



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 DAN PERANCANGAN APLIKASI

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini antara lain adalah sebagai berikut

1. Identifikasi Masalah

Penulis mengidentifikasi gejala-gejala yang ada pada pasien autisme serta penanganan yang tepat dengan konsultasi bersama pakar.

2. Studi Literatur

Penulis melakukan studi terhadap referensi-referensi yang berhubungan dengan penelitian ini, seperti penelitian dengan metode yang sama yang sudah dilakukan sebelumnya, artikel, *website*, buku, dan lain-lainnya

3. Pengumpulan Data

Pada tahap ini, peneliti akan mengumpulkan data-data yang berhubungan dengan penelitian dari ahli pakar sebagai informasi yang akan dimasukkan ke dalam *knowledge base*. Untuk proses pengumpulan data, peneliti menggunakan metode wawancara dengan pakar.

4. Perancangan dan Analisa Program

Perancangan dibuat dengan mengimplementasikan algoritma Rete pada sistem pakar berbasis web.

5. Pembangunan Aplikasi

Aplikasi berbasis web ini dibangun menggunakan Notepad++ dan bahasa pemrograman html, mysql, css dan php.

6. Analisa dan pengambilan kesimpulan.

Pada tahap ini, peneliti akan melakukan uji coba terhadap aplikasi dan mencari beberapa responden untuk berperan sebagai pengguna. Dari uji coba tersebut, peneliti mengharapkan adanya *feedback* dari pengguna. Pengumpulan *feedback* didapatkan dengan metode pengisian kuisioner

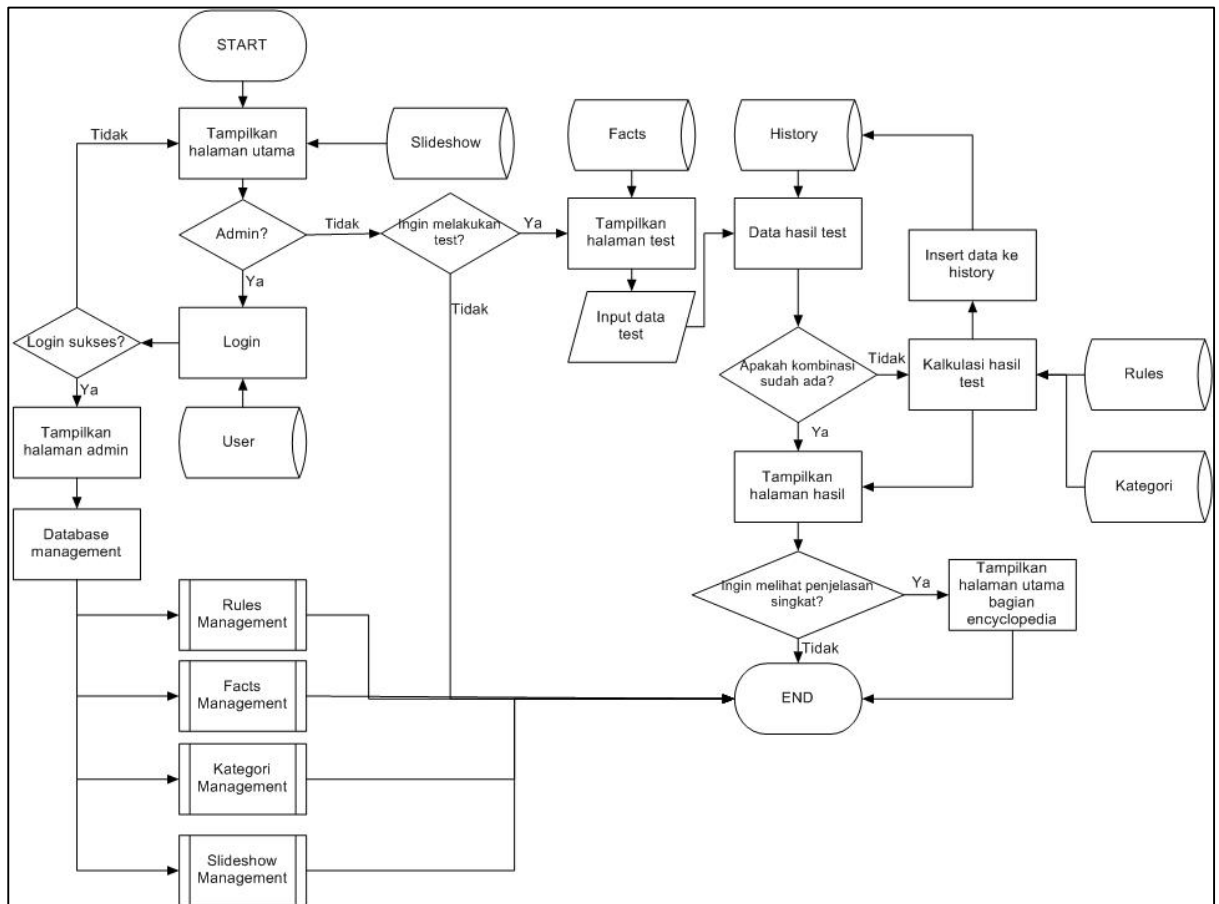
3.2 Perancangan Aplikasi

Dalam membuat aplikasi ini, digunakanlah Diagram alir (*Flowchart Diagram*), *Entity Relationship Diagram* dan *Data Flow Diagram*. Diagram alir adalah suatu tipe diagram yang melambangkan suatu proses menggunakan simbol-simbol yang berbeda yang mengandung informasi mengenai langkah-langkah dari suatu prosedur (Janssen, 2010). *Entity Relationship Diagram* adalah suatu model data yang secara grafis mengilustrasikan entitas dari suatu sistem informasi dan hubungan antar entitas tersebut, sementara *Data Flow Diagram* adalah representasi dari alur dan pertukaran informasi di dalam sistem (Janssen, 2010).

3.2.1 Diagram Alir

Diagram alir pada aplikasi ini dibagi menjadi lima bagian, yaitu diagram alir sistem secara umum, diagram alir untuk pengaturan *rules*, diagram alir untuk pengaturan kategori, diagram alir untuk pengaturan *facts* dan diagram alir untuk pengaturan *slideshow*

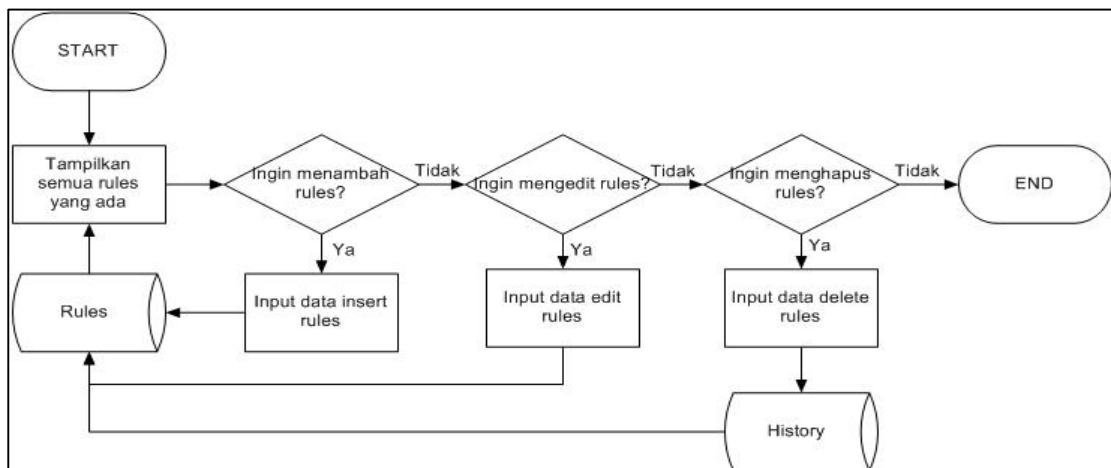
A. Diagram alir sistem



Gambar 3.1 Diagram alir sistem

Diagram alir ini menggambarkan cara kerja sistem secara umum. Hanya admin yang bisa *login* dan melakukan proses *database management*. Pengunjung bisa melakukan *test* dan sistem akan melakukan kalkulasi hasil *test* yang dimasukkan oleh pengguna. Setelah mendapat hasil *test*, maka pengguna bisa melihat penjelasan singkat pada halaman utama di bagian *encyclopedia*.

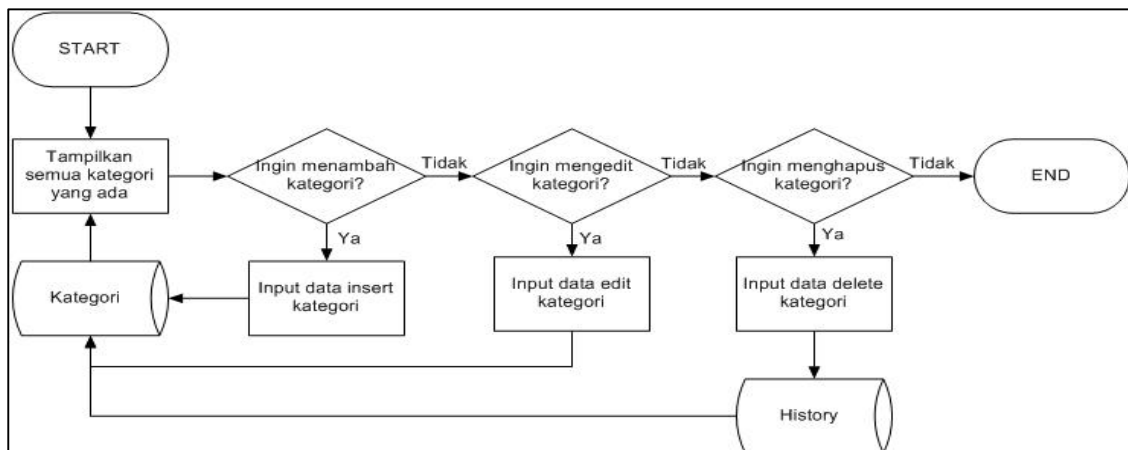
B. Diagram alir pengaturan *rules*



Gambar 3.2 Diagram alir *rules*

Diagram alir ini menggambarkan proses *rules management*. Proses ini hanya bisa dilakukan oleh *admin* yang telah *login*. *Admin* bisa menambahkan, meng-*edit* dan menghapus *rules* yang ada. Jika *admin* menghapus *rules* yang ada, maka jenis autisme yang bersangkutan juga akan dihapus pada tabel *history*.

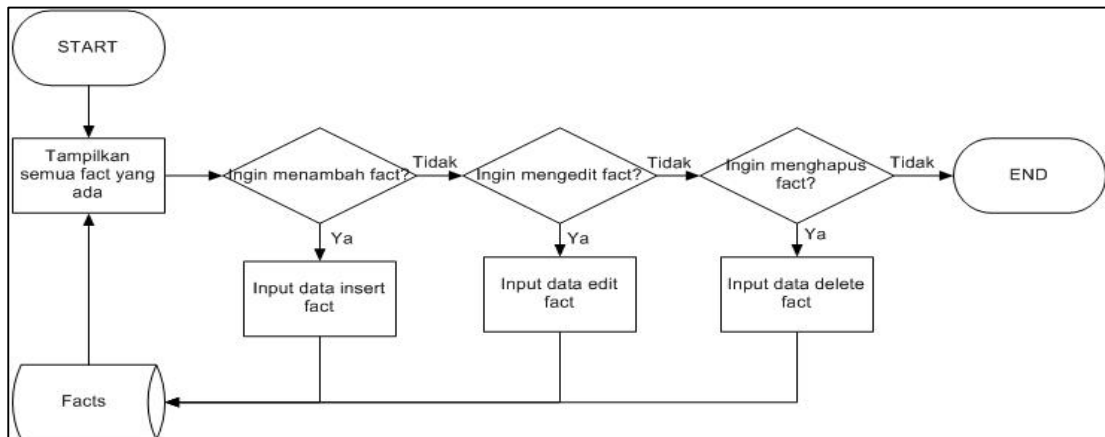
C. Diagram alir pengaturan kategori



Gambar 3.3 Diagram alir kategori

Diagram alir ini menggambarkan proses kategori *management*. Proses ini hanya bisa dilakukan oleh *admin* yang telah *login*. *Admin* bisa menambahkan, meng-*edit* dan menghapus kategori yang ada. Jika *admin* menghapus kategori yang ada, maka jenis autisme yang bersangkutan juga akan dihapus pada tabel *history*.

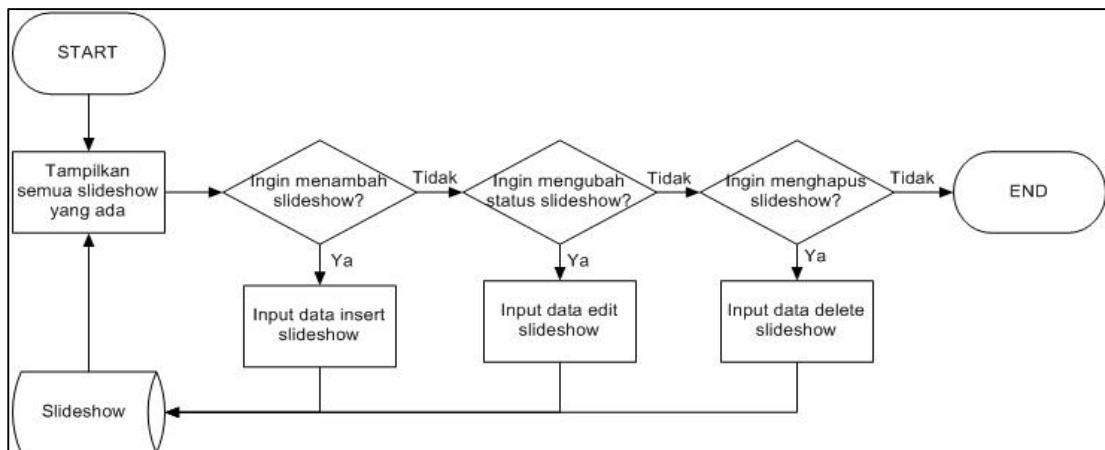
D. Diagram alir pengaturan *facts*



Gambar 3.4 Diagram alir *facts*

Diagram alir ini menggambarkan proses *facts management*. Proses ini hanya bisa dilakukan oleh *admin* yang telah *login*. *Admin* bisa menambahkan, meng-*edit* dan menghapus *fact* yang ada.

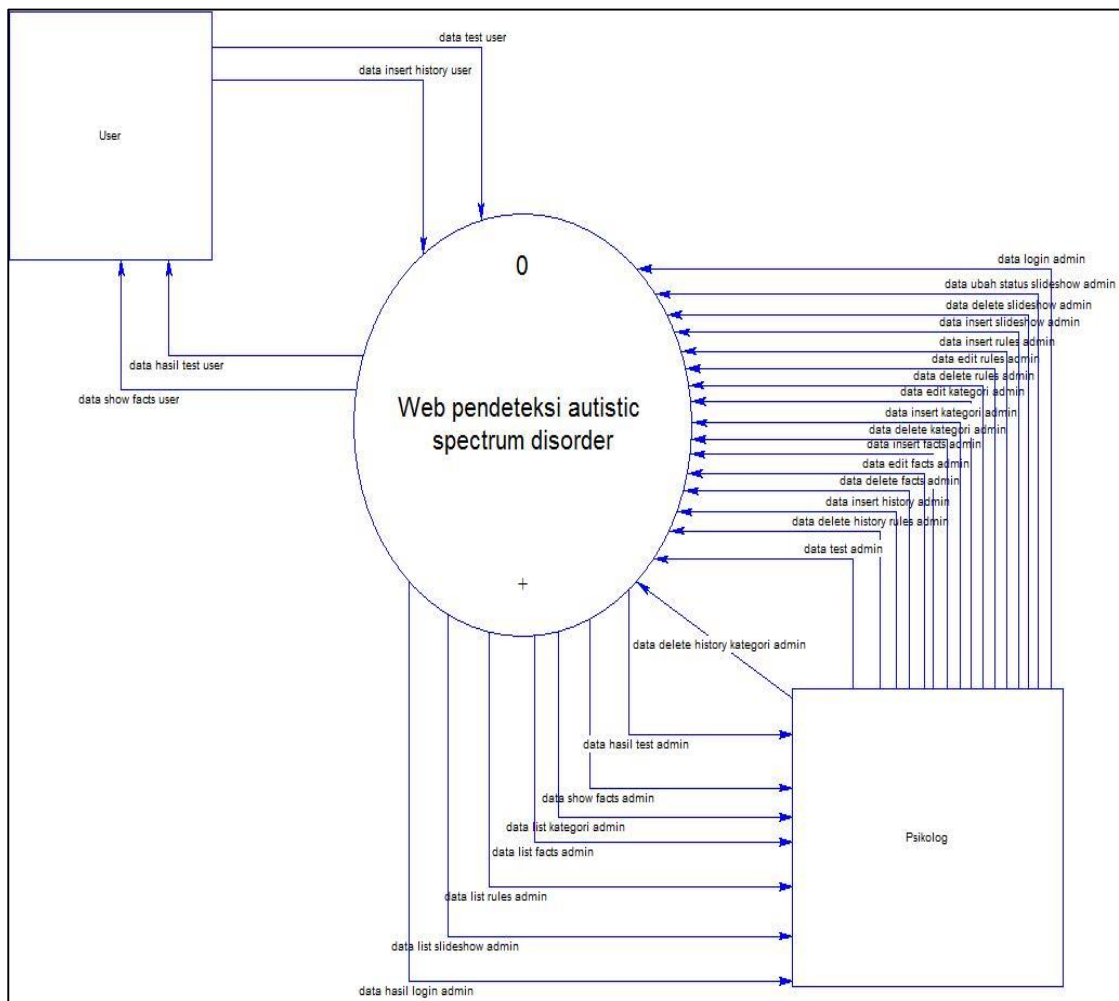
E. Diagram Alir pengaturan *slideshow*



Gambar 3.5 Diagram alir *slideshow*

Diagram alir ini menggambarkan proses *slideshow management*. Proses ini hanya bisa dilakukan oleh *admin* yang telah *login*. *Admin* bisa menambahkan, mengubah status dan menghapus *slideshow* yang ada.

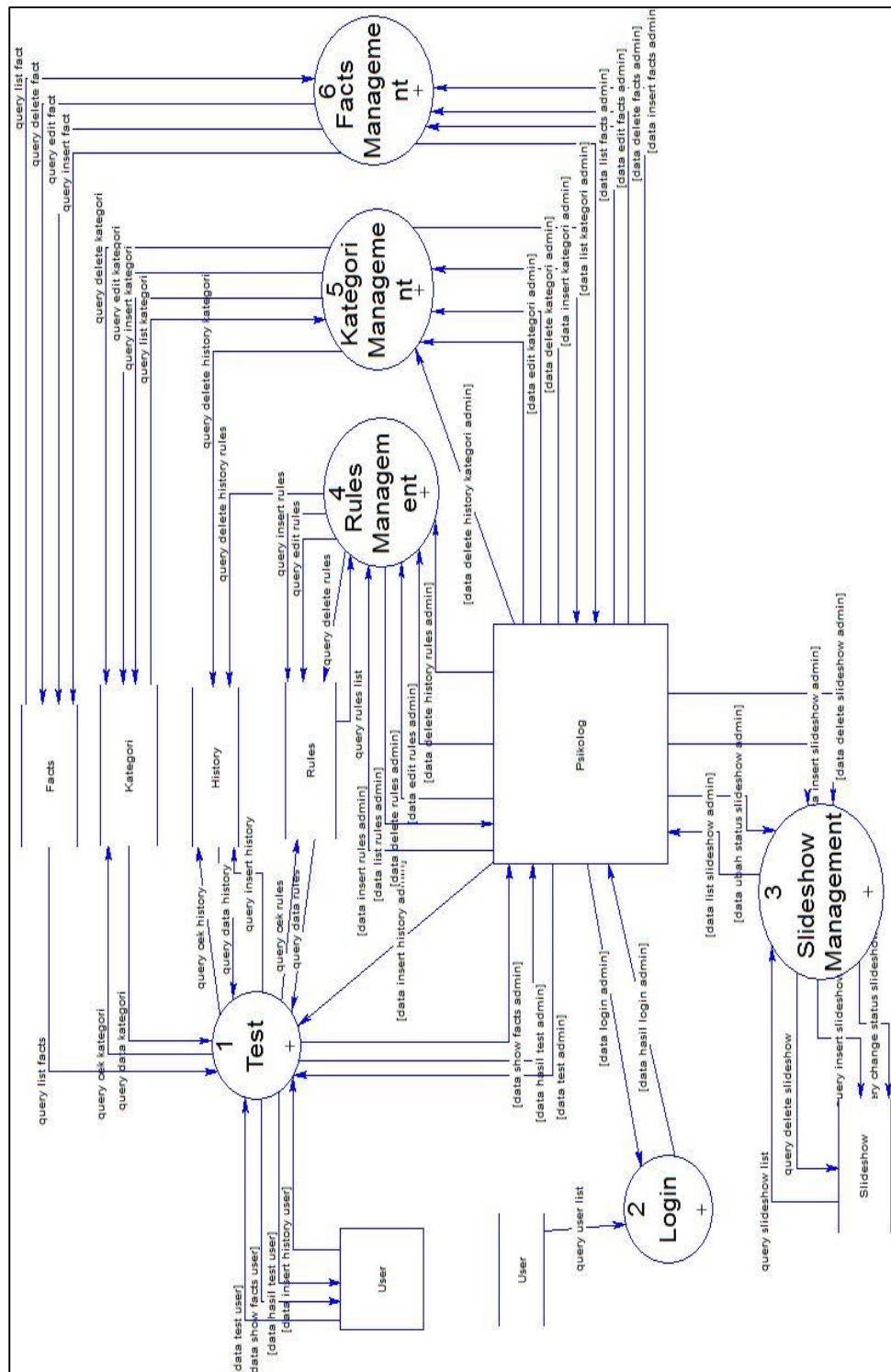
3.2.2 Data Flow Diagram



Gambar 3.6 Context Diagram

Context diagram untuk aplikasi ini terdiri dari dua entitas eksternal, yaitu psikolog dan *user*. Berdasarkan pada gambar di atas, *context diagram* ini dipecah menjadi enam proses pada DFD *level 1*, yaitu proses *test*, proses *login*, proses *slideshow management*, proses *rules management*, proses *kategori management* dan proses *facts management*.

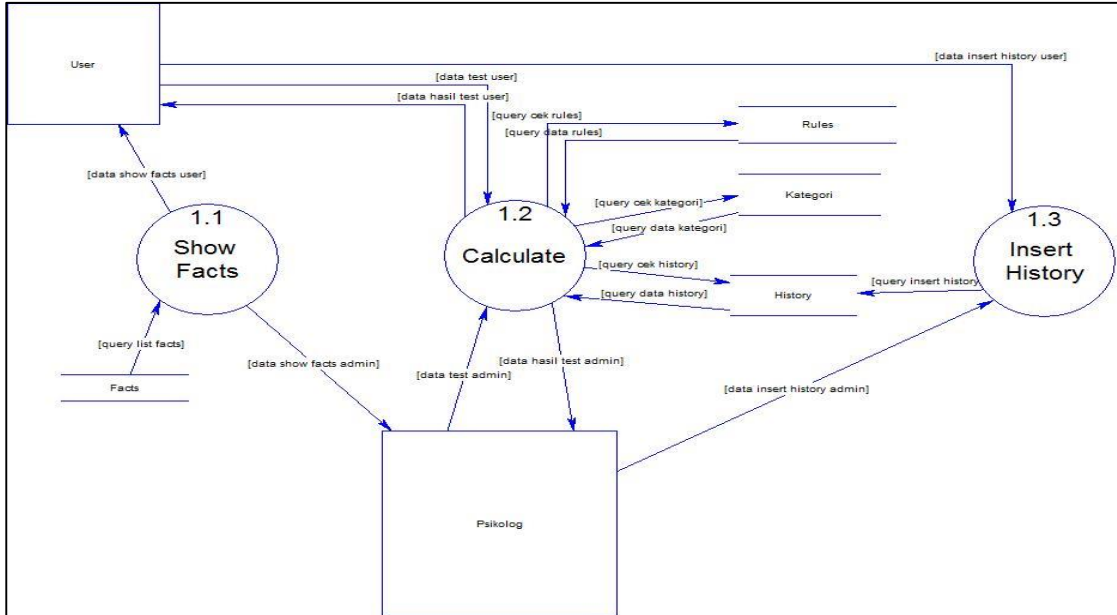




Gambar 3.7 DFD Level 1

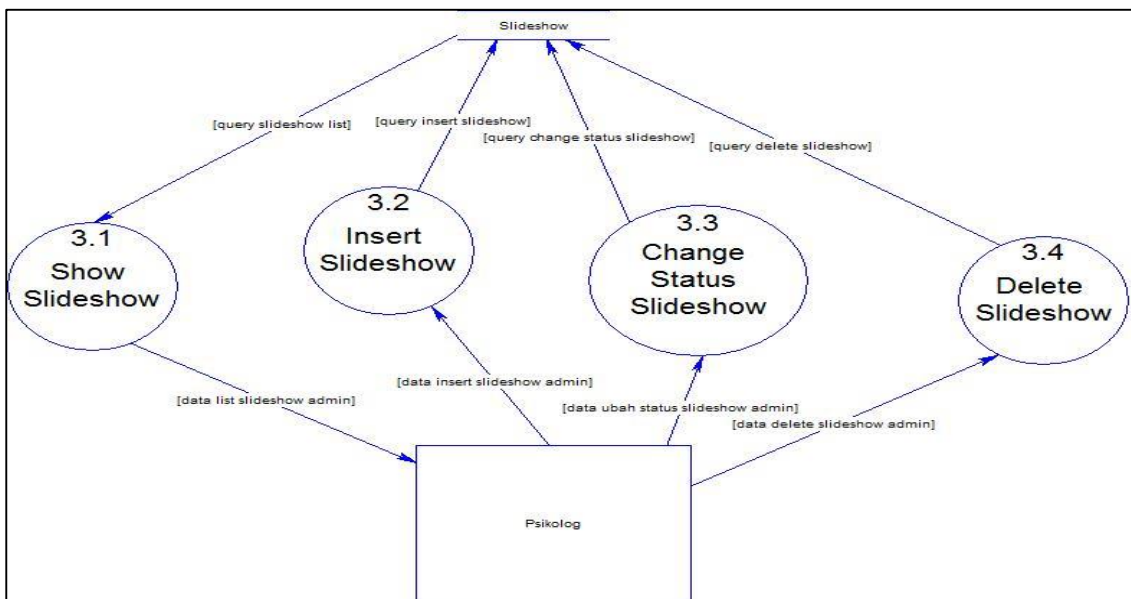
Pada *level 2* terdapat sembilan belas subproses dari enam proses pada level 1.

Pada proses *test* terdapat tiga subproses, yaitu *show facts*, *calculate*, dan *insert history*.



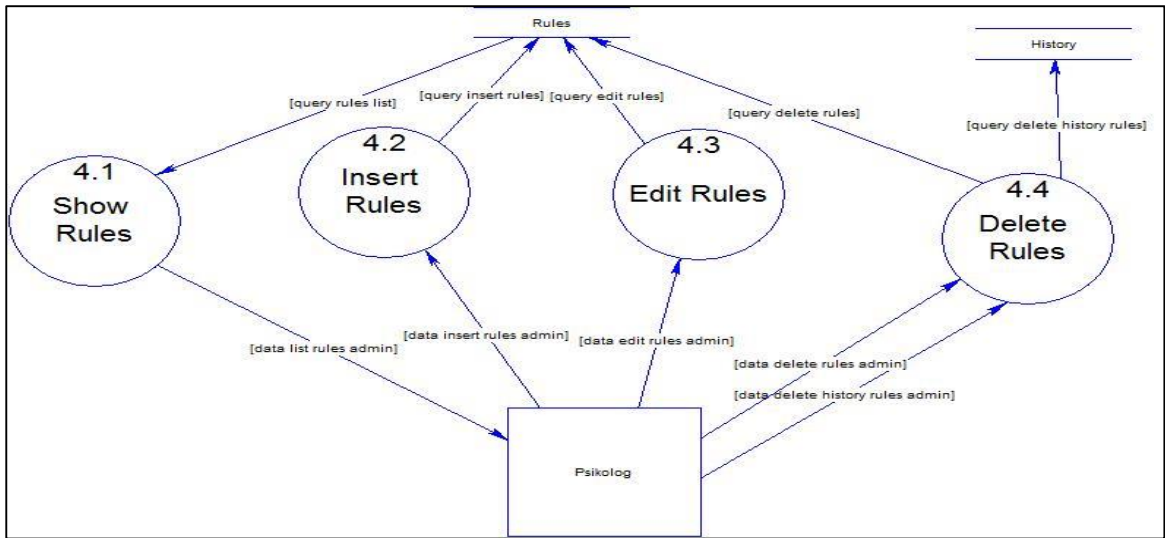
Gambar 3.8 DFD *Level 2* Proses *Test*

Pada proses *slideshow management* terdapat empat subproses, yaitu *show slideshow*, *insert slideshow*, *edit slideshow*, dan *delete slideshow*.



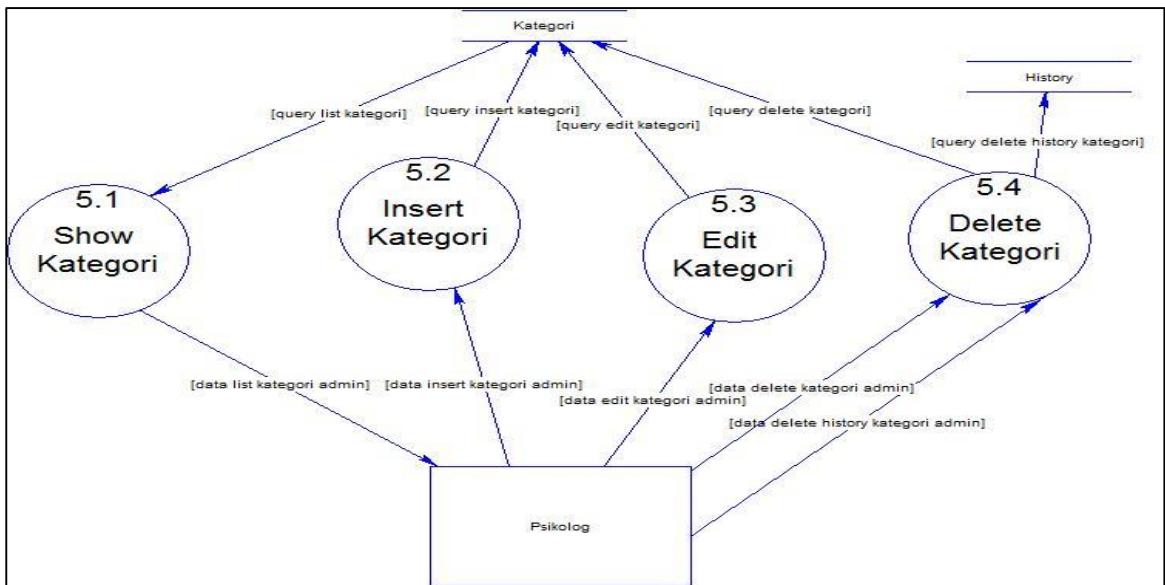
Gambar 3.9 DFD *Level 2* Proses *Slideshow Management*

Pada proses *rules management* terdapat empat subproses, yaitu *show rules*, *insert rules*, *edit rules*, dan *delete rules*.



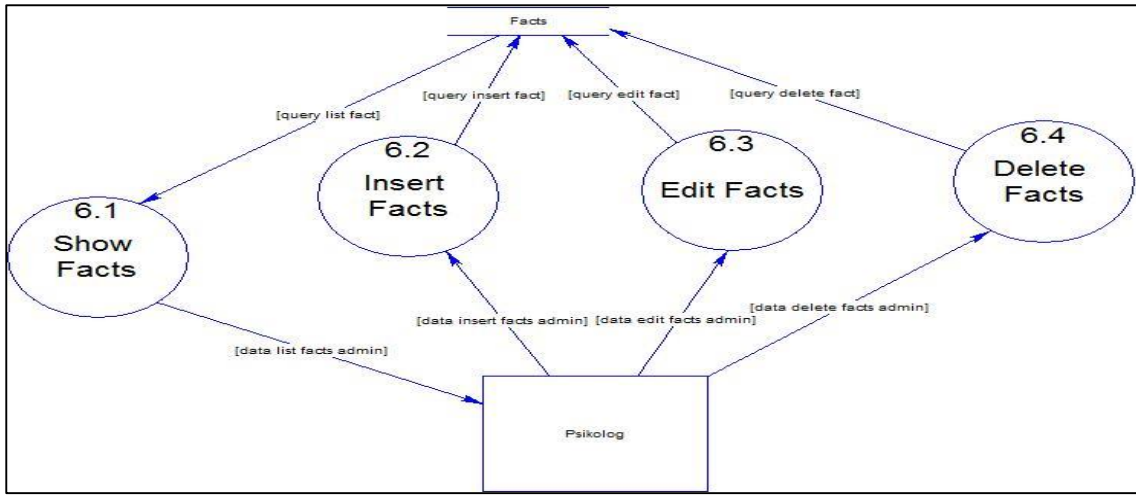
Gambar 3.10 DFD *Level 2* Proses *Rules Management*

Pada proses *kategori management* terdapat empat subproses yaitu *show kategori*, *insert kategori*, *edit kategori*, dan *delete kategori*.



Gambar 3.11 DFD *Level 2* Proses *Kategori Management*

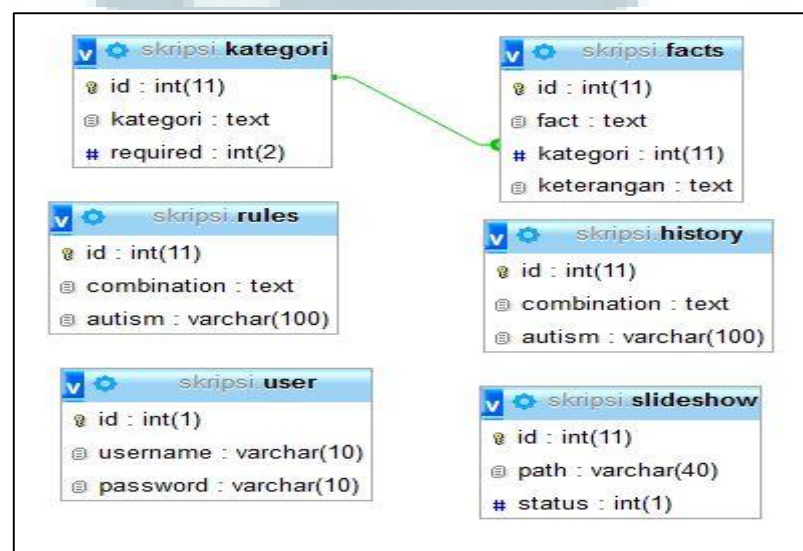
Pada proses *facts management* terdapat empat subproses yaitu *show facts*, *insert fact*, *edit fact*, dan *delete fact*.



Gambar 3.12 DFD Level 2 Proses Facts Management

3.2.3 Entity Relationship Diagram

Database yang digunakan oleh sistem adalah MySQL. Untuk menjelaskan hubungan antar objek data dalam database maka digambarkan Entity Relationship Diagram (ERD). Terdapat enam buah entitas yakni facts, kategori, rules, history, user, dan slideshow. Hubungan antar entitas dan atributnya digambarkan pada Gambar 3.13.



Gambar 3.13 Entity Relationship Diagram

Dari gambar 3.13 dapat dilihat bahwa hanya tabel kategori dan *facts* yang berhubungan. Kolom id dalam tabel kategori berperan sebagai *primary key* dan kolom kategori dalam tabel *facts* berperan sebagai *foreign key*.

1. Nama Tabel : *Facts*

Fungsi : menyimpan data-data fakta yang akan digunakan.

Primary Key : ID

Foreign Key : kategori

Tabel 3.1 Tabel *Facts*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID fakta
2	fact	text		Fakta
3	kategori	int	11	ID kategori
4	keterangan	text		

2. Nama Table : Kategori

Fungsi : menyimpan data-data kategori.

Primary Key : ID

Foreign Key : -

Tabel 3.2 Tabel Kategori

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID kategori
2	kategori	text		Nama kategori
3	required	int	2	Jumlah fakta yang harus dipenuhi untuk memenuhi syarat

3. Nama Table : *Rules*

Fungsi : menyimpan data-data *rules*.

Primary Key : ID

Foreign Key : -

Tabel 3.3 Tabel *Rules*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID rules
2	combination	text		Kombinasi kategori yang harus dipenuhi
3	autism	varchar	100	Jenis autisme yang diderita jika kombinasi terpenuhi

4. Nama Table : *History*

Fungsi : menyimpan data-data kombinasi *input* yang pernah dimasukkan.

Primary Key : ID

Foreign Key : -

Tabel 3.4 Tabel *History*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID rules
2	combination	text		Kombinasi input yang pernah digunakan
3	autism	varchar	100	Jenis autisme yang diderita jika kombinasi terpenuhi

5. Nama Table : *User*

Fungsi : menyimpan data pengguna yang memiliki hak untuk *login*.

Primary Key : ID

Foreign Key : -

Tabel 3.5 Tabel *User*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	1	ID user
2	username	varchar	10	Nama pengguna
3	password	varchar	10	Password

6. Nama Table : *Slideshow*

Fungsi : menyimpan data gambar yang digunakan sebagai *slideshow*.

Primary Key : ID

Foreign Key : -

Tabel 3.6 Tabel *Slideshow*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID slideshow
2	path	varchar	40	Lokasi gambar
3	status	int	1	Status gambar apakah akan ditampilkan atau tidak

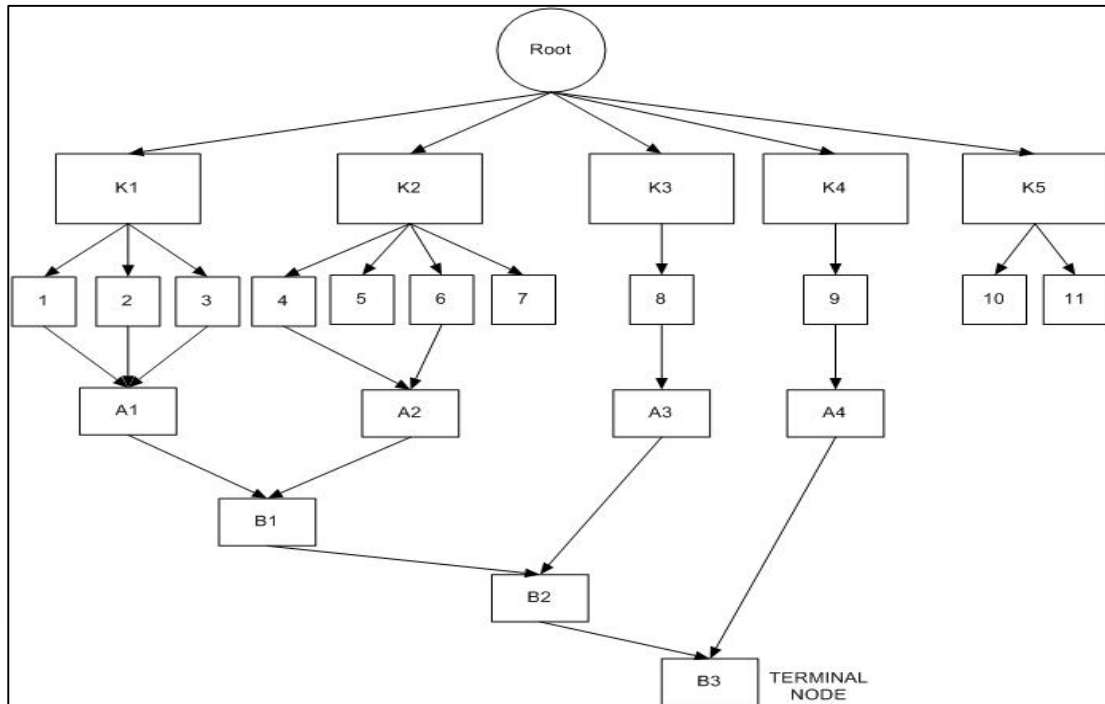
3.2.4 Rete Network

Jaringan Rete adalah suatu *graph* yang terdiri dari *nodes* yang melambangkan pola dalam kondisi aturan. *Nodes* dalam jaringan Rete terdiri dari 5 *nodes* yaitu *alpha node*, *beta node*, *kind node*, *root node*, dan *terminal node*. *Alpha node* akan mengevaluasi kondisi yang ada apakah sesuai dengan *rules* yang ada. *Rules* yang digunakan dalam aplikasi ini adalah sebagai berikut

Tabel 3.7 Tabel Daftar *Rules*

No	Rules	Autism
1	K1 K2 K3 K4	<i>Autism spectrum disorder</i>
2	K1 K2 K3 K4 K5	<i>Rett Syndrome</i>

Untuk mengeksekusi *rules* yang ada, aturan – aturan yang ada harus dipenuhi. Dalam aplikasi ini, aturan – aturan tersebut adalah kategori yang ada harus bernilai *true*. Berikut adalah contoh ilustrasi dari jaringan Rete



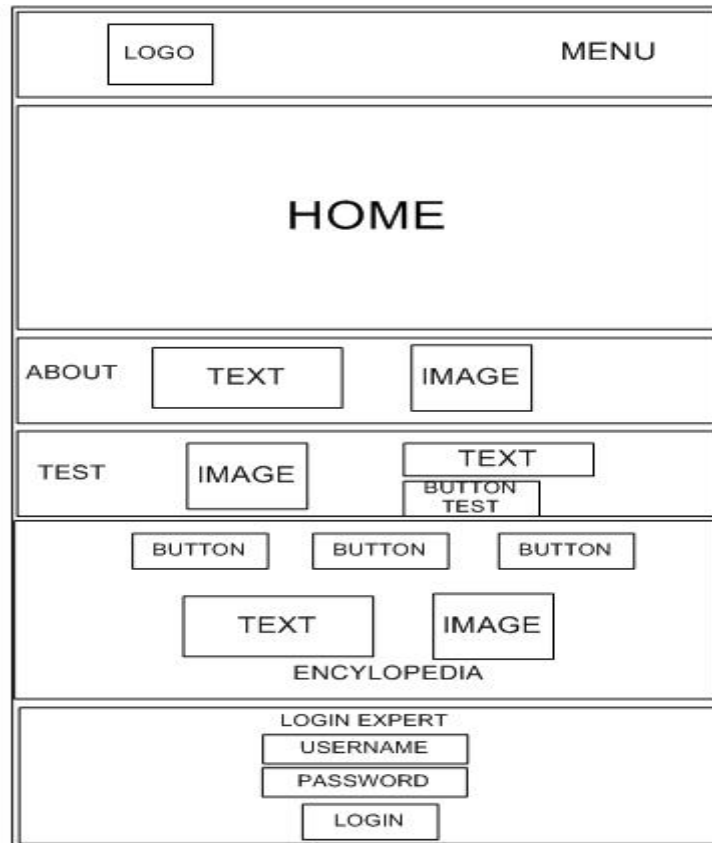
Gambar 3.14 Ilustrasi jaringan Rete

Dalam ilustrasi yang ditunjukkan pada gambar 3.14 menunjukkan jaringan Rete yang terbentuk jika pertanyaan 1,2,3,4,6,8 dan 9 pada tabel 2.3 dijawab ya sehingga *rule autism spectrum disorder* terpenuhi dan bisa di *fire*. Awalnya semua *input* akan ditampung dalam *root*. Kemudian *root* akan memisahkan *input* yang ada ke dalam *kind node* yang bersangkutan. Dalam aplikasi ini, *kind node* adalah kategori. Jika *input* yang ada memenuhi syarat setiap *kind node* maka *alpha node* akan terbentuk. *Alpha node* yang ada akan digabung menjadi *beta node*. Setiap ronde hanya dua *node* yang dapat digabung. Penggabungan akan terus dilakukan hingga semua *alpha node* telah digabung dan menghasilkan *terminal node*.

3.2.5 Design Interface

- Halaman Utama

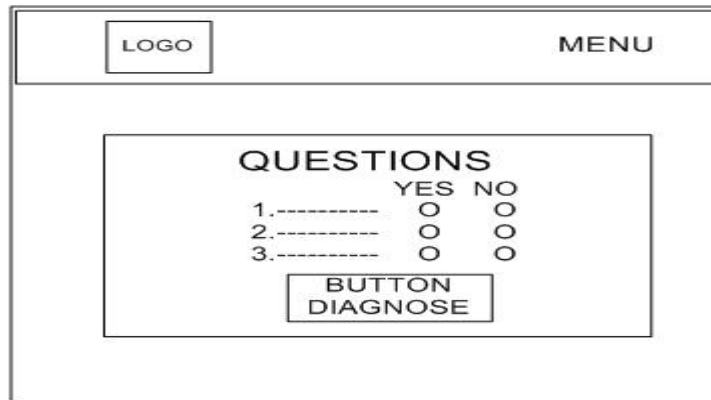
Halaman utama dibagi menjadi beberapa bagian yaitu bagian *home*, bagian *about*, bagian *test*, bagian *encyclopedia*, dan bagian *login*. Menu akan selalu mengikuti pengguna di semua bagian halaman utama.



Gambar 3.15 Home Interface

- Halaman Test

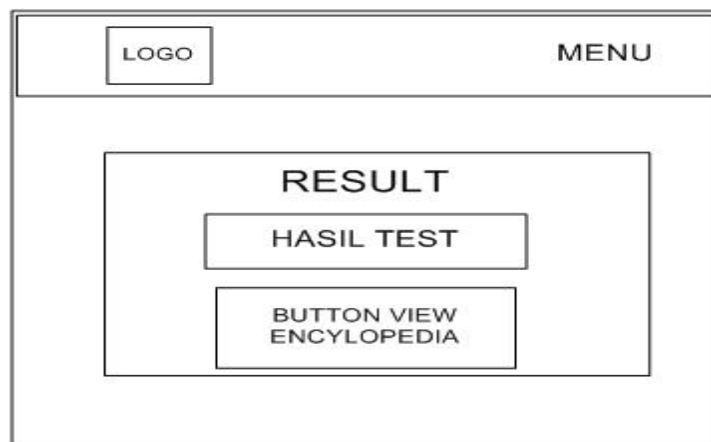
Halaman *test* akan menampilkan daftar pertanyaan yang akan menentukan hasil prediksi sistem. Untuk menjawab pertanyaan, pengguna cukup memilih opsi ya dan tidak yang telah disediakan.



Gambar 3.16 *Test Page Interface*

- Halaman *Result*

Halaman *result* akan menampilkan hasil prediksi sistem berdasarkan hasil *test* yang telah dilakukan pengguna. Di bawah hasil *test*, terdapat tombol *view encyclopedia* yang berfungsi membawa pengguna ke halaman utama bagian *encyclopedia*.

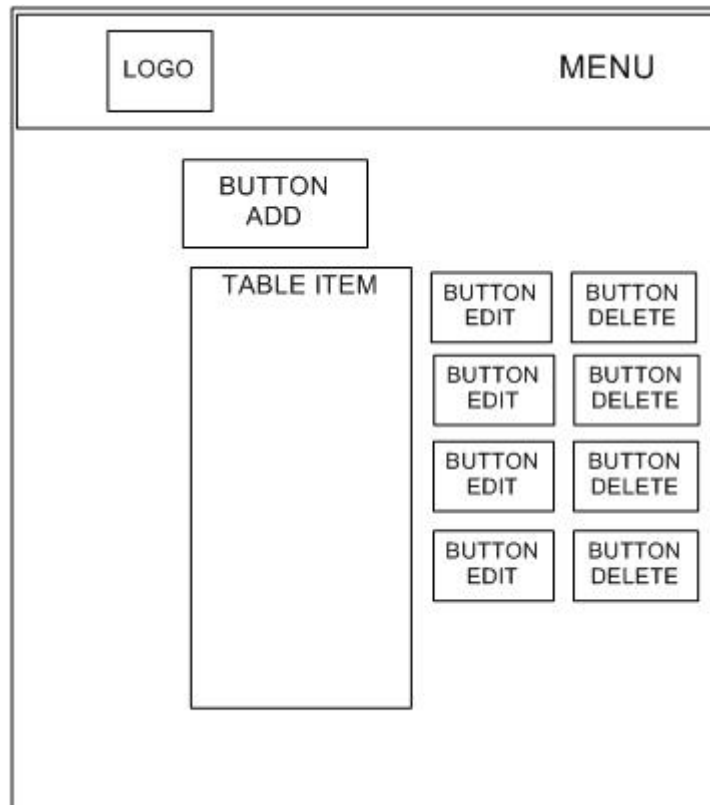


Gambar 3.17 *Result Page Interface*

- Halaman *List Database Management*

Halaman *list* pada *database management* akan menampilkan daftar data-data yang ada di dalam *database*. Terdapat tombol *add* di atas *table* yang berfungsi untuk menambahkan data ke dalam *table*. Di samping setiap data yang ditampilkan di dalam *table* terdapat tombol *edit* dan *delete*. Tombol *edit*

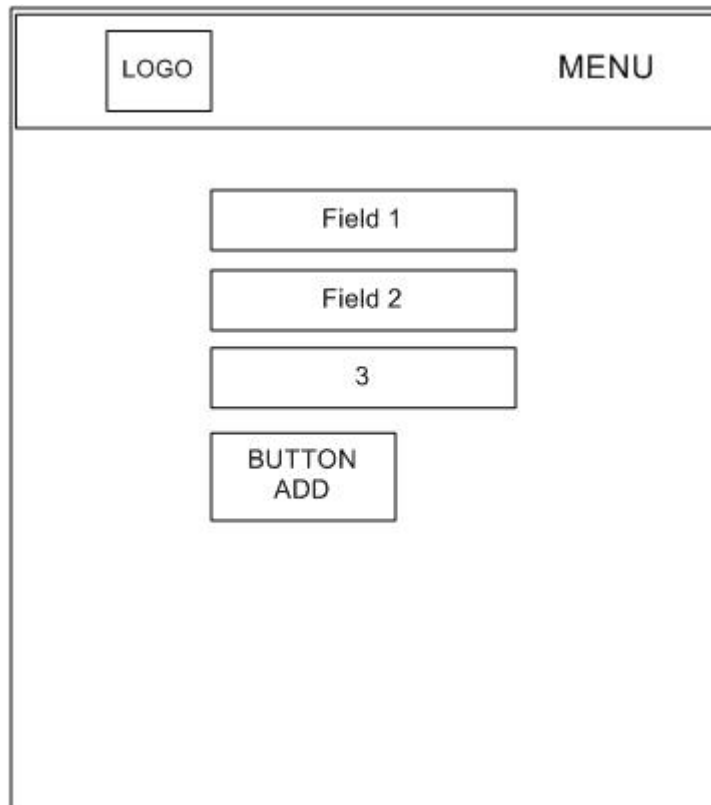
berfungsi untuk meng-*edit* data dan tombol *delete* berfungsi untuk menghapus data. *Interface* ini digunakan untuk *slideshow management*, *rules management*, kategori *management*, dan *facts management*.



Gambar 3.18 *List Database Management Interface*

- Halaman *Add* dan *Edit Database Management*

Halaman *add* dan *edit* pada *database management* berfungsi untuk menambahkan data baru dan mengubah data yang dipilih. *Interface* ini digunakan untuk *slideshow management*, *rules management*, kategori *management*, dan *facts management*.



Gambar 3.19 *Add and Edit Database Management Interface*