

BAB IV

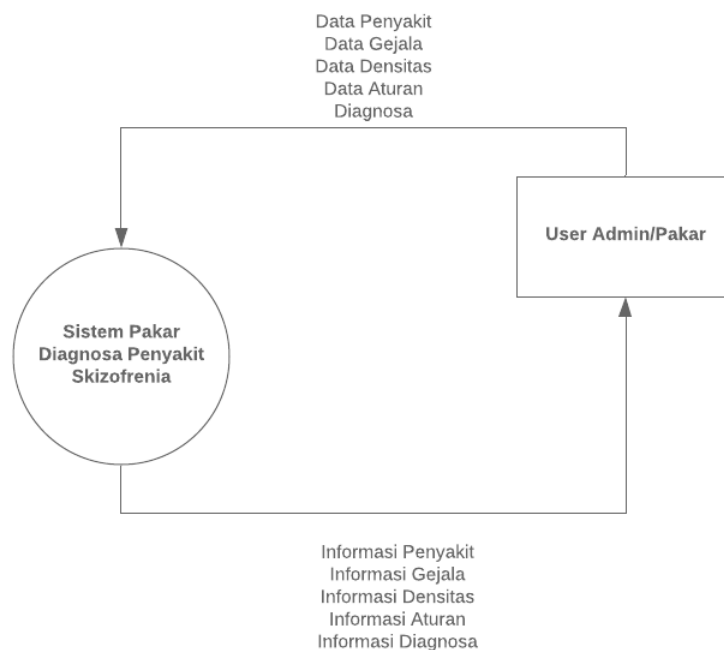
PERANCANGAN

4.1 Perancangan Data Flow Diagram

Data Flow Diagram adalah sebuah diagram yang digunakan untuk menggambarkan suatu sistem bekerja, yang dipresentasikan dalam bentuk aliran data agar lebih mudah dipahami.

4.1.1 DFD Level 0

DFD level 0 adalah sebuah diagram dimana pada tahap ini menggambarkan seluruh proses yang terjadi dalam suatu sistem dan menggambarkan interaksi antara entitas dan proses. Untuk lebih jelasnya dapat dilihat pada Gambar 4.1

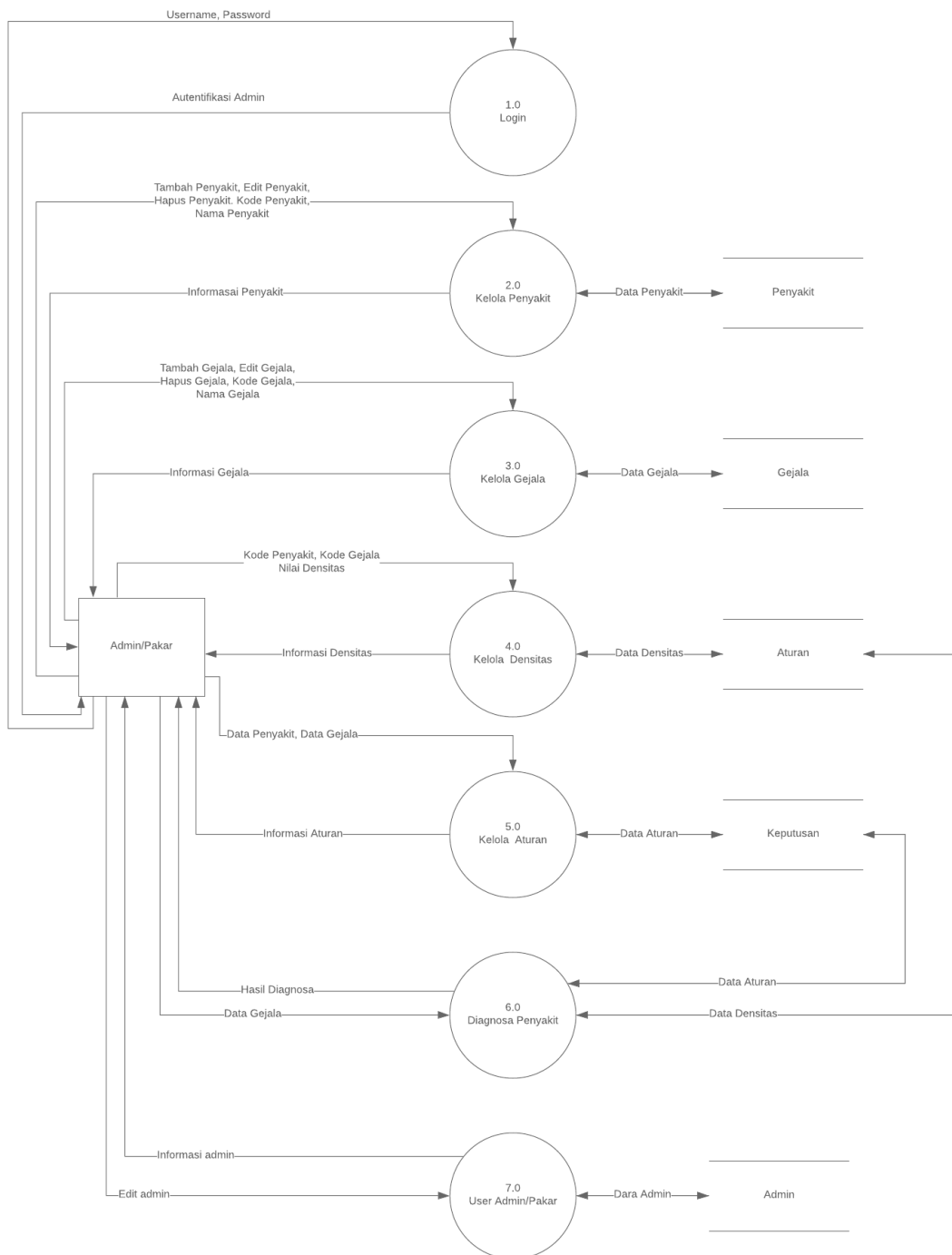


Gambar 4.1 DFD Level 0

Pada Gambar 4.1 menjelaskan interaksi antara admin/pakar dengan sistem pakar. Admin/pakar dapat memasukan data penyakit, data gejala, data densitas, data aturan, dan melakukan diagnosa, lalu akan menerima informasi penyakit, informasi gejala, informasi densitas, informasi aturan, dan informasi diagnosa.

4.1.2 DFD Level 1

DFD Level 1 adalah sebuah diagram dimana pada tahap ini akan menjelaskan lebih detail mengenai proses yang terjadi pada DFD level 0. Untuk lebih jelasnya dapat dilihat pada Gambar 4.2

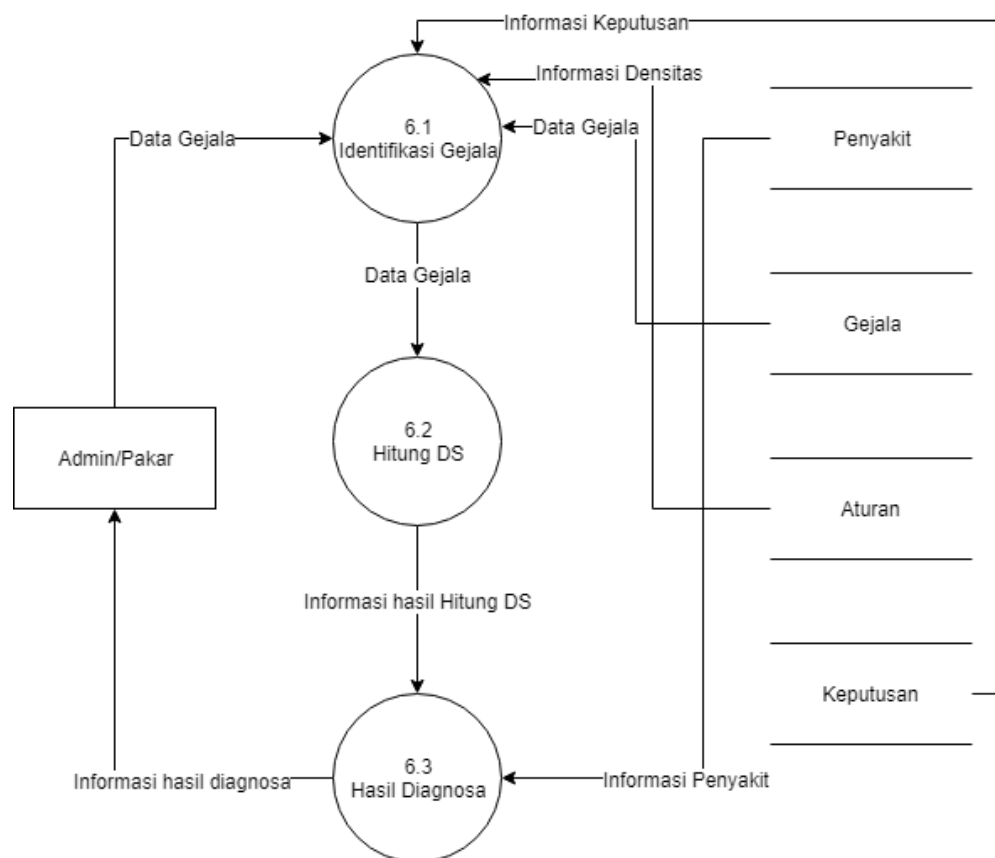


Gambar 4.2 DFD Level 1

Proses yang terbentuk pada Gambar 4.2 yaitu proses login, proses kelola penyakit, proses kelola gejala, proses kelola densitas, proses kelola aturan dan proses diagnosa penyakit. Pada proses kelola penyakit dan kelola gejala admin/pakar dapat menginputkan tambah, edit, dan hapus. Pada proses kelola densitas admin/pakar dapat menginputkan nilai densitas. Pada proses kelola aturan admin/pakar dapat menginputkan data penyakit dan data gejala untuk dijadikan aturan. Pada proses diagnosa penyakit admin/pakar dapat melakukan diagnosa.

4.1.3 DFD Level 2 Proses Diagnosa Penyakit

Pada DFD tahap ini, akan menjelaskan proses apa saja yang terjadi pada DFD level 1 sehingga proses lebih detail dan terperinci. Untuk lebih jelasnya dapat dilihat pada Gambar 4.3



Gambar 4.3 DFD Level 2 Proses Diagnosa Penyakit

Pada proses diagnosa penyakit terdapat proses yang lebih detail diantaranya proses identifikasi gejala, proses hitung nilai DS, dan proses hasil dari diagnosa penyakit. Pada proses identifikasi gejala admin/pakar akan menginputkan gejala apa saja yang dialami kedalam sistem. Lalu pada proses hitung nilai DS sistem akan menghitung nilai DS terhadap setiap gejala yang dipilih. Pada proses hasil diagnosa sistem akan menampilkan hasil diagnosa sesuai dengan gejala yang dipilih yang sudah diperhitungkan dengan perhitungan *Dempster Shafer*.

4.2 Perancangan Basisdata

Perancangan basisdata digunakan sebagai penyimpanan apa saja yang dibutuhkan untuk mendukung pembuatan sistem. Adapun tabel-tabel yang terbentuk untuk disimpan ke dalam sistem sebagai berikut.

4.2.1 Struktur Tabel

Struktur tabel adalah tempat penyimpanan data yang sudah terstruktur ke dalam tabel. Adapun tabel yang terbentuk menjadi 5 tabel yaitu:

a. Tabel Admin

Tabel admin digunakan untuk penyimpanan data admin. Tabel Admin memiliki 4 kolom yaitu id_admin, Username, Password, dan Nama_admin. Untuk lebih jelasnya dapat dilihat pada Tabel 4.1 di bawah ini:

Tabel 4.1 Tabel Admin

No	Nama Field	Tipe Data	Ukuran	Peran
1	id_admin	Integer	11	Primary Key
2	Username	Varchar	225	
3	Password	Varchar	225	
4	Nama_admin	Varchar	225	

Kolom id_admin memiliki tipe data integer dengan ukuran sepanjang 11 karakter dan memiliki peran sebagai *primary key*. Kolom username memiliki tipe data varchar dengan ukuran sepanjang 225 karakter. Kolom password memiliki tipe data varchar dengan ukuran sepanjang 225 karakter. Kolom nama_admin memiliki tipe data varchar dengan ukuran sepanjang 225 karakter.

b. Tabel Penyakit

Tabel penyakit digunakan untuk penyimpanan data penyakit skizofrenia. Tabel penyakit memiliki 3 kolom yaitu `id_penyakit`, `kode_penyakit`, dan `nama_penyakit`. Untuk lebih jelasnya dapat dilihat pada Tabel 4.2 di bawah ini:

Tabel 4.2 Tabel Penyakit

No	Nama Field	Tipe Data	Ukuran	Peran
1	<code>id_penyakit</code>	Integer	11	Primary Key
2	<code>Kode_penyakit</code>	Varchar	225	
3	<code>Nama_penyakit</code>	Varchar	225	

Kolom `id_penyakit` memiliki tipe data integer dengan ukuran sepanjang 11 karakter dan memiliki peran sebagai *primary key*. Kolom `kode_penyakit` memiliki tipe data varchar dengan ukuran sepanjang 225 karakter. Kolom `nama_penyakit` memiliki tipe data varchar dengan ukuran sepanjang 225 karakter.

c. Tabel Gejala

Tabel gejala digunakan untuk penyimpanan data gejala. Tabel gejala memiliki 4 kolom yaitu `id_gejala`, `kode_gejala`, `nama_gejala`, dan `klasifikasi`. Untuk lebih jelasnya dapat dilihat pada Tabel 4.3 di bawah ini:

Tabel 4.3 Tabel Gejala

No	Nama Field	Tipe Data	Ukuran	Peran
1	<code>id_gejala</code>	Integer	11	Primary Key
2	<code>kode_gejala</code>	Varchar	225	
3	<code>nama_gejala</code>	Text		
4	<code>klasifikasi</code>	Integer	11	

d. Tabel Aturan

Tabel aturan digunakan untuk menyimpan data densitas dari setiap gejala terhadap suatu penyakit. Tabel aturan memiliki 4 kolom yaitu `id_aturan`, `id_penyakit`, `id_gejala`, dan `densitas`. Untuk lebih jelasnya dapat dilihat pada Tabel 4.4 di bawah ini:

Tabel 4.4 Tabel Aturan

No	Nama Field	Tipe Data	Ukuran	Peran
1	id_aturan	Integer	11	Primary Key
2	id_penyakit	Integer	11	
3	id_gejala	Integer	11	
4	densitas	Float		

e. Tabel Keputusan

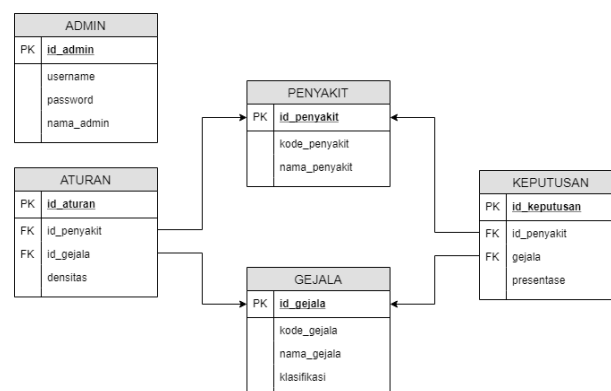
Tabel keputusan digunakan untuk menyimpan data keputusan atau aturan yang terbentuk untuk menentukan suatu penyakit terhadap gejala yang dipilih. Tabel keputusan memiliki 4 kolom yaitu id_keputusan, id_penyakit, gejala, dan presentase. Tabel keputusan dapat dilihat pada Tabel 4.5

Tabel 4.5 Tabel Keputusan

No	Nama Field	Tipe Data	Ukuran	Peran
1	id_keputusan	Integer	11	Primary Key
2	id_penyakit	Integer	11	
3	gejala	Varchar	225	
4	presentase	Float		

4.2.2 Relasi Antar Tabel

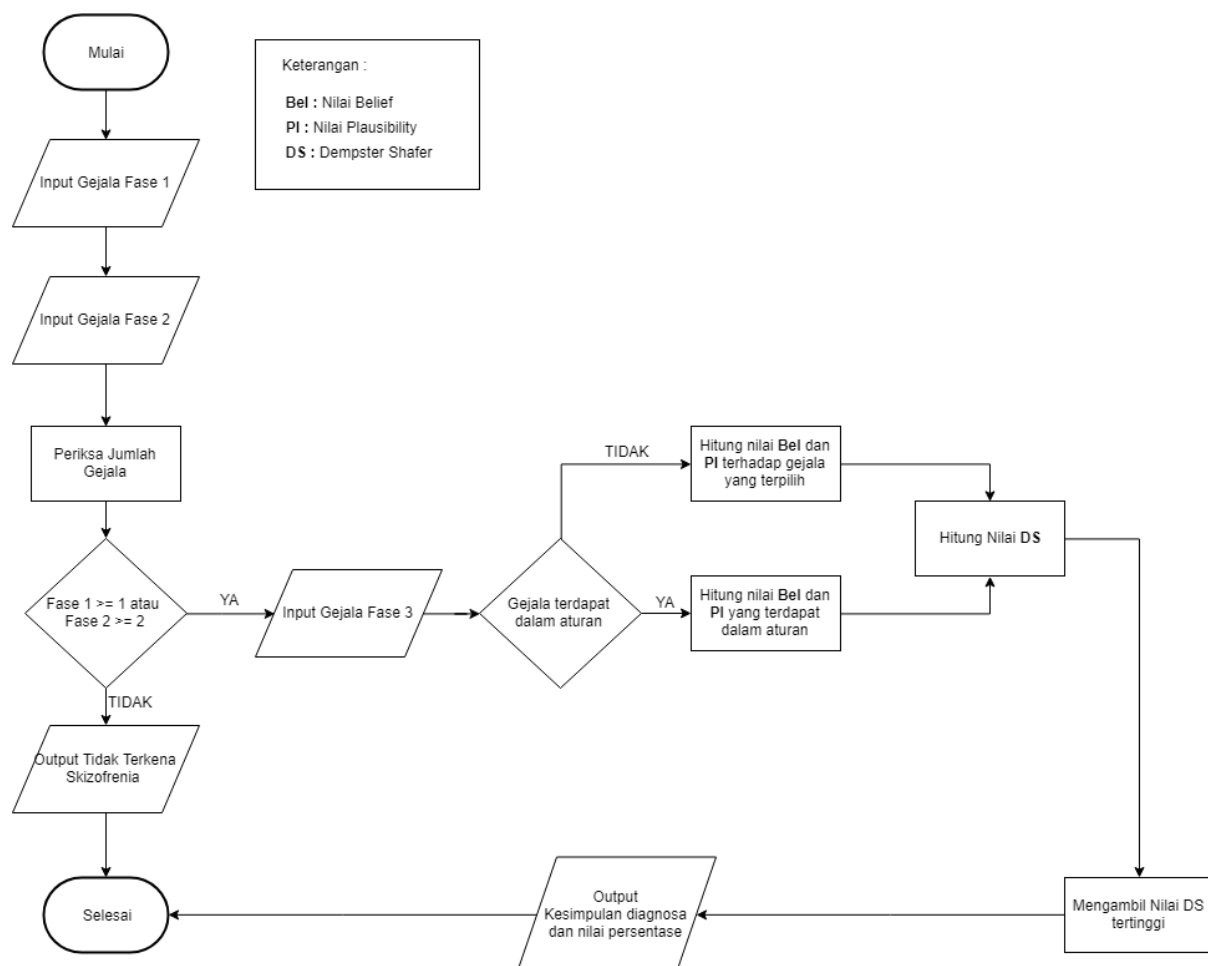
Relasi antar tabel merupakan hubungan yang terjadi antar satu tabel dengan tabel lainnya yang ada di dalam basisdata untuk mengetahui apakah terdapat keterkaitan antara satu tabel dengan tabel lainnya. Relasi antar tabel yang terbentuk dapat dilihat pada Gambar 4.4



Gambar 4.4 Relasi Antar Tabel

4.3 Flowchart

Flowchart merupakan suatu bagan atau simbol yang digunakan untuk menggambarkan urutan suatu proses atau alur yang terjadi pada suatu sistem. Pada flowchart ini akan menjelaskan bagaimana alur atau urutan dalam melakukan diagnosis penyakit skizofrenia. Alur flowchart didapatkan dari hasil diskusi dan wawancara bersama dr. Wikan Ardinigrum M.Sc., Sp.KJ . Dalam menentukan diagnosis dini secara manual terdapat 2 fase yaitu fase wawancara anamnesis dan wawancara klinis. Pada proses wawancara anamnesis pihak keluarga akan bercerita tentang pasien, dan pada proses wawancara klinis pihak dokter akan bertanya tentang keadaan pasien. Sistem ini akan ditempatkan pada fase wawancara klinis seperti layaknya dokter sehingga terbentuk flowchart yang dapat dilihat pada Gambar 4.5



Gambar 4.5 Flowchart Proses Diagnosis Penyakit Skizofrenia

Pada Gambar 4.5 Gambar 4.5 Flowchart Proses Diagnosis Penyakit Skizofrenia dapat diketahui alur proses diagnosis penyakit skizofrenia. Adapun penjelasan dari flowchart tersebut yaitu:

1. Pakar akan memasukan gejala fase 1 dan gejala fase 2 yang dialami pasien
2. Sistem akan memproses hasil dari gejala fase 1 dan fase 2. Apabila gejala pada fase 1 terdapat minimal 1 gejala atau gejala pada fase 2 terdapat minimal 2 gejala, maka pakar akan memasukan gejala fase 3. Apabila gejala pada fase 1 tidak terdapat minimal 1 gejala atau gejala pada fase 2 tidak terdapat minimal 2 gejala, maka dapat disimpulkan pasien tidak terkena penyakit skizofrenia.
3. Sistem akan memeriksa gejala yang terpilih ke dalam aturan. Apabila terdapat pada aturan, maka sistem akan menghitung nilai *belief* dan *plausibility* yang ada dalam aturan. Apabila tidak terdapat dalam aturan, maka sistem akan menghitung nilai *belief* dan *plausibility* yang terpilih.
4. Setelah mendapatkan nilai *belief* dan *plausibility* maka sistem akan menghitung nilai Dempster Shafer.
5. Setelah mendapatkan nilai Dempster Shafer maka dicari nilai tertinggiya dan menampilkan hasil kesimpulan diagnosa dan persentase.

4.4 Perancangan Sistem Pakar Pada Sistem

Dalam menyelesaikan masalah metode yang digunakan adalah ketidakpastian dengan pendekatan *Dempster Shafer*. Proses diagnosis dini akan menggunakan metode *Dempster Shafer* sehingga *Dempster Shafer* perlu dimasukan kedalam sistem. Perancangan *Dempster Shafer* dibagi menjadi 2 fungsi yaitu:

a. Fungsi Hitung_ds

Fungsi *hitung_ds* berguna untuk mendefinisikan setiap gejala yang dipilih dan memasukan nilai densitas pada setiap gejala kedalam sebuah array. Untuk algoritmanya dapat dilihat pada Gambar 4.6

```
function hitung_ds($jawaban) {
    foreach ($jawaban as $key => $id_gejala) {
        $this->db->where('id_gejala', $id_gejala);
        $aturan = $this->db->get('aturan')->result_array();

        foreach ($aturan as $no => $value) {
```



```

        $penyakit_gejala[$value['id_gejala']]['penyakit'][] =
value['id_penyakit'];

        $penyakit_gejala[$value['id_gejala']]['densitas'] =
$value['densitas'];
    }
}

$data_awal = array();

foreach ($penyakit_gejala as $id_gejala => $value) {
    // mengambil isi dari penyakit dan densitas
    $data_awal[] = $value;
}

$jml_gejala = count($data_awal);

foreach ($data_awal as $key => $value) {
    // menggabungkan semua id_penyakit
    $penyakit = implode($value['penyakit'], ',');
    $datahitung[$key][$penyakit] = $value['densitas'];
    $datahitung[$key]['teta'] = 1-$value['densitas'];
}

for($i=0;$i<$jml_gejala-1;$i++){

    if (empty($_SESSION['kombinasi'])) {
        $data1a[0] = $datahitung[0];
        $data1a[1] = $datahitung[1];
        $_SESSION['kombinasi'][0]['dataawal'] = $data1a;

        $this->perhitungan_kombinasi($_SESSION['kombinasi'],0);
    } else {
        $data1a[0] = $_SESSION['kombinasi'][$i-1]['jumlahakhir'];
        $data1a[1] = $datahitung[$i+1];
        $_SESSION['kombinasi'][$i]['dataawal'] = $data1a;

        $this->perhitungan_kombinasi($_SESSION['kombinasi'],$i);
    }
}

```

Gambar 4.6 Perancangan *Dempster Shafer* Fungsi Hitung_ds

b. Fungsi Perhitungan_kombinasi

Fungsi `perhitungan_kombinasi` digunakan untuk menghitung hasil kombinasi yang dilakukan pada setiap gejala dan membuat irisan untuk mengetahui hasil akhir. Untuk algoritamnya dapat dilihat pada Gambar 4.7

```

function perhitungan_kombinasi($kombinasi, $index){

    foreach ($kombinasi[$index]['dataawal'][0] as $baris => $isi_baris)
    {

```

```

        foreach ($kombinasi[$index]['dataawal'][1] as $kolom =>
$isi_kolom) {

            $data2a[$kolom][$baris] = $isi_baris*$isi_kolom;
        }
    }

    foreach ($data2a as $key => $value) {

        foreach ($value as $data_penyakit => $densitas) {

            if ($key!="teta" AND $data_penyakit!="teta") {
                $penyakit1 = explode(",", $key);
                $penyakit2 = explode(",", $data_penyakit);

                $array_sama = array_intersect($penyakit1, $penyakit2);

                if (!empty($array_sama)) {
                    $penyakit_sama = implode(" ", $array_sama);
                    $data3a[$penyakit_sama]=$densitas;

                    $data2a1[$key][$data_penyakit] = $penyakit_sama;
                } else {
                    $data3a['konflik'] = $densitas;

                    $data2a1[$key][$data_penyakit] = 'konflik';
                }
            } elseif ($key!="teta" AND $data_penyakit=="teta") {

                $data3a[$key] = $densitas;

                $data2a1[$key][$data_penyakit] = $key;
            } elseif ($key=="teta" AND $data_penyakit!="teta") {

                $data3a[$data_penyakit] = $densitas;

                $data2a1[$key][$data_penyakit] = $data_penyakit;
            } else {

                $data3a['teta'] = $densitas;

                $data2a1[$key][$data_penyakit] = "teta";
            }
        }
    }

    foreach ($data3a as $key => $value) {

        foreach ($value as $key_2 => $densitas) {

            $data4a[$key_2][$key] = $densitas;
        }
    }

    foreach ($data4a as $key => $value) {

        $data5a[$key] = 0;
    }
}

```

```

        foreach ($value as $key_2 => $densitas) {
            $data5a[$key]+=$densitas;
        }
    }

    foreach ($data5a as $key_1 => $value_1) {

        if (!empty($data5a['konflik'])) {

            $jumlahkonflik = $data5a['konflik'];
        } else {
            $jumlahkonflik = 0;
        }

        if ($key_1!=='konflik') {
            $data6a[$key_1] = $value_1 / (1-$jumlahkonflik);
        }
    }

    foreach ($data4a as $key_1 => $value_1) {

        foreach ($value_1 as $key_2 => $value_2) {

            $datapembulatan[$key_1][$key_2] = round($value_2, 3);
        }
    }

    foreach ($datapembulatan as $key_1 => $value_1) {
        if (!empty($data5a['konflik'])) {
            $jumlahkonflik = round($data5a['konflik'], 3);
        } else {
            $jumlahkonflik = 0;
        }

        if ($key_1!=='konflik') {

            $rumusjumlah = implode($value_1, " + ");
            $data7a[$key_1][0]="(".$rumusjumlah.")/(1-".$jumlahkonflik.")
";
        }
    }

    $_SESSION['kombinasi'][$index]['perkalian'] = $data2a;
    $_SESSION['kombinasi'][$index]['irisan'] = $data2a1;
    $_SESSION['kombinasi'][$index]['ubahkey1'] = $data3a;
    $_SESSION['kombinasi'][$index]['ubahkey2'] = $data4a;
    $_SESSION['kombinasi'][$index]['penjumlahan'] = $data5a;
    $_SESSION['kombinasi'][$index]['jumlahakhir'] = $data6a;
    $_SESSION['kombinasi'][$index]['rumus'] = $data7a;
}

```

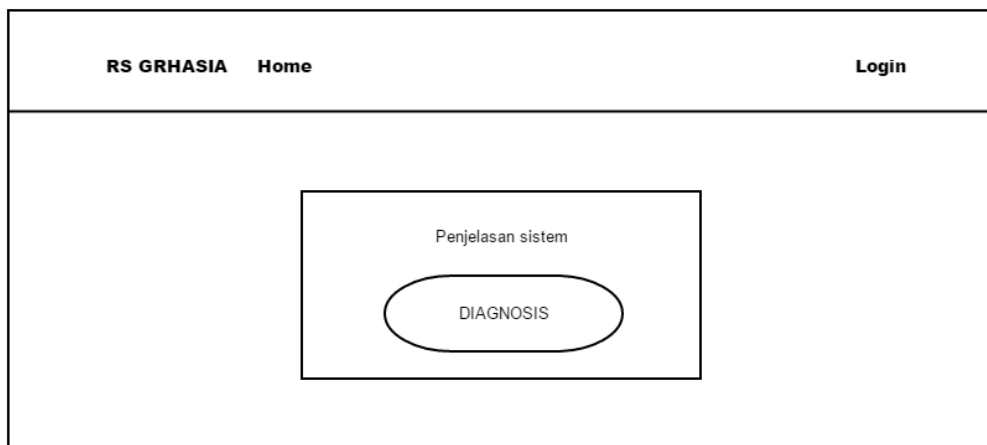
Gambar 4.7 Perancangan *Dempster Shafer* Fungsi Perhitungan_kombinasi

4.5 Perancangan Antarmuka

Perancangan antarmuka adalah perancangan yang digunakan untuk menggambarkan atau mengilustrasikan bagaimana sistem akan dibentuk secara kasar agar memudahkan dalam pembuatan sistem.

4.5.1 Perancangan Antarmuka Halaman Home

Halaman home merupakan halaman yang akan ditampilkan pertamakali apabila sistem dijalankan. Pada halaman home memiliki menu login dan juga terdapat penjelasan singkat tentang sistem. Adapun perancangan antarmuka halaman home dapat dilihat pada Gambar 4.8



Gambar 4.8 Perancangan halaman home

4.5.2 Perancangan Antarmuka Halaman Login


Halaman login merupakan halaman yang berguna untuk melakukan login terhadap sistem. Pada halaman login admin akan diminta untuk memasukan username dan password untuk dapat memasuki sistem. Adapun perancangan antarmuka halaman login dapat dilihat pada Gambar 4.9

RS GRHASIA Home	Login
<div style="border: 1px solid black; padding: 10px; margin: 20px auto; width: 300px;"><p>Login</p><div style="margin-bottom: 5px;"><input type="text" value="Username"/></div><div style="margin-bottom: 5px;"><input type="password" value="Password"/></div><div style="text-align: center;"><input type="button" value="Login"/></div></div>	

Gambar 4.9 Perancangan halaman login

4.5.3 Perancangan Antarmuka Halaman Utama Admin

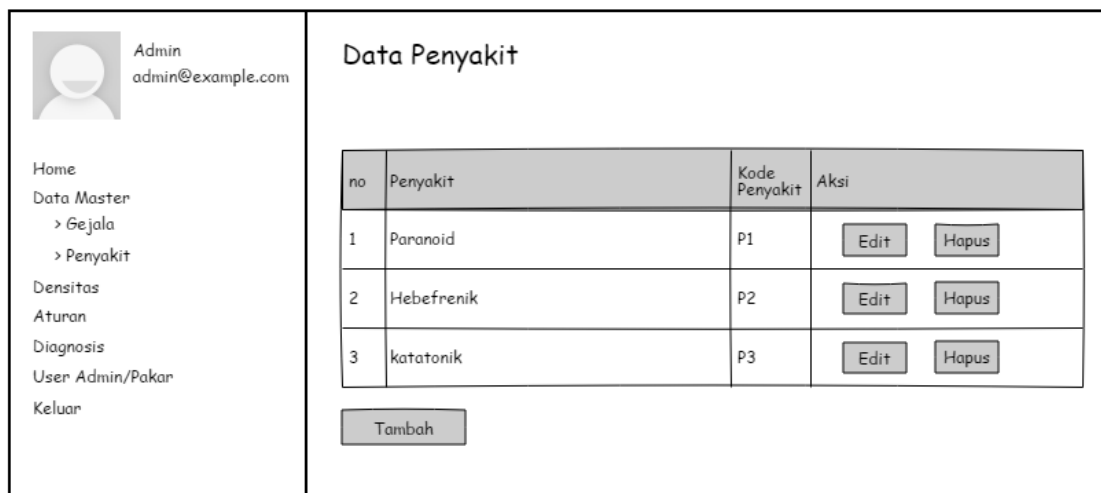
Halaman utama admin merupakan halaman yang akan pertama kali muncul ketika admin atau pakar telah berhasil login. Pada halaman utama admin terdapat beberapa menu diantaranya yaitu home, data master, densitas, aturan, diagnosis, user admin/pakar, dan keluar. Setiap menu memiliki fungsi yang berbeda. Pada halaman utama admin terdapat penjelasan penggunaan sistem. Adapun perancangan antarmuka halaman utama admin dapat dilihat pada Gambar 4.10

<div style="text-align: center;"><div>Admin admin@example.com</div></div> <div style="margin-top: 10px;"><p>Home</p><p>Data Master</p><p>Densitas</p><p>Aturan</p><p>Diagnosis</p><p>User Admin/Pakar</p><p>Keluar</p></div>	
---	--

Gambar 4.10 Perancangan Halaman utama admin

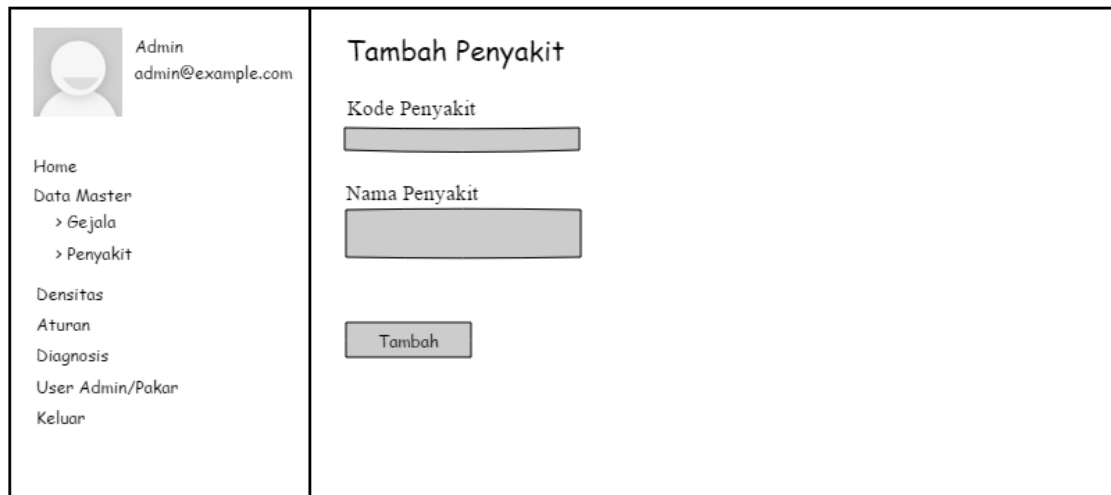
4.5.4 Perancangan Antarmuka Halaman Data Master

Halaman data master merupakan sebuah halaman yang berisikan data penyakit dan data gejala. Halaman data master akan menampilkan data yang sudah masuk ke dalam sistem dalam bentuk tabel. Tabel tersebut berisikan kolom penyakit, kolom kode penyakit, dan kolom aksi. Adapun perancangan antarmuka halaman data master dapat dilihat pada Gambar 4.11



Gambar 4.11 Perancangan Halaman Data Master

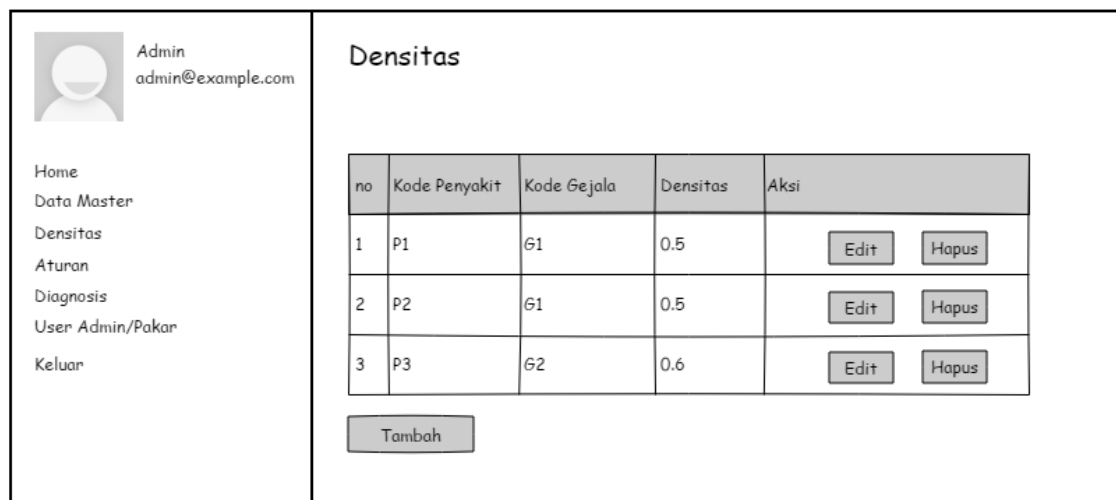
Pada setiap data master dapat melakukan penambahan, pengeditan, dan penghapusan data. Pada halaman ini sistem akan menyediakan kolom kode penyakit, dan nama penyakit untuk ditambah atau diedit kedalam sistem. Adapun perancangan antarmuka halaman penambahan dan pengeditan dapat dilihat pada Gambar 4.12



Gambar 4.12 Perancangan halaman penambahan dan pengeditan data

4.5.5 Perancangan Antarmuka Halaman Densitas

Halaman densitas merupakan sebuah halaman yang berisi nilai kepercayaan setiap gejala terhadap suatu penyakit. Halaman densitas akan menampilkan data densitas yang sudah masuk ke dalam sistem dalam bentuk tabel. Tabel tersebut memiliki kolom kode penyakit, kolom kode gejala, kolom densitas, dan kolom aksi. Adapun perancangan antarmuka halaman densitas dapat dilihat pada Gambar 4.13



no	Kode Penyakit	Kode Gejala	Densitas	Aksi
1	P1	G1	0.5	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	P2	G1	0.5	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	P3	G2	0.6	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Gambar 4.13 Perancangan halaman densitas

Pada halaman densitas dapat melakukan penambahan data densitas, pengeditan data densitas dan penghapusan data densitas. Pada halaman ini sistem akan menyediakan kolom

penyakit, kolom gejala, dan kolom densitas untuk ditambah atay diedit ke dalam sistem. Adapun perancangan penambahan dan pengeditan data densitas dapat dilihat pada Gambar 4.14

Gambar 4.14 Perancangan halaman penambahan dan pengeditan data densitas

4.5.6 Perancangan Antarmuka Halaman Aturan

Halaman aturan merupakan halamana yang berisikan aturan yang dapat dikatakan terkena skizofrenia pada setiap kemungkinan gejala yang dialami. Halaman aturan akan menampilkan data aturan yang sudah masuk ke dalam sistem. Tabel tersebut berisikan kolom nama penyakit, kolom gejala yang dialami, dan kolom aksi. Adapun perancangan antarmuka halaman aturan dapat dilihat pada Gambar 4.15

no	Kode Penyakit	Kode Gejala	Densitas	Aksi
1	P1	G1	0.5	<button>Edit</button> <button>Hapus</button>
2	P2	G1	0.5	<button>Edit</button> <button>Hapus</button>
3	P3	G2	0.6	<button>Edit</button> <button>Hapus</button>

Gambar 4.15 Perancangan halaman aturan

Pada halaman aturan dapat melakukan penambahan data, pengeditan data, dan penghapusan data. Pada halama ini sistem akan menyediakan kolom penyakit, kolom gejala, dan kolom densitas untuk ditambah atau diedit ke dalam sistem. Adapun perancangan antarmuka halaman penambahan dan pengeditan data aturan dapat dilihat pada Gambar 4.16

Gambar 4.16 Perancangan halaman penambahan dan pengeditan data aturan

4.5.7 Perancangan Antarmuka Halaman Diagnosis

Halaman diagnosis merupakan halaman untuk melakukan diagnosis yang berisikan gejala-gejala skizofrenia. Halaman diagnosis akan menampilkan data gejala dalam bentuk tabel. Tabel tersebut memiliki kolom kode gejala, kolom nama gejala, dan kolom pilih. Adapun perancangan antarmuka halaman diagnosis dapat dilihat pada Gambar 4.17

no	Kode Gejala	Nama Gejala	Pilih
1	G1	Sering Halusinasi	<input type="checkbox"/>
2	G1	Suka Berhayal	<input type="checkbox"/>
3	G2	Suka menyendiri	<input type="checkbox"/>

Gambar 4.17 Perancangan halaman diagnosis

Pada halaman diagnosis, setelah pakar memilih semua gejala yang dialami dan menekan tombol 'selesai' maka sistem akan menampilkan hasil diagnosis. Hasil diagnosis berisikan data gejala yang dipilih dan menampilkan penyakit skizofrenia serta tingkat kepercayaan terhadap penyakit skizofrenia dalam bentuk persen. Adapun perancangan antarmuka halaman hasil diagnosis dapat dilihat pada Gambar 4.18

Gambar 4.18 Perancangan halaman hasil diagnosis

4.5.8 Perancangan Antarmuka Halaman User Admin/Pakar

Halaman user admin/pakar merupakan halaman yang digunakan untuk mengubah nama pengguna, username, dan password baru. Halaman user admin/pakar akan menampilkan kolom nama, kolom username, dan kolom password untuk mengubah data user admin/pakar. Adapun perancangan antarmuka halaman user admin/pakar dapat dilihat pada Gambar 4.19

Gambar 4.19 Perancangan Halaman User Admin/Pakar