

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian yang Digunakan

Untuk mendapatkan hasil seperti yang diinginkan dalam perancangan sistem pendukung keputusan ini tentunya membutuhkan komponen penunjang dalam proses pengerjaannya, yaitu antara lain sebagai berikut :

3.1.1 Teknik Pengolahan Data

Metodologi penelitian merupakan cara yang digunakan dalam memperoleh berbagai data untuk diproses menjadi informasi yang lebih akurat sesuai permasalahan yang akan diteliti dan mendeskripsikan masalah yang dilengkapi dengan penyajian diagram alur pelaksanaan penelitian untuk memudahkan pemahaman tahapan penelitian.

Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan sebelumnya. Dalam penyusunan tugas akhir ini akan melalui beberapa tahapan yang membentuk sebuah alur yang sistematis seperti dibawah ini:

1. Pengumpulan data.
2. Melakukan analisis kasus.
3. Pembangunan sistem dengan metode sekuensial linear.
4. Uji implementasi sistem
5. Kesimpulan dan Saran



Gambar. 3.1 Desain Tahapan Penelitian

Pada tahap ini dilakukan pengumpulan data yang diperlukan dalam penelitian yang dilakukan. Semua tahap pada proses pengumpulan data tersebut diperoleh dari pengumpulan data rekam medis dan studi pustaka.

a. Pengumpulan Data Rekam Medis

Data yang dikumpulkan yaitu data kasus yang terdiri dari nama penyakit, gejala dan solusi. Data tersebut di peroleh dengan melakukan pencarian dan penelitian.

b. Studi Pustaka

Studi pustaka dilakukan dengan cara mempelajari buku, artikel, dan jurnal yang berhubungan dengan penelitian untuk membantu menyelesaikan permasalahan yang akan diteliti, serta mendapatkan dasar referensi yang kuat terutama dalam menerapkan Metode *Case Based Reasoning* (CBR) sebagai metode yang digunakan dalam penelitian sistem pakar penyakit kulit kepala ini.

3.2 Spesifikasi Kebutuhan *Hardware* dan *Software*

Pada penelitian ini dibangun dengan bahasa pemrograman *HTML* dan *PHP* menggunakan spesifikasi perangkat lunak (*software*) sebagai berikut:

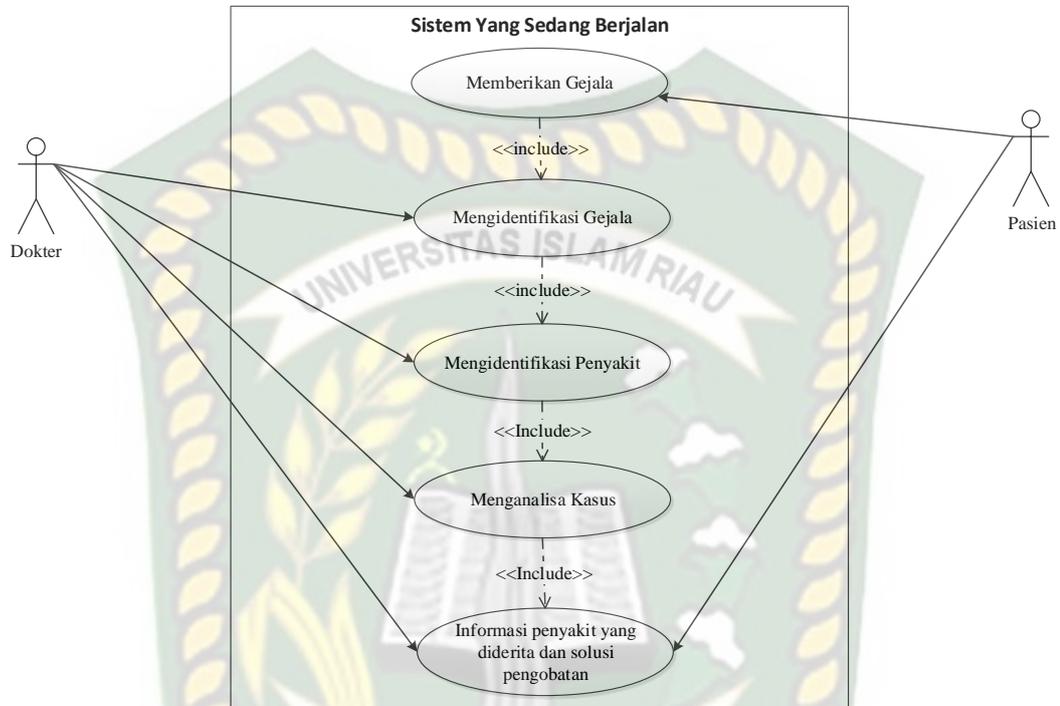
1. Sistem operasi menggunakan Windows 7 Ultimate 32-bit.
2. *Notepad ++* yang merupakan *tools* bahasa pemrograman.
3. *Database Manajemen System* menggunakan *Mysql*.

Adapun perangkat keras (*hardware*) yang digunakan adalah komputerspesifikasi sebagai berikut:

1. Processor Intel Core i3
2. 320 GB HDD
3. RAM 2048 MB

3.3 Analisa Sistem yang Sedang Berjalan

Adapun gambaran proses sistem yang sedang berjalan adalah sebagai berikut :



Gambar. 3.2 Analisa Sistem yang Sedang Berjalan

Proses manual yang terjadi saat ini dimulai dari dokter/pakar mengidentifikasi gejala-gejala yang diberikan oleh pasien, dari gejala tersebut dokter/pakar melanjutkan mengidentifikasi penyakit yang diderita oleh pasien dilihat dari gejala dan kasus-kasus yang pernah terjadi sebelumnya, setelah mendapatkan penyakit yang diderita, dokter/pakar akan memberikan informasi nama penyakit dan solusi pengobatan pada pasien.

3.4 Perancangan Sistem

Setelah melakukan kegiatan menganalisa sistem yang sedang berjalan, maka analisis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan

sebagai tindak lanjut penyelesaian masalah sistem pakar diagnosa penyakit kulit kepala.

Dalam membangun atau mengembangkan sistem pakar tentunya diperlukan perancangan yang tepat dan terstruktur dengan baik sehingga sistem pakar yang akan di bangun dapat berjalan dengan baik dan memberikan hasil diagnosa sesuai yang di harapkan. Tahap ini disebut dengan perancangan sistem yang dimulai dari perancangan konteks diagram yang nantinya dapat menggambarkan keseluruhan system pakar berbasis kasus diagnosa penyakit kulit kepala.

3.4.1 Analisa Sistem Baru



Gambar. 3.3 Analisa Sistem Baru

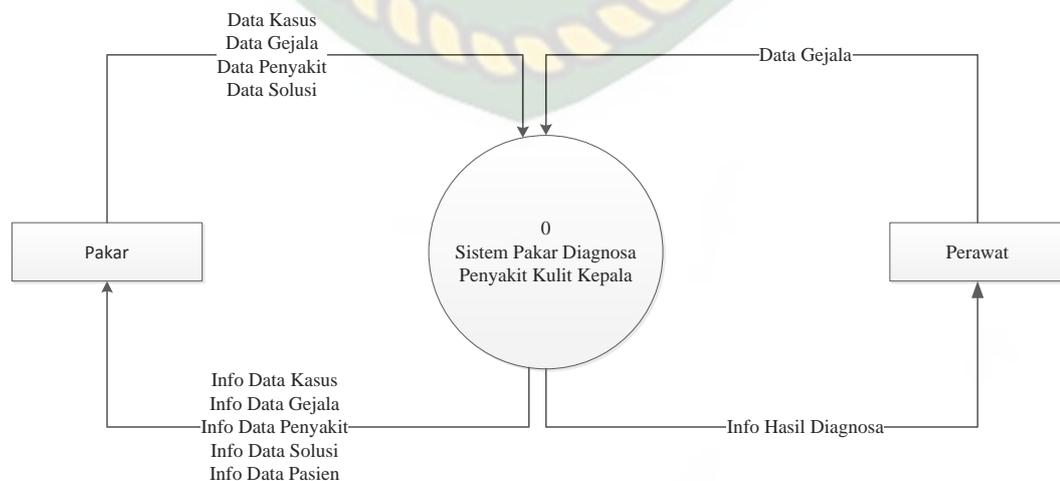
Keterangan :

1. Pakar melakukan login kedalam sistem untuk dapat menginputkan data penyakit, gejala dan solusi.

2. Selanjutnya pakar menginputkan basis kasus yang didapat dari kasus-kasus yang pernah terjadi sebelumnya
3. Perawat menjalankan sistem dengan menginputkan gejala-gejala yang di alami oleh pasien
4. Sistem akan melakukan proses diagnosa menggunakan gejala-gejala yang di inputkan oleh perawat dan melakukan perhitungan similarity (CBR).
5. Setelah melakukan proses perhitungan similarity (CBR) sistem akan menampilkan hasil diagnosa berupa nama penyakit, definisi, gambar, solusi pencegahan dan pengobatan penyakit tersebut.

3.4.2 Konteks Diagram (*Diagram Context*)

Konteks Diagram merupakan alat untuk struktur analisis, pendekatan struktur ini untuk menggambarkan sistem secara keseluruhan. Pada diagram konteks ini menggambarkan sistem pakar yang di butuhkan dan tujuan yang akan di hasilkan. Untuk lebih jelasnya dapat dilihat pada gambar 3.4.

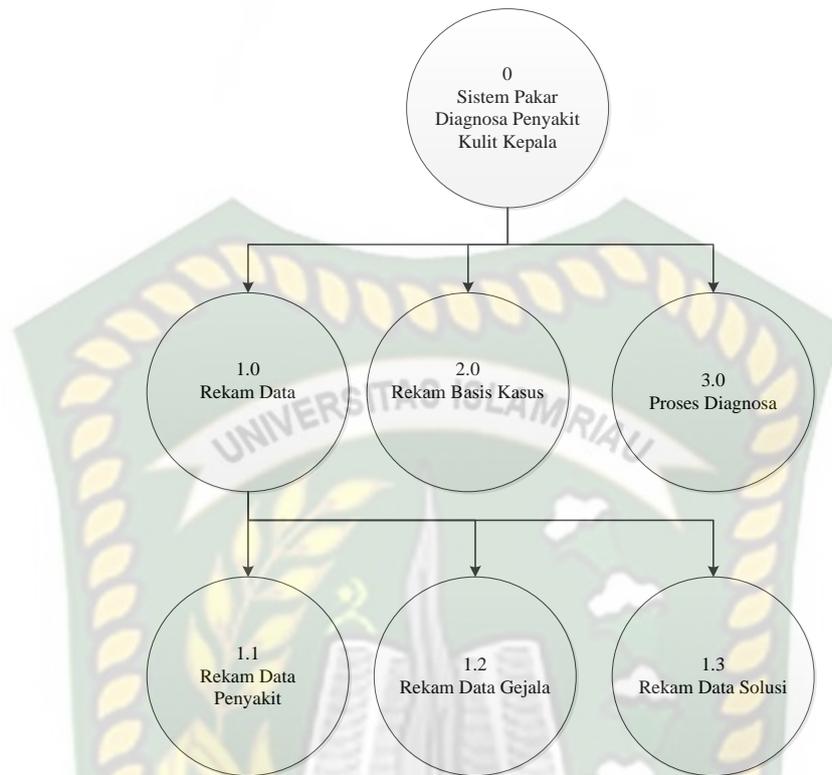


Gambar 3.4 Konteks Diagram

Didalam konteks diagram terdapat dua pelaku pengguna sistem dan satu proses sistem. Pelaku pertama adalah pakar yang mengelola sistem, pelaku kedua adalah perawat, dan proses sistem adalah sistem pakar diagnosa penyakit kulit kepala. Pakar memberikan data ke sistem berupa data kasus-kasus penyakit kulit kepala yang pernah terjadi, data gejala-gejala yang terjadi pada penyakit kulit kepala, data jenis penyakit kulit kepala beserta gambar dan keterangannya, dan data solusi pengobatan serta pencegahan data ini akan di simpan ke dalam sistem sehingga pakar dapat melihat info data kasus, info data gejala, info data penyakit, dan info data solusi. Selanjutnya pelaku kedua yaitu perawat, menginputkan data gejala yang di alami pasien ke dalam sistem, lalu sistem akan memproses sehingga menghasilkan data hasil diagnosa yang akan diberikan pada perawat.

3.4.3 *Hierarchy Chart*

Hirarchy Chart adalah suatu diagram yang menggambarkan permasalahan-permasalahan yang kompleks yang di uraikan pada elemen-elemen yang bersangkutan. Berikut ini adalah gambaran *Hirarchy Chart* pada system pakar diagnosa penyakit kulit kepala, dapat dilihat pada gambar 3.5



Gambar 3.5 Hierarchy Chart

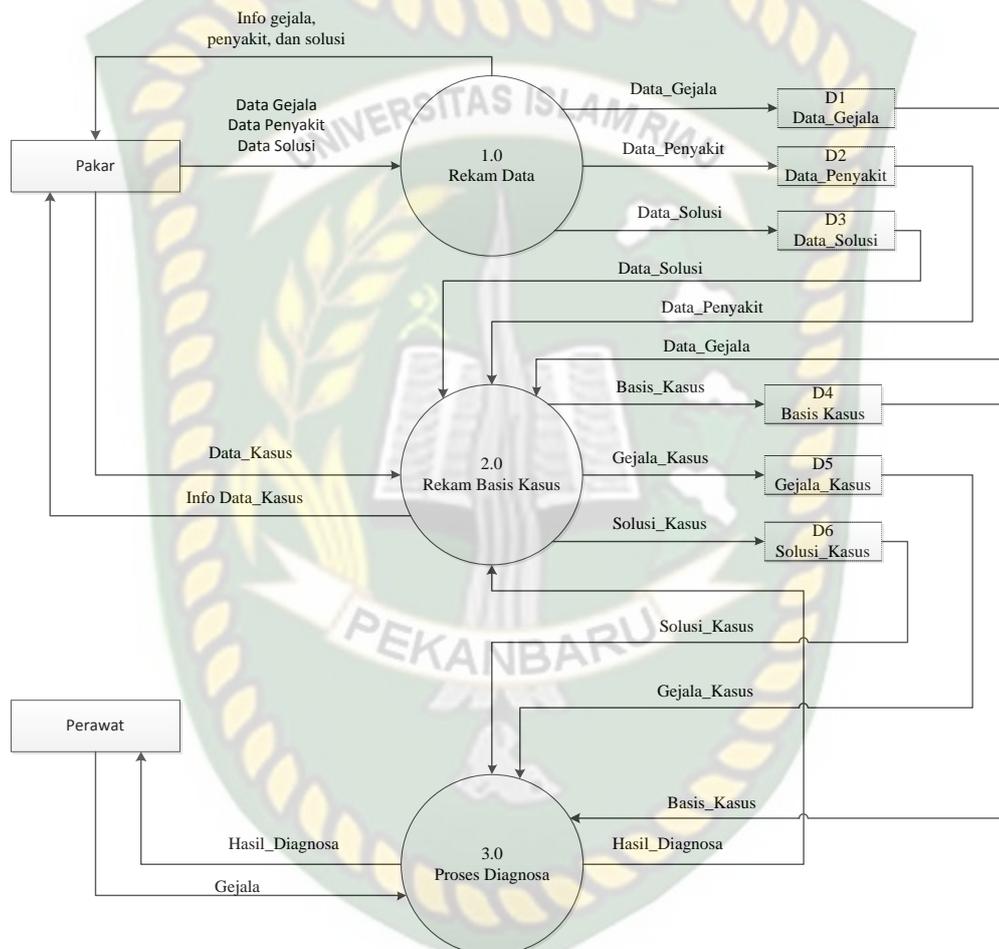
Dari gambar 3.5 *Hierarchy chart* dapat dilihat bahwa pada sistem yang akan dibangun terdapat 3 proses dan 3 sub proses. Proses yang pertama yaitu rekam data, proses kedua yaitu rekam basis kasus, dan yang terakhir yaitu proses diagnosa. Pada proses rekam data terdapat 3 sub proses yaitu rekam data penyakit, rekam data gejala, rekam data solusi.

3.4.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) berfungsi untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa memperhatikan lingkungan fisik dimana data tersebut mengalir. Pada sistem pakar ini memiliki beberapa level proses yaitu :

1. DFD Level 0

DFD level 0 membahas tentang penjabaran sistem yang akan dirancang berdasarkan rancangan pada konteks diagram. Adapun rancangannya seperti pada gambar 3.6 berikut :



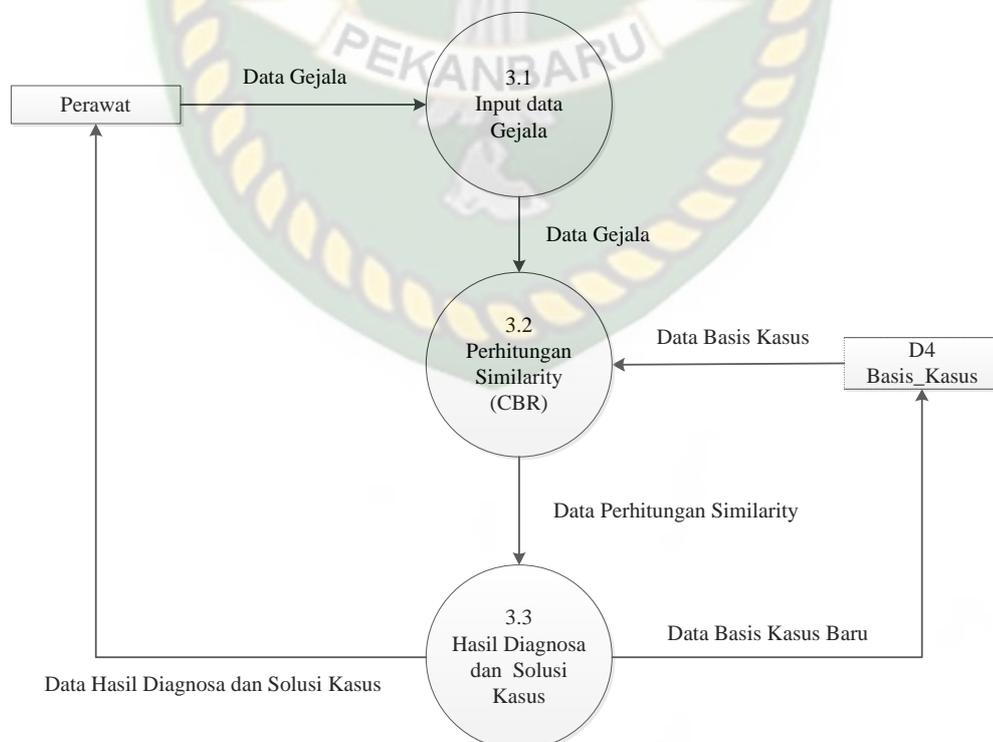
Gambar 3.6 Data Flow Diagram Level 0

Keterangan DFD Level 0 adalah sebagai berikut :

1. Pakar merekam data pada proses 1.0, data akan disimpan ke dalam tabel Data_Gejala, Data_Penyakit, dan Data_Solusi, sistem akan memberikan informasi data yang sudah di rekam kepada pakar.

2. Pakar akan menginputkan data kasus pada proses 2.0, data kasus terdiri dari data gejala, penyakit dan solusi masing-masing akan disimpan kedalam tabel Basis_Kasus, Gejala_Kasus, dan Solusi_Kasus. Sistem akan memberikan informasi data kasus yang sudah di simpan kepada pakar.
3. Perawat melakukan diagnosa penyakit pada proses 3.0 pada proses diagnosa, perawat akan memberikan gejala apa yang dialami pasien kepada sistem selanjutnya sistem akan memproses dan memberikan informasi hasil diagnosa kepada perawat, jika hasil diagnosa memiliki hasil minimal 80% maka pakar dapat merevisi dan menyimpan kedalam basis kasus untuk digunakan kembali.

2. DFD Level 1 Proses 3.0 Proses Diagnosa



Gambar 3.7 Data Flow Diagram Level 1 Proses Diagnosa

Pada level 1 proses 3.0, Sistem akan memberikan pertanyaan kepada perawat berbagai macam gejala, lalu perawat akan menginputkan gejala yang diderita pasien kedalam sistem, sistem akan mengolah data yang di inputkan oleh perawat dengan data pada basis kasus yang sudah ada sesuai dengan metode similarity CBR. Hasil dari proses diagnosa akan diberikan kepada perawat berupa nama penyakit yang diderita dan solusinya, hasil diagnosa ini dapat di revisi dan disimpan oleh pakar ke dalam basis_kasus apabila memiliki hasil minimal 80%.

3.4.5 Desain Output

Desain *output* pada sistem pakar ini terdiri dari :

1. Rancangan *Output* Data Penyakit

HEADER				
DATA PENYAKIT				
<input type="button" value="Tambah"/>				
No	Kode Penyakit	Nama Penyakit	Definisi Penyakit	Aksi (Edit/Hapus)
FOOTER				

Gambar 3.8 Rancangan *Output* Data Penyakit

Gambar 3.8 diatas adalah rancangan tampilan data penyakit. Informasi yang ditampilkan kusus untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah informasi tentang data penyakit berupa kode penyakit, nama penyakit, definisi penyakit, tombol tambah untuk menambah data penyakit, dan tombol aksi untuk mengedit dan menghapus data penyakit.

2. Rancangan *Output* Data Gejala

HEADER			
DATA GEJALA			
<input type="button" value="Tambah"/>			
No	Kode Gejala	Gejala	Aksi (Edit/Hapus)
FOOTER			

Gambar 3.9 Rancangan *Output* Data Gejala Penyakit

Gambar 3.9 diatas adalah rancangan tampilan data gejala. Informasi yang ditampilkan pada halaman ini kusus untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah informasi tentang data gejala berupa kode gejala, gejala, tombol tambah untuk menambah data gejala, dan tombol aksi untuk mengedit dan menghapus data gejala.

3. Rancangan *Output* Data Solusi Penyakit

HEADER			
DATA SOLUSI PENYAKIT			
<div style="border: 1px solid black; display: inline-block; padding: 2px 10px; margin-bottom: 5px;">Tambah</div>			
No	Kode Solusi	Solusi	Aksi (Edit/Hapus)
FOOTER			

Gambar 3.10 Rancangan *Output* Data Solusi Penyakit

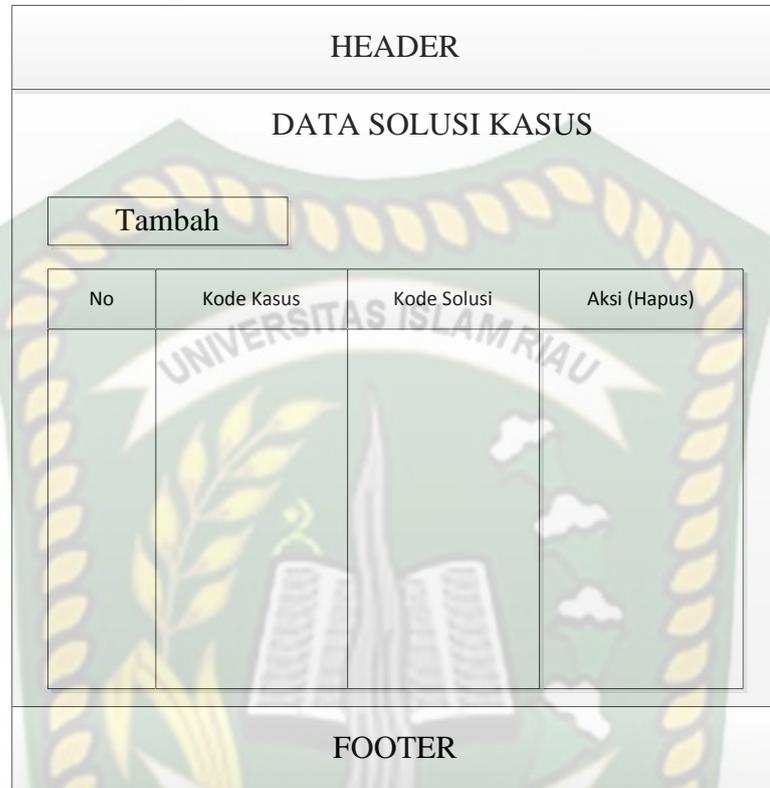
Gambar 3.10 diatas adalah rancangan tampilan data solusi penyakit. Informasi yang ditampilkan pada halaman ini kusus untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah informasi tentang data solusi penyakit berupa kode solusi, solusi, tombol tambah untuk menambah data solusi, dan tombol aksi untuk mengedit dan menghapus data solusi.

4. Rancangan *Output* Data Gejala kasus

HEADER			
DATA GEJALA KASUS			
<input type="button" value="Tambah"/>			
No	Kode Kasus	Kode Gejala	Aksi (Hapus)
FOOTER			

Gambar 3.11 Rancangan *Output* Data Gejala Kasus

Gambar 3.11 diatas adalah rancangan tampilan data gejala kasus. Informasi yang ditampilkan pada halaman ini kasus untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah informasi tentang data gejala kasus berupa kode kasus, kode gejala, tombol tambah untuk menambah data gejala kasus, dan tombol aksi untuk menghapus data gejala kasus.

5. Rancangan *Output* Data Solusi Kasus

HEADER			
DATA SOLUSI KASUS			
<input type="button" value="Tambah"/>			
No	Kode Kasus	Kode Solusi	Aksi (Hapus)
FOOTER			

Gambar 3.12 Rancangan *Output* Data Solusi Kasus

Gambar 3.12 diatas adalah rancangan tampilan data solusi kasus. Informasi yang ditampilkan pada halaman ini kasus untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah informasi tentang data solusi kasus berupa kode kasus, kode solusi, tombol tambah untuk menambah data solusi kasus, dan tombol aksi untuk menghapus data solusi kasus.

6. Rancangan *Output* Data Basis Kasus

HEADER

DATA BASIS KASUS

Tambah

Kode Kasus

Kode Penyakit

Nama Penyakit

Solusi

No	Kode Gejala	Nama Gejala	Aksi (Edit/Hapus)

FOOTER

Gambar 3.13 Rancangan *Output* Basis Kasus

Gambar 3.13 diatas adalah rancangan tampilan data basis kasus. Informasi yang ditampilkan pada halaman ini kusus untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah informasi tentang data basis kasus berupa kode kasus, kode penyakit, nama_penyakit, solusi, kode_gejala, nama_gejala, aksi, dan tombol tambah untuk menambah data basis kasus, dan tombol aksi untuk mengedit dan menghapus data gejala pada basis kasus.

7. Rancangan *Output* Hasil Diagnosa



Gambar 3.14 Rancangan *Output* Hasil Diagnosa

Gambar 3.14 diatas adalah rancangan tampilan data hasil diagnosa. Informasi yang ditampilkan pada halaman ini untuk perawat. Adapun informasi yang ditampilkan pada halaman ini adalah informasi hasil diagnosa berupa gejala, penyakit yang diderita serta solusi yang diberikan, dan tombol diagnosa kembali untuk perawat mendiganosa kembali.

8. Rancangan *Output Revise Kasus Baru*

HEADER					
Data Kasus Baru					
No	Kode Kasus	Gejala	Nama Penyakit	Nama Solusi	Revise
					Yes/No

FOOTER

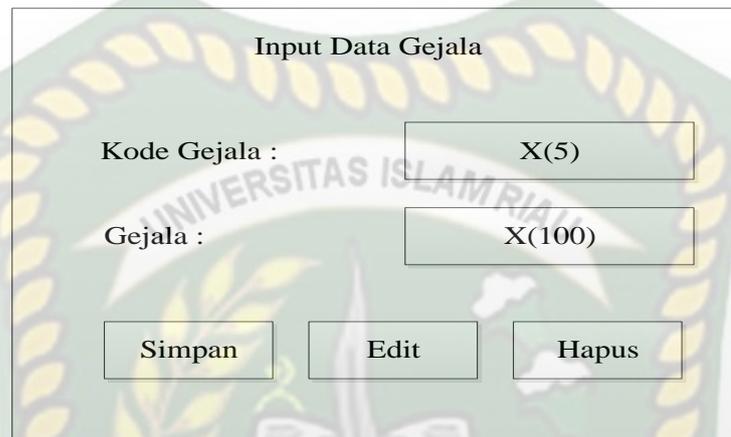
Gambar 3.15 Rancangan *Output Revise Kasus Baru*

Gambar 3.15 diatas adalah rancangan tampilan data hasil diagnose yang dapat di revise oleh pakar jika hasil diagnosa kasus baru memiliki kemiripian minimal 80%. Informasi yang ditampilkan pada halaman ini untuk pakar. Adapun informasi yang ditampilkan pada halaman ini adalah kode kasus, gejala, nama penyakit, dan nama solusi. Pakar dapat menghapus, menyimpan dan mengedit gejala dan solusi sebelum di tambahkan ke basis kasus.

3.4.6 Desain *Input*

Desain input pada sistem pakar ini terdiri dari :

1. Rancangan *input* data gejala



Input Data Gejala

Kode Gejala : X(5)

Gejala : X(100)

Simpan Edit Hapus

Gambar 3.16 Rancangan *Input* Data Gejala

Pada gambar 3.16 dapat dilihat bahwa pakar dapat memasukkan kode gejala dan gejala. Terdapat juga tombol simpan, edit dan hapus untuk mengolah data gejala.

2. Rancangan *input* data penyakit



Input Data Penyakit

Kode Penyakit : X(5)

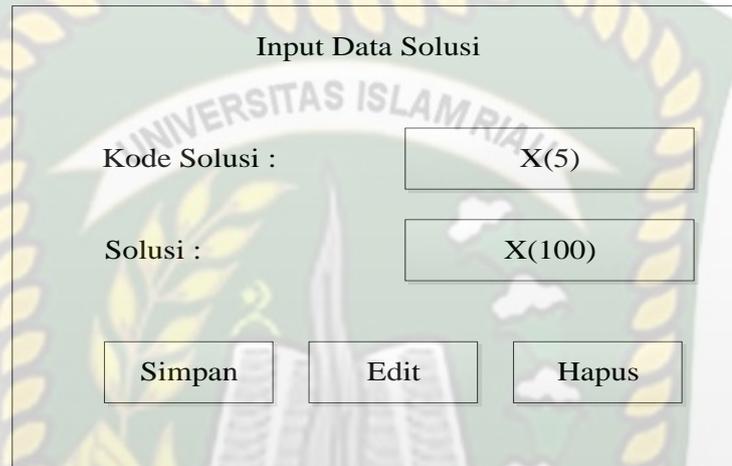
Nama Penyakit : X(100)

Simpan Edit Hapus

Gambar 3.17 Rancangan *Input* Data Penyakit

Pada gambar 3.17 dapat dilihat bahwa pakar dapat memasukkan kode penyakit dan nama penyakit. Terdapat juga tombol simpan, edit dan hapus untuk mengolah data penyakit.

3. Rancangan *input* data solusi



Input Data Solusi

Kode Solusi :

Solusi :

Gambar 3.18 Rancangan *Input* Data Solusi

Pada gambar 3.18 dapat dilihat bahwa pakar dapat memasukkan kode solusi dan solusi. Terdapat juga tombol simpan, edit dan hapus untuk mengolah data solusi.

4. Rancangan *input* data gejala kasus



Input Data Gejala Kasus

Kode Kasus :

Kode Gejala :

Gambar 3.19 Rancangan *Input* Data Gejala Kasus

Pada gambar 3.19 dapat dilihat bahwa pakar dapat memasukkan kode kasus dan kode gejala. Terdapat juga tombol simpan, edit dan hapus untuk mengolah data gejala kasus.

5. Rancangan *input* data solusi kasus

Input Data Solusi Kasus

Kode Kasus :

Kode Solusi :

Gambar 3.20 Rancangan *Input* Data Solusi Kasus

Pada gambar 3.20 dapat dilihat bahwa pakar dapat memasukkan kode kasus dan kode solusi. Terdapat juga tombol simpan, edit dan hapus untuk mengolah data solusi kasus.

6. Rancangan *input* data basis kasus

Input Data Basis Kasus

Kode Kasus :

Jenis Kulit :

Kode Penyakit:

Gambar 3.21 Rancangan *Input* Data Basis Kasus

Pada gambar 3.21 dapat dilihat bahwa pakar dapat memasukkan kode kasus dan kode penyakit. Terdapat juga tombol simpan, edit dan hapus untuk mengolah data basis kasus.

7. Rancangan *input data revise* kasus baru

Input Data Revise

Kode Kasus :

Gejala :

Solusi :

Gambar 3.22 Rancangan *Input Data Revise* Kasus Baru

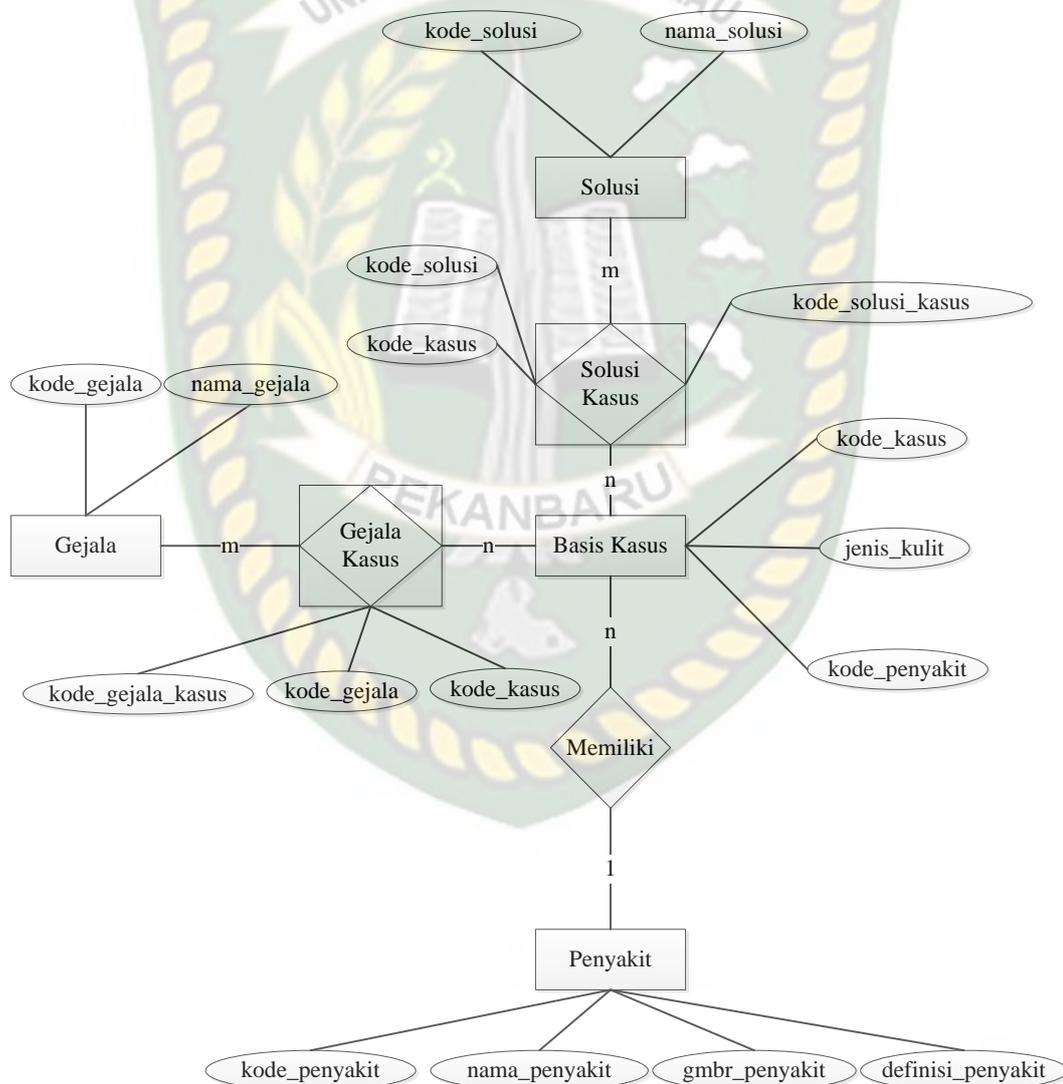
Pada gambar 3.22 dapat dilihat bahwa pakar dapat memasukkan kode kasus baru, menambah gejala, dan solusi. Terdapat juga tombol *revise*, dan batal untuk mengolah data basis kasus.

3.4.7 Perancangan *Database*

Rancangan *database* bertujuan untuk membangun basis data pada sistem. Sub bagian dari pengerjaan basis data meliputi perancangan *Entity Relationship Diagram* (ERD) dan rancangan tabel.

3.4.7.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek data yang mempunyai hubungan antar relasi. ERD untuk memodelkan data digunakan beberapa notasi dan simbol. ERD pada sistem pakar diagnosa penyakit kulit kepala ini untuk lebih jelasnya dapat dilihat pada gambar 3.22.



Gambar 3.23 ERD Sistem Pakar Diagnosa Penyakit Kulit Kepala

3.4.7.2 Rancangan Tabel

Rancangan tabel dibuat pada database MySQL, dengan nama database db_kulit_kepala yang terdiri dari enam tabel yang dapat di uraikan sebagai berikut:

1. Tabel Pakar

Tabel ini diberi nama tabel pakar, digunakan untuk login pakar ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.1

Tabel 3.1 Rancangan Tabel Pakar

No	Field Name	Type	Size	Description
1	Username	Varchar	20	Username (Primary Key)
2	Password	Varchar	50	Password

Tabel ini memiliki 2 *field*. *Field* pertama adalah *field username* bertipe varchar dengan ukuran 20 karakter, *field* kedua adalah *field password* pakar bertipe varchar dengan ukuran 50 karakter

2. Tabel Penyakit

Tabel ini diberi nama tabel penyakit, digunakan untuk menyimpan data penyakit ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.2

Tabel 3.2 Rancangan Tabel Penyakit

No	Field Name	Type	Size	Description
1	kode_penyakit	Char	4	Kode Penyakit (Primary Key)
2	nama_penyakit	Varchar	100	Nama Penyakit
3	definisi_penyakit	Varchar	200	Definisi Penyakit
4	gmbr_penyakit	Varchar	100	Gambar Penyakit

Tabel ini memiliki 4 *field*. *Field* pertama adalah *field kode_penyakit* bertipe varchar dengan ukuran 4 karakter. *field* kedua adalah *field nama_penyakit* bertipe varchar dengan ukuran 100 karakter, *field* ketiga adalah *field definisi_penyakit*

bertipe varchar dengan ukuran 200 karakter, dan *field* terakhir adalah *field* *gmr_penyakit* bertipe varchar dengan ukuran 100 karakter.

3. Tabel Gejala

Tabel ini diberi nama tabel gejala, digunakan untuk menyimpan data gejala ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.3

Tabel 3.3 Rancangan Tabel Gejala

No	Field Name	Type	Size	Description
1	kode_gejala	Char	4	Kode Gejala (<i>Primary Key</i>)
2	nama_gejala	Varchar	100	Nama Gejala

Tabel ini memiliki 2 *field*. *Field* pertama adalah *field* *kode_gejala* bertipe varchar dengan ukuran 4 karakter, dan *field* kedua adalah *field* *nama_gejala* bertipe varchar dengan ukuran 100 karakter.

4. Tabel Solusi

Tabel ini diberi nama tabel solusi, digunakan untuk menyimpan data solusi ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.4

Tabel 3.4 Rancangan Tabel Solusi

No	Field Name	Type	Size	Description
1	kode_solusi	Char	4	Kode Solusi (<i>Primary Key</i>)
2	nama_solusi	Varchar	100	Nama Solusi

Tabel ini memiliki 2 *field*. *Field* pertama adalah *field* *kode_solusi* bertipe varchar dengan ukuran 4 karakter, dan *field* kedua adalah *field* *nama_solusi* bertipe varchar dengan ukuran 100 karakter.

5. Tabel Basis Kasus

Tabel ini diberi nama tabel basis kasus, digunakan untuk menyimpan data basis kasus ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.5

Tabel 3.5 Rancangan Tabel Basis Kasus

<i>No</i>	<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
1	kode_kasus	Char	4	Kode Kasus (<i>Primary Key</i>)
2	jenis_kulit	Varchar	20	Jenis Kulit Kepala
3	kode_penyakit	Char	4	Kode Penyakit

Tabel ini memiliki 3 *field*. *Field* pertama adalah *field* kode_kasus bertipe varchar dengan ukuran 4 karakter, *field* kedua adalah *field* jenis_kulit bertipe varchar dengan ukuran 20 karakter, dan *field* terakhir adalah *field* kode_penyakit bertipe varchar dengan ukuran 4 karakter.

6. Tabel Gejala Kasus

Tabel ini diberi nama tabel gejala kasus, digunakan untuk menyimpan data gejala kasus ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.6

Tabel 3.6 Rancangan Tabel Gejala Kasus

<i>No</i>	<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
1	kode_kasus	Char	4	Kode Kasus
2	kode_gejala	Char	4	Kode Gejala

Tabel ini memiliki 2 *field*. *Field* pertama adalah *field* kode_kasus bertipe varchar dengan ukuran 4 karakter, dan *field* kedua adalah *field* kode_gejala bertipe varchar dengan ukuran 4 karakter.

7. Tabel Solusi Kasus

Tabel ini diberi nama tabel solusi kasus, digunakan untuk menyimpan data solusi kasus ke dalam sistem. Adapun rancangannya dapat dilihat pada tabel 3.7

Tabel 3.7 Rancangan Tabel Solusi Kasus

<i>No</i>	<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
1	kode_kasus	Char	4	Kode Kasus
2	kode_solusi	Char	4	Kode Solusi

Tabel ini memiliki 2 *field*. *Field* pertama adalah *field* kode_kasus bertipe varchar dengan ukuran 4 karakter, dan *field* kedua adalah *field* kode_solusi bertipe varchar dengan ukuran 4 karakter.

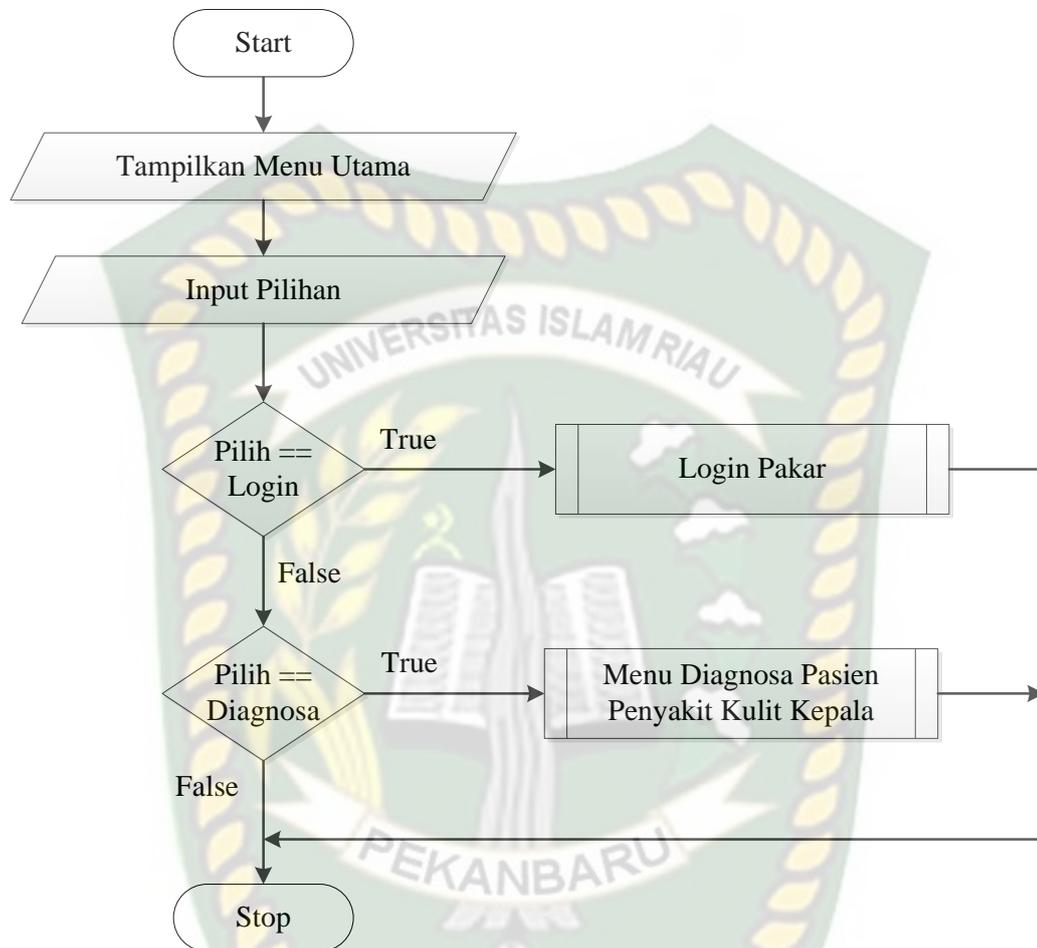
3.4.8 Desain Logika Program

Desain logika program yang akan dikembangkan adalah sebagai berikut :

3.4.8.1 Flowchart Program

Flowchart merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan untuk proses pengolahan data serta hubungan antara peralatan tersebut. Adapun alur logika program digambarkan melalui *flowchart* berikut :

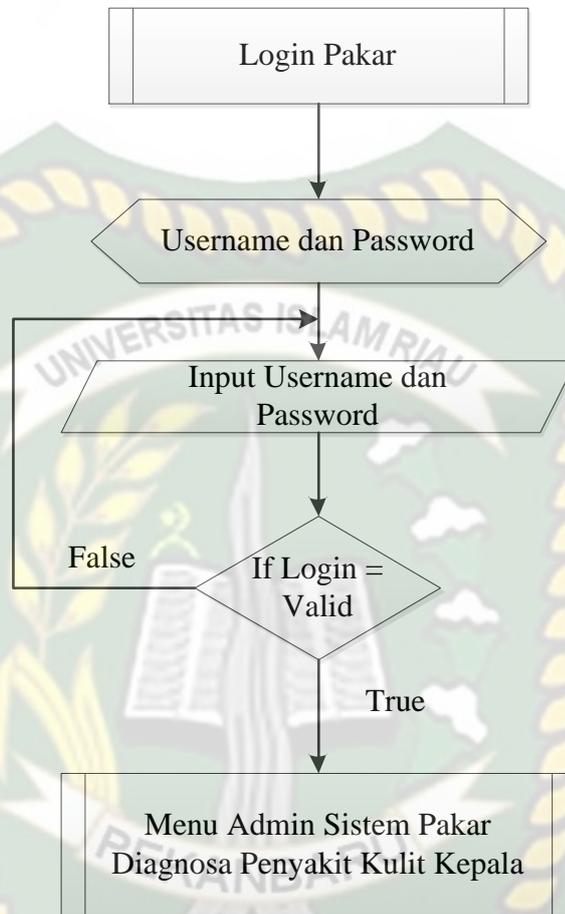
1. Flowchart Utama Program



Gambar 3.24 Flowchart Utama Program

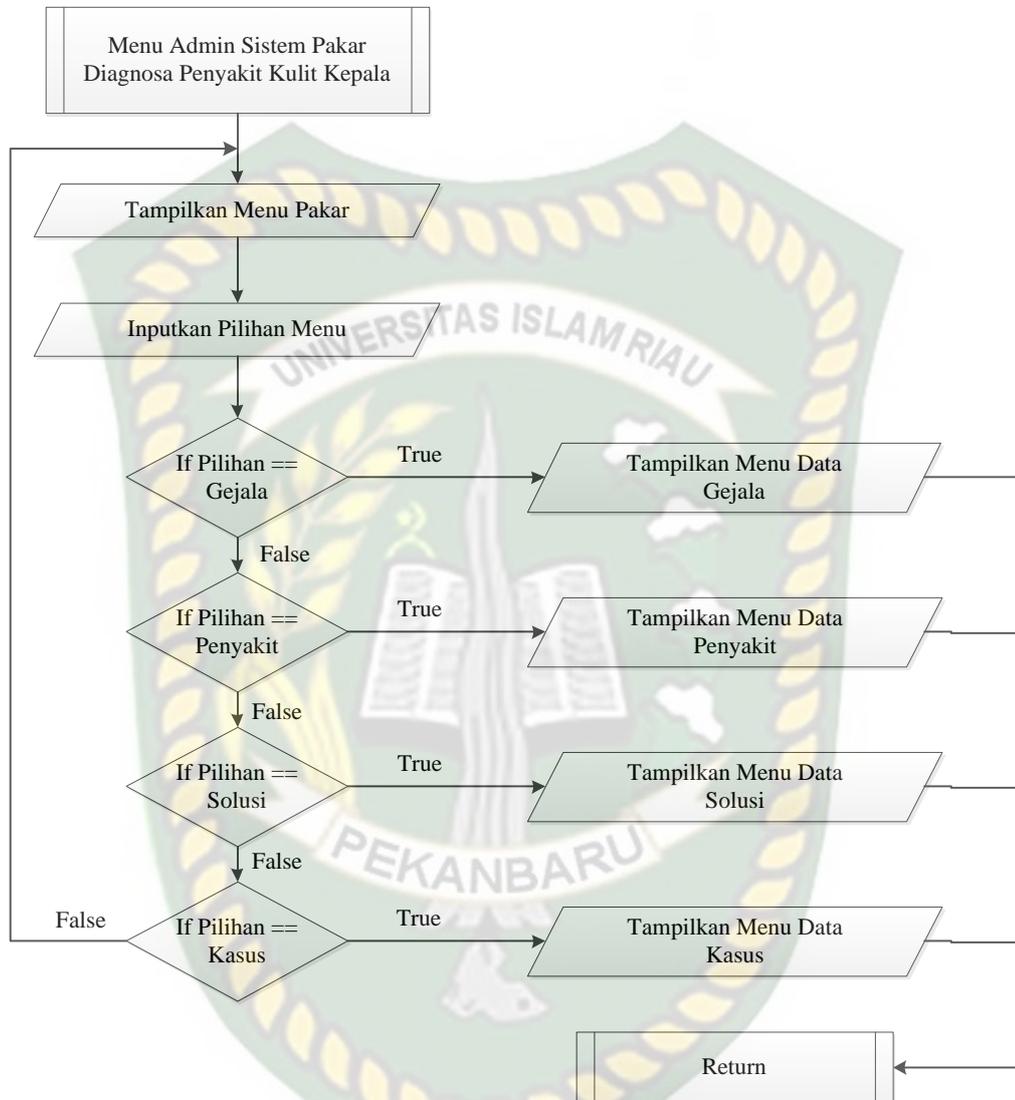
Berdasarkan gambar 3.24, terdapat dua pilihan menu yakni menu diagnosa Pasien Penyakit Kulit Kepala dan menu login pakar, pengguna yang memilih menu login akan masuk kehalaman login pakar, dan bagi pengguna yang memilih menu diagnosa pasien penyakit kulit kepala akan masuk kehalaman menu pasien.

2. Flowchart Login Pakar

**Gambar 3.25 Flowchart Login Pakar**

Berdasarkan Gambar 3.25, pengguna dapat menginputkan username dan passwordnya. Apabila valid pengguna akan di arahkan ke halaman menu pakar.

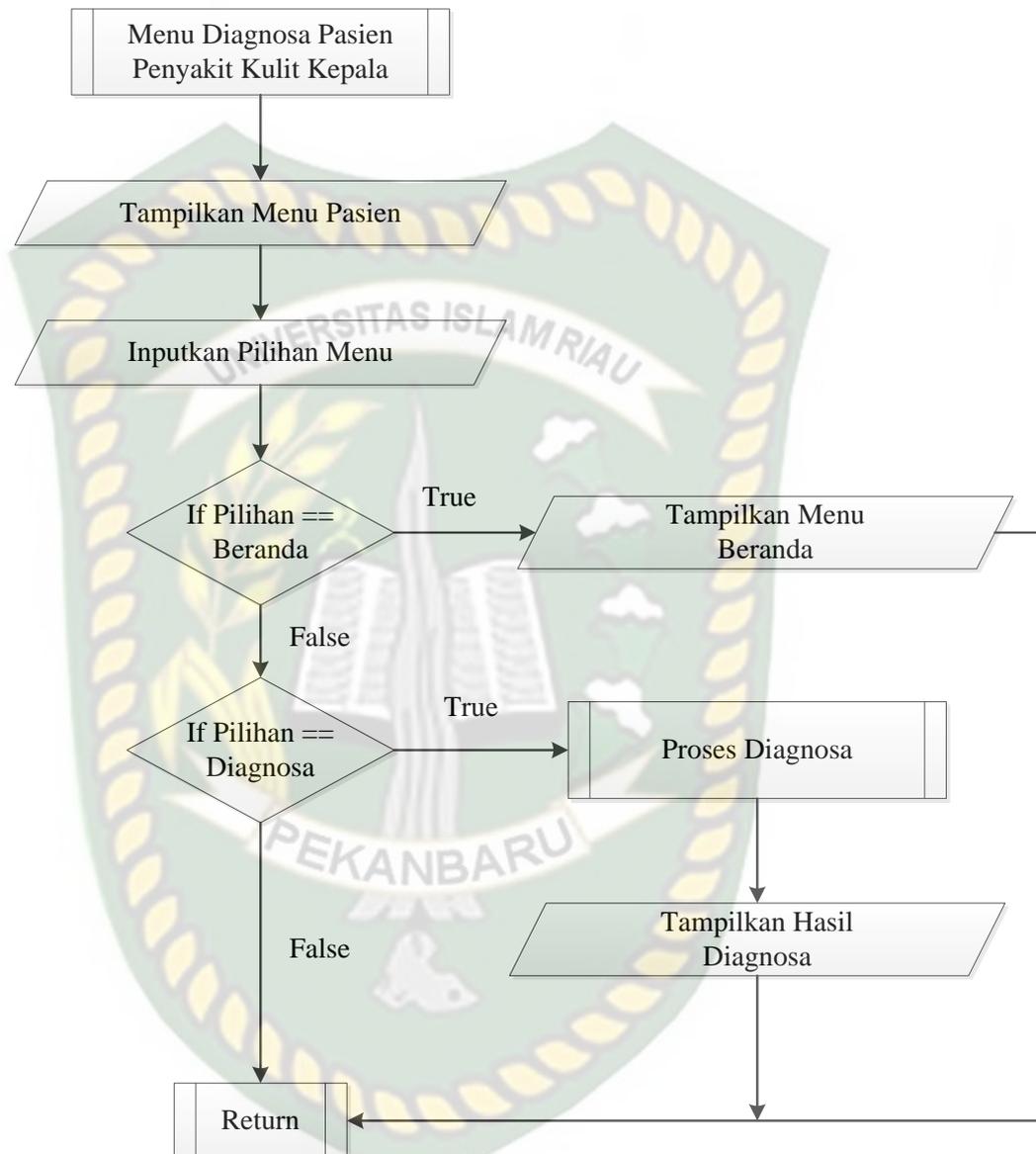
3. Flowchart Menu Pakar



Gambar 3.26 Flowchart Menu Pakar

Berdasarkan gambar 3.26 pada menu pakar terdapat 4 pilihan, jika memilih gejala akan masuk ke halaman menu data gejala, jika memilih penyakit akan masuk ke halaman menu data penyakit, jika memilih solusi akan masuk ke halaman menu data solusi, dan jika memilih kasus akan masuk ke dalam halaman menu data kasus. Pakar dapat mengolah data di setiap halaman menu yang dipilih.

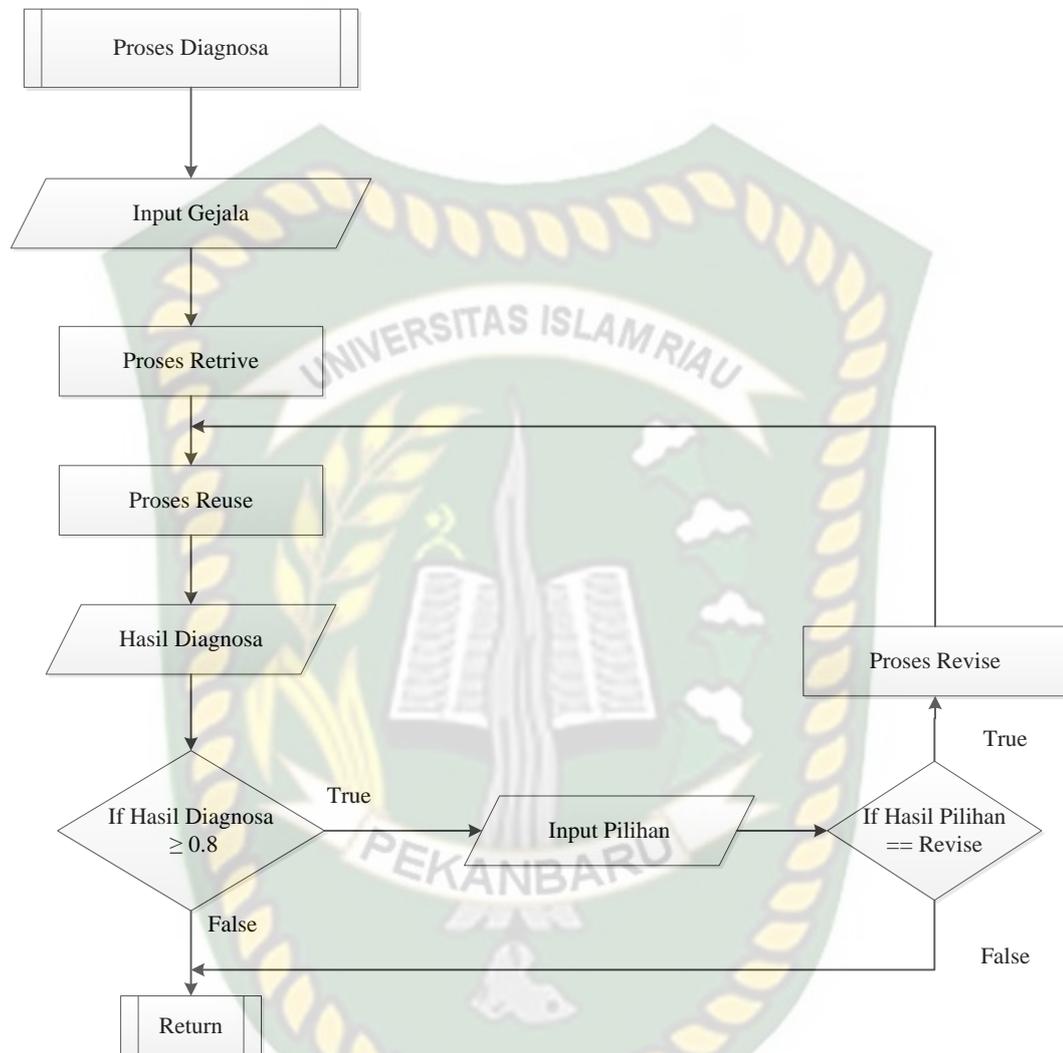
4. Flowchart Menu Pasien



Gambar 3.27 Flowchart Menu Pasien

Berdasarkan gambar 3.27 dapat dilihat bahwa perawat dapat menampilkan menu, setelah menu ditampilkan perawat dapat menginputkan menu pilihan, jika menu pilihan adalah beranda maka sistem akan menampilkan menu beranda dan jika menu pilihan adalah diagnosa maka sistem akan melakukan proses diagnosa kemudian hasil diagnosa akan ditampilkan kembali kepada perawat.

5. Flowchart Proses Diagnosa



Gambar 3.28 Flowchart Proses Diagnosa

Berdasarkan gambar 3.28 pada proses diagnosa perawat dapat menginputkan gejala yang di alami pasien ke dalam sistem, lalu sistem akan melakukan proses retrieve untuk mencari kesamaan gejala pada data kasus yang ada dalam database, selanjutnya data kasus akan digunakan kembali untuk mencari nilai kemiripan kasus yang baru pada proses reuse. Setelah proses reuse maka sistem akan mendapatkan hasil diagnosa, jika hasil diagnosa ini memiliki kemiripan minimal

80% akan disimpan ke dalam sistem yang nantinya akan di proses oleh pakar, pakar dapat memilih ingin merevisi atau tidak kasus baru tersebut, jika direvisi maka kasus baru akan disimpan ke dalam basis kasus sebagai kasus baru, dan pada akhirnya akan digunakan kembali dalam proses reuse.



Dokumen ini adalah Arsip Miik :

Perpustakaan Universitas Islam Riau