



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODE PENELITIAN & PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang metode penelitian yang dilakukan dan langkah-langkah dalam perancangan sistem yang meliputi analisis sistem dan perancangan sistem.

3.1 Metode Penelitian

Metode yang akan digunakan dalam penelitian tugas akhir ini terdiri dari.

1. Studi Literatur

Melakukan tinjauan pustaka melalui hasil penelitian orang lain maupun artikel-artikel lainnya yang relevan serta mempelajari teori-teori tentang sistem pakar dan teknik yang tepat untuk diimplementasikan dalam aplikasi untuk membuat sistem pakar ini.

2. Observasi

Melakukan pengamatan dan pengujian terhadap beberapa aplikasi sistem pakar yang sudah ada di internet. Dengan melakukan hal ini, akan didapat pengetahuan tambahan mengenai bentuk sistem yang sudah ada dan telah diimplementasikan.

3. Analisis dan Perancangan Sistem

Melakukan analisis terhadap masalah yang ditemukan serta menentukan batasan-batasan dan kebutuhan yang diperlukan dalam membangun aplikasi untuk membuat sistem pakar. Kemudian

melakukan perancangan sistem yang dapat menyelesaikan masalah yang ditemukan.

4. Implementasi

Mengimplementasikan rancangan sistem yang telah dirancang sebelumnya untuk membangun aplikasi ini.

5. Pengujian

Melakukan pengujian dengan menguji beberapa data dan menganalisis terhadap hasil dari aplikasi yang dibuat.

6. Perbaikan/penambahan data

Setelah sistem diuji coba dan terjadi beberapa kekurangan/kelemahan, maka perlu diperbaiki lagi sehingga sistem yang dibuat menjadi lebih baik.

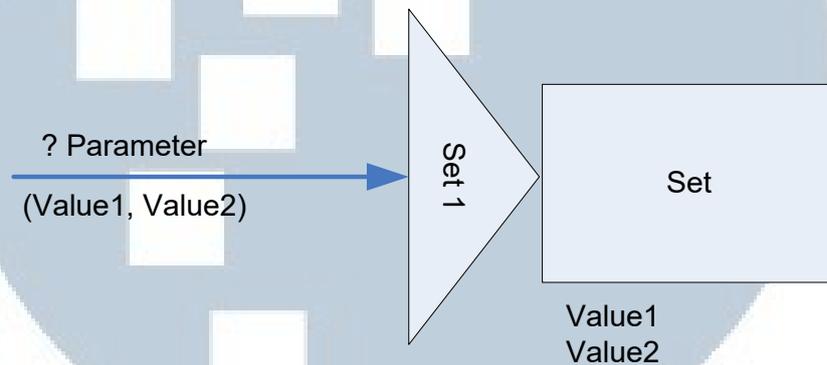
3.2 Analisis Sistem

Dalam rancang bangun aplikasi yang dapat membuat sistem pakar untuk penyelesaian kasus sistem berbasis aturan diperlukan langkah awal, yaitu menganalisis sistem. Pada bagian ini diuraikan kebutuhan dasar implementasi aplikasi ini dan juga akan dibahas tentang segala aturan yang diperlukan agar dapat membentuk suatu sistem berbasis aturan yang memenuhi persyaratannya.

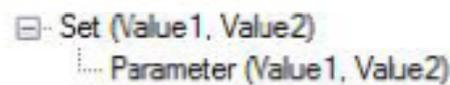
1. Dependency Diagram

Langkah pertama yang dilakukan dalam membuat sistem berbasis aturan adalah membuat *dependency diagram*. Di dalam aplikasi ini, *dependency diagram* direpresentasikan dalam bentuk *tree view*.

Dependency diagram diperlukan untuk menggambar susunan parameter beserta valuenya. Parameter yang memiliki cabang di bawahnya disebut sebagai set. Di dalam sebuah *dependency diagram*, diperlukan minimal dua buah parameter dan salah satunya berfungsi sebagai set dan nama parameter yang berada dalam satu set yang sama tidak boleh sama.



Gambar 3.1 Syarat minimal *Dependency diagram*



Gambar 3.2 Syarat minimal *Dependency diagram* dalam *tree view*

Sebuah parameter harus memiliki minimal dua buah value yang berbeda. *User* yang ingin membuat sistem berbasis aturan harus memasukkan sebuah pertanyaan pada setiap parameter yang nantinya akan ditampilkan kepada *user*. Pertanyaan tersebut digunakan untuk memperoleh fakta yang akan digunakan oleh sistem untuk mencari kesimpulan dan solusi dari kasus yang ada. Selain itu *user* juga harus mengisi solusi atau saran pada saat mengisi value dari parameter yang paling atas pada *dependency diagram*. Sistem tidak bisa memeriksa

kebenaran solusi atau saran yang dimasukkan oleh *user* untuk setiap kesimpulan.

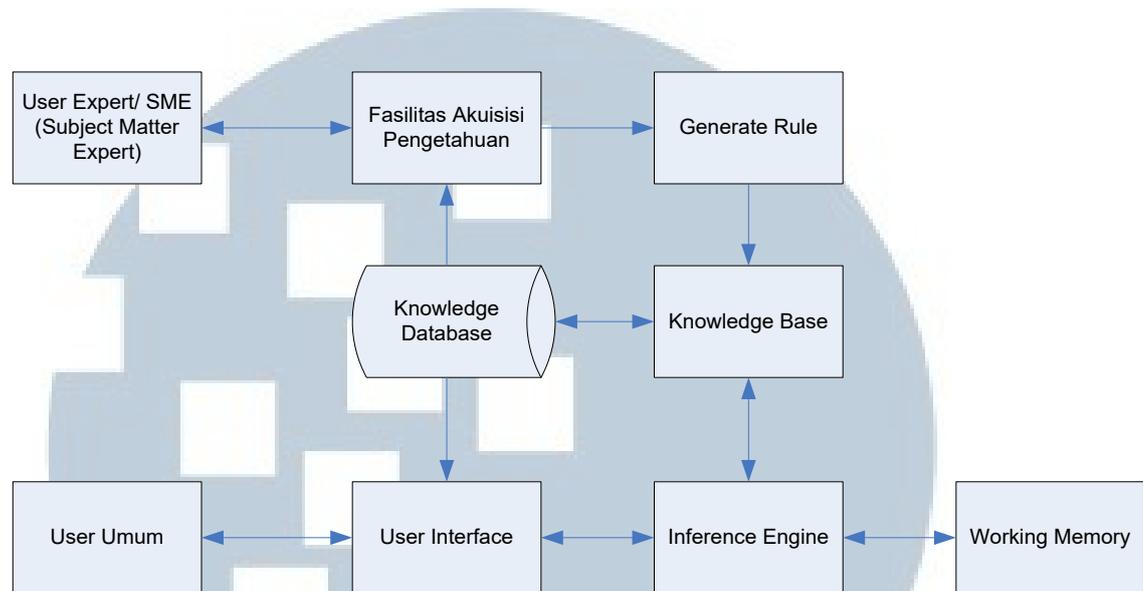
2. Decision Table

Setelah *dependency diagram* selesai dibuat, dilanjutkan dengan mengisi *decision table* yang direpresentasikan dengan *datagrid view*. Setiap set dalam *dependency diagram* akan ditampilkan dalam *decision table* tersendiri. *Decision table* dibuat secara otomatis oleh sistem setelah *user* selesai membuat *dependency diagram*. *User* akan mengisi *decision table* dengan memilih pilihan dalam bentuk *combobox* yang nilainya diambil dari value-value pada set yang diinput oleh *user* pada saat membuat *dependency diagram*. Sistem kemudian membuat *rule-rule* yang diambil dari *decision table* yang telah selesai diisi.

Hal utama yang perlu diingat adalah, *user expert* dalam aplikasi ini seharusnya adalah seorang pakar atau memiliki kompetensi dan pengalaman dalam kasus yang dibuatnya. Selain itu juga wajib memiliki pengetahuan dasar tentang sistem berbasis aturan. Hal ini diperlukan dalam pembuatan *dependency diagram* dan *decision table*.

3.3 Perancangan Sistem

Perancangan sistem aplikasi ini terdiri dari rancangan arsitektur dan perancangan proses yang akan digambarkan dalam bentuk *flowchart*. Rancangan arsitektur aplikasi untuk membuat sistem berbasis aturan dapat dilihat pada Gambar 3.3.



Gambar 3.3 Rancangan Arsitektur Sistem Berbasis Pengetahuan
(Sumber: Arisudana, 2010)

Penjelasan dari rancangan arsitektur sistem berbasis pengetahuan adalah.

- a. Fasilitas Akuisisi Pengetahuan adalah media yang digunakan oleh pakar untuk merepresentasikan pengetahuannya ke dalam *dependency diagram* dan *decision table*.
- b. *Generate Rule* adalah komponen yang berfungsi untuk membuat *rule-rule* dari *decision table* yang telah diisi semuanya.
- c. *Knowledge Base* adalah komponen yang berisi fungsi-fungsi untuk menyimpan ilmu pengetahuan berupa *rules* ke dalam database.

Selain itu, komponen ini juga berisi fungsi untuk mengambil ilmu berupa *rules* dari database sesuai *query* yang dilakukan.

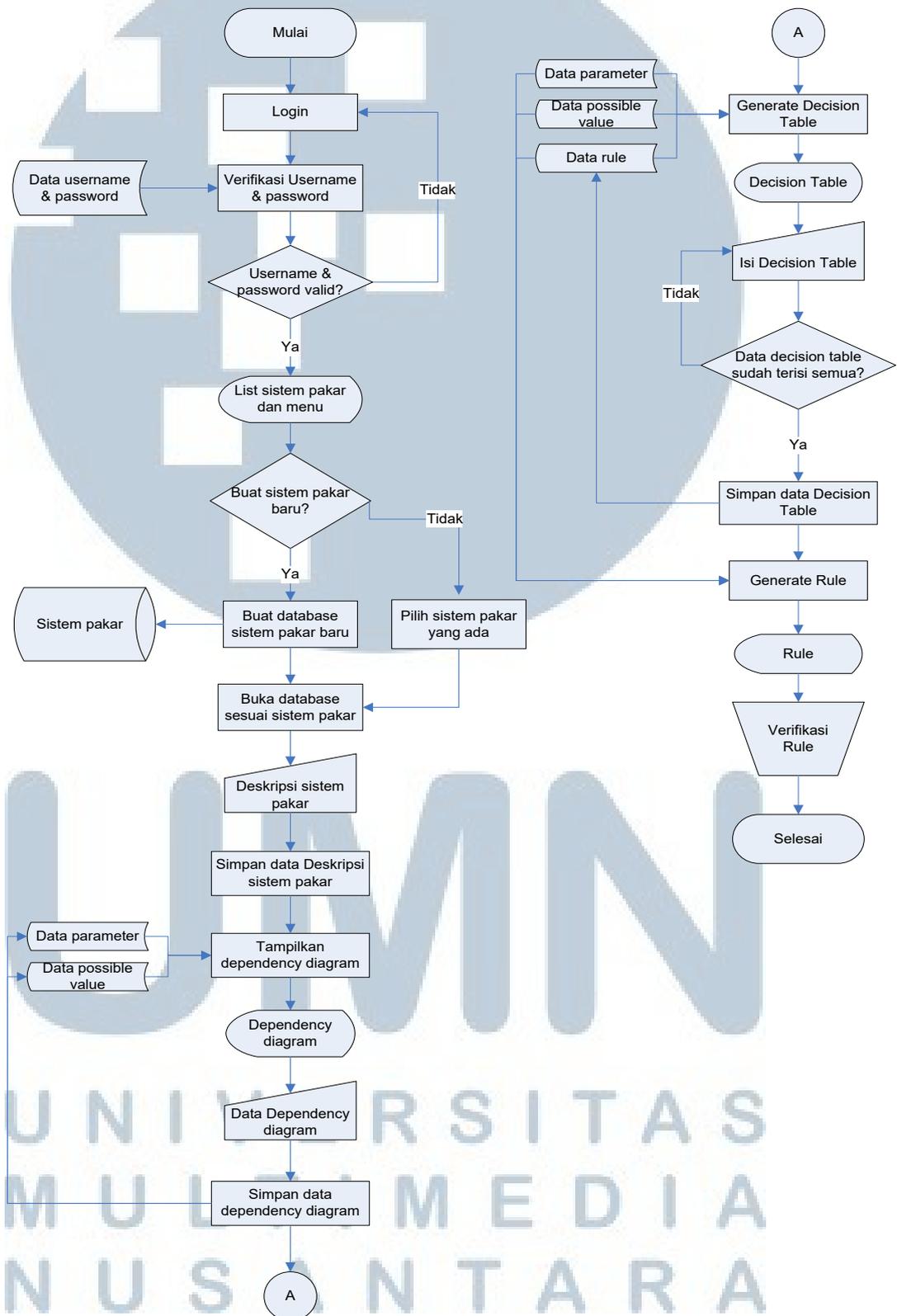
- d. *User Interface* adalah komponen yang berisi fungsi-fungsi untuk interaksi antara *user umum* dengan aplikasi.

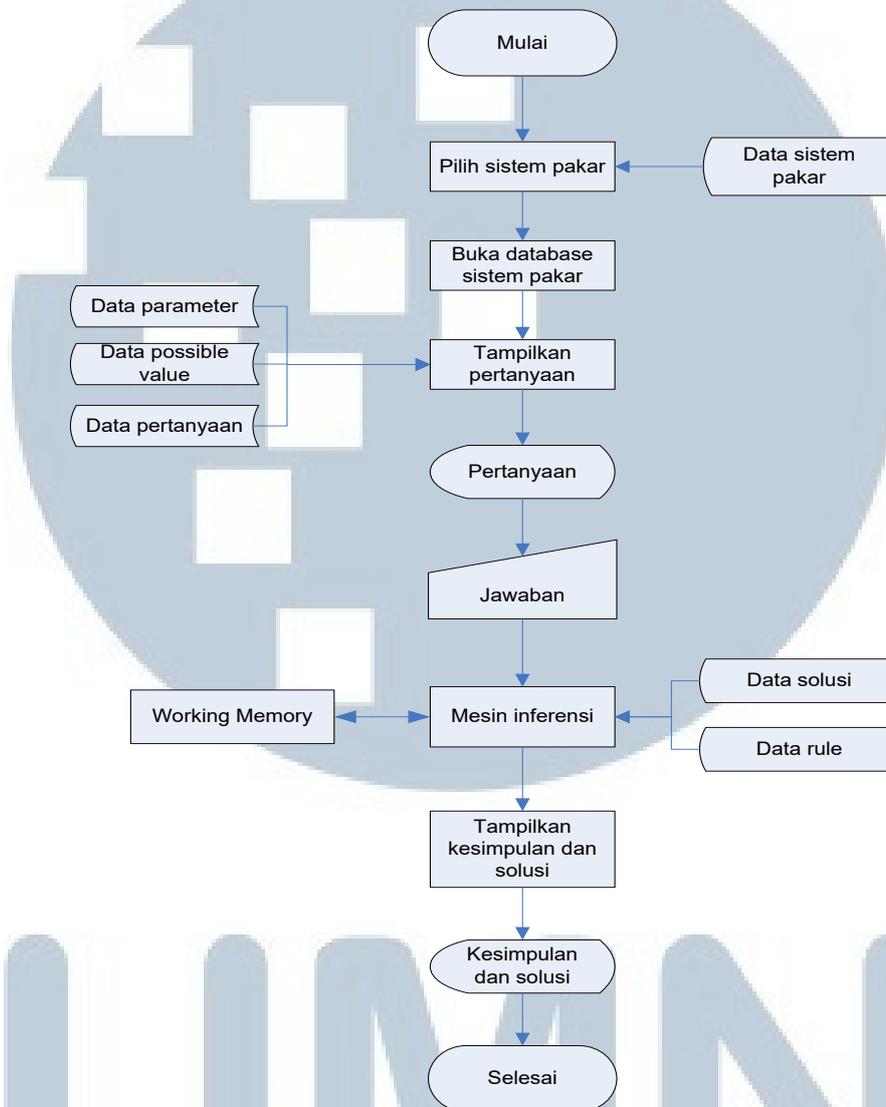
- e. *Inference Engine* adalah komponen yang berisi fungsi-fungsi untuk proses pemecahan masalah menggunakan metode *forward chaining* dan kemudian menghasilkan solusi atau kesimpulan.
- f. *Working Memory* merupakan tempat penyimpanan fakta-fakta yang diketahui dari jawaban *user* umum atas pertanyaan yang diberikan oleh sistem.
- g. *Output* adalah media yang digunakan untuk menampilkan hasil konsultasi yang dilakukan oleh *user* umum, sedangkan *output* untuk pakar atau *user expert* adalah *rules* yang telah dibuat.

Setelah melakukan analisis terhadap sistem dan membuat rancangan arsitektur, maka langkah selanjutnya adalah perancangan proses-proses utama yang terdapat di dalam aplikasi. Proses-proses yang ada di dalam aplikasi ini digambarkan dalam bentuk *Flow Chart*. *Flow Chart* yang ada yaitu *flow chart* untuk *user expert* atau pakar dan *flow chart* untuk *user* umum.

Gambar 3.4 merupakan *flow chart* untuk *user expert* yang menggambarkan tentang proses akuisisi pengetahuan dari seorang pakar. Pengetahuan dari seorang pakar tersebut direpresentasikan dalam bentuk *dependency diagram* dan *decision table*. *User expert* juga bisa melakukan perubahan terhadap data yang telah ada sebelumnya.

Gambar 3.4 Flowchart untuk user expert atau pakar



Gambar 3.5 *Flowchart* untuk *user* umum

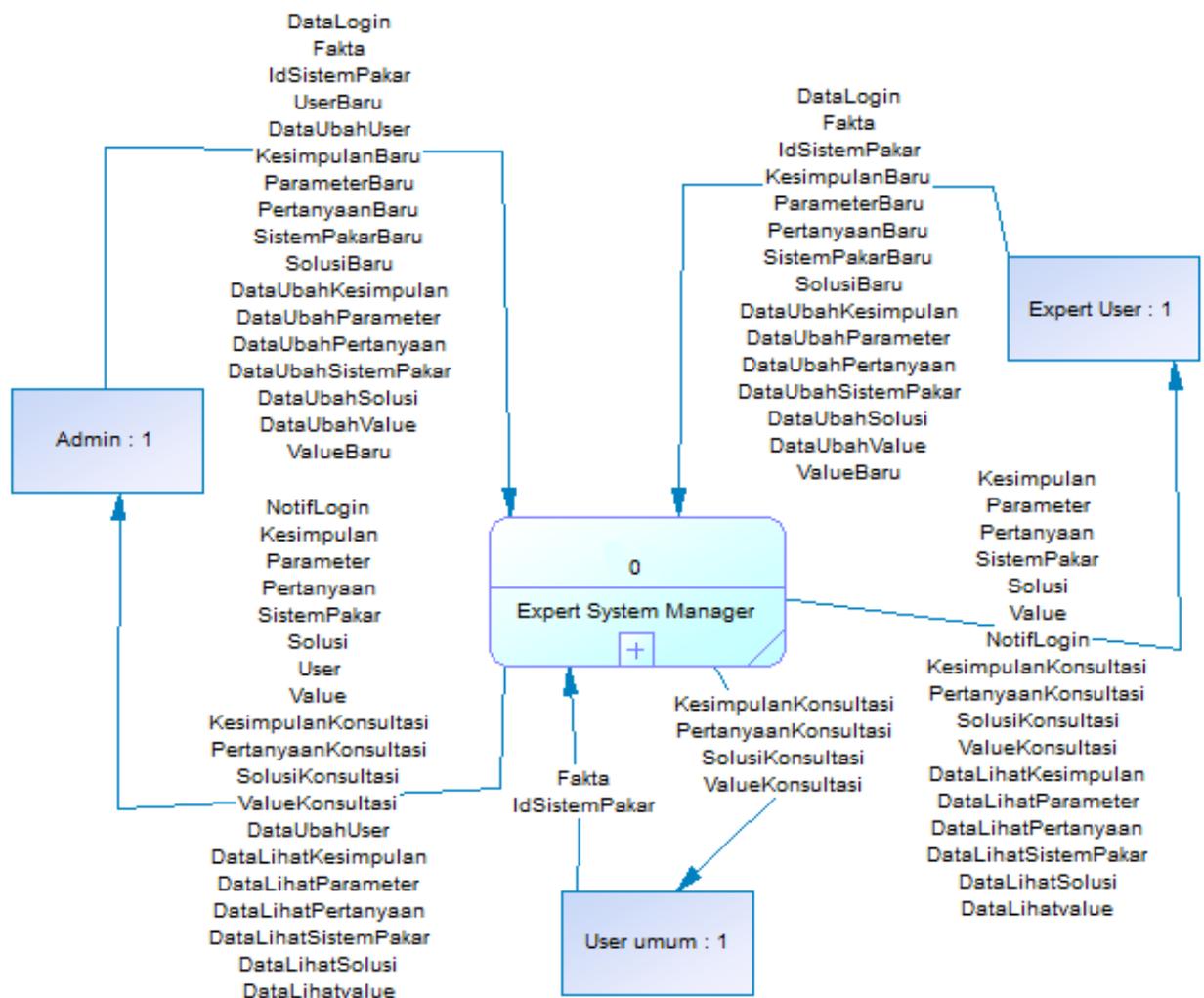
Gambar 3.5 merupakan gambar *flow chart* untuk *user* umum yang

menjelaskan tentang proses yang terjadi saat *user* umum melakukan konsultasi dengan menjawab pertanyaan yang diberikan oleh sistem. *User* umum terlebih dahulu memilih sistem pakar yang ada di dalam *knowledge database*, kemudian sistem akan memberikan pertanyaan dan *user* umum harus menjawab semua pertanyaan yang diberikan. Setelah itu sistem akan

menampilkan kesimpulan dan solusi berdasarkan data yang ada di dalam *knowledge database*.

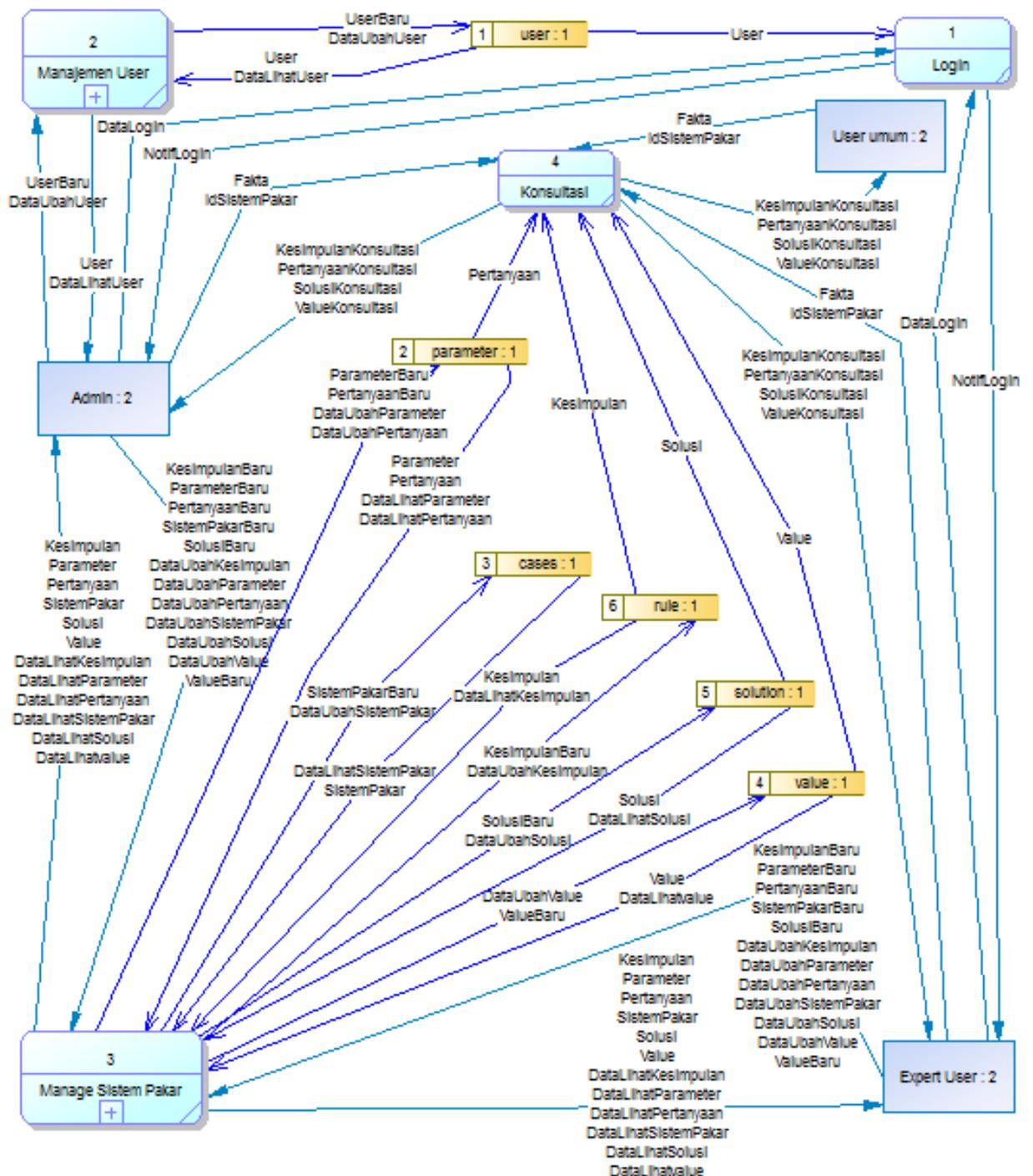
3.4 Data Flow Diagram

Setelah membuat *flowchart*, *Data Flow Diagram* dibuat untuk menggambarkan aliran-aliran data yang ada di dalam sistem. Berikut adalah gambar diagram-diagram *Data Flow Diagram* dari aplikasi *Expert System Manager*.



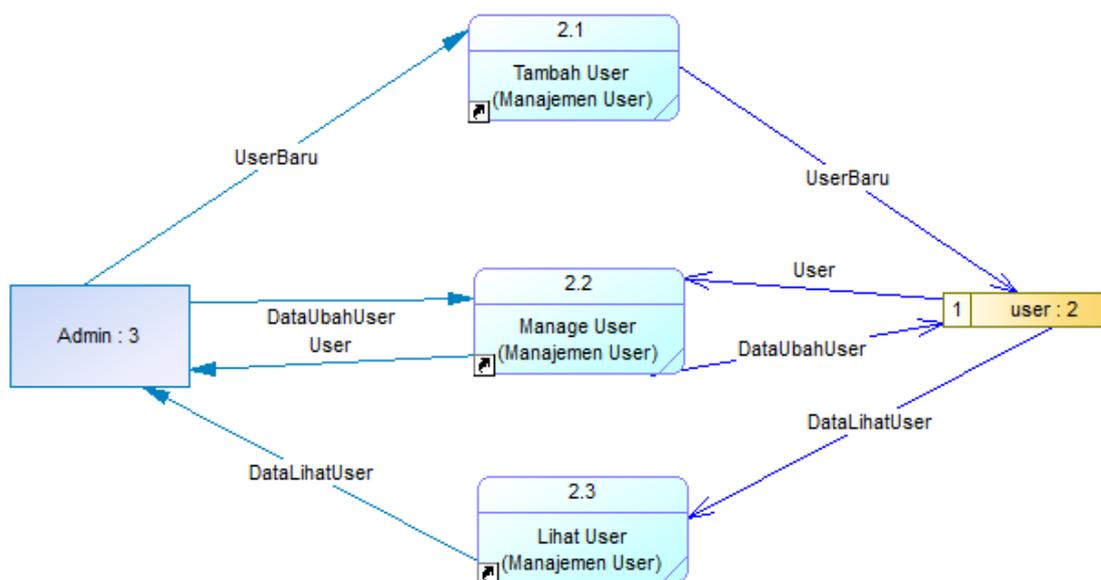
Gambar 3.6 Context Diagram *Expert System Manager*

Pada gambar tersebut menerangkan bahwa sistem berinteraksi dengan tiga sumber data atau tujuan data, yaitu Admin, *user* umum dan *user expert*. Tanda panah menunjukkan keluaran dan masukkan sistem.



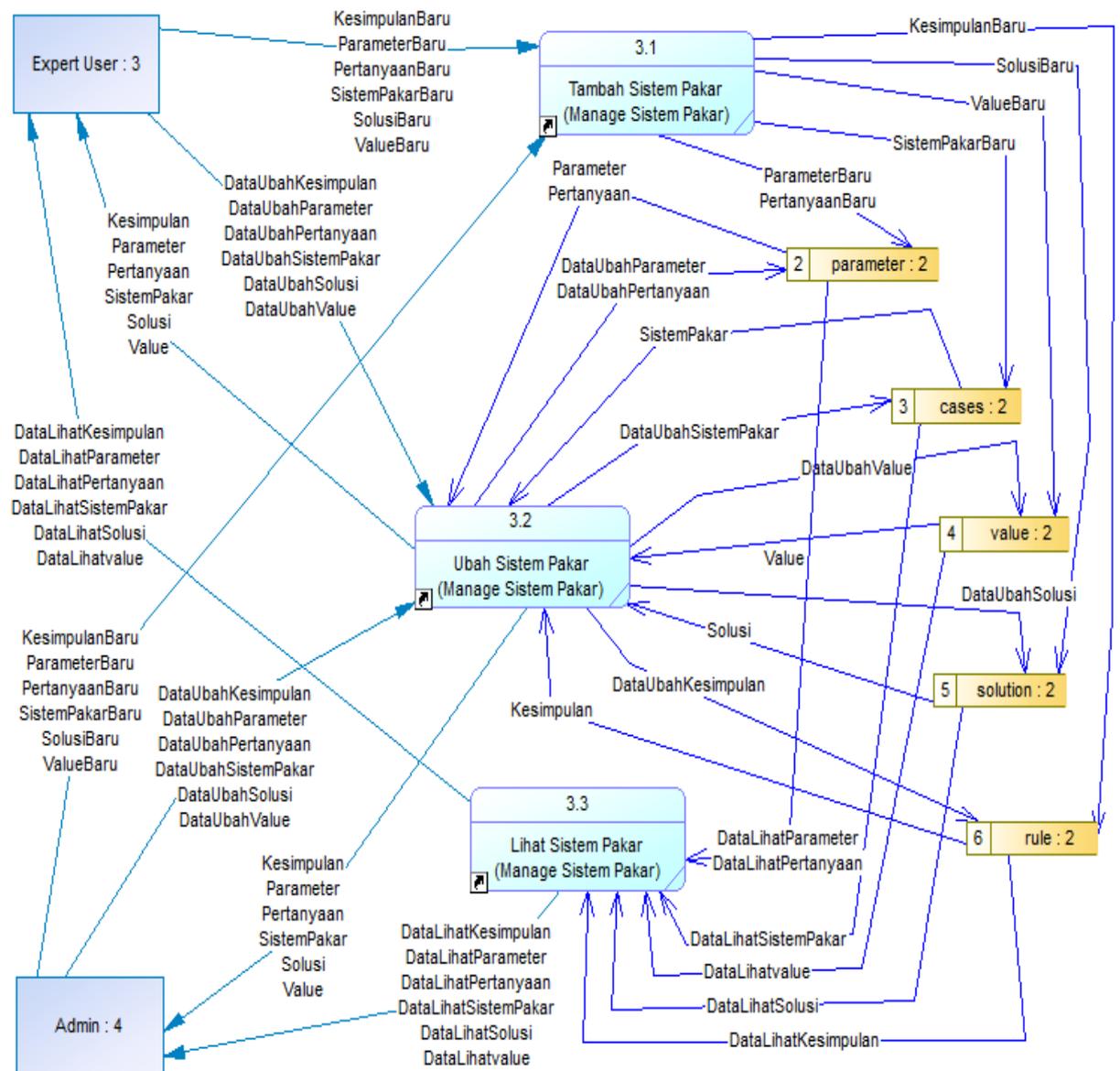
Gambar 3.7 Level 1 Expert System Manager

Pada gambar 3.7, terdapat 4 proses yaitu Login, Manajemen *User*, Manage Sistem Pakar dan Konsultasi. Keempat proses tersebut merupakan proses yang telah diturunkan dari *Context Diagram*. Terdapat 6 data store yang digunakan sebagai tempat penyimpanan data dan pengambilan data. Entitas *Admin* melakukan transaksi data dengan semua proses yang ada, entitas *expert user* melakukan transaksi data dengan semua proses kecuali Manajemen *User*, sedangkan entitas *user* umum hanya melakukan transaksi dengan proses Konsultasi.



Gambar 3.8 Level 2 Diagram rinci Manajemen User

Gambar di atas merupakan turunan dari proses Manajemen *User* yang terdapat pada Level 1 yang terbagi menjadi 3 proses, yaitu Tambah *User*, Manage *User* dan Lihat *User*. Setiap proses melakukan transaksi dengan entitas *Admin* dan *data store* user.

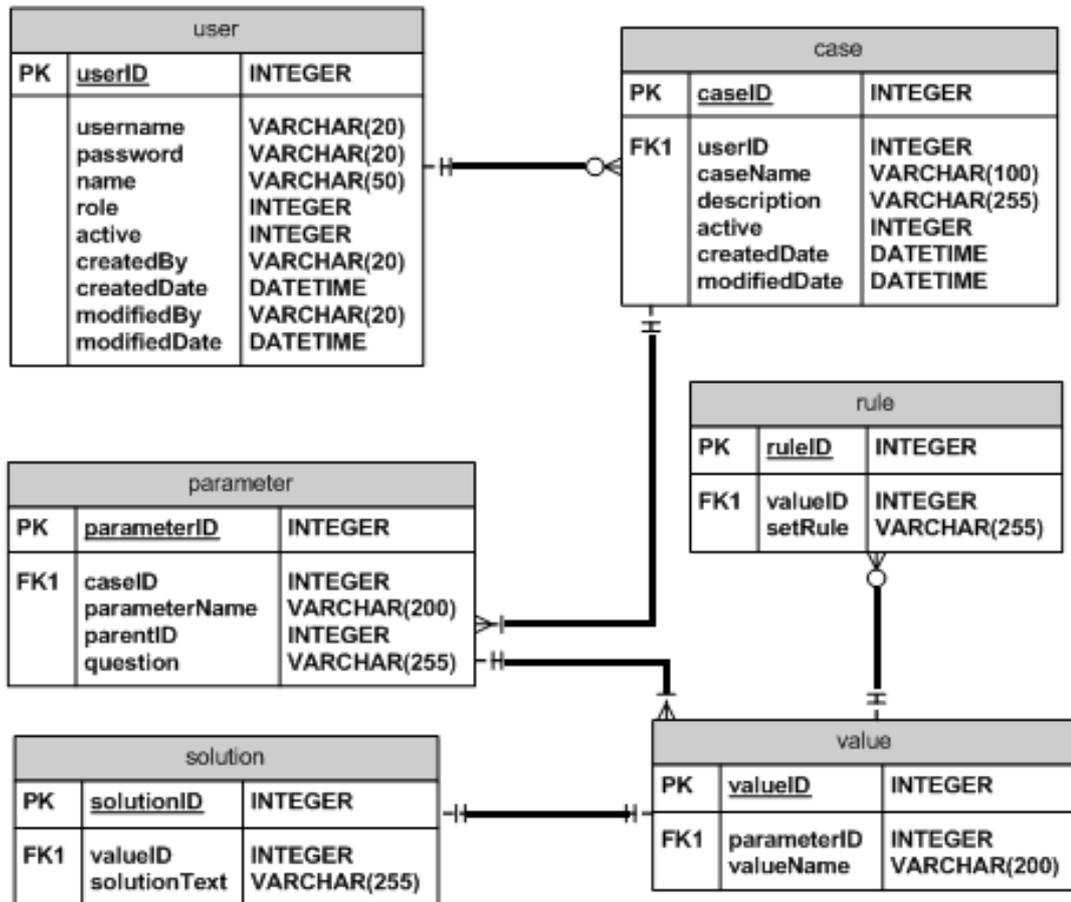


Gambar 3.9 Level 2 Diagram rinci Manage Sistem Pakar

Gambar di atas merupakan turunan dari proses Manage Sistem Pakar yang terdapat pada Level 1 yang terbagi menjadi 3 proses, yaitu Tambah Sistem Pakar, Ubah Sistem Pakar dan Lihat Sistem Pakar. Setiap proses melakukan transaksi dengan entitas *Admin*, entitas *Expert User* dan 5 data store yang terdiri dari *parameter*, *cases*, *value*, *solution*, *rule*.

3.5 Pemodelan Database

3.5.1. Entity Relationship Diagram



Gambar 3.10 Entity Relationship Diagram

Uraian hubungan antar entitas dari Entity Relationship Diagram aplikasi ini dapat dilihat pada tabel berikut ini :

Tabel 3.1 Uraian Entity Relationship Diagram

No	Entity name	Multi-plicity	Relationship	Multi-plicity	Entity name
1	user	1..1	memiliki/dimiliki	0..N	case
2	case	1..1	memiliki/dimiliki	2..N	parameter
3	parameter	1..1	memiliki/dimiliki	2..N	value
4	value	1..1	memiliki/dimiliki	0..N	rule
5	value	1..1	memiliki/dimiliki	1..1	solution

3.5.2. Struktur Tabel

Dengan menggunakan *Entity Relationship Diagram* yang sudah dibuat, maka dibentuklah database dengan menggunakan *My Structured Query Language (MySQL)* dan terbentuklah 6 tabel yang meliputi :

a. Tabel user

Nama Tabel : user

Kegunaan : menyimpan data mengenai user yang akan menggunakan aplikasi ini

Primary Key : userID

Foreign Key : -

Tabel 3.2 Tabel user

No	Field Name	Type	Length	Information
1	userID	INTEGER	11	Id user berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi userID
2	username	VARCHAR	20	Username ketika <i>user</i> login ke dalam aplikasi
3	password	VARCHAR	20	kata sandi untuk melakukan login
4	name	VARCHAR	50	Name adalah nama dari <i>user</i>
5	role	INTEGER	11	Role adalah aturan atau mode berupa <i>integer</i> untuk <i>user</i> mendapatkan hak-hak tertentu
6	active	INTEGER	11	Active menjadi 1 ketika <i>user</i> dengan <i>userID</i> tertentu masih aktif dan akan menjadi 0 ketika <i>userID</i> ini dihapus
7	createBy	VARCHAR	20	createBy adalah <i>username</i> yang membuat <i>userID</i> tertentu
8	createDate	DATETIME	-	createDate adalah waktu dari pembuatan <i>userID</i>
9	modifiedBy	VARCHAR	20	createBy adalah <i>username</i> yang mengubah <i>userID</i> tertentu
10	modifiedDate	DATETIME	-	createDate adalah waktu dari perubahan <i>userID</i>

b. Tabel cases

Nama Tabel : cases

Kegunaan : menyimpan data dari kasus sistem berbasis aturan.

Primary Key : caseID

Foreign Key : userID

Tabel 3.3 Tabel cases

No	Field Name	Type	Length	Information
1	caseID	INTEGER	11	Id case berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi caseID
2	userID	INTEGER	11	Id <i>User</i> yang sedang login di dalam aplikasi yang membuat <i>caseID</i> . Diambil dari tabel user
3	caseName	VARCHAR	100	Nama dari kasus sistem berbasis aturan
4	description	VARCHAR	255	Deskripsi mengenai kasus yang dibuat
5	active	INTEGER	11	Active menjadi 0 ketika <i>caseID</i> dihapus. 1 ketika <i>caseID</i> belum dihapus
6	createdDate	DATETIME	-	createDate adalah waktu dari pembuatan <i>caseID</i>
7	modifiedDate	DATETIME	-	createDate adalah waktu dari perubahan <i>caseID</i>

c. Tabel parameter

Nama Tabel : parameter

Kegunaan : menyimpan data parameter berdasarkan *tree view* yang dibuat

Primary Key : parameterID

Foreign Key : caseID

Tabel 3.4 Tabel parameter

No	Field Name	Type	Length	Information
1	parameterID	INTEGER	11	Id parameter berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi parameterID
2	caseID	INTEGER	11	Id kasus sistem berbasis aturan dimana parameter digunakan. Diambil dari tabel cases
3	parameterName	VARCHAR	200	Nama dari parameter di dalam <i>tree view</i>
4	parentID	VARCHAR	255	parentID di-generate oleh sistem sebagai induk dari parameter sesuai dengan susunan pada <i>tree view</i> . Induk paling atas pada <i>tree view</i> bernilai 0
5	question	INTEGER	11	Pertanyaan yang akan ditampilkan untuk <i>user</i> umum ketika proses konsultasi dilakukan

d. Tabel value

Nama Tabel : value

Kegunaan : menyimpan data value dari masing-masing parameter

Primary Key : valueID

Foreign Key : parameterID

Tabel 3.5 Tabel value

No	Field Name	Type	Length	Information
1	valueID	INTEGER	11	Id value berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi valueID
2	parameterID	INTEGER	11	Id parameter yang memiliki <i>valueID</i> tertentu. Diambil dari tabel parameter
3	valueName	VARCHAR	200	Nama dari value

e. Tabel rule

Nama Tabel : rule

Kegunaan : menyimpan data rule hasil generate sistem

Primary Key : ruleID

Foreign Key : valueID

Tabel 3.6 Tabel rule

No	Field Name	Type	Length	Information
1	ruleID	INTEGER	11	Id rule berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi ruleID
2	valueID	INTEGER	11	Value yang merupakan kesimpulan untuk suatu aturan. Diambil dari tabel value
3	setRule	VARCHAR	255	Kombinasi aturan yang di- <i>generate</i> otomatis berupa kombinasi ID value-value yang dapat menghasilkan satu kesimpulan

f. Tabel solution

Nama Tabel : solution

Kegunaan : menyimpan data solusi dari setiap kesimpulan akhir sebagai saran atau jalan keluar

Primary Key : solutionID

Foreign Key : valueID

Tabel 3.7 Tabel solution

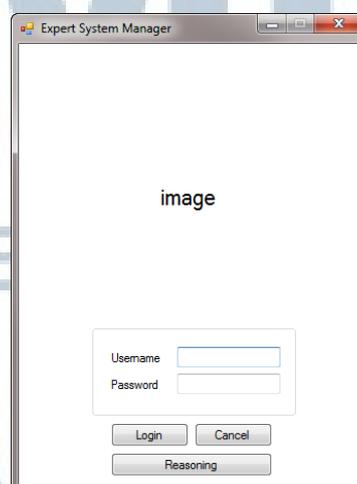
No	Field Name	Type	Length	Information
1	solutionID	INTEGER	11	Id solution berupa <i>auto number</i> sehingga <i>user</i> tidak perlu mengisi solutionID
2	valueID	INTEGER	11	Value dimana solutionID berada. Diambil dari tabel value
3	solutionText	VARCHAR	255	Isi dari solusi yang dimasukkan oleh <i>user</i> .

3.6 Desain User Interface

Setelah desain struktur dibuat, barulah masuk ke tahap desain interface yaitu dengan membuat tampilan tiap-tiap halaman yang akan dibuat di dalam aplikasi. Tampilan-tampilan umum yang dibuat yaitu.

a. Halaman Login

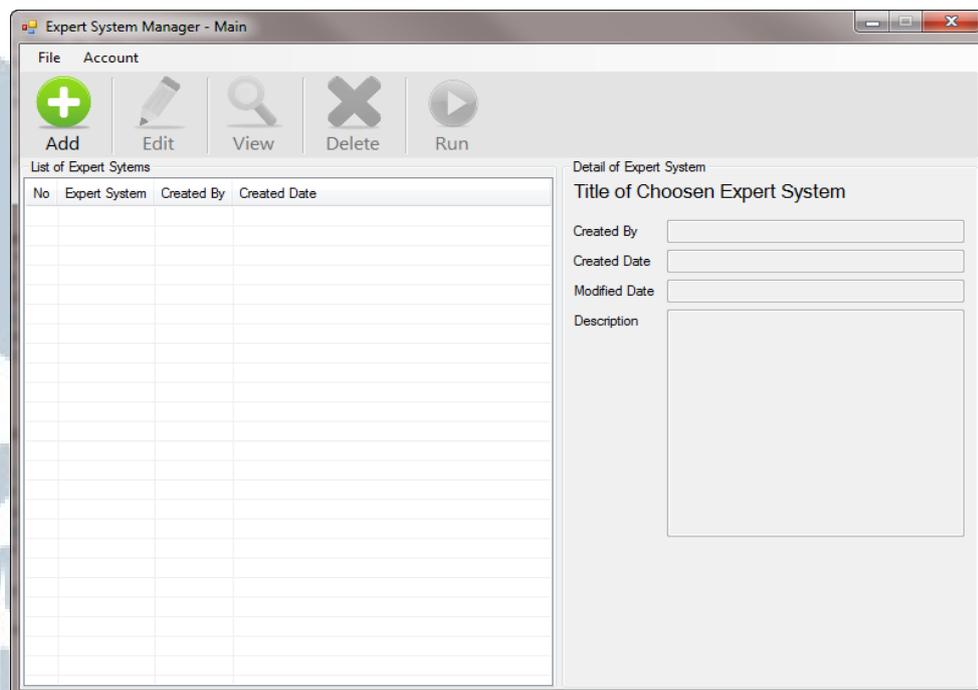
Pada halaman ini user diminta untuk mengisi username dan password untuk masuk ke dalam sistem. Terdapat 2 jenis user yang memiliki hak akses yang berbeda, yaitu user untuk *user expert* dan user untuk admin. Hak akses *user expert* dan admin hampir sama, yang membedakan adalah admin dapat melakukan pengaturan data *user* sedangkan *user expert* tidak. Jika user memasukkan username atau password yang salah, maka akan muncul peringatan. Jika benar, maka akan muncul konfirmasi bahwa user berhasil masuk ke dalam sistem. Sedangkan *user* umum dapat masuk ke dalam sistem dengan menggunakan tombol *reasoning*. *User* umum hanya dapat melakukan konsultasi atau proses *reasoning*.



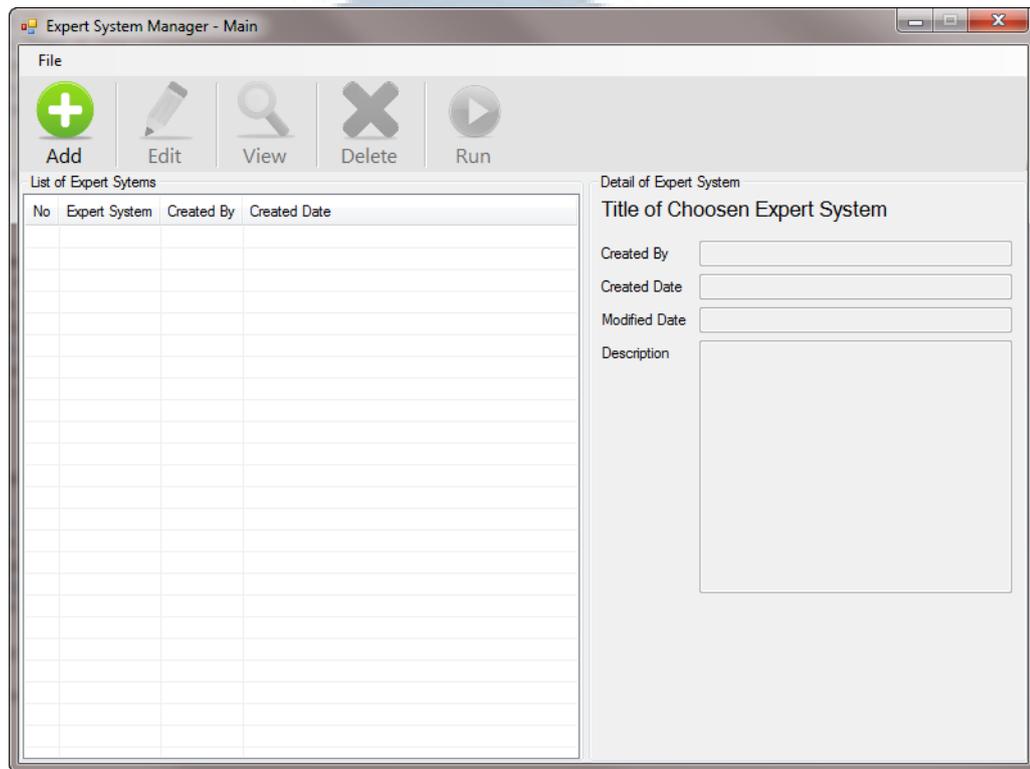
Gambar 3.11 Halaman Login

b. Halaman Main (menu utama)

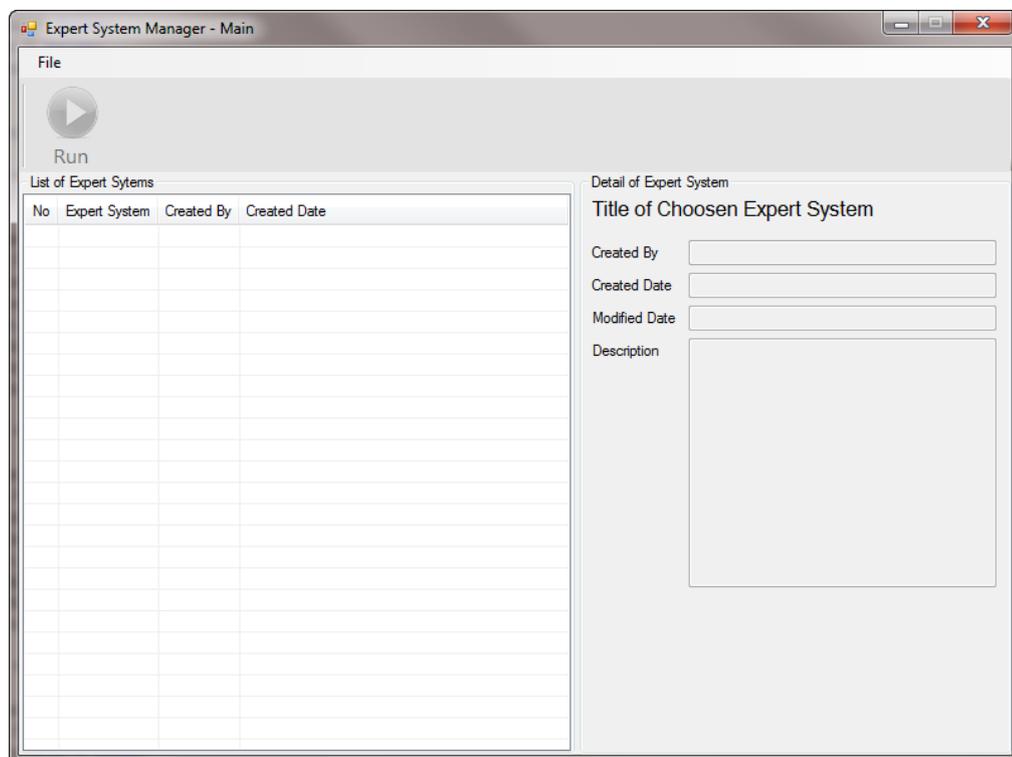
Pada Halaman ini, *user expert* dan admin dapat melihat daftar sistem pakar yang dibuat oleh *user* yang sedang login. Sedangkan *user umum* yang masuk melalui tombol *reasoning* dapat melihat daftar sistem pakar yang dibuat oleh semua *user*. *User expert* dan admin dapat melakukan penambahan, pengubahan, penghapusan sistem pakar atau kasus dengan menggunakan menu-menu yang ada dan juga dapat melakukan proses reasoning dengan menggunakan tombol *run* setelah memilih salah satu sistem pakar atau kasus yang ada di daftar tersebut. Pada tampilan untuk *user expert* tidak terdapat menu *Account*, sedangkan pada admin terdapat menu *Account* yang digunakan untuk mengatur data *user*. Tampilan untuk *user* umum hanya terdapat tombol *run* untuk melakukan proses *reasoning*.



Gambar 3.12 Halaman Main untuk Admin



Gambar 3.13 Halaman Main untuk *User Expert*



Gambar 3.14 Halaman Main untuk *User umum*

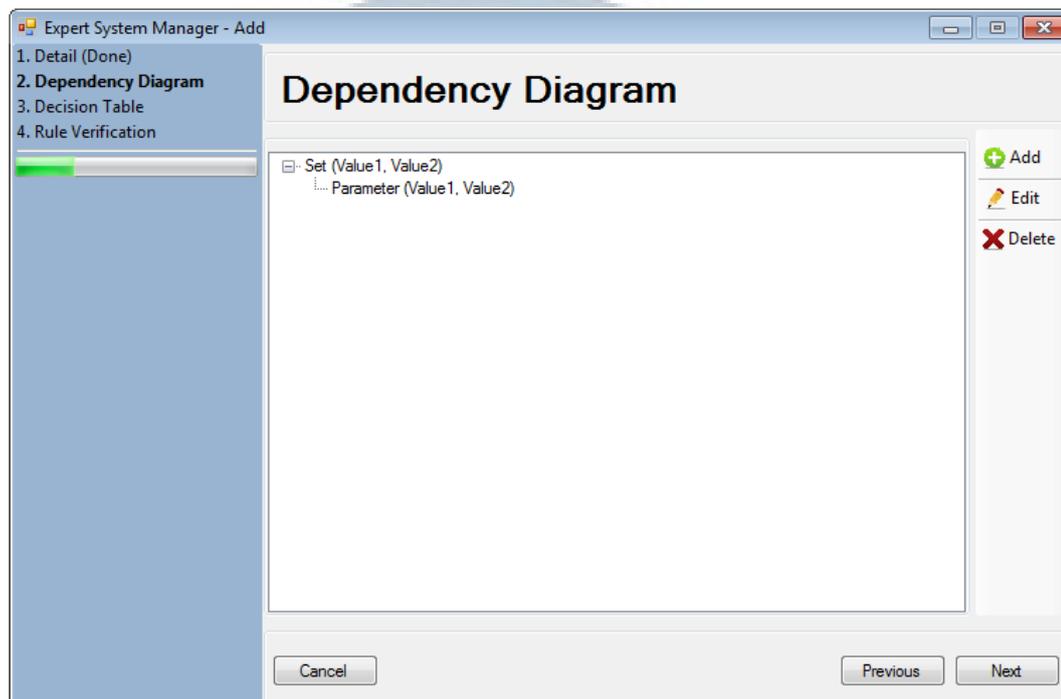
c. Halaman *Add, Edit, View Detail Expert System*

Halaman *Add, Edit, View Detail Expert System* memiliki tampilan yang sama, namun memiliki fungsi dan tingkah laku (*behaviour*) yang berbeda. *Add* untuk melakukan fungsi penambahan data baru, *edit* untuk mengubah data yang sudah ada dan *view* hanya untuk melihat data sehingga *user* tidak dapat mengubah atau mengisi data. Halaman *Add, Edit, View Detail Expert System* digunakan untuk mengatur detail dari sistem pakar seperti nama dari sistem pakar, deskripsi dari sistem pakar, pembuat sistem pakar, waktu sistem pakar dibuat dan waktu sistem pakar diubah. Tombol *next* digunakan untuk menuju ke halaman *Add, Edit, View Dependency Diagram*, sedangkan tombol *cancel* digunakan membatalkan proses dan kembali ke menu utama. *Progress bar* di bagian kiri digunakan untuk mengetahui tahapan-tahapan yang akan dilakukan sampai sebuah sistem pakar berhasil dibuat atau diubah.

Gambar 3.15 Halaman *Add, Edit, View Detail Expert System*

d. Halaman *Add, Edit, View Dependency Diagram*

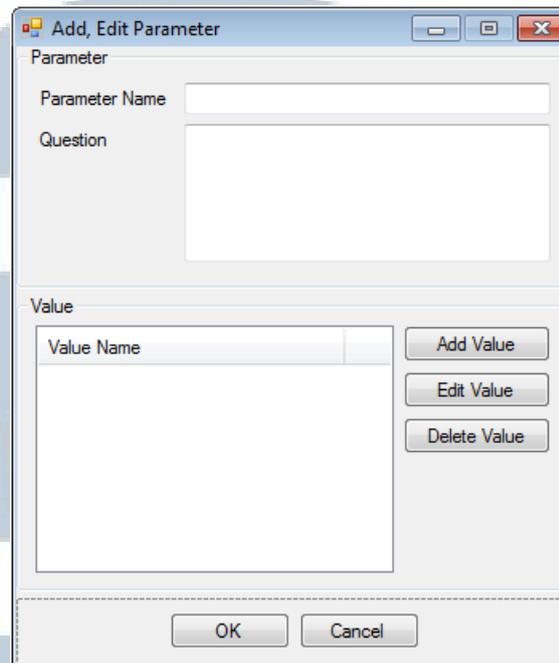
Halaman *Add, Edit, View Dependency Diagram* memiliki tampilan yang sama, namun memiliki fungsi dan tingkah laku (*behaviour*) yang berbeda. *Add* untuk melakukan fungsi penambahan data baru, *edit* untuk mengubah data yang sudah ada dan *view* hanya untuk melihat data sehingga tidak dapat mengubah atau mengisi data. Halaman *Add, Edit, View Dependency Diagram* digunakan untuk mengatur *dependency diagram* dalam bentuk *tree view*. *User* dapat menambah, mengubah dan menghapus parameter dengan menggunakan tombol *add, edit, delete* yang tersedia di sebelah kanan *tree view*. Tombol *next* digunakan untuk menuju ke halaman *Add, Edit, View Decision Table*, tombol *Previous* digunakan untuk kembali ke halaman *Add, Edit, View Detail Expert System* sedangkan tombol *cancel* digunakan membatalkan proses dan kembali ke menu utama. Pada Gambar 3.16, *progress bar* di bagian kiri digunakan untuk mengetahui tahapan-tahapan yang akan dilakukan sampai sebuah sistem pakar berhasil dibuat atau diubah.



Gambar 3.16 Halaman *Add, Edit, View Dependency Diagram*

e. Halaman *Add, Edit Parameter*

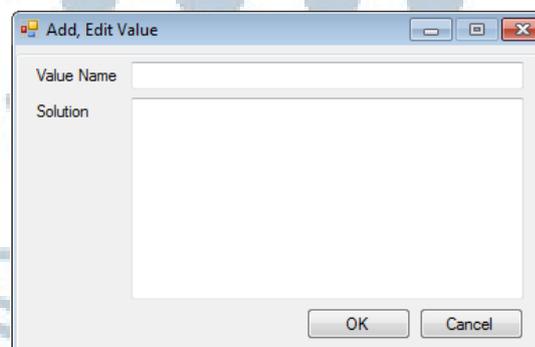
Halaman *Add, Edit Parameter* digunakan untuk menambah atau mengubah data parameter seperti nama parameter dan pertanyaan yang akan ditampilkan saat proses *reasoning*. Halaman ini juga menampilkan daftar value yang terdapat pada parameter yang sedang dipilih dan terdapat tombol *add value*, *edit value* dan *delete value* yang berfungsi untuk mengatur value-value dari parameter yang sedang dipilih. Tombol OK digunakan untuk menyimpan data parameter dan kembali ke halaman *dependency diagram*, sedangkan tombol *cancel* digunakan untuk kembali ke halaman *dependency diagram* tanpa menyimpan data parameter.



Gambar 3.17 Halaman *Add, Edit Parameter*

f. Halaman *Add, Edit Value*

Halaman *Add, Edit Value* digunakan untuk menambah atau mengubah nama value dan solusi. Solusi akan ditampilkan pada akhir proses *reasoning*. Solusi hanya dapat diisi apabila parameter merupakan parameter yang paling atas di dalam *tree view*. Nama value dalam satu parameter yang sama harus berbeda. Bila *user* memasukkan nama value yang sama maka akan muncul peringatan.

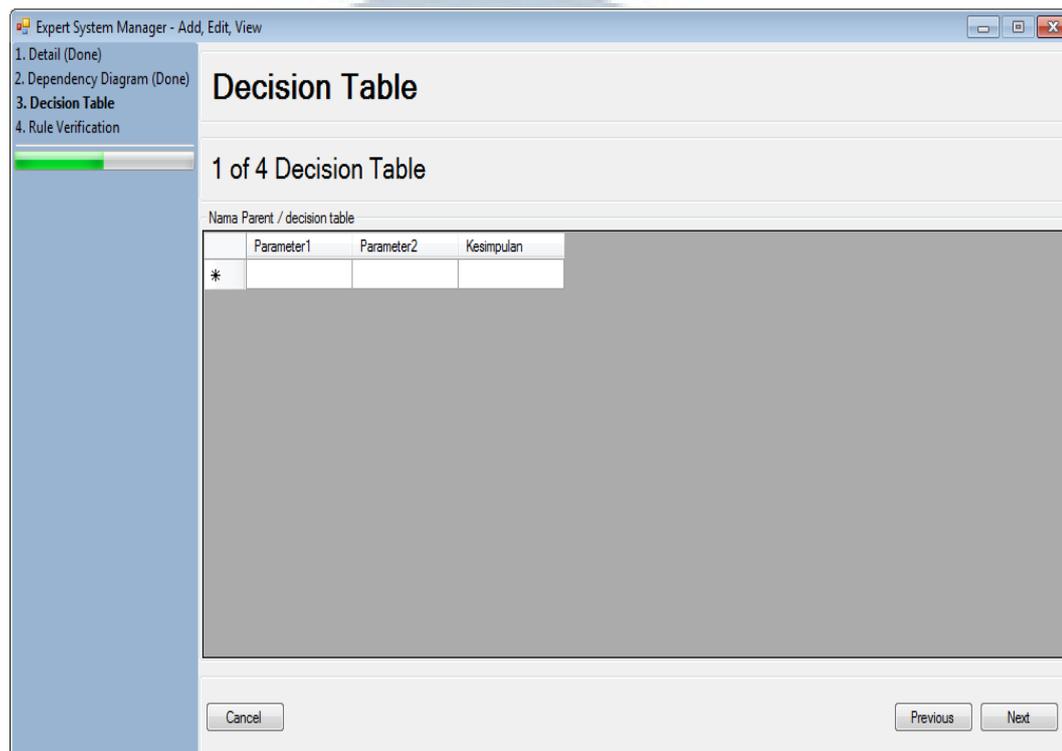


Gambar 3.18 Halaman *Add, Edit Value*

g. Halaman *Add, Edit, View Decision Table*

Halaman *Add, Edit, View Decision Table* memiliki tampilan yang sama, namun memiliki fungsi dan tingkah laku (*behaviour*) yang berbeda. *Add* untuk melakukan fungsi penambahan data baru, *edit* untuk mengubah data yang sudah ada dan *view* hanya untuk melihat data sehingga tidak dapat mengubah atau mengisi data. Halaman *Add, Edit, View Decision Table* digunakan untuk mengisi kesimpulan dari *rule-rule* yang di-*generate* oleh sistem dalam bentuk *datagrid view*. User dapat memilih kesimpulan yang sudah ada dalam bentuk *combobox* Halaman ini akan muncul sebanyak jumlah tabel yang perlu diisi oleh *user* yaitu sebanyak set (parameter yang memiliki anak/turunan). Tombol *next* digunakan untuk menuju ke halaman *Rule Verification*, tombol *Previous* digunakan untuk kembali ke halaman *Add, Edit, View Dependency Diagram* sedangkan tombol *cancel* digunakan membatalkan proses dan kembali ke menu utama. *Progress bar* di bagian kiri digunakan untuk mengetahui tahapan-tahapan yang akan dilakukan sampai sebuah sistem pakar berhasil dibuat atau diubah.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

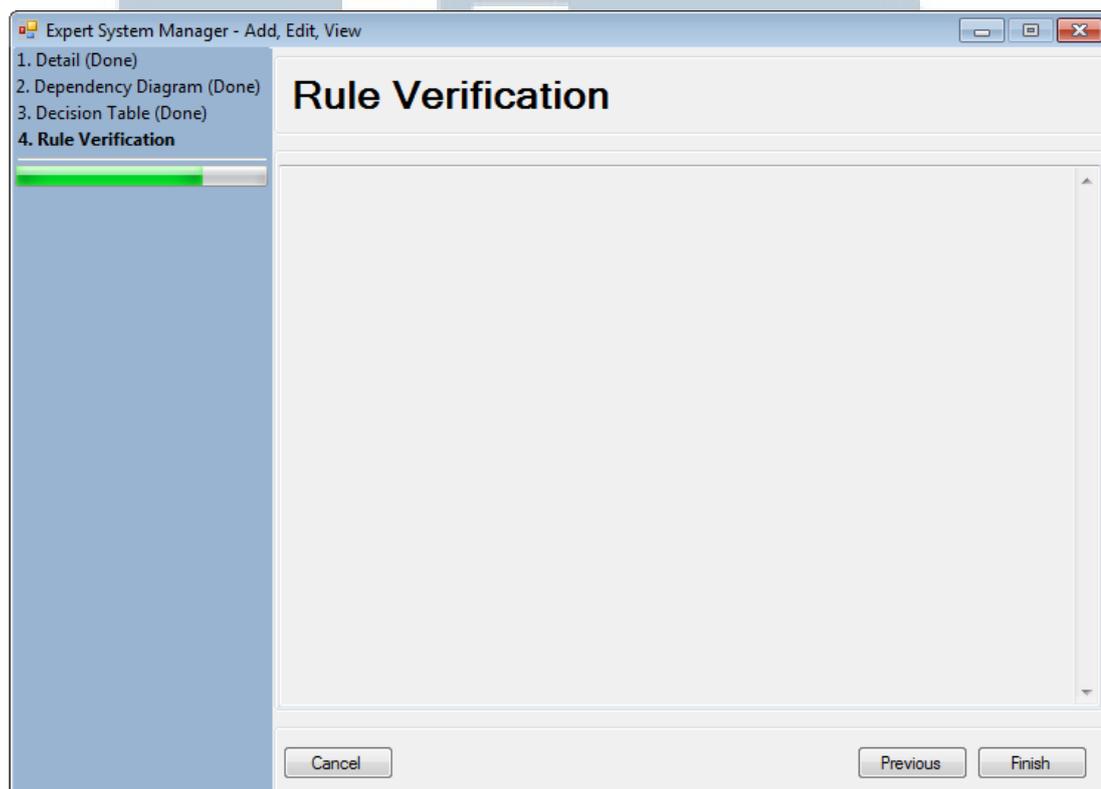


Gambar 3.19 Halaman *Add, Edit, View Decision Table*

h. Halaman *Rule Verification*

Halaman *Rule Verification* adalah halaman terakhir dari proses *Add, Edit* dan *View*. Halaman *Rule Verification* digunakan *user* untuk memeriksa secara manual *rule-rule* yang di-generate oleh sistem berdasarkan data yang dimasukkan oleh *user* pada halaman *dependency diagram* dan *decision table*. *Rule-rule* tersebut direpresentasikan dengan *IF-THEN* dalam *textbox* yang tidak dapat diisi atau diubah. Tombol *finish* digunakan apabila *user* sudah selesai memeriksa *rule-rule* tersebut dan menyelesaikan proses *Add* atau *Edit* atau *View*. *User* akan kembali ke menu utama setelah menekan tombol *finish*. Tombol *Previous* digunakan untuk kembali ke halaman *Add,*

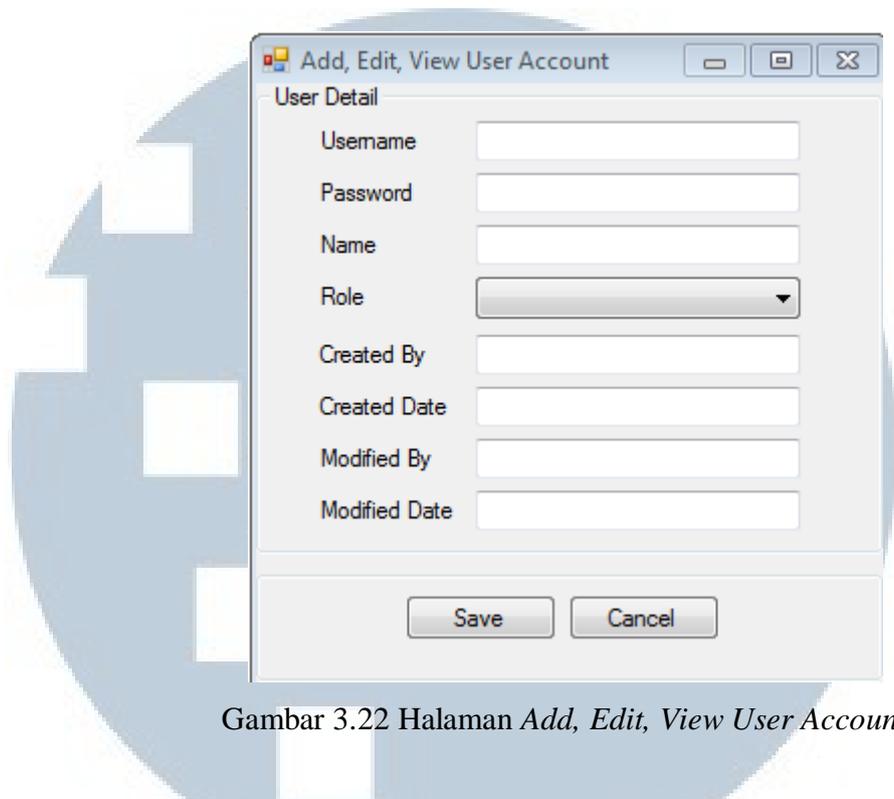
Edit, *View Decision Table* sedangkan tombol *cancel* digunakan membatalkan proses dan kembali ke menu utama. *Progress bar* di bagian kiri digunakan untuk mengetahui tahapan-tahapan yang akan dilakukan sampai sebuah sistem pakar berhasil dibuat atau diubah.



Gambar 3.20 Halaman *Rule Verification*

i. Halaman *User Account*

Halaman *User Account* digunakan untuk mengatur data *user*. Pada halaman ini Admin dapat melihat daftar *user* dan dapat menambah, mengubah, menghapus dan melihat *user* dengan menggunakan tombol *Add*, *Edit*, *Delete* dan *View*.

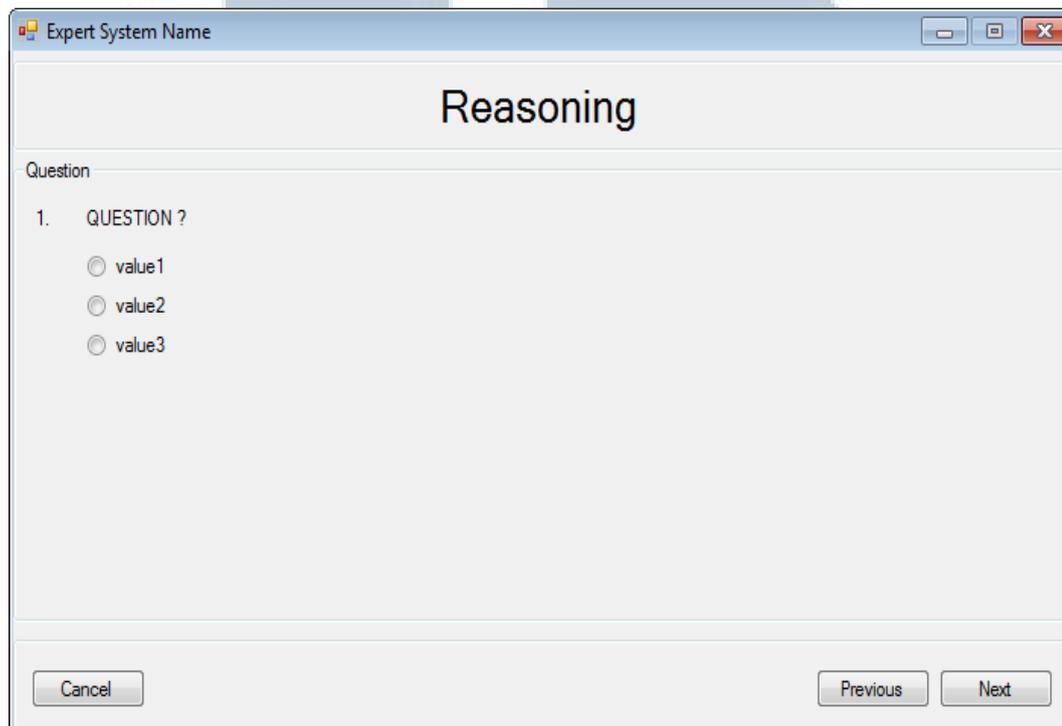


Gambar 3.22 Halaman *Add, Edit, View User Account*

k. Halaman *Reasoning*

Halaman *Reasoning* dapat digunakan oleh semua *user* untuk melakukan proses konsultasi. *User* yang melakukan proses konsultasi akan diberikan pertanyaan oleh sistem sebanyak parameter yang tidak memiliki anak atau turunan dengan jawaban yang berasal dari value dari masing-masing parameter. *User* harus menjawab setiap pertanyaan satu-persatu untuk mendapatkan hasil akhir dari proses konsultasi berupa kesimpulan dan solusi. Tombol *Next* digunakan untuk pindah ke halaman pertanyaan berikutnya setelah *user* memilih salah satu jawaban, ketika *user* berada di pertanyaan terakhir, maka tombol *next* akan memindahkan *user* ke halaman *conclusion*. Tombol *previous* digunakan untuk kembali ke halaman pertanyaan sebelumnya bila *user* ingin mengubah jawaban yang telah dipilih. Tombol *cancel*

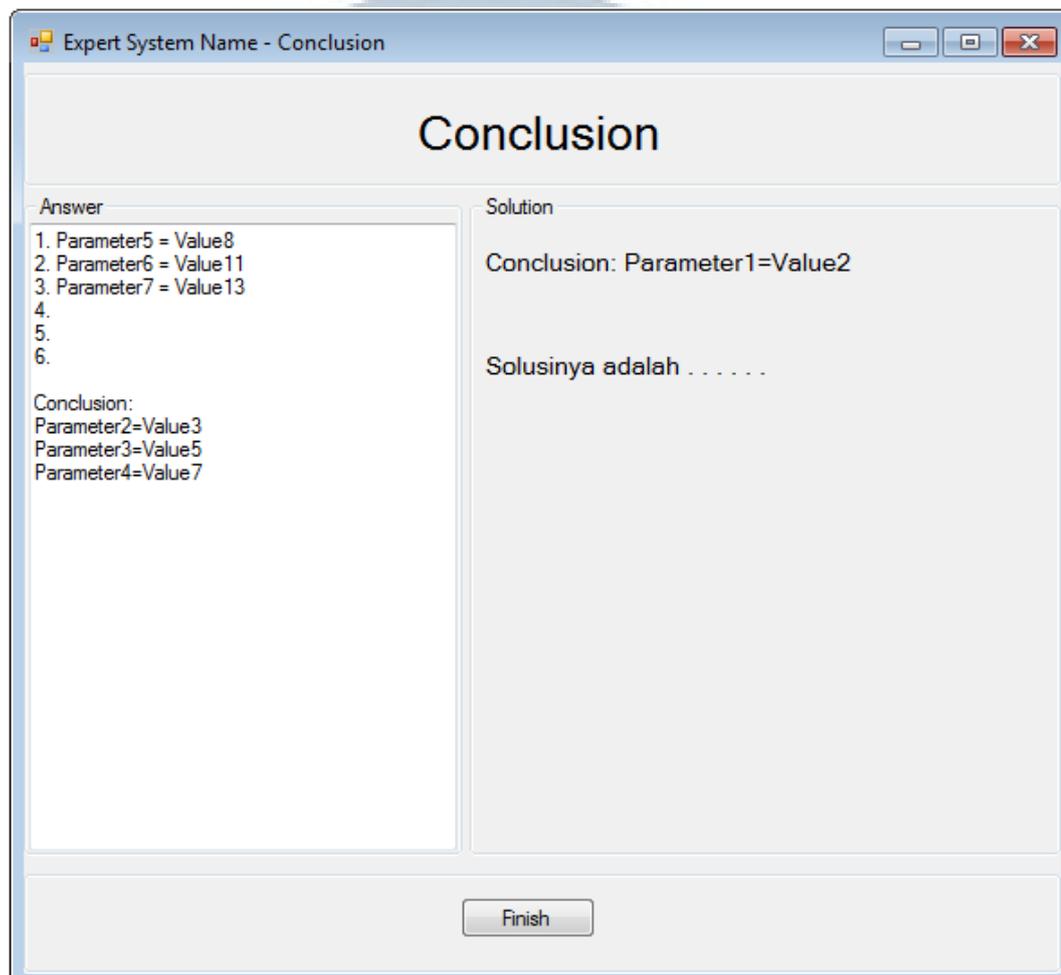
digunakan untuk membatalkan semua proses dan kembali ke menu utama.



Gambar 3.23 Halaman *Reasoning*

1. Halaman *Conclusion*

Halaman *Conclusion* merupakan halaman akhir dari proses konsultasi atau *reasoning* yang menampilkan jawaban-jawaban dari *user* atas pertanyaan-pertanyaan yang diberikan oleh sistem dan menampilkan kesimpulan (*conclusion*) dan solusi berdasarkan jawaban yang dipilih oleh *user*. Tombol *finish* digunakan untuk menyelesaikan proses *reasoning* dan kembali ke menu utama.

Gambar 3.24 Halaman *Conclusion*

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA