



### **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 SISTEM

#### 3.1 Metode Penelitian

Langkah-langkah yang dilakukan dalam pembangunan sistem pakar dalam penelitian ini. Penjelasan adalah sebagai berikut.

##### 1. Studi Literatur

Studi literatur dilakukan dengan cara mencari informasi secara mandiri mengenai teori dasar yang digunakan dalam penelitian berupa pembangunan suatu sistem pakar untuk mendiagnosa penyakit jantung berbasis *website*. Pencarian informasi dilakukan melalui sumber buku, jurnal, majalah, dan juga internet. Hal ini dilakukan untuk menambah wawasan dan pengetahuan mengenai topik penelitian agar hasil penelitian bersifat *valid* dan benar adanya, serta untuk mempermudah dalam proses pembangunan sistem pakar.

##### 2. Pengumpulan Data

Data yang dibutuhkan dalam pembangunan program ini adalah data mengenai gejala yang dapat dilihat dan dirasakan secara fisik. Data lainnya adalah data riwayat penyakit keturunan, riwayat penyakit dahulu, dan wawancara langsung dari dokter spesialis jantung dan pembuluh darah.

##### 3. Analisa Model Sistem

Pada tahap ini, dilakukan penentuan gejala dan kebiasaan yang digunakan dalam sistem. Penentuan gejala dan kebiasaan tersebut juga telah dikonsultasikan dengan dokter spesialis jantung dan pembuluh darah.

Selanjutnya, dilakukan analisis model sistem secara terstruktur, yaitu dengan mempertimbangkan keterkaitan data dengan proses yang dapat mengubah data sebagai *entities* terpisah

#### 4. Perancangan Sistem

Perancangan sistem dimulai dengan merancang antarmuka *website*. Kemudian dilanjutkan dengan perancangan *database* dan perancangan alur sistem. Perancangan sistem dilakukan dengan mengimplementasikan seluruh kebutuhan yang terdapat pada proses analisis model, sehingga seluruh kebutuhan sistem dapat terpenuhi.

#### 5. Pembuatan Program

Program dibangun dengan bahasa pemrograman PHP dan *database* MySQL. Program yang dibangun merupakan program berbasis *web* yang dapat diakses melalui semua perangkat yang terhubung dengan internet. *Website* dibangun berdasarkan rancangan antarmuka yang telah dibuat pada tahap perancangan sistem. Pada tahap ini, rancangan yang telah dibuat sebelumnya divisualisasikan menggunakan dokumen HTML berserta dengan dokumen CSS yang digunakan untuk membangun tampilan website yang menarik. Kemudian dilanjutkan dengan proses penerapan metode *dempster shafer* pada program.

#### 6. Pengujian

Pada tahap ini dilakukan pengujian program yang telah dibuat. Pengujian dilakukan mulai dari awal sampai akhir program. Hal ini dilakukan untuk memeriksa apakah ada tampilan atau proses dalam program yang masih mengalami *error*. Selain itu, Pengujian juga dilakukan untuk melihat apakah

program sudah berjalan sesuai dengan apa yang dirancang dan juga mencari kekurangan dari program agar dapat dikembangkan dan direvisi lebih lanjut.

#### 7. Penarikan Simpulan

Penarikan simpulan merupakan tahapan terakhir dari penelitian. Pada tahap ini dilakukan proses penarikan kesimpulan dan pemberian saran dari sistem yang telah dibuat. Kemudian dilanjutkan dengan proses penulisan laporan mengenai sistem yang telah dibuat.

### 3.2 Analisis Perancangan

Dalam membangun sistem pakar diagnosa penyakit jantung, diperlukan suatu analisis dan perancangan terlebih dahulu. Diagram yang dibuat dalam analisis dan perancangan sistem adalah Data Flow Diagram (DFD), *Entity Relation Diagram* (ERD), *flowchart*, struktur tabel, dan desain antarmuka (*interface*).

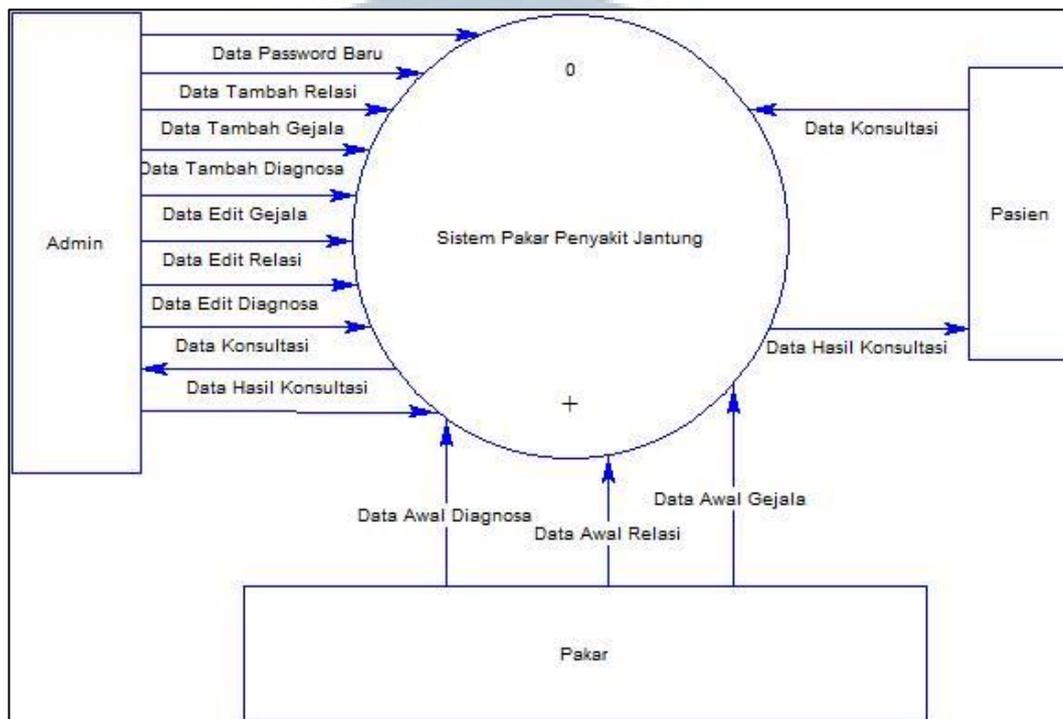
#### 3.2.1 Data Flow Diagram

*Data Flow Diagram* (DFD) merupakan suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur, dan jelas.

Terdapat 2 level di dalam suatu DFD yaitu diagram level 0 dan diagram level 1.

##### A. Diagram Level Nol

Diagram konteks pada sistem digambarkan pada Gambar 3.1.

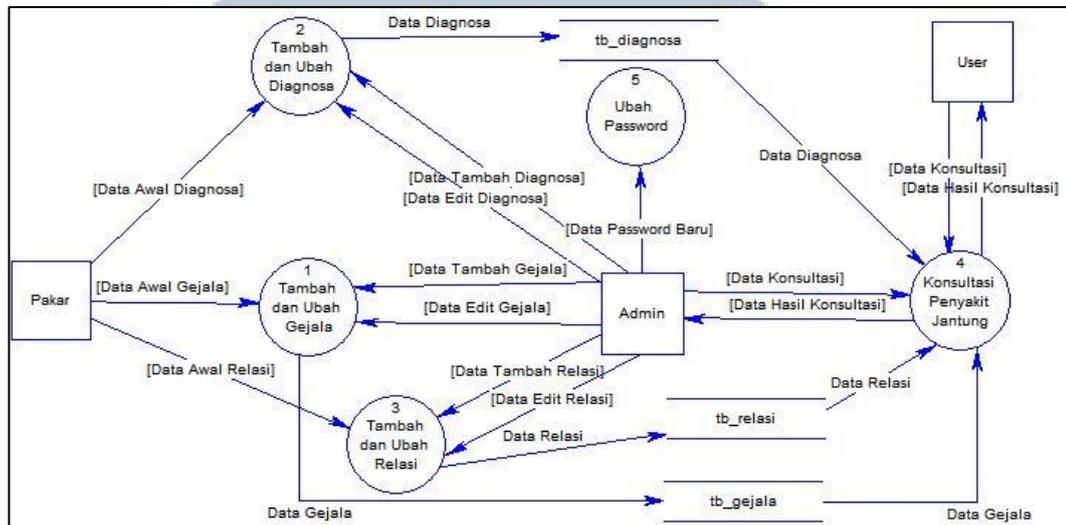


Gambar 3.1 Diagram level 0

Terdapat tiga entitas eksternal dalam sistem ini, yaitu *admin*, pakar dan *user*. *Admin* dapat berinteraksi dengan memberikan data password baru, data tambah relasi, data tambah gejala, data tambah diagnosa, data edit gejala data edit relasi, data edit diagnosa, dan data konsultasi. Sebaliknya sistem pakar penyakit jantung memeberikan data hasil konsultasi kepada *admin*. Pakar berinteraksi dengan memberikan data gejala, data penyakit, dan data aturan kepada sistem. *User* dapat berinteraksi dengan sistem dengan memberikan data konsultasi dan sistem pakar penyakit jantung akan memberikan data hasil konsultasi. Setiap *input* dan *Output* yang terjadi pada diagram konteks harus digambarkan kembali pada diagram nol.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## B. Diagram Level Satu

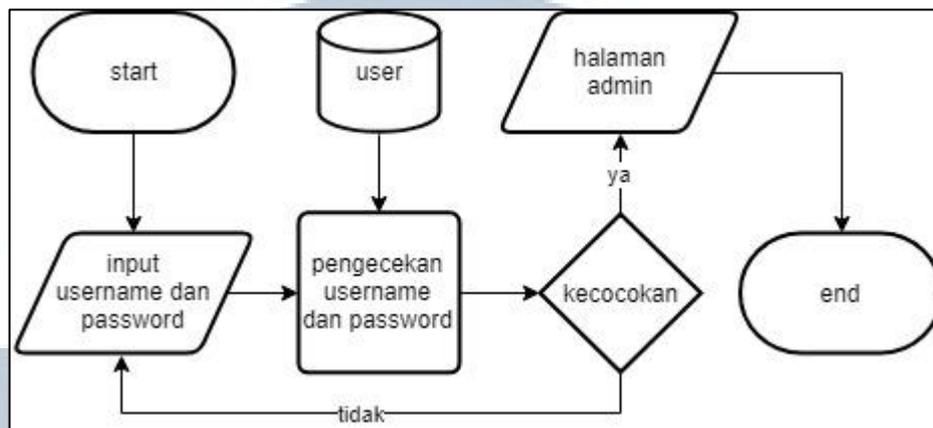


Gambar 3.2 Diagram level 1

Diagram level 1 Merupakan perincian dari diagram konteks. Pada diagram ini sistem terbagi menjadi 5 proses utama yaitu 'tambah dan ubah gejala', 'tambah dan ubah diagnosa', 'tambah dan ubah relasi', 'konsultasi penyakit jantung', dan 'ubah password'. Terdapat pula *data store* yang terhubung dengan salah satu proses yang ada. Entitas *admin*, *pakar*, dan *user* memiliki hubungan dengan satu atau lebih proses dari semua proses yang ada.

### 3.2.2 Flowchart

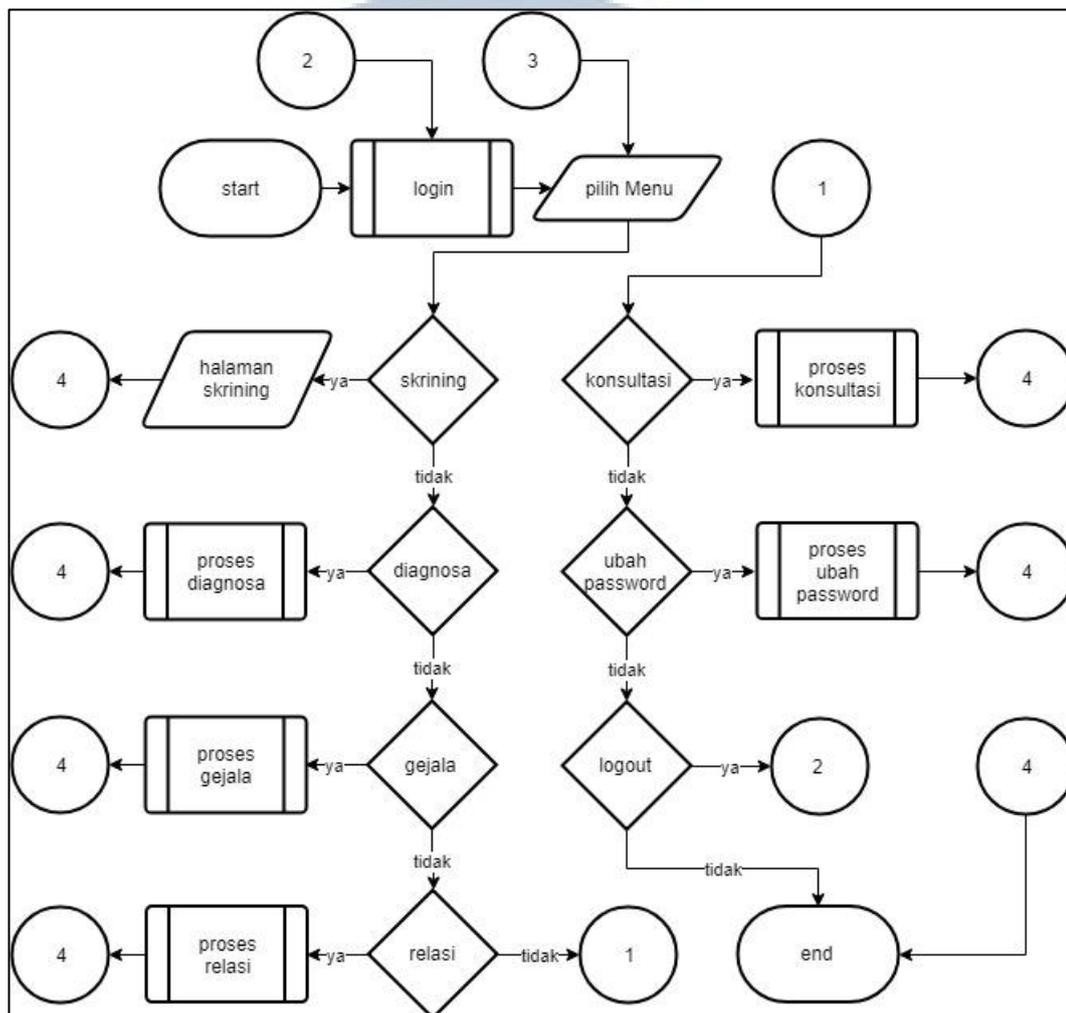
*Flowchart* merupakan diagram yang menggambarkan hubungan antar proses secara urut dan mendetail yang terjadi pada suatu sistem. Dalam menyusun program dari sistem. *Flowchart* sistem dalam melakukan login *admin* digambarkan pada Gambar 3.3.



Gambar 3.3 Flowchart melakukan *login* admin

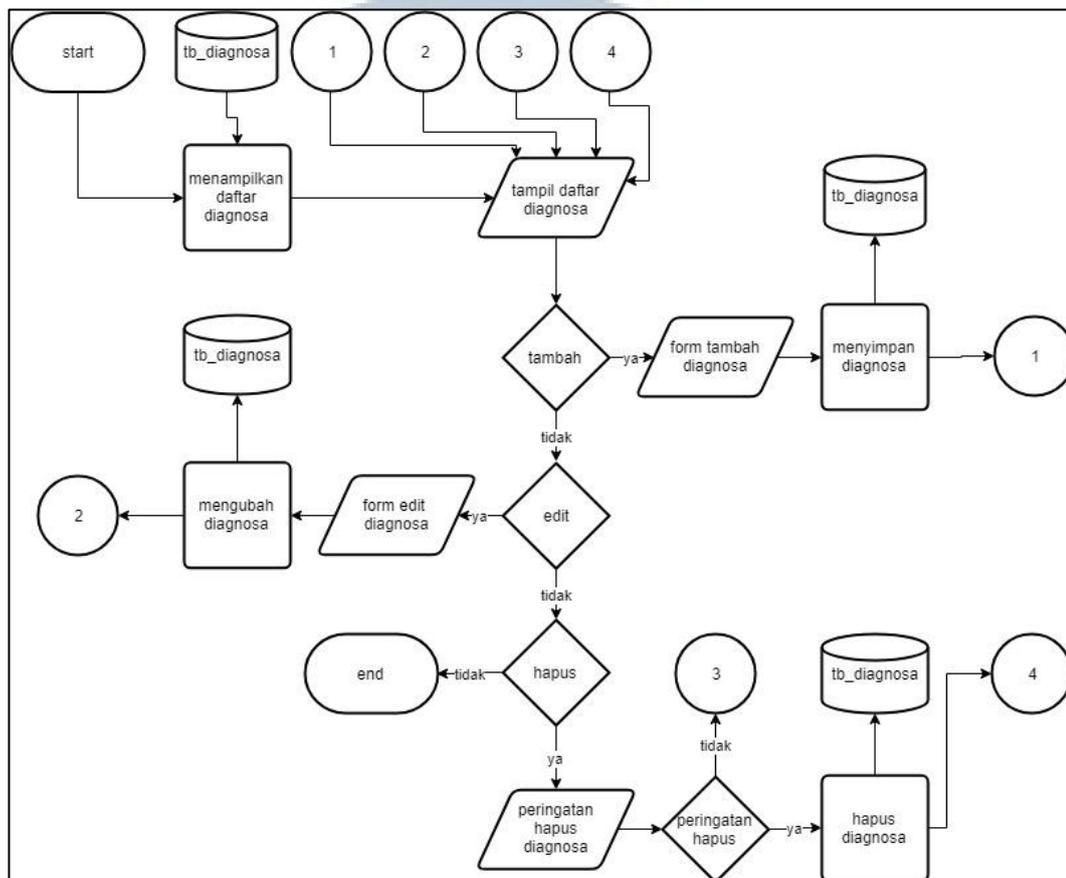
Proses *admin* dilakukan untuk mengolah data-data sistem pakar seperti jenis diagnosa, gejala, dan relasi. Sebelum melakukan pengolahan data sistem pakar penyakit jantung terlebih dahulu harus melakukan *login* sebagai *admin*.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.4 Flowchart aktivitas menu admin

Halaman *admin* mempunyai *menu* skrining, diagnosa, *menu* gejala, dan *menu* relasi yang masing-masing dapat melakukan tambah, edit, dan hapus data yang berguna untuk melakukan konsultasi, *menu* konsultasi berfungsi untuk melakukan konsultasi secara *self service*, *menu* ubah *password* berfungsi untuk mengganti *password admin*, dan *menu* *logout* untuk menutup *menu admin*.



Gambar 3.5 Flowchart diagnosa admin

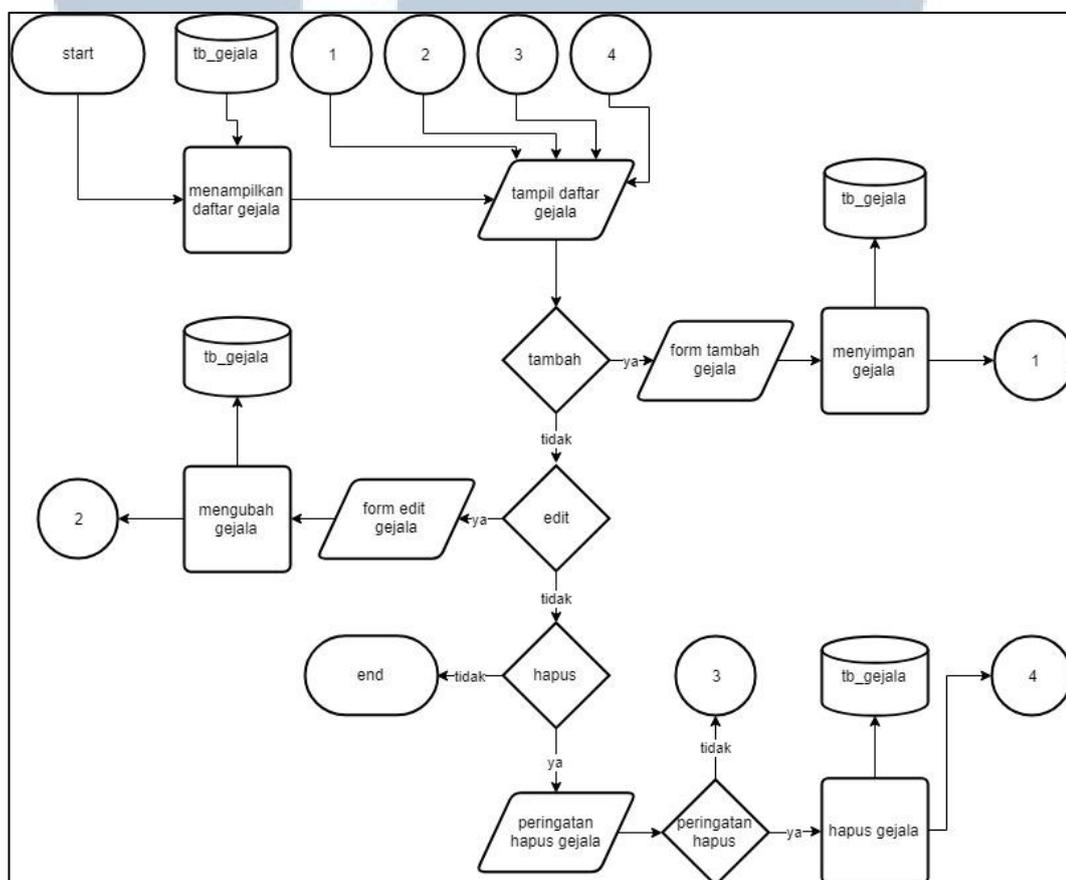
Diagnosa *admin* menampilkan daftar-daftar diagnosa, di daftar diagnosa memiliki tambah diagnosa, edit diagnosa dan hapus diagnosa yang masing masing memiliki fungsinya sendiri.

Tambah diagnosa di halaman *admin* melakukan aktifitas tambah diagnosa yang berfungsi untuk menambahkan jenis penyakit jantung, Isi form diagnosa untuk mengisi form diagnosa yang ingin di buat, data diagnosa untuk menyimpan data yang sudah di input dari form diagnosa ke database, daftar tabel diagnosa untuk menampilkan list tabel dari database tabel *tb\_diagnosa*.

Edit diagnosa di halaman *admin* melakukan aktifitas edit diagnosa yang berfungsi untuk mengubah nama jenis penyakit jantung, edit form diagnosa untuk

mengubah diagnosa yang ada di *database*, data diagnosa untuk menyimpan data yang sudah diubah dari edit form diagnosa ke *database*, daftar tabel diagnosa untuk menampilkan list tabel dari database tabel *tb\_diagnosa* yang sudah diubah.

Hapus diagnosa di halaman *admin* melakukan aktifitas menghapus data yang berada di *database* bagian tabel *tb\_diagnosa*, daftar diagnosa untuk menampilkan daftar dari *database* bagian tabel *tb\_diagnosa*, hapus data mempunyai peringatan setiap ingin melakukan penghapusan data.



Gambar 3.6 Flowchart gejala admin

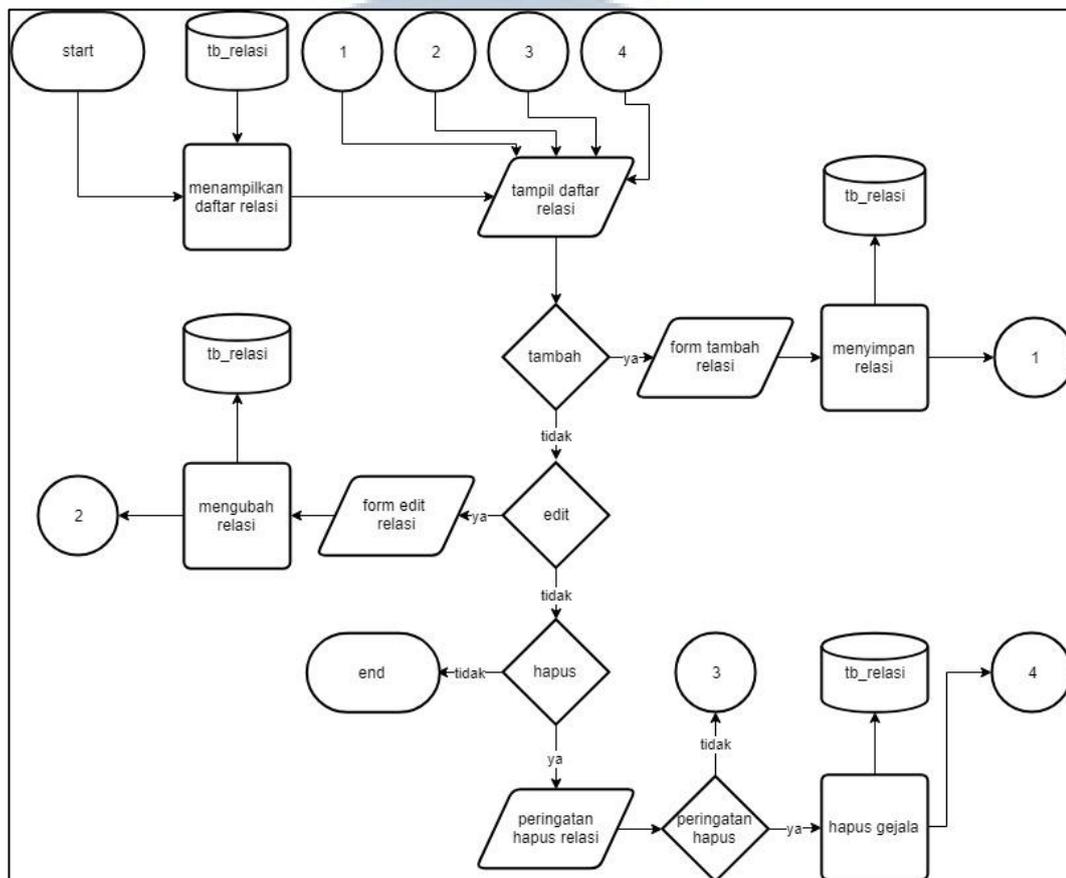
Gejala *admin* menampilkan daftar-daftar gejala. di daftar gejala memiliki tambah gejala, edit gejala dan hapus gejala yang masing masing memiliki fungsinya sendiri.

Tambah gejala di halaman *admin* melakukan aktifitas tambah gejala yang berfungsi untuk menambahkan gejala-gejala yang berhubungan dengan penyakit jantung, form gejala untuk mengisi gejala yang ingin di buat, data gejala untuk menyimpan data yang sudah di input dari form gejala ke database, daftar tabel gejala untuk menampilkan list tabel dari database tabel *tb\_gejala*.

Edit gejala di halaman *admin* melakukan aktifitas edit gejala yang berfungsi untuk mengubah gejala penyakit jantung, edit form gejala untuk mengubah gejala yang ada di *database*, data gejala untuk menyimpan data yang sudah diubah dari edit form diagnosa ke *database*, daftar tabel gejala untuk menampilkan list tabel dari database tabel *tb\_gejala* yang sudah diubah.

Hapus gejala di halaman *admin* melakukan aktifitas menghapus data yang berada di *database* bagian tabel *tb\_gejala*, daftar diagnosa untuk menampilkan daftar dari *database* bagian tabel *tb\_gejala*, hapus data mempunyai peringatan setiap ingin melakukan penghapusan data.





Gambar 3.7 Flowchart Relasi admin

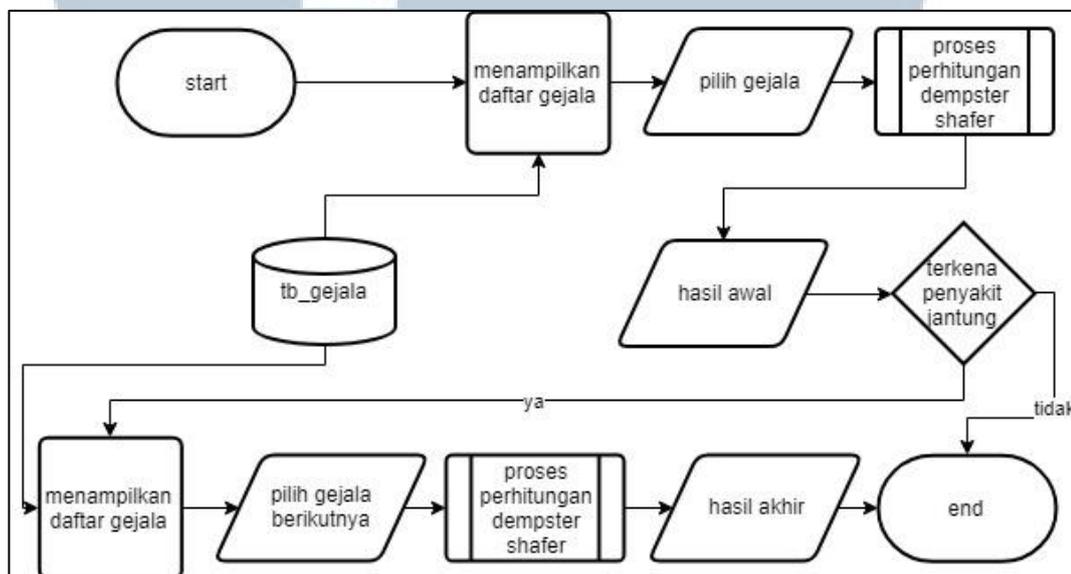
Relasi *admin* menampilkan daftar-daftar relasi. di daftar relasi memiliki tambah relasi, edit relasi dan hapus relasi yang masing masing memiliki fungsinya sendiri.

Tambah relasi di halaman *admin* melakukan aktifitas tambah relasi yang berfungsi untuk menambahkan hubungan diagnosa dan gejala, form relasi untuk memilih dari daftar diagnosa dan daftar gejala, data relasi untuk menyimpan data yang sudah dimasukkan dari form relasi ke database, daftar tabel relasi untuk menampilkan daftar tabel dari database tabel *tb\_relasi*

Edit relasi di halaman *admin* melakukan aktifitas edit relasi yang berfungsi untuk mengubah relasi diagnosa dan gejala, edit form relasi untuk mengubah relasi

yang ada di *database*, data relasi untuk menyimpan data yang sudah diubah dari edit form diagnosa ke *database*, daftar tabel relasi untuk menampilkan list tabel dari *database* tabel *tb\_relasi* yang sudah diubah.

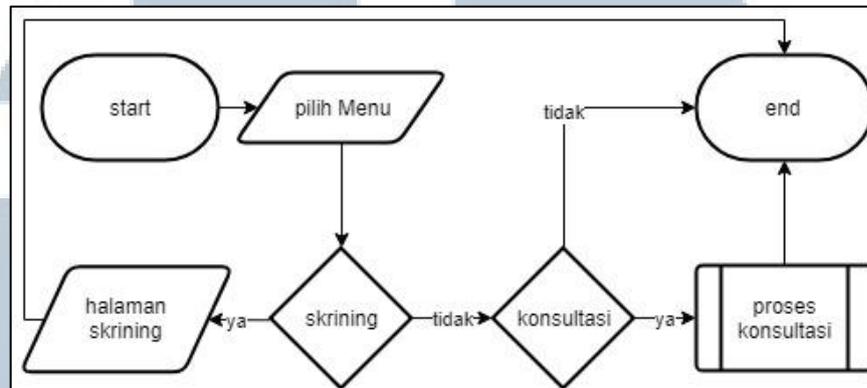
Hapus relasi di halaman *admin* melakukan aktifitas menghapus data yang berada di *database* bagian tabel *tb\_relasi*, daftar diagnosa untuk menampilkan daftar dari *database* bagian tabel *tb\_relasi*, hapus data mempunyai peringatan setiap ingin melakukan penghapusan data.



Gambar 3.8 Flowchart konsultasi admin

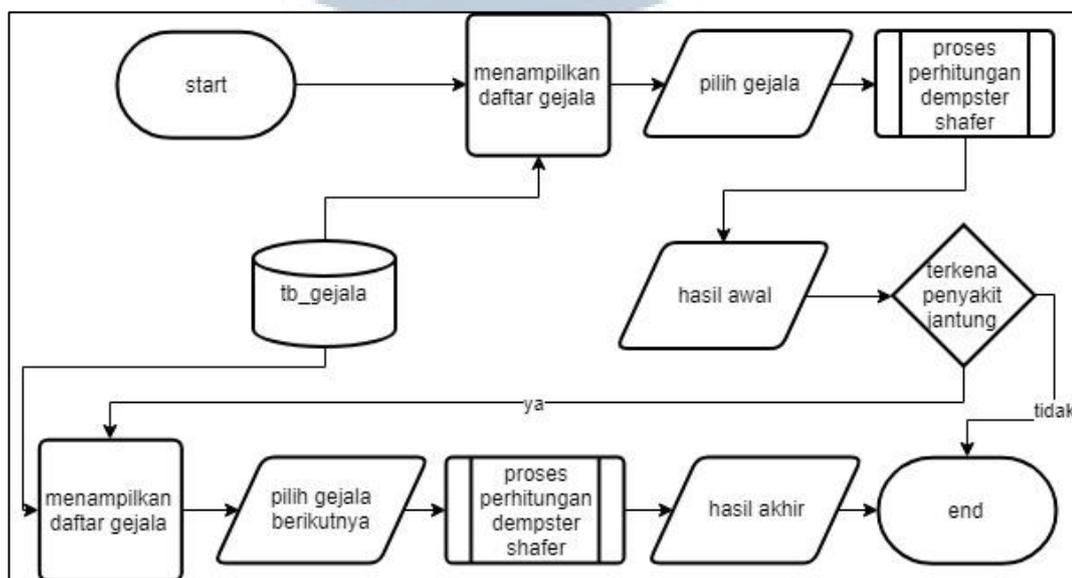
Konsultasi di halaman *admin* melakukan aktifitas mendiagnosa penyakit jantung, form konsultasi tahap awal untuk memilih gejala yang diderita pasien, hasil konsultasi tahap awal berguna untuk menghitung hasil dari gejala yang dipilih menggunakan algoritma *dempster shafer* dan jika terkena penyakit jantung maka sistem akan melanjutkan ke konsultasi tahap selanjutnya, jika tidak terkena penyakit jantung maka konsultasi selesai, form konsultasi tahap akhir untuk

memilih gejala-gejala lebih lanjut, hasil konsultasi tahap akhir berguna untuk menghitung hasil dari gejala yang dipilih menggunakan algoritma *dempster shafer*.



Gambar 3.9 Flowchart aktivitas halaman asisten dokter

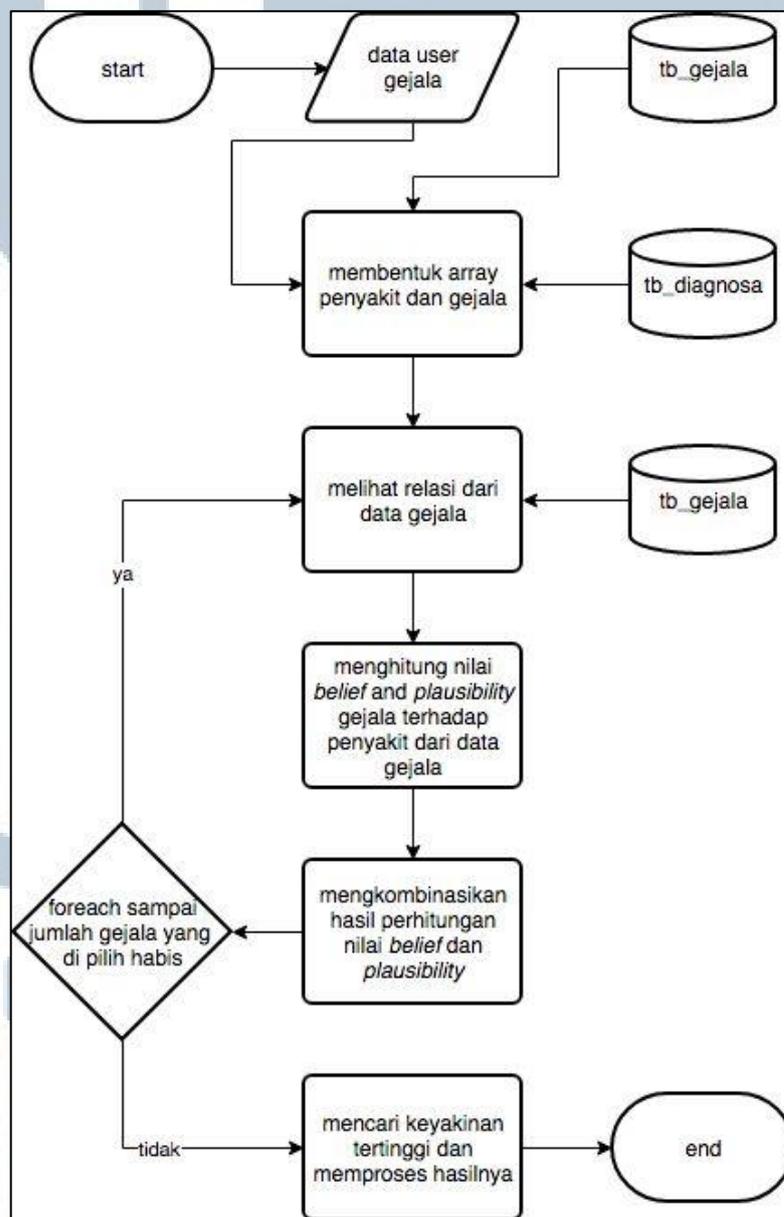
Halaman *admin* mempunyai *menu* skrining dan konsultasi, *menu* skrining untuk menjelaskan tentang penyakit jantung dan pembuluh darah secara singkat, *menu* konsultasi berfungsi untuk melakukan konsultasi secara *self service*.



Gambar 3.10 Flowchart aktivitas konsultasi asisten dokter

Konsultasi di halaman asisten dokter melakukan aktifitas mendiagnosa penyakit jantung. form konsultasi tahap awal untuk memilih gejala yang diderita pasien. hasil konsultasi tahap awal berguna untuk menghitung hasil dari gejala yang

dipilih menggunakan algoritma *dempster shafer* dan jika terkena penyakit jantung, maka sistem akan meminta untuk melakukan konsultasi tahap selanjutnya, selain itu jika tidak terkena penyakit jantung, maka konsultasi selesai. formulir konsultasi tahap akhir untuk memilih gejala-gejala lebih lanjut. hasil konsultasi tahap akhir berguna untuk menghitung hasil dari gejala yang dipilih menggunakan algoritma *dempster shafer*.



Gambar 3.11 Flowchart algoritma dempster shafer

Algoritma *dempster shafer* dimulai dengan data gejala diterima lalu membuat array untuk menampung data penyakit dan gejala dari database. Kemudian melihat relasi dari gejala user dan menghitung nilai *belief* dan *plausibility* dan diulang hingga semua gejala telah dikombinasikan. Kemudian mengambil nilai yang tertinggi.

### 3.2.3 Struktur Tabel

Terdapat 8 tabel, keempat tabel yang digunakan dalam pembuatan sistem ini adalah sebagai berikut.

Nama Tabel : *tb\_admin*.

Fungsi : menyimpan *user* dan *password* admin.

*Primary Key* : tidak ada.

*Foreign Key* : tidak ada.

Struktur dari tabel admin dapat dilihat pada Tabel 3.1.

Tabel 3.1 Struktur Tabel admin

Nama Kolom	Tipe	Panjang	Keterangan
user	varchar	16	Not Null
pass	varchar	16	Not Null
level	varchar	16	Not Null

Nama Tabel : *tb\_diagnosa*.

Fungsi : menyimpan data jenis penyakit jantung.

*Primary Key* : tidak ada.

*Foreign Key* : tidak ada.

Struktur dari tabel diagnosa dapat dilihat pada Tabel 3.2.

Tabel 3.2 Struktur Tabel tb\_diagnosa

Nama Kolom	Tipe	Panjang	Keterangan
Kode_diagnosa	varchar	16	Not null
Nama_diagnosa	varchar	256	null
keterangan	text		null

Nama Tabel : tb\_gejala.

Fungsi : menyimpan data gejala penyakit.

*Primary Key* : tidak ada.

*Foreign Key* : tidak ada.

Struktur dari tabel gejala dapat dilihat pada Tabel 3.3.

Tabel 3.3 Struktur Tabel tb\_gejala

Nama Kolom	Tipe	Panjang	Keterangan
Kode_gejala	varchar	16	Not null
Nama_gejala	varchar	255	Not null
bobot	double		Not null

Nama Tabel : tb\_relasi.

Fungsi : menyimpan relasi dari tb\_diagnosa dan tb\_gejala.

*Primary Key* : ID.

*Foreign Key* : tidak ada.

Struktur dari tabel relasi dapat dilihat pada Tabel 3.4.

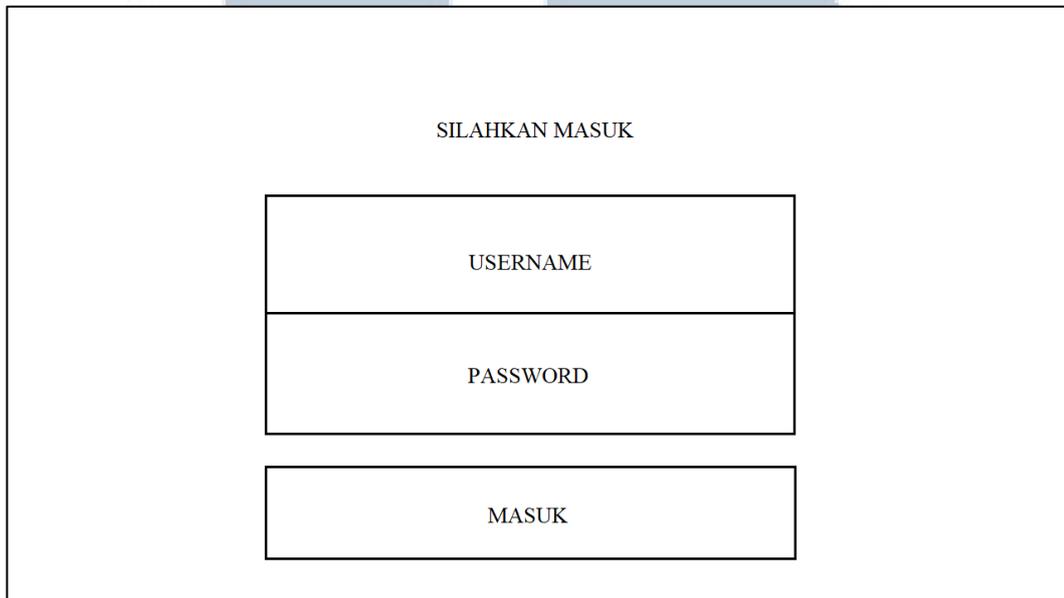
Tabel 3.4 Struktur Tabel tb\_relasi

Nama Kolom	Tipe	Panjang	Keterangan
ID	int	11	Not null dan auto increment
Kode_diagnosa	varchar	16	Not null
Kode_gejala	varchar	16	Not null

### 3.2.4 Perancangan Antarmuka Website

Sistem pakar dalam penelitian ini dibuat dalam bentuk *website*, sehingga dapat diakses melalui semua perangkat yang dapat terhubung langsung dengan

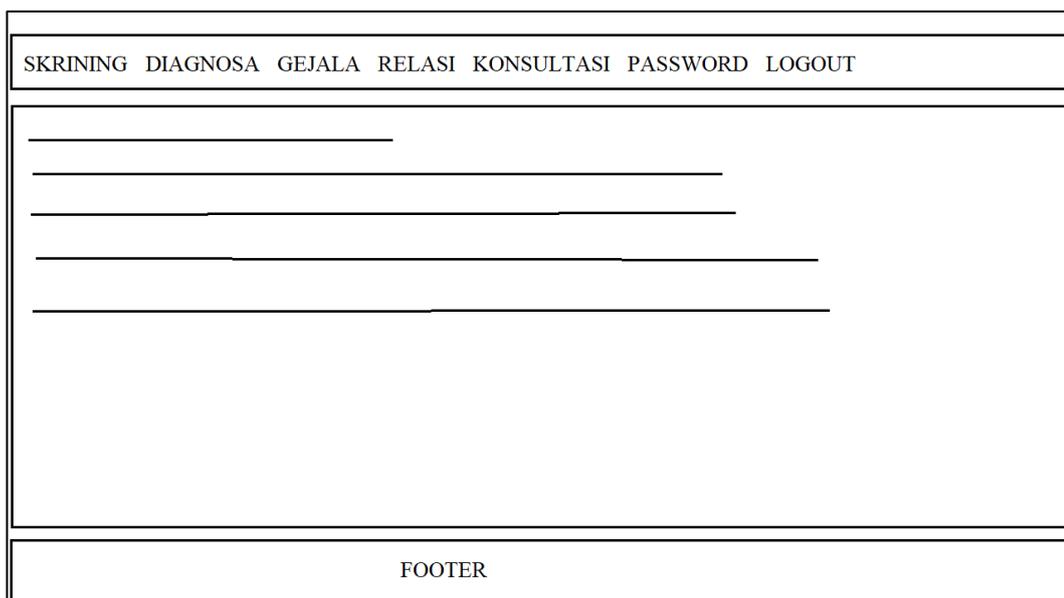
internet. Seluruh rancangan *website* ini dirancang agar dapat tampil *responsive* pada segala resolusi layar menggunakan *framework bootstrap*.



The image shows a wireframe for an admin login page. It consists of a central rectangular box containing the following elements from top to bottom: the text 'SILAHKAN MASUK', a text input field labeled 'USERNAME', a text input field labeled 'PASSWORD', and a button labeled 'MASUK'.

Gambar 3.12 Rancangan halaman login *admin*

Dimulai dari halaman login *admin* dengan memasukan *username* dan *password* untuk masuk ke halaman utama *admin*.



The image shows a wireframe for an admin dashboard page. It features a top navigation bar with the following menu items: 'SKRINING', 'DIAGNOSA', 'GEJALA', 'RELASI', 'KONSULTASI', 'PASSWORD', and 'LOGOUT'. Below the navigation bar is a large content area with five horizontal lines representing text or data. At the bottom of the page is a footer section labeled 'FOOTER'.

Gambar 3.13 Rancangan halaman *admin*

Halaman *admin* memiliki menu yang diletakan di *header*. Untuk *footer* berisikan informasi tentang pengembang. Untuk konten awal halaman adalah isi dari menu skrining. Skrining adalah informasi tentang penggunaan tes atau metode diagnosis untuk mengetahui apakah seseorang memiliki penyakit atau kondisi tertentu sebelum menyebabkan gejala apapun.

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT				
DIAGNOSA				
<input type="text" value="pencarian..."/> <input type="button" value="refresh"/> <input type="button" value="tambah"/>				
no	kode	nama diagnosa	keterangan	aksi
—	—	—	—	<input type="button" value="edit"/> <input type="button" value="hapus"/>
—	—	—	—	<input type="button" value="edit"/> <input type="button" value="hapus"/>
—	—	—	—	<input type="button" value="edit"/> <input type="button" value="hapus"/>
—	—	—	—	<input type="button" value="edit"/> <input type="button" value="hapus"/>
FOOTER				

Gambar 3.14 Rancangan menu diagnosa *admin*

Menu diagnosa terdapat di halaman *admin*. Berisikan sebuah data diagnosa penyakit yang ditampilkan dalam bentuk tabel *list*. Memiliki fitur pencarian, *refresh*, tambah, edit dan hapus data.

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

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT
<p>TAMBAH DIAGNOSA</p> <p>kode</p> <input type="text"/> <p>nama diagnosa</p> <input type="text"/> <p>keterangan</p> <input type="text"/> <p> <input type="button" value="simpan"/> <input type="button" value="kembali"/> </p>
FOOTER

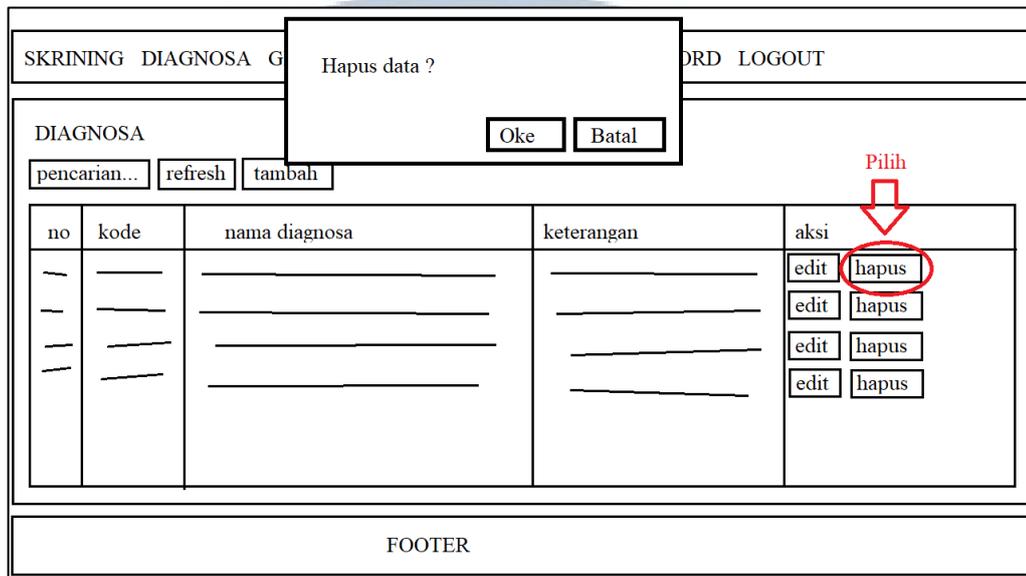
Gambar 3.15 Rancangan tambah diagnosa *admin*

Tambah diagnosa *admin* menampilkan tampilan formulir tambah diagnosa untuk memasukan data kode, nama diagnosa dan keterangan ke dalam database.

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT
<p>UBAH DIAGNOSA</p> <p>kode</p> <input type="text" value="A"/> <p>nama diagnosa</p> <input type="text" value="penyakit jantung"/> <p>keterangan</p> <input type="text" value="penyakit jantung"/> <p> <input type="button" value="simpan"/> <input type="button" value="kembali"/> </p>
FOOTER

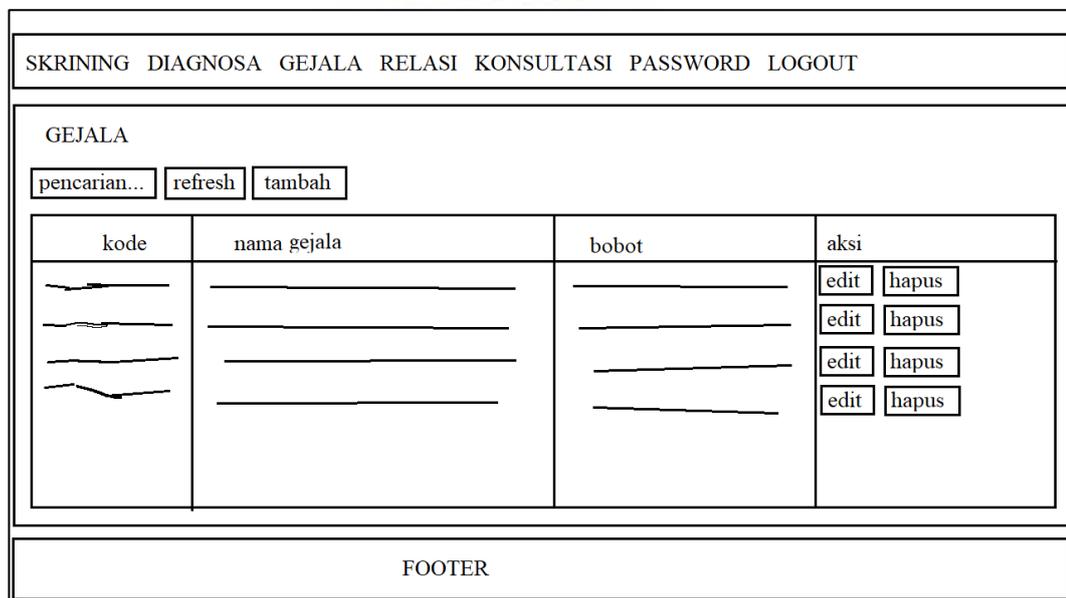
Gambar 3.16 Rancangan ubah diagnosa *admin*

Ubah diagnosa *admin* menampilkan formulir untuk mengubah diagnosa penyakit dengan mengubah isi formulir nama diagnosa dan keterangan. Formulir bagian kode tidak boleh diubah.



Gambar 3.17 Rancangan hapus diagnosa *admin*

Tombol hapus diagnosa *admin* terdapat di kolom aksi, jika tombol hapus dipilih akan menampilkan peringatan yang bertuliskan “Hapus data?” dan harus memilih antara tombol oke atau batal.



Gambar 3.18 Rancangan menu gejala *admin*

MULTIMEDIA  
NUSANTARA

Menu gejala terdapat di halaman *admin*. Berisikan data-data seluruh gejala penyakit yang ditampilkan dalam bentuk tabel *list*. Memiliki fitur pencarian, *refresh*, tambah, edit dan hapus data.

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT
<b>TABAH GEJALA</b> kode <input type="text"/> nama gejala <input type="text"/> bobot <input type="text"/> <input type="button" value="simpan"/> <input type="button" value="kembali"/>
FOOTER

Gambar 3.19 Rancangan tambah gejala *admin*

Tambah gejala *admin* menampilkan formulir kode, nama gejala, dan bobot yang dimasukan dan disimpan ke *database*.

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT
<b>UBAH GEJALA</b> kode <input type="text"/> Nama Gejala <input type="text" value="usia &gt; 40 tahun"/> <input type="text" value="0.2"/> <input type="button" value="simpan"/> <input type="button" value="kembali"/>
FOOTER

Gambar 3.20 Rancangan ubah gejala *admin*

Ubah gejala *admin* menampilkan formulir untuk mengubah nama gejala dan bobot untuk diupdate ke *database*. Untuk formulir kode tidak bisa diubah.

The screenshot shows a web interface for 'GEJALA' (Symptoms) management. At the top, there is a navigation bar with 'SKRINING DIAGNOSA GEJALA' and 'RD LOGOUT'. Below this is a section titled 'GEJALA' containing search and management buttons: 'pencarian...', 'refresh', and 'tambah'. A modal dialog box titled 'Hapus data?' is open, with 'Oke' and 'Batal' buttons. The main content is a table with columns: 'kode', 'nama gejala', 'bobot', and 'aksi'. The 'aksi' column contains 'edit' and 'hapus' buttons for each row. A 'FOOTER' section is at the bottom.

kode	nama gejala	bobot	aksi
_____	_____	_____	edit hapus
_____	_____	_____	edit hapus
_____	_____	_____	edit hapus
_____	_____	_____	edit hapus

Gambar 3.21 Rancangan hapus gejala *admin*

Hapus gejala *admin* menampilkan sebuah peringatan “hapus data?” jika tombol yang berada di tabel bagian kolom aksi dipilih.

The screenshot shows a web interface for 'RELASI' (Relationships) management. At the top, there is a navigation bar with 'SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT'. Below this is a section titled 'RELASI' containing search and management buttons: 'pencarian...', 'refresh', and 'tambah'. The main content is a table with columns: 'no', 'diagnosa', 'gejala', 'bobot', and 'aksi'. The 'aksi' column contains 'edit' and 'hapus' buttons for each row. A 'FOOTER' section is at the bottom.

no	diagnosa	gejala	bobot	aksi
_____	_____	_____	_____	edit hapus
_____	_____	_____	_____	edit hapus
_____	_____	_____	_____	edit hapus
_____	_____	_____	_____	edit hapus

Gambar 3.22 Rancangan menu relasi *admin*.

Menu relasi terdapat di halaman *admin*. Berisikan data diagnosa dan gejala yang sudah direlasikan. Ditampilkan dalam bentuk tabel *list*. Memiliki fitur pencarian, *refresh*, tambah, edit dan hapus data.

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT
<p>TAMBAH RELASI</p> <p>diagnosa</p> <input type="text"/>
<p>gejala</p> <input type="text"/>
<input type="button" value="simpan"/> <input type="button" value="kembali"/>
FOOTER

Gambar 3.23 Rancangan tambah relasi admin

Tambah relasi menampilkan dua *combo box* yang harus diisi dengan cara memilih dari *list combo box* diagnosa dan *list combo box* gejala.

SKRINING DIAGNOSA GEJALA RELASI KONSULTASI PASSWORD LOGOUT
<p>UBAH RELASI</p> <p>diagnosa</p> <input type="text" value="[A] penyakit jantung"/>
<p>gejala</p> <input type="text" value="A001] usia &gt; 40 tahun"/>
<input type="button" value="simpan"/> <input type="button" value="kembali"/>
FOOTER

Gambar 3.24 Rancangan ubah relasi *admin*

Ubah relasi *admin* menampilkan 2 *combo box* diagnosa dan gejala yang isi dari kedua *combo box* sudah dipilih sebelumnya dan dapat diubah lagi.

The screenshot shows a web application interface with a navigation bar containing 'SKRINING', 'DIAGNOSA', 'GEJALA', and 'LOGOUT'. Below the navigation bar is a section titled 'RELASI'. This section includes a search bar labeled 'pencarian...', a 'refresh' button, and a 'tambah' button. The main content is a table with the following structure:

no	diagnosa	gejala	bobot	aksi
---	---	---	---	edit hapus
---	---	---	---	edit hapus
---	---	---	---	edit hapus
---	---	---	---	edit hapus

Below the table is a 'FOOTER' section. A modal dialog box titled 'Hapus data?' is overlaid on the table, containing 'Oke' and 'Batal' buttons.

Gambar 3.25 Rancangan hapus relasi admin

Hapus relasi *admin* menampilkan peringatan “Hapus data?” jika memilih tombol hapus yang terdapat di tabel di bagian kolom aksi.

The screenshot shows a web application interface with a navigation bar containing 'SKRINING', 'DIAGNOSA', 'GEJALA', 'RELASI', 'KONSULTASI', 'PASSWORD', and 'LOGOUT'. Below the navigation bar is a section titled 'KONSULTASI'. This section contains a form with a list of checkboxes and a 'submit diagnosa' button. The form is set against a background of a large 'NUSANTARA' watermark.

Gambar 3.26 Rancangan menu konsultasi.

Menu konsultasi berisikan sebuah pertanyaan yang dapat dipilih sesuai gejala yang diderita dengan menandai checkbox yang tersedia. Setelah menandai *checkbox* yang dipilih kemudian pilih submit diagnosa untuk melakukan kalkulasi menggunakan algoritma *dempster shafer*.

SKRINING DIAGNOSA
<hr/> <hr/> <hr/> <hr/> <hr/>
FOOTER

Gambar 3.27 Rancangan Halaman utama asisten dokter

Halaman asisten dokter memiliki menu yang diletakkan di *header*. Untuk *footer* berisikan informasi tentang pengembang. Untuk konten awal halaman adalah isi dari menu skrining. Skrining adalah penggunaan tes atau metode diagnosis lain untuk mengetahui apakah seseorang memiliki penyakit atau kondisi tertentu sebelum menyebabkan gejala apapun. Halaman asisten dokter hanya memiliki menu skrining dan konsultasi.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

SKRINING    KONSULTASI
<p>KONSULTASI</p> <div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> <li><input type="checkbox"/> _____</li> </ul> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px; margin-top: 5px;">submit diagnosa</div> </div>
FOOTER

Gambar 3.28 Rancangan menu konsultasi asisten dokter

Menu konsultasi berisikan sebuah pertanyaan yang dapat dipilih sesuai gejala yang diderita dengan menandai *checkbox* yang tersedia. Setelah menandai *checkbox* yang dipilih kemudian pilih submit diagnosa untuk melakukan kalkulasi menggunakan algoritma *dempster shafer*.

