkan *rule* yang benar harus diverifikasi terlebih dahulu. Langkah pertama yang harus dilakukan adalah memilih *set* yang akan diverifikasi. Setelah *set* dipilih, tekan *button generate decision table* untuk menampilkan *decision table* yang telah direduksi dari *form Tabel Keputusan*. Setelah itu, pakar menekan *button generate rule* untuk menampilkan *rule* beserta jawabannya kemudian tekan *button simpan*. Desain *form Rule* dapat dilihat pada Gambar 3.21 dan fungsi dari setiap obyeknya dapat dilihat pada Tabel 3.20.

...: Desain Rule ...:

Set

Reduced decision table		

Daftar rule

Generate Decision Table Generate Rule Simpan Keluar

Gambar 3.21 Desain *Form Rule*Tabel 3.20 Fungsi Obyek Desain *Form Rule*

No	Nama Obyek	Tipe Obyek	Fungsi
1	<i>Set</i>	<i>ComboBox</i>	Memilih <i>set</i>
2	<i>Reduced Decision table</i>	<i>DataGridView</i>	Menampilkan hasil dari mereduksi tabel pada <i>form</i> Tabel Keputusan
3	<i>Generate Decision Table</i>	<i>Button</i>	Menampilkan <i>form</i> Tabel Keputusan sesuai dengan <i>set</i> yang dipilih
4	<i>Generate rule</i>	<i>Button</i>	Membuat rule berdasarkan <i>reduced decision table</i>
5	<i>Daftar rule</i>	<i>RichTextBox</i>	Berisi <i>rules</i> hasil <i>generate rule</i>
6	<i>Simpan</i>	<i>Button</i>	Menyimpan <i>rule</i>
7	<i>Keluar</i>	<i>Button</i>	Keluar dari <i>form Rule</i>

12. Desain *Form* Tabel Keputusan

Desain *form* Tabel Keputusan ini digunakan oleh pakar untuk memasukkan konklusi pada *decision table* yang ditampilkan otomatis dari sistem sesuai dengan *set* yang dipilih pada *form Rule*. *Button* reduksi digunakan untuk menghilangkan *unnecessary if condition* dan secara otomatis akan menghilangkan premis yang tidak perlu di kondisikan. Desain *form* Tabel Keputusan dapat dilihat pada Gambar 3.22 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.21.

... Solusi ...

Nama Penyakit	Penyebab	Gejala	Tindakan Pengendalian

Gambar 3.23 Desain *Form Solusi*Tabel 3.22 Fungsi Obyek Desain *Form Solusi*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Nama Penyakit	<i>DataGridView</i>	Berisi nama penyakit, <i>double</i> klik untuk menampilkan <i>form Isi Solusi</i>
2	Penyebab	<i>DataGridView</i>	Berisi penyebab penyakit
3	Gejala	<i>DataGridView</i>	Berisi gejala-gejala penyakit
4	Tindakan Pengendalian	<i>DataGridView</i>	Berisi tindakan pengendalian
5	Keluar	<i>Buttuon</i>	Keluar dari <i>form Solusi</i> dan menampilkan <i>form Menu</i>

14. Desain *Form Isi Solusi*

Desain *form Isi Solusi* digunakan untuk memasukkan solusi tindakan sesuai dengan nama penyakit yang dipilih dari *form Solusi*. Desain *form Isi Solusi* dapat dilihat pada Gambar 3.24 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.23.

... Isi Solusi ...

Nama Penyakit

Penyebab

Gejalanya:

Tindakan Pengendalian

Tambahkan Gambar

Gambar 3.24 Desain *Form* Isi SolusiTabel 3.23 Fungsi Obyek Desain *Form* Isi Solusi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Field Nama Penyakit	<i>TextBox</i>	Nama penyakit tanaman kopi yang dipilih sebelumnya pada <i>form</i> Solusi
2	Field Penyebab	<i>TextBox</i>	Memasukkan penyebab penyakit tanaman kopi
3	Field Gejalanya	<i>RischTextBox</i>	Memasukkan Gejala-gejala sesuai dengan penyakit tanaman kopi yang terpilih
4	Field Tindakan Pengendalian	<i>RichTextBox</i>	Memasukkan tindakan pengendalian penyakit pada tanaman kopi
5	Tambahkan Gambar	<i>Button</i>	Untuk memilih dan menampilkan Gambar
6	GAMBAR	<i>PictureBox</i>	Tampilan gambar yang telah dipilih
7	OK	<i>Button</i>	Menyimpan data solusi dan menampilkan <i>form</i> Solusi
8	Batal	<i>Button</i>	Membatalkan proses dan keluar dari <i>form</i> Isi Solusi

15. Desain *Form* Konsultasi

Desain *form* Konsultasi ini digunakan oleh pengguna untuk melakukan konsultasi diagnosis penyakit tanaman kopi. Konsultasi dilakukan dengan memilih terlebih dahulu bagian dari tanaman kopi yang ada, kemudian pengguna dapat memilih gejala-gejala yang ditampilkan oleh sistem. Setelah itu pengguna menekan *button* diagnosis untuk menampilkan *form* Hasil Konsultasi yang berisi kesimpulan akhir. Desain *form* Konsultasi dapat dilihat pada Gambar 3.25 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.24.

Gambar 3.25 Desain *Form* Konsultasi

Tabel 3.24 Fungsi Obyek Desain *Form* Konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Pertanyaan	<i>DataGridView</i>	Berisi daftar pertanyaan
2	Gambar	<i>DataGridView</i>	Berisi Gambar dari setiap pertanyaan yang ada.
3	Jawaban	<i>CheckBox</i>	Digunakan oleh pengguna untuk memilih jawaban
4	Daftar Gejala Yang Dipilih	<i>ListView</i>	Berisi daftar gejala penyakit tanaman kopi yang dipilih
5	Diagnosis	<i>Button</i>	Mendiagnosis penyakit dan menampilkan <i>form</i> Hasil Konsultasi
6	Keluar	<i>Button</i>	Keluar dari <i>form</i> Konsultasi dan menampilkan <i>form</i> Menu

16. Desain *Form* Hasil Konsultasi

Desain *form* Hasil Konsultasi digunakan untuk menampilkan hasil diagnosis penyakit berupa kesimpulan akhir yang berisi nama penyakit. Sistem akan menampilkan tiga kesimpulan jika gejala yang dipilih oleh pengguna berada pada ketiga jenis penyakit yang ada. Pengguna dapat melihat detail penyakitnya dengan menekan *link* yang tersedia. Desain *form* Hasil Konsultasi dapat dilihat pada Gambar 3.26 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.25.

Gambar 3.26 Desain *Form* Hasil Konsultasi

Tabel 3.25 Fungsi Obyek Desain *Form* Hasil Konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Jawaban user	<i>RichTextBox</i>	Berisi jawaban atau daftar gejala penyakit yang dipilih oleh <i>user</i>
2	Kesimpulan 1	<i>Label</i>	Berisi nama penyakit yang pertama
3	Kesimpulan 2	<i>Label</i>	Berisi nama penyakit yang kedua
4	Kesimpulan 3	<i>Label</i>	Berisi nama penyakit yang ke tiga
5	<u>Klik untuk melihat detail penyakit</u>	<i>LinkLable</i>	Untuk menampilkan <i>form</i> detail penyakit

17. Desain *Form* Detil Penyakit

Desain *form* ini digunakan untuk menampilkan detil penyakit dari setiap kesimpulan yang terlihat pada *form* Hasil Konsultasi. Detil penyakit yang ditampilkan berupa penjelasan dari masing-masing penyakit, yaitu penyebabnya, gejalanya dan cara pengendaliannya. Desain *form* Detil penyakit ini dapat dilihat pada Gambar 3.27 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.26

Gambar 3.27 Desain *Form* Detil Penyakit.

Tabel 3.26 Fungsi Obyek Desain *Form* Detil konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Kesimpulan 1	<i>RichTextBox</i>	Berisi Penjelasan dari peyakit pertama
2	Kesimpulan 2	<i>RichTextBox</i>	Berisi Penjelasan dari peyakit kedua
3	Kesimpulan 3	<i>RichTextBox</i>	Berisi Penjelasan dari peyakit ketiga
4	OK	<i>Button</i>	Untuk menampilkan <i>form</i> laporan hasil konsultasi

18. Desain *Form* Laporan *User*

Desain *form* Laporan *User* ini digunakan untuk menampilkan data *user* pakar yang tersimpan dalam *database*. Laporan ini bisa menampilkan data *user* berdasarkan level pengguna dengan memilih pada bagian level, atau bisa juga ditampilkan berdasarkan periode dengan memilih pada periode yang ada. Dan juga laporan ini bisa ditampilkan dengan memilih keduanya, yaitu berdasarkan level pengguna dan periodenya. Desain *form* Laporan *User* ini dapat dilihat pada Gambar 3.28 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.27.

Gambar 3.28 Desain *Form* Laporan *User*

Tabel 3.27 Fungsi Obyek Desain *Form* Laporan *User*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Tampilkan semua	<i>ComboBox</i>	Berisi pilihan untuk menampilkan laporan
2	Level	<i>ComboBox</i>	Berisi pilihan level pengguna dan menampilkan laporan <i>user</i> sesuai dengan level yang dipilih.
3	Periode	<i>DateTimePicker</i>	Untuk memilih menampilkan laporan berdasarkan tanggal.
4	Tampilkan	<i>Button</i>	Menampilkan laporan sesuai dengan pilihannya.

19. Desain *Form* Laporan Hasil Konsultasi

Desain *form* Laporan Hasil Konsultasi ini digunakan untuk menampilkan laporan hasil konsultasi yang dilakukan oleh pengguna. Laporan ini dapat ditampilkan dengan memilih pada *comboBox* yang tersedia. Tampilan laporan ini dapat dipilih berdasarkan nama penyakit atau tanggal konsultasi, atau kombinasi keduanya dan juga dapat ditampilkan keseluruhan hasil konsultasi yang pernah dilakukan oleh pengguna yang sedang menggunakan aplikasi ini. Desain *form* Laporan Hasil Konsultasi dapat dilihat pada Gambar 3.29 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.28.

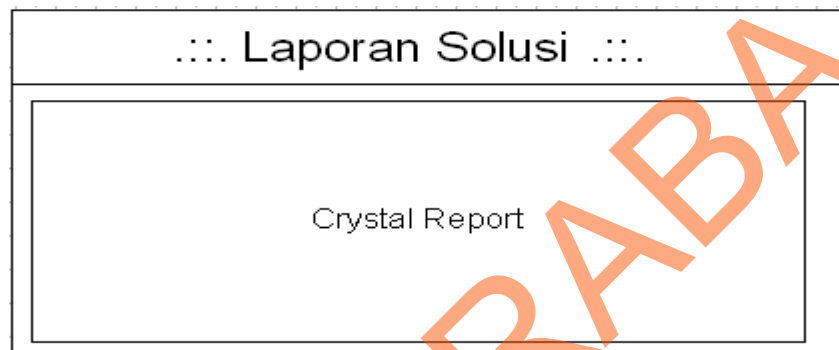
Gambar 3.29 Desain *Form* Laporan Hasil Konsultasi

Tabel 3.28 Fungsi Obyek Desain *form* Laporan Hasil Konsultasi

No	Nama Obyek	Tipe Obyek	Fungsi
1	Keduanya	<i>RadioButton</i>	Untuk menampilkan laporan berdasarkan nama penyakit dan juga tanggal konsultasinya
2	Nama Penyakit	<i>ComboBox</i>	Berisi nama-nama penyakit pada tanaman kopi
3	Tanggal Konsultasi	<i>DateTimePicker</i>	Untuk menampilkan laporan berdasar tanggal konsultasi
4	Tampilkan	<i>Button</i>	Untuk menampilkan laporan sesuai dengan pilihannya

20. Desain *Form* Laporan Solusi

Desain *form* Laporan Solusi ini digunakan untuk menampilkan laporan penjelasan dari setiap penyakit pada tanaman kopi. Penjelasannya berupa nama penyakit, penyebabnya dan cara pengendaliannya. Desain *form* Laporan Solusi ini dapat dilihat pada Gambar 3.30.



Gambar 3.30 Desain *Form* Laporan Solusi

20. Desain *Form* Laporan Grafik Penyakit

Desain *form* Laporan ini digunakan untuk menampilkan laporan penyakit yang sering muncul. Grafik penyakit ini dapat ditampilkan perbulan atau juga pertahun. Desain *form* ini dapat dilihat pada Gambar 3.31 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.29.



Gambar 3.31 Desain *Form* Laporan Grafik Penyakit

Tabel 3.29 Fungsi Obyek Desain *Form* Laporan Grafik Penyakit

No	Nama Obyek	Tipe Obyek	Fungsi
1	Bulan	<i>CheckBox</i>	Untuk memilih menampilkan Grafik perbulan
2	Tahun	<i>CheckBox</i>	Untuk menampilkan Grafik pertahun
3	Oktober	<i>ComboBox</i>	Berisi pilihan bulan untuk menampilkan grafik berdasarkan bulan yang dipilih
4	2011	<i>ComboBox</i>	Berisi pilihan tahun untuk menampilkan grafik penyakit setiap bulannya dalam tahun yang dipilih.
5	Tampilkan	<i>Button</i>	Untuk menampilkan laporan sesuai dengan pilihan.

21. Desain *Form* Laporan Rule

Desain *form* Laporan ini digunakan untuk menampilkan *rules* dari setiap penyakit yang ada. Desain *form* ini dapat dilihat pada Gambar 3.32 dan fungsi setiap obyeknya dapat dilihat pada Tabel 3.30.

Gambar 3.32 Desain *Form* Laporan RuleTabel 3.30 Fungsi Obyek Desain *Form* Laporan *Rule*

No	Nama Obyek	Tipe Obyek	Fungsi
1	Nama Penyakit	<i>ComboBox</i>	Berisi pilihan jenis penyakit yang akan ditampilkan laporan <i>rules</i>
2	Tampilkan	<i>Button</i>	Untuk menampilkan laporan sesuai dengan pilihan.

3.3 Perancangan Evaluasi Sistem

3.3.1 Perancangan Uji Coba Sistem

Perancangan uji coba sistem ini berisi tentang rancangan uji coba pada *form* yang telah dibuat. Uji coba ini dilakukan dengan menggunakan *Black Box Testing*. Aplikasi ini akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang dibuat sesuai dengan tujuan.

A Rancangan Uji Coba *Form Login*

Proses *login* dilakukan oleh *user* dengan cara memasukkan *username* dan *password*. Contoh data uji coba *login* yang digunakan untuk proses *login* ini dapat dilihat pada Tabel 3.31 dan untuk rancangan uji coba *form Login* dapat dilihat pada Tabel 3.32.

Tabel 3.31 Data Uji Coba *Login*

No	Nama Field	Data 1	Data 2
1	Username	Cupid	Dewi
2	Password	Cupit123	D3w1

Tabel 3.32 Rancangan Uji Coba *Form Login*

No	Tujuan	Input	Output yang diharapkan
1	Mendeskripsikan <i>Username</i> dan <i>Password</i> yang <i>valid</i>	Memasukkan data 1 seperti pada Tabel 3.31	Muncul Pesan “Login anda berhasil” dan <i>menu</i> aktif sesuai dengan level pengguna
2	Mendeskripsikan <i>Username</i> atau <i>Password</i> yang <i>invalid</i>	Memasukkan data <i>Username</i> : Dewi dan <i>Password</i> : believe	Muncul Pesan <i>Error</i> “Password anda salah”

B Rancangan Uji Coba *Form Data User*

Proses dari *form Data User* ini adalah menyimpan dan mengubah data *user*. Contoh uji coba data *user* dapat dilihat pada Tabel 3.33 dan untuk rancangan uji coba *form Data User* dapat dilihat pada Tabel 3.34.

Tabel 3.33 Data Uji Coba *User*

No	Nama Field	Data 1
1	Nama	Novita Hosama
2	Username	Vitazone
3	Level	PAKAR
4	Password	Vita
5	Alamat	Jln. Penjaringan Sari H/1 Perum. Pondok Citra Eksekutif (PCE), Surabaya
6	No. Telp	085722233344

Tabel 3.34 Rancangan Uji Coba *Form Data User*

No	Tujuan	Input	Output yang diharapkan
1	Menyimpan data <i>User</i> baru ke dalam <i>database</i>	Memasukkan data 1 seperti pada Tabel 3.33, kemudian menekan tombol Simpan	Muncul Pesan “Data berhasil disimpan”
2	Menghindari ada <i>field</i> yang kosong saat penyimpanan	Memasukkan data kosong pada salah satu <i>field</i> , kemudian menekan tombol Simpan	Muncul Pesan <i>Error</i> “Isian tidak boleh ada yang kosong, Data harus diisi semua”
3	Menghindari pengisian <i>username</i> yang sama	Memasukan <i>Username</i> yang sama dengan <i>username</i> sebelumnya, kemudian tekan Tombol Simpan	Muncul Pesan <i>Error</i> “Username sudah ada, Silahkan ganti Username anda”
4	Mengubah data <i>User</i>	Mengubah No. Telp data 1 pada Tabel 3.33 menjadi 085731096123. Tekan tombol Cari untuk menampilkan <i>form Cari User</i> dan memilih data <i>user</i> sesuai dengan nama, kemudian tekan <i>enter</i> , pilih datanya dengan <i>double klik</i> , ubah masukan pada <i>field</i> No.Telp kemudian tekan tombol Ubah	Muncul Pesan “Data berhasil diubah”

C Rancangan Uji Coba *Form Treeview*

Form ini digunakan untuk menampilkan *dependency diagram* dalam bentuk *treeview*. Proses yang dilakukan pada *form* ini adalah menghapus parameter. Rancangan uji coba *form Treeview* ini dapat dilihat pada Tabel 3.35.

Tabel 3.35 Rancangan Uji Coba *Form Treeview*

No	Tujuan	Input	Output yang diharapkan
1	Nama Parameter akan berubah ketika melakukan perubahan nama parameter	Pilih parameter yang akan diubah kemudian tekan Ubah Parameter. Ubah data pada <i>form</i> Parameter.	Nama Parameter berubah sesuai dengan data yang telah dimasukkan.
2	Parameter pada <i>form Treeview</i> akan bertambah ketika ada penambahan Parameter	Pilih parameter yang akan dijadikan induk pada <i>form Treeview</i> kemudian tekan Tambah Parameter. Isikan data pada <i>form</i> Parameter	Muncul parameter baru sesuai dengan data yang telah ditambahkan.
3	Menghindari proses hapus parameter yang memiliki cabang dibawahnya	Memilih nama parameter yang memiliki <i>child</i> kemudian tekan tombol Hapus Parameter	Muncul Pesan <i>Error</i> "Parameter tidak dapat dihapus karena memiliki cabang dibawahnya, Hapus parameter dibawahnya satu per satu"
4	Menghindari proses hapus parameter yang paling atas (nama penyakit)	Memilih nama parameter yang paling atas pada <i>treeview</i> (Jenis Penyakit Tanaman kopi) kemudian menekan tombol Hapus Parameter	Muncul Pesan <i>Error</i> "Parameter tidak dapat dihapus"
5	Menghapus parameter	Memilih parameter pada <i>treeview</i> yang akan dihapus kemudian tekan tombol Hapus Parameter	Muncul Pesan "anda yakin melakukan penghapusan parameter", jika memilih <i>No</i> , maka parameter tidak dihapus, jika memilih <i>Yes</i> , maka muncul pesan "Parameter telah Dihapus"

D Rancangan Uji Coba *Form* Parameter

Proses dari *form* Parameter ini adalah menyimpan dan mengubah data parameter serta menambah, mengubah dan menghapus *value* atau jawaban.

Rancangan uji coba *form* Parameter dapat dilihat pada Tabel 3.36.

Tabel 3.36 Rancangan Uji Coba *Form* Parameter

No	Tujuan	Input	Output yang diharapkan
1	Menambah parameter baru	Pilih parameter yang akan dijadikan induk pada <i>treeview</i> , kemudian tekan tombol Tambah Parameter masukan data parameter baru pada <i>field</i> nama parameter dan pertanyaan kemudian tekan tombol Tambah	Muncul Pesan "Parameter berhasil disimpan"
2	Menghindari parameter kosong saat penyimpanan	Memasukkan data kosong pada <i>field</i> nama parameter kemudian tekan tombol Tambah	Muncul Pesan <i>Error</i> "Parameter tidak boleh kosong"
3	Menghindari jawaban kurang dari dua	Memasukkan satu jawaban pada <i>field</i> jawaban, kemudian tekan tombol OK	Muncul Pesan <i>Error</i> "Jawaban minimal harus dua"
4	Menghindari memasukkan Parameter yang sama	Mengisi nama parameter yang sama atau yang sudah ada dalam <i>Database</i> kemudian menekan tombol Tambah	Muncul Pesan <i>Error</i> "nama Parameter tidak boleh sama. dengan nama parameter yang lainnya"
5	Menghindari memasukkan jawaban yang sama	Menambah jawaban yang sama atau sudah ada pada <i>field</i> Option Jawaban kemudian tekan <i>enter</i>	Muncul Pesan <i>Error</i> "Jawaban sudah ada"
6	Mengubah Jawaban	Memilih jawaban pada option jawaban dan menekan <i>enter</i>	Muncul pesan "Jawaban telah diubah"
7	Mengubah data parameter	Memilih nama parameter pada <i>treeview</i> yang akan diubah kemudian tekan tombol Ubah Parameter, ubah nama kemudian tekan tombol Ubah	Muncul Pesan "Parameter berhasil diedit"
8	Menghindari option jawaban yang kosong ketika menekan tombol OK	Menekan tombol OK tanpa mengisi Jawaban	Muncul Pesan <i>Error</i> "Jawaban Tidak Boleh Kosong"
9	Menghapus Jawaban	Memilih jawaban pada option jawaban, kemudian tekan tombol yang bergambar tanda kurang (-)	Muncul pesan "Data telah dihapus"

E Rancangan Uji Coba *Form Rule*

Menu *generate rule* digunakan untuk membuat *rule* secara otomatis dari *reduced decision table* yang dilakukan pada *form* Tabel Keputusan dengan menekan tombol Generate Rule. Rancangan uji coba menu *generate rule* dapat dilihat pada Tabel 3.37.

Tabel 3.37 Rancangan Uji Coba *Form Rule*

No	Tujuan	Input	Output yang diharapkan
1	Menghindari simpan <i>rule</i> tanpa menekan tombol Generate Rule	Tidak menekan tombol Generate Rule untuk menghasilkan Rule-Rule	Tombol Simpan Rule tidak aktif
2	Keluar <i>form rule</i> tanpa melakukan penyimpanan rule	Tekan tombol keluar sebelum menyimpan hasil generate rule.	Muncul pesan <i>error</i> "Rule belum disimpan, silakan tekan tombol Simpan Rule".

F Rancangan Uji Coba *Form* Tabel Keputusan

Form Tabel Keputusan ini digunakan untuk memasukkan konklusi pada *decision table* (tabel keputusan) dan mereduksi *decision table* tersebut. Rancangan uji coba *form* Tabel Keputusan ini dapat dilihat pada Tabel 3.38.

Tabel 3.38 Rancangan Uji Coba *Form* Tabel Keputusan

No	Tujuan	Input	Output yang diharapkan
1	Menghasilkan <i>reduced decision table</i>	Memasukkan konklusi dengan nama set yang terpilih kemudian tekan tombol Reduksi	Menghasilkan <i>reduced decision table</i>
2	Menghindari proses <i>reduced decision table</i> tanpa mengisi konklusi	Menekan tombol Reduksi tanpa mengisi konklusi	Muncul Pesan <i>Error</i> "ada kesimpulan yang belum dipilih"

Tabel 3.38 Rancangan Uji Coba *Form* Tabel Keputusan (Lanjutan)

No	Tujuan	Input	Output yang diharapkan
3	Menghindari menekan tombol OK dan Batal sebelum mereduksi table	Mengisi konklusi dan tidak menekan tombol Reduksi	Tombol OK dan Batal tidak aktif.
4	Menghindari menekan tombol Reduksi setelah proses reduksi tabel dilakukan	Mengisi semua konklusi dan Menekan tombol Reduksi Tabel	Tombol OK aktif dan Tombol Reduksi tidak aktif

G Rancangan Uji Coba *Form* Solusi

Form Solusi ini digunakan untuk mengisi solusi dari setiap jenis penyakit tanaman kopi yang ada. Untuk mengisi solusi/kesimpulan tersebut, pengguna harus mengklik pada tabel solusi yang ada kemudian mengisi data solusinya pada *form* Isi Solusi. Contoh data uji coba *form* Solusi ini dapat dilihat pada Tabel 3.39 dan rancangan uji coba *form* Solusi ini dapat dilihat pada Tabel 3.40.

Tabel 3.39 Data Uji Coba *Form* Solusi

Nama Penyakit	Penyebab	Gejala	Tindakan Pengendalian
Cendawan Akar Coklat	Jamur <i>Phellinus noxius</i>	<ol style="list-style-type: none"> Daunnya menguning, layu dan gugur. Pada akar tertutup kerak yang terdiri atas butir-butir tanah yang melekat sangat kuat, sehingga tidak dapat terlepas. Di antara butir-butir tanah tersebut tampak jaringan jamur yang berwarna coklat tua sampai coklat kehitaman. Akar menjadi busuk, kering dan lunak. 	<ol style="list-style-type: none"> Melakukan pembongkaran pada tanaman yang sakit, sisa-sisa akar diambil dan dibakar Membuat saluran isolasi di tempat yang terinfeksi Melakukan peremajaan, dengan membongkar tanaman yang sudah tua hingga tidak dijumpai tunggul pohon-pohon tua.

Tabel 3.40 Rancangan Uji Coba *Form Solusi*

No	Tujuan	Input	Output yang diharapkan
1	Mengisi data Solusi	Memilih nama penyakit “Cendawan Akar Coklat “ pada <i>form Solusi</i> , isi data pada Tabel 3.38 kemudian tekan tombol OK	Solusi terisi sesuai dengan data yang dimasukkan
2	Menyimpan hasil pengisian data solusi	Memilih nama penyakit pada <i>form Solusi</i> , isi data pada Tabel 3.38 pada <i>form Isi Solusi</i> , tekan tombol OK	Muncul Pesan “Data Solusi berhasil diedit”
3	Menghindari <i>field</i> yang kosong ketika menekan tombol OK	Menekan tombol OK tanpa mengisi <i>field</i> yang ada	Muncul pesan <i>Error</i> “ Isian tidak boleh ada yang kosong”

H Rancangan Uji Coba *Form Konsultasi*

Form Konsultasi ini digunakan untuk melakukan konsultasi diagnosis penyakit dengan sistem. Rancangan uji coba *Form* Konsultasi ini dapat dilihat pada Tabel 3.41.

Tabel 3.41 Rancangan Uji Coba *Form Konsultasi*

No	Tujuan	Input	Output yang diharapkan
1	Menghindari proses diagnosis sebelum memilih gejala penyakit	Menekan tombol Diagnosis tanpa memilih gejala penyakit	Muncul Pesan <i>Error</i> “Pilih Gejala penyakit terlebih dahulu”
2	Menyaring daftar pertanyaan yang bersesuaian	Memilih pada <i>comboBox</i> yang tersedia	Daftar pertanyaan akan tersaring sesuai dengan pilihan.
3	Menampilkan pesan saat menekan tombol Keluar setelah memilih gejala penyakit	Memilih gejala-gejala penyakit pada <i>form</i> Konsultasi dan menekan tombol Keluar	Muncul Pesan <i>error</i> “Anda yakin akan membatalkan konsultasi?” jika menjawab <i>yes</i> maka keluar dari <i>form</i> , jika <i>cancel</i> maka melanjutkan proses konsultasi.
4	Melakukan proses Diagnosis penyakit	Memilih pertanyaan	Menghasilkan diagnosis penyakit berupa nama penyakit sesuai dengan gejala-gejala yang dipilih.

Tabel 3.41 Rancangan Uji Coba *Form* Konsultasi (Lanjutan)

No	Tujuan	Input	Output yang diharapkan
5	Menampilkan <i>Form</i> detil penyakit beserta dengan penjelasannya	menekan <i>link</i> Lihat Detil pada <i>form</i> Hasil Konsultasi	Form Detil Penyakit beserta dengan penjelasannya
6	Menampilkan laporan Hasil konsultasi ketika memilih OK	Memilih Tombol OK pda <i>form</i> detil Penyakit	Laporan yang ditampilkan adalah laporan dari hasil konsultasi yang baru saja dilakukan.

I Rancangan Uji Coba *Form* Ubah Password

Form Ubah Password ini digunakan untuk mengubah *password* dari user umum yang ingin mengubah *password*. Rancangan uji coba *Form* Ubah Password ini dapat dilihat pada Tabel 3.42.

Tabel 3.42 Rancangan Uji Coba *Form* Ubah Password

No	Tujuan	Input	Output yang diharapkan
1	Menghindari <i>field</i> yang kosong ketika mengubah <i>password</i>	Menekan tombol OK tanpa mengisi <i>fields</i> yang ada	Muncul Pesan <i>Error</i> "Password tidak boleh kosong"
2	Menghindari memasukan <i>password</i> baru sama dengan <i>password</i> yang lama	Mengisi <i>password</i> yang baru sama persis dengan <i>password</i> yang lama, kemudian tekan tombol OK	Muncul Pesan <i>Error</i> "password yang baru tidak boleh sama dengan password yang lama"
3	Menghindari memasukan <i>password</i> yang beda pada <i>field</i> ulangi <i>password</i> dengan <i>password</i> pada <i>field</i> <i>password</i> baru	Mengisi data pada <i>field</i> Ulangi <i>Password</i> yang tidak sesuai dengan <i>Password</i> baru kemudian menekan tombol OK	Muncul Pesan <i>Error</i> "Password anda tidak sesuai, silakan masukan ulang password sesuai dengan password baru anda"
4	Proses ubah <i>password</i> berhasil dilakukan	Mengisi <i>password</i> sesuai dengan <i>fields</i> yang ada, kemudian tekan tombol OK.	Muncul pesan "Password anda telah diubah"

3.3.2 Perancangan Uji Coba Kasus Jenis Penyakit Tanaman Kopi

Selain melakukan uji coba terhadap beberapa menu utama, sistem juga dievaluasi dengan melakukan konsultasi untuk mendiagnosis penyakit dengan

metode *forward chaining* berdasarkan kasus pada beberapa jenis penyakit tanaman kopi. Perancangan dimulai dari melakukan konsultasi samapai menghasilkan hasil diagnosis. Jenis penyakit tanaman kopi yang digunakan adalah penyakit Cendawan Akar Coklat, penyakit Akar Putih dan penyakit Busuk Akar.

A Rancangan Uji Coba Kasus Untuk Penyakit Cendawan Akar Coklat

Uji coba dilakukan untuk menguji sistem dalam mendiagnosis penyakit ini sesuai dengan gejala-gejalanya. Data uji coba yang digunakan untuk melakukan konsultasi dapat dilihat pada Tabel 3.43.

Tabel 3.43 Data Uji Coba Untuk Melakukan Konsultasi Dengan Kasus Untuk Penyakit Cendawan Akar Coklat

Nama Parameter	Jawabannya
Akar tertutup kerak yang terdiri atas butir-butir tanah	Ya
Akar membusuk	Ya
Akar mengering	Ya
Tampak Jaringan Jamur berwarna coklat tua pada akar	Ya
Akar menjadi Lunak	Ya
Daun Menguning	Ya
Daun layu	Ya
daun gugur	Ya

B Rancangan Uji Coba Kasus Untuk Penyakit Akar Putih

Uji coba dilakukan untuk menguji sistem dalam mendiagnosis penyakit ini sesuai dengan gejala-gejalanya. Data uji coba yang digunakan untuk melakukan konsultasi dapat dilihat pada Tabel 3.44.

Tabel 3.44 Data Uji Coba Untuk Melakukan Konsultasi Dengan Kasus Untuk Penyakit Akar Putih

Nama Parameter	Jawabannya
Akar membusuk	Ya
Akar diselubungi selaput miselium jamur mirip jala putih	Ya
Daun Menguning	Ya
Daun gugur	Ya
Berbuah sebelum umurnya	Ya
Tanaman menjadi roboh karena infeksi berat	Ya

C Rancangan Uji Coba Kasus Untuk Penyakit Busuk Akar

Uji coba dilakukan untuk menguji sistem dalam mendiagnosis penyakit ini sesuai dengan gejala-gejalanya. Data uji coba yang digunakan untuk melakukan konsultasi dapat dilihat pada Tabel 3.45.

Tabel 3.45 Data Uji Coba Untuk Melakukan Konsultasi Dengan Kasus Untuk Penyakit Busuk Akar

Nama Parameter	Jawabannya
Akar mengeluarkan cairan kental berwarna coklat kekuningan	Ya
Muncul retakan pada batang	Ya
Daun Menguning	Ya
Daun layu	Ya
Tampak tudung jamur yang bergerombol di dasar tanaman	Ya