



### **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**

Berikut adalah metode penelitian dan perancangan sistem yang akan dilakukan penulis dalam mengembangkan aplikasi ini:

##### **1. Studi Literatur**

Pada tahap studi literatur dilakukan pencarian dan pemahaman literatur-literatur yang bersangkutan dengan penelitian yang dilakukan baik berupa buku, jurnal, naskah publikasi, maupun literatur-literatur lain.

##### **2. Analisis Kebutuhan**

Pada tahapan ini dilakukan pengumpulan informasi yang dibutuhkan untuk pengembangan sistem dengan wawancara dengan pakar yang bersangkutan yaitu dokter spesialis kulit.

##### **3. Perancangan Sistem**

Pada tahap perancangan sistem dilakukan perancangan bagaimana aplikasi bekerja, perancangan alur dan hubungan antar data yang digunakan, perancangan basis data, dan perancangan antarmuka aplikasi.

##### **4. Pemrograman Sistem**

Pada tahap ini sistem mulai dibangun sesuai perancangan yang dilakukan menggunakan bahasa pemrograman yang telah dipilih.

##### **5. Pengujian**

Pada tahap pengujian, program diuji untuk memastikan bahwa setiap komponen program dapat bekerja dengan benar dan memberi hasil sesuai dengan yang diharapkan.

## 6. Penerapan Sistem

Pada tahap ini sistem diimplementasikan pada perangkat mobile sesuai spesifikasi yang dibutuhkan dan sudah siap digunakan oleh pengguna.

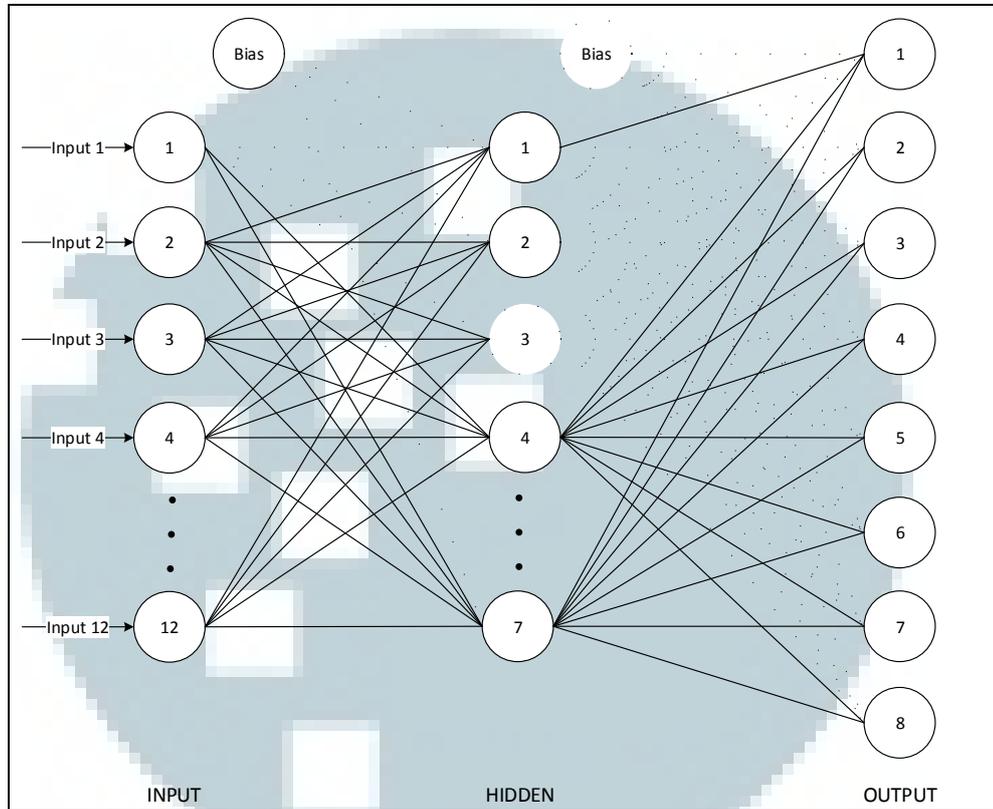
### 3.2. Perancangan Sistem

Pada sistem pakar ini diperlukan pelatihan sistem terlebih dahulu sebelum sistem layak digunakan. Pelatihan ini digunakan untuk menentukan bobot dan bias yang tepat pada jaringan saraf tiruan sehingga bisa memberi hasil yang tepat menggunakan algoritma *backpropagation*. Pada sistem ini digunakan dua program berbeda untuk pelatihan dan untuk deteksi penyakit kulit bagi pengguna. Kedua program merupakan aplikasi android namun proses pelatihan dan penyimpanan data dilakukan di *server* menggunakan *script* PHP dan basis data MySQL. Proses pelatihan dilakukan di *server* supaya tidak membebani perangkat yang digunakan. Hasil pelatihan kemudian juga disimpan di *server* agar pengguna dapat langsung menggunakan fungsi deteksi penyakit pada aplikasi sistem pakar dan tidak perlu melakukan dan menunggu proses pelatihan jaringan. Pakar yang terlibat pada penelitian ini adalah dr. Glennarda Triharsa Hadisaputra, Sp.KK, dokter spesialis kulit dan kelamin dari klinik perawatan kulit Derma Farma dan RSUD dr. M. Ashari Pematang.

#### 3.2.1. Rancangan Jaringan Saraf Tiruan

Jaringan saraf tiruan yang digunakan pada penelitian ini adalah *multi-layer perceptron* dengan tiga *layer* yaitu *input layer*, sebuah *hidden layer*, dan *output layer*. *Input layer* memiliki 12 *neuron* sesuai jumlah masukan dari pengguna. Jumlah *neuron hidden layer* akan disesuaikan melalui beberapa kali uji coba

sehingga bisa didapat hasil terbaik. *Output layer* terdiri dari delapan *neuron* sesuai jumlah penyakit yang dicakup. Berikut adalah rancangan jaringan saraf tiruan yang digunakan.



Gambar 3.1. Rancangan arsitektur jaringan saraf tiruan yang digunakan

Variabel *input* merupakan 12 kriteria untuk mendeteksi penyakit kulit yang masing-masing mewakili 1 *neuron input*. *Input* yang digunakan pada jaringan saraf tiruan bernilai antara 0 sampai dengan 1 untuk usia dan nilai biner 0 untuk Tidak dan 1 untuk Ya untuk variabel lain. Berikut adalah tabel variabel *input* dari jaringan saraf tiruan yang digunakan.

Tabel 3.1. Tabel variabel *input*

Input #	Keterangan	Variabel	Nilai
1	Usia Penderita	0-5 Tahun	0
		6-13 Tahun	0.25
		14-30 Tahun	0.5
		31-40 Tahun	0.75
		41 Tahun Ke atas	1

Tabel 3.1. Tabel variabel *input* (lanjutan)

<i>Input</i> #	Keterangan	Variabel	Nilai
2	Kulit terasa gatal	Tidak	1
		Ya	0
3	Kulit terasa nyeri atau sakit	Tidak	1
		Ya	0
4	Penderita sering berkontak langsung dengan bahan kimia tertentu	Tidak	1
		Ya	0
5	Penderita memiliki riwayat penyakit sistemik (diabetes, hipertensi, gangguan ginjal, dan sebagainya)	Tidak	1
		Ya	0
6	Penderita sering terpapar sinar matahari	Tidak	1
		Ya	0
7	Terdapat bercak merah pada kulit	Tidak	1
		Ya	0
8	Terdapat bercak putih pada kulit	Tidak	1
		Ya	0
9	Terdapat bercak hitam pada kulit	Tidak	1
		Ya	0
10	Terdapat plenting berisi air pada kulit	Tidak	1
		Ya	0
11	Terdapat benjolan pada kulit	Tidak	1
		Ya	0
12	Kulit kering atau bersisik	Tidak	1
		Ya	0

*Output* dari jaringan saraf tiruan merupakan delapan penyakit yang dicakup dalam penelitian ini. Variabel *output* masing-masing bernilai 0 untuk penyakit yang tidak mungkin diderita dan 1 untuk penyakit yang mungkin diderita. Berikut adalah tabel variabel *output* dari jaringan saraf tiruan yang digunakan.

Tabel 3.2. Tabel variabel *output*

<i>Output</i> #	Keterangan	Variabel	Nilai
1	Dermatitis Atopik	Tidak Mungkin	0
		Mungkin	1
2	Dermatitis Kontak	Tidak Mungkin	0
		Mungkin	1
3	Jerawat (Akne Vulgaris)	Tidak Mungkin	0
		Mungkin	1
4	Melasma (Flek)	Tidak Mungkin	0
		Mungkin	1

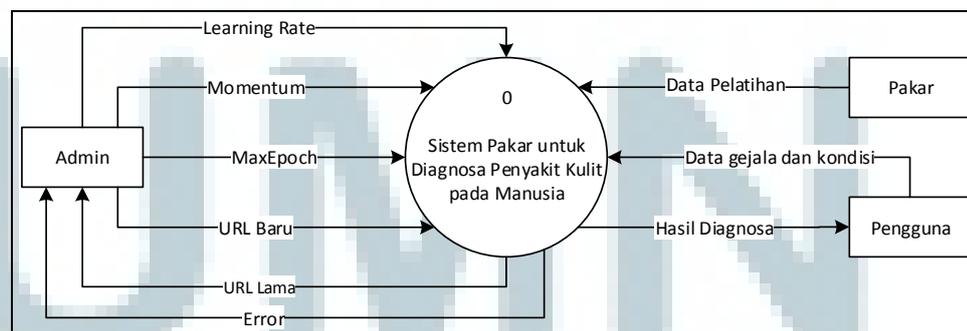
Tabel 3.2. Tabel variabel *output* (lanjutan)

<i>Output #</i>	Keterangan	Variabel	Nilai
5	Tinea (Jamur Kulit)	Tidak Mungkin	0
		Mungkin	1
6	Pioderma	Tidak Mungkin	0
		Mungkin	1
7	Tumor Kulit	Tidak Mungkin	0
		Mungkin	1
8	Dermatitis akibat penyakit sistemik	Tidak Mungkin	0
		Mungkin	1

### 3.2.2. Data Flow Diagram (DFD)

#### A. Diagram Konteks

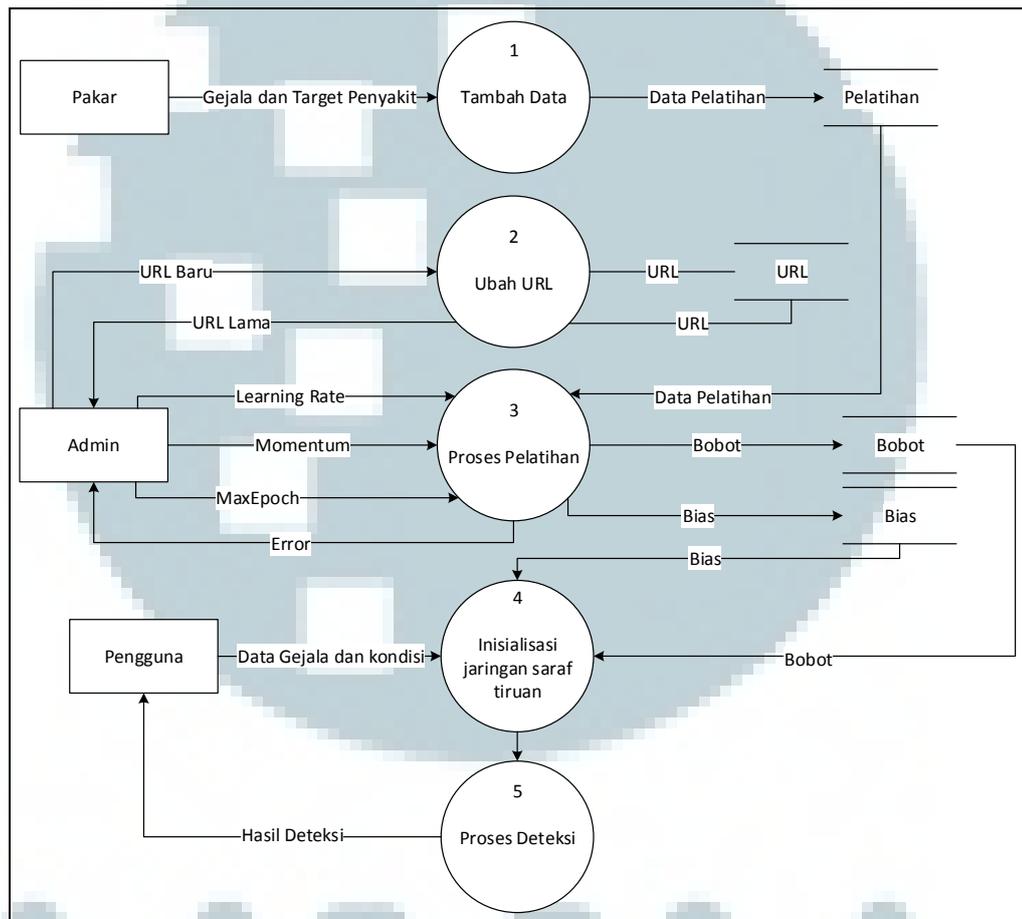
Diagram konteks sistem pakar ditunjukkan pada Gambar 3.2. Terdapat dua entitas yang terlibat dalam tahap pelatihan jaringan saraf tiruan untuk sistem pakar yaitu pakar dan admin. Pakar memberikan data pelatihan berupa kriteria gejala dan kondisi dari tiap-tiap penyakit yang mampu dideteksi sistem. Admin memberikan data-data berupa parameter untuk melatih jaringan saraf tiruan menggunakan backpropagation dan menerima hasil pelatihan berupa nilai error yang didapat. Pengguna memberikan data berupa gejala dan kondisi sebagai masukan untuk dideteksi dan mendapat keluaran berupa hasil dari sistem.



Gambar 3.2. Diagram konteks sistem pakar

DFD *level 1* sistem ditunjukkan pada Gambar 3.3. Data pelatihan dari pakar dimasukkan dari proses tambah data kemudian disimpan di tabel pelatihan. Isi tabel

pelatihan ini digunakan pada proses pelatihan. Dari hasil pelatihan, admin memutuskan apakah bobot dan bias disimpan atau tidak. Pengguna memberikan data gejala dan kondisi untuk proses inialisasi jaringan saraf tiruan dengan menggunakan bobot dan bias yang tersimpan di basis data yang kemudian diproses pada proses deteksi dan mengembalikan hasil deteksi ke pengguna.



Gambar 3.3. Data Flow Diagram level 1 sistem pakar

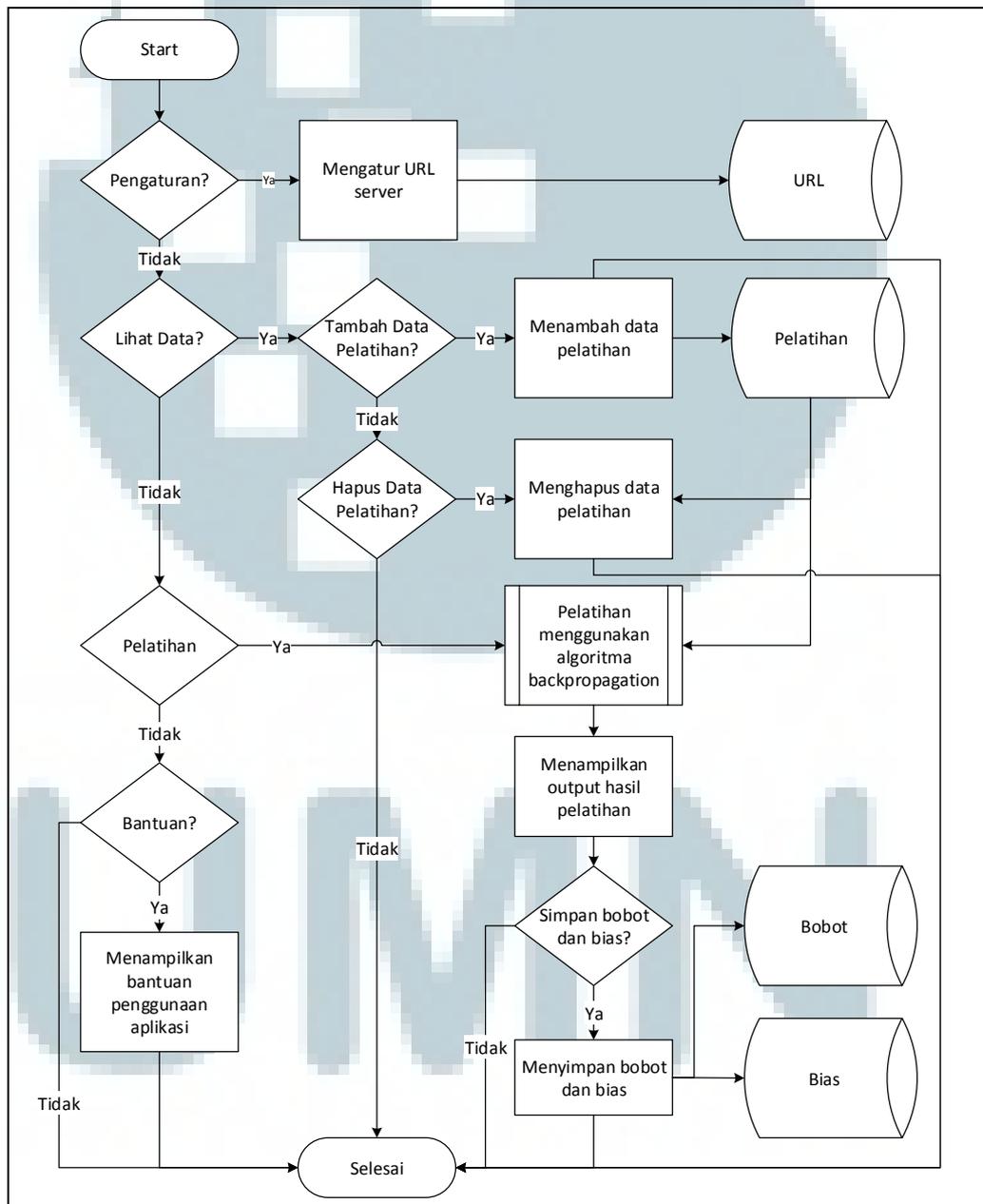
### 3.2.3. Flowchart

#### A. Flowchart Aplikasi untuk Tahap Pelatihan

Pada aplikasi untuk tahap pelatihan terdapat tiga buah pilihan yaitu melihat data pelatihan, melakukan pelatihan jaringan, dan melihat bantuan penggunaan aplikasi. Ketika memilih pilihan Lihat Data Pelatihan, maka selain hanya melihat data-data pelatihan, dapat pula menambah data pelatihan maupun menghapus data

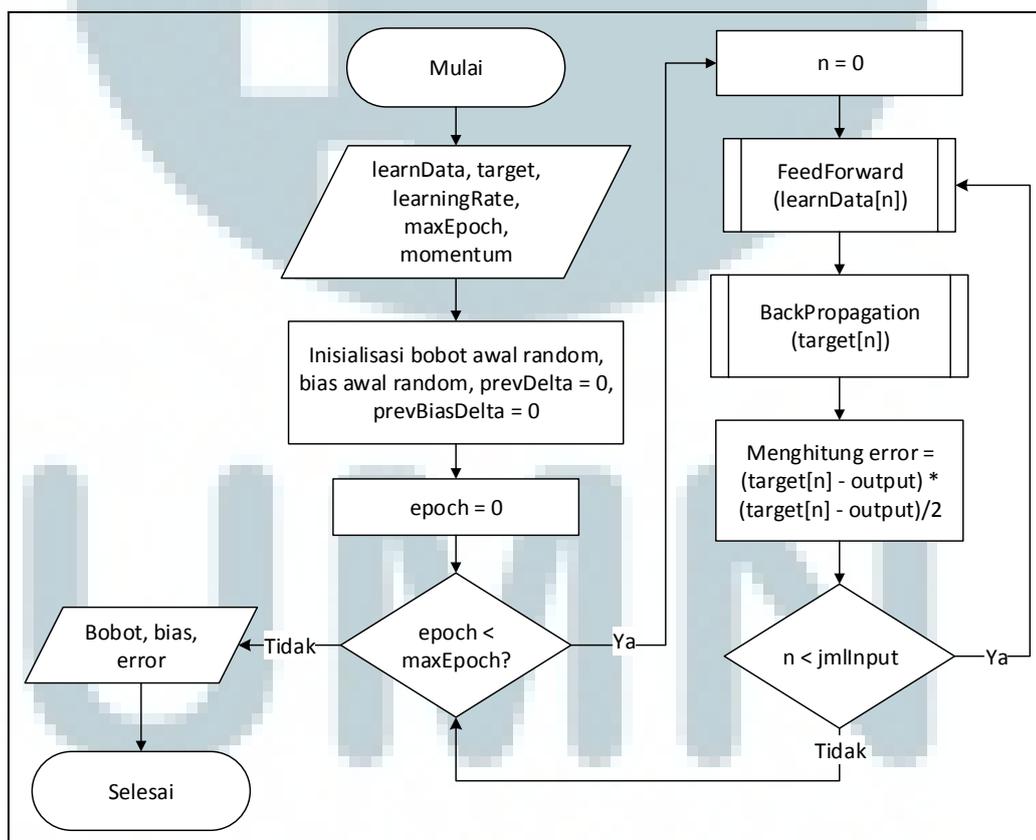
pelatihan. Jika memilih Latih Jaringan, maka akan dilakukan proses pelatihan jaringan saraf tiruan menggunakan *backpropagation* menggunakan data yang tersimpan. Setelah pelatihan selesai, bobot dan bias hasil pelatihan dapat disimpan maupun tidak. Jika tidak, maka bobot dan bias yang tersimpan tidak berubah. Jika disimpan, maka bobot dan bias tersimpan akan dihapus dan diganti yang baru.

Flowchart aplikasi untuk tahap pelatihan ditunjukkan Gambar 3.4.



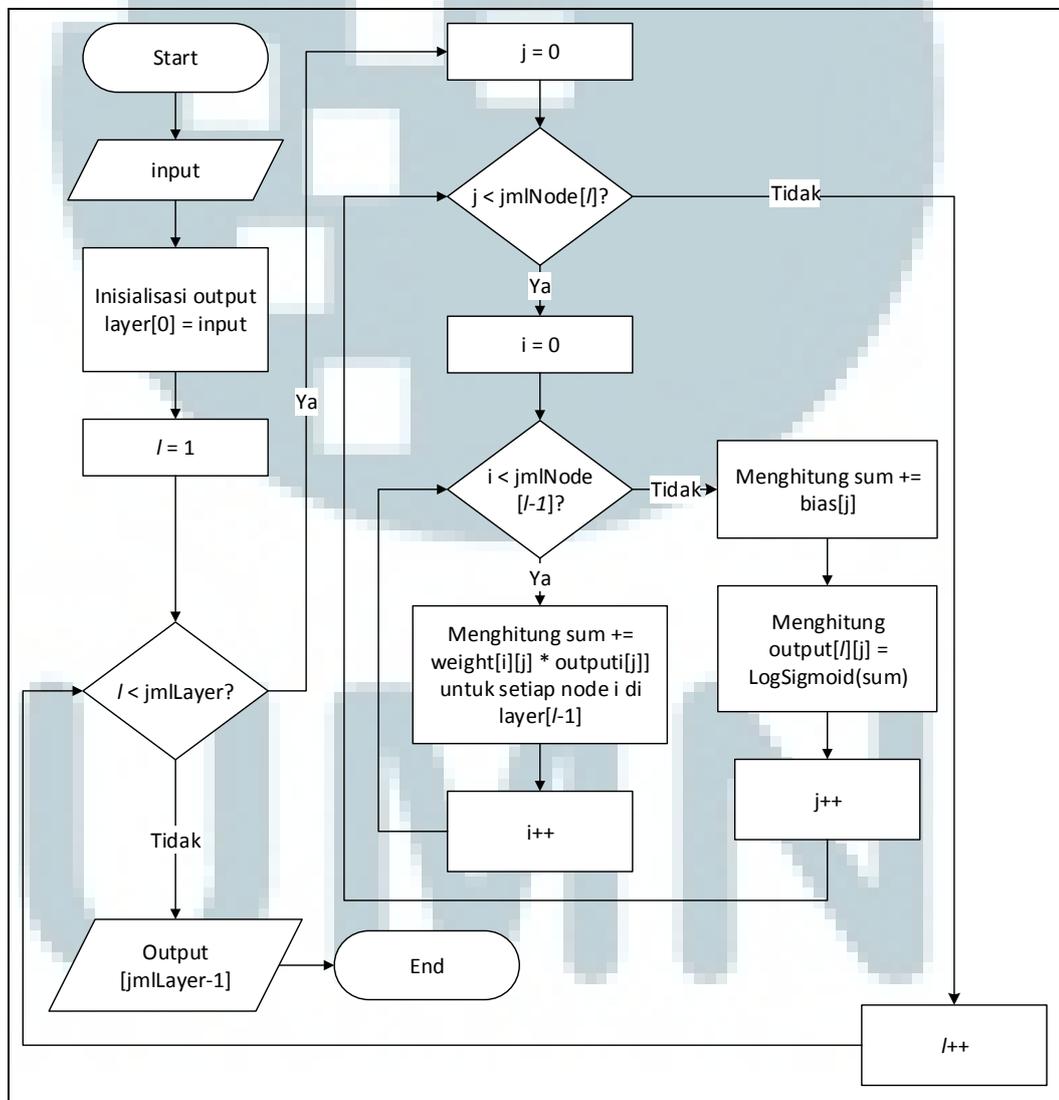
Gambar 3.4. Flowchart aplikasi untuk tahap pelatihan

Proses pelatihan jaringan saraf tiruan menggunakan *backpropagation* ditunjukkan Gambar 3.5. Pertama-tama dilakukan inisialisasi data-data yang diperlukan untuk pelatihan. *Input* learnData dan target berisi data pelatihan yang diambil dari tabel pelatihan. *Input* learningRate, maxEpoch, momentum, dan hiddenNode didapat dari masukan pengguna. Kemudian dilakukan inisialisasi variabel lain yaitu bobot dan bias awal bernilai *random*, *delta* bobot sebelumnya (prevDelta) dan *delta* bias sebelumnya (prevBiasDelta) bernilai 0 (nol). Kemudian dilakukan proses pelatihan untuk setiap data pelatihan yang terdiri dari tahap *feed forward* (propagasi maju) dan *back propagation* (propagasi balik) secara berulang sesuai jumlah *epoch* maksimal yang ditentukan. Setelah mencapai *epoch* maksimal, didapat nilai error, bobot, dan bias akhir.



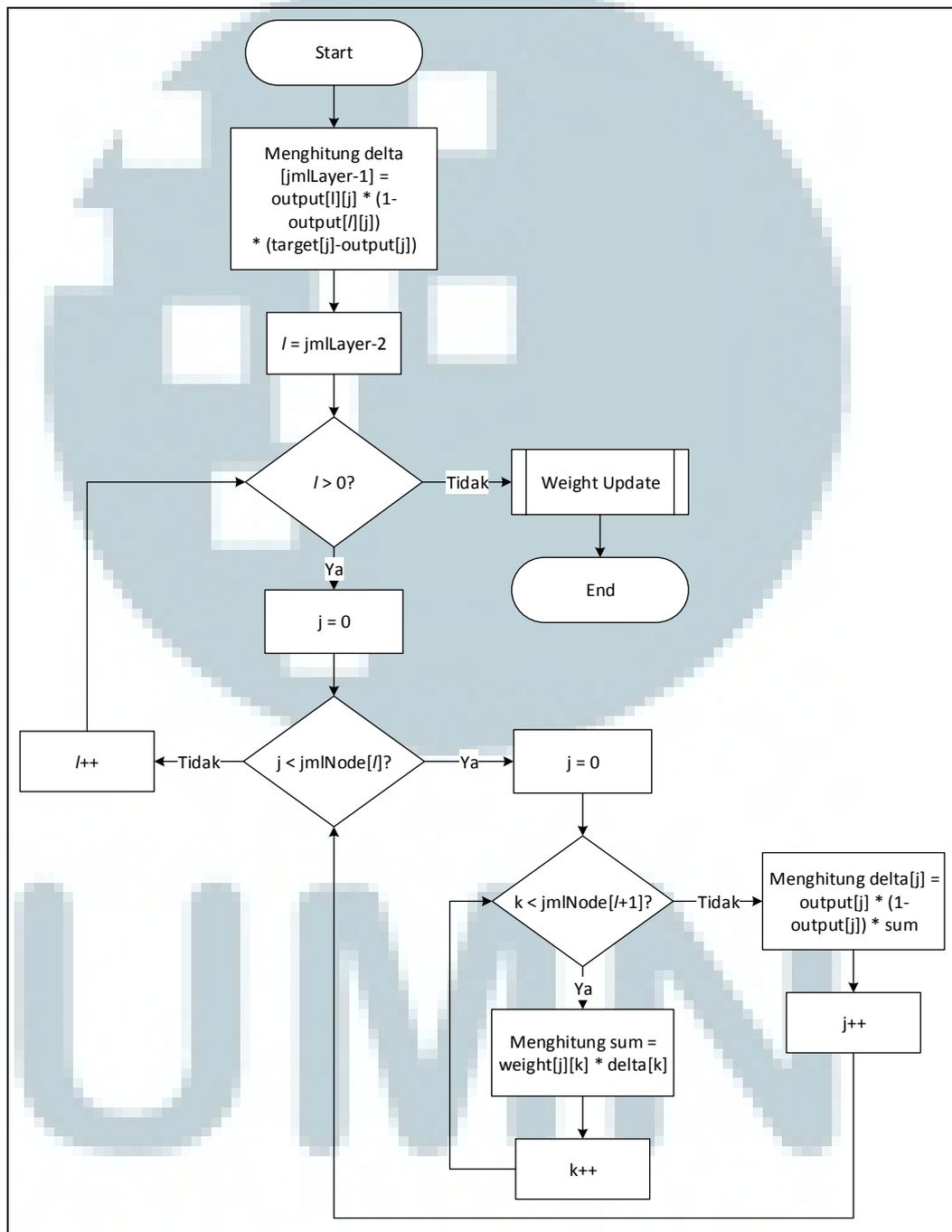
Gambar 3.5. Flowchart Proses Pelatihan Menggunakan *Backpropagation*

Pada tahap *feed forward*, pertama-tama akan dilakukan inisialisasi data pelatihan ke *input layer*. Kemudian dilakukan perhitungan propagasi maju untuk mendapat nilai *output* menggunakan bobot dan bias saat itu. *Output* didapatkan dari perhitungan jumlah perkalian bobot dengan *input* pada setiap *node* di *layer l* ditambah bias dari tiap *node* tersebut yang dimasukkan ke dalam fungsi aktivasi LogSigmoid. Tahap *feed forward* akan menghasilkan nilai *output* sebenarnya dengan bobot dan bias saat itu. *Output* ini akan digunakan pada tahap berikutnya, yaitu *back propagation*.



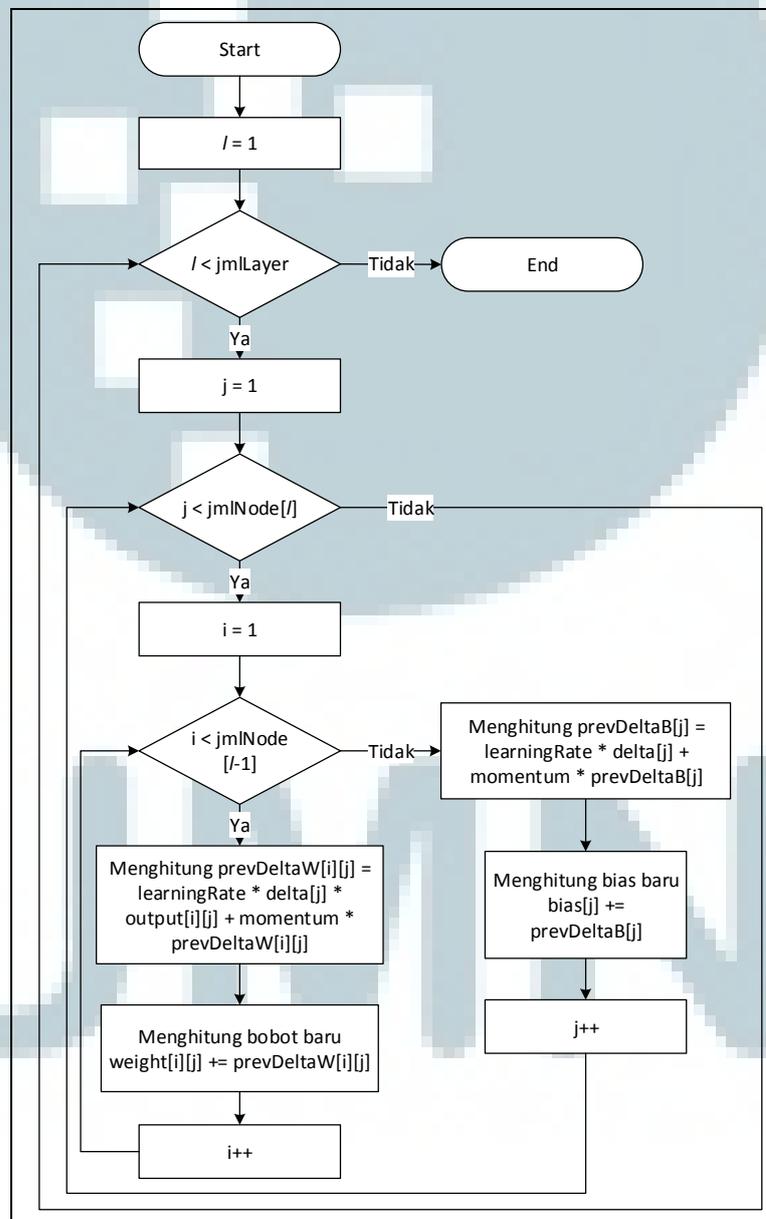
Gambar 3.6. Flowchart Proses Feed Forward dari Pelatihan Backpropagation

Tahap *back propagation* dimulai dengan menghitung *delta layer output* yaitu perkalian dari hasil turunan fungsi aktivasi dengan selisih target dengan *output*. Kemudian *delta layer-layer* sebelumnya dihitung dengan mengalikan hasil turunan fungsi aktivasi dengan jumlah perkalian bobot dengan *delta layer* berikutnya.



Gambar 3.7. Flowchart Proses Backpropagation dari Pelatihan Backpropagation

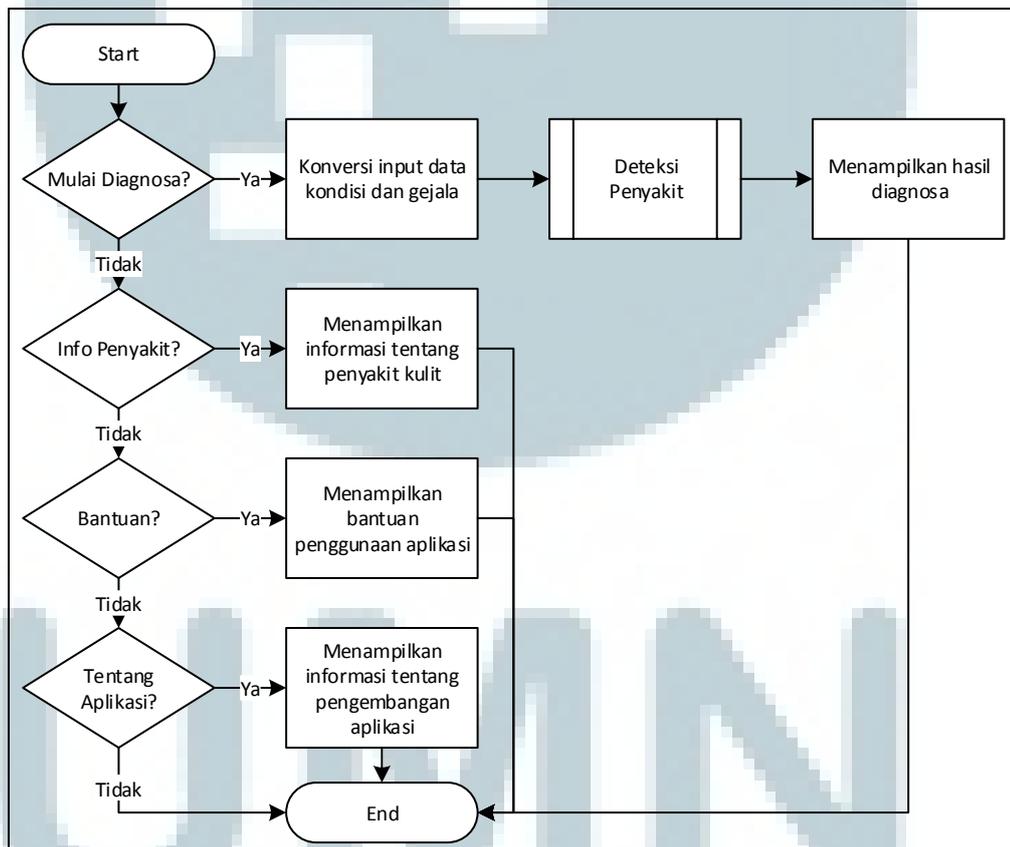
Setelah  $\delta$  dari setiap *node* pada setiap *layer* dihitung, nilai  $\delta$  tersebut digunakan untuk memperbarui bobot dan bias agar bisa memberi hasil sesuai target. Bobot baru dihitung dengan menambahkan bobot awal dengan  $\delta$  bobot sebelumnya yang telah dihitung menggunakan  $\delta$  yang didapat dari tahap *backpropagation*. Bias baru juga diperbarui dengan menambahkan bias awal dengan  $\delta$  bias sebelumnya yang dihitung menggunakan  $\delta$  yang didapat pada tahap *backpropagation* juga.



Gambar 3.8. Flowchart Proses Weight Update pada Proses Backpropagation

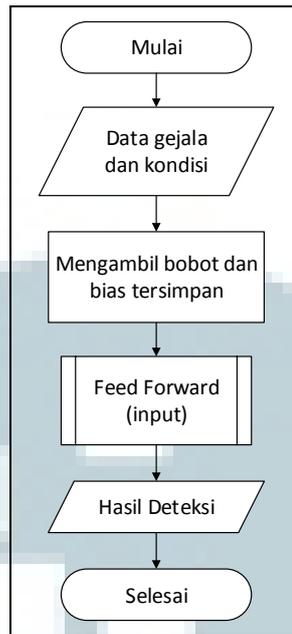
## B. Flowchart Aplikasi Sistem Pakar

Pada aplikasi sistem pakar terdapat empat buah pilihan, yaitu Mulai Diagnosa, Info Penyakit Kulit, Bantuan, dan Tentang Aplikasi. Ketika memilih Mulai Diagnosa, maka masukan dari pengguna akan dikonversi untuk dideteksi. Kemudian hasil yang didapat akan diproses dan ditampilkan beserta keterangannya. Jika Info Penyakit dipilih, maka akan ditampilkan informasi singkat tentang penyakit kulit, begitu pula pada pilihan Bantuan dan Tentang Aplikasi, maka masing-masing akan ditampilkan cara penggunaan aplikasi secara sederhana dan informasi singkat tentang pengembangan aplikasi.



Gambar 3.9. Flowchart Aplikasi Sistem Pakar

Proses deteksi penyakit dilakukan dengan menggunakan bobot dan bias hasil pelatihan yang tersimpan menggunakan *feed forward* pada Gambar 3.6. dengan *input* data dari pengguna yang sudah dikonversi untuk didapat *output* sebenarnya.



Gambar 3.10. *Flowchart* Proses Deteksi pada Aplikasi Sistem Pakar

### 3.2.4. Struktur Tabel

Sistem yang dibangun menggunakan dua jenis *database*, yaitu *database* MySQL pada *server* dan *database* SQLite pada aplikasi. *Database* pada *server* berperan untuk menyimpan data pelatihan serta hasilnya, sehingga hasil pelatihan berupa bobot dan bias bisa digunakan pada sistem pakar. Pada aplikasi sistem pakar juga terdapat *database* untuk menyimpan data penyakit. Keempat tabel ini tidak memiliki hubungan satu sama lain sehingga sistem ini tidak memiliki *Entity Relationship Diagram* (ERD).

#### A. Tabel pada aplikasi sistem pakar

Nama Tabel	: penyakit
Primary Key	: id
Fungsi	: menyimpan nama, keterangan, dan solusi penyakit untuk ditampilkan di hasil deteksi

Tabel 3.3. Tabel penyakit di program sistem pakar

No.	Nama Field	Tipe	Deskripsi
1	id	Integer	Id penyakit
2	penyakit	Text	Nama penyakit
3	Keterangan	Text	Keterangan penyakit
4	Solusi	Text	Solusi penanganan penyakit

## B. Tabel pada server

1. Nama Tabel : pelatihan

Primary Key : idPelatihan

Fungsi : menyimpan data-data pelatihan jaringan saraf tiruan

Tabel 3.4. Tabel data pelatihan

No.	Nama Field	Tipe	Deskripsi
1	idPelatihan	Integer(11)	Id data pelatihan
2	usia	Double	Nilai kategori usia (kriteria 0)
3	i1	TinyInt(1)	Nilai <i>input</i> kriteria 1
4	i2	TinyInt(1)	Nilai <i>input</i> kriteria 2
5	i3	TinyInt(1)	Nilai <i>input</i> kriteria 3
6	i4	TinyInt(1)	Nilai <i>input</i> kriteria 4
7	i5	TinyInt(1)	Nilai <i>input</i> kriteria 5
8	i6	TinyInt(1)	Nilai <i>input</i> kriteria 6
9	i7	TinyInt(1)	Nilai <i>input</i> kriteria 7
10	i8	TinyInt(1)	Nilai <i>input</i> kriteria 8
11	i9	TinyInt(1)	Nilai <i>input</i> kriteria 9
12	i10	TinyInt(1)	Nilai <i>input</i> kriteria 10
13	i11	TinyInt(1)	Nilai <i>input</i> kriteria 11
14	target	String(8)	String biner target

2. Nama Tabel : bobot

Primary Key : id

Fungsi : menyimpan bobot hasil pelatihan

Tabel 3.5. Tabel data bobot

No.	Nama Field	Tipe	Deskripsi
1	id	Integer(11)	Id bobot
2	layer	Integer(11)	Layer

Tabel 3.5. Tabel data bobot (lanjutan)

No.	Nama Field	Tipe	Deskripsi
3	nodeIn	Integer(11)	Node layer sebelumnya
4	nodeOut	Integer(11)	Node layer ini
5	bobot	Double	Nilai bobot

3. Nama Tabel : bias

Primary Key : id

Fungsi : menyimpan bias hasil pelatihan

Tabel 3.6. Tabel data bias

No.	Nama Field	Tipe	Deskripsi
1	id	Integer(11)	Id bobot
2	layer	Integer(11)	Layer
3	node	Integer(11)	Node layer ini
4	bias	Double	Nilai bias

### 3.2.5. Desain Antar Muka

Penelitian ini terdiri dari 2 aplikasi yaitu aplikasi pelatihan dan aplikasi sistem pakar. Tampilan halaman utama aplikasi pelatihan ditunjukkan pada Gambar 3.11.

The screenshot shows a software interface for skin diagnosis training. The main window has a title bar with 'Diagnosa Kulit' and 'BANTUAN'. The content area is divided into several sections: a central text area with a welcome message, a 'Masukkan Parameter' section on the right with three input fields (Learning Rate: 0.1, Epoch Maksimal: 1000, Momentum: 0.1) and 'Batal' and 'Latih' buttons, and a bottom section with 'Lihat Data Pelatihan' and 'Latih Jaringan' buttons. An arrow points from the 'Latih Jaringan' button to the 'Latih' button in the parameter section.

Gambar 3.11. Tampilan utama aplikasi pelatihan jaringan

Tampilan halaman utama hanya terdiri dari sebuah *textview* untuk memuat *welcome message* serta dua tombol dan satu tombol menu pada *action bar* untuk navigasi seperti pada Gambar 3.11. Tombol latih jaringan saat ditekan akan menampilkan *dialog* untuk meminta *input* parameter pelatihan. Setelah semua parameter diisi, saat menekan tombol latih pada *dialog*, maka sistem akan melakukan pelatihan di *server* kemudian akan mengembalikan hasil pelatihan. Admin dapat memilih untuk menyimpan hasil itu atau tidak. Tombol lihat data saat ditekan akan bernavigasi ke halaman daftar data pelatihan pada Gambar 3.12, sementara tombol menu bantuan akan bernavigasi ke halaman bantuan pada Gambar 3.14.

Latih Jaringan	KOSONGKAN	+
Data Latih 1		
Data Latih 2		
Data Latih 3		
Data Latih 4		
Data Latih 5		

Gambar 3.12. Tampilan halaman pelatihan

Di halaman pelatihan terdapat sebuah *listview* untuk mengatur data pelatihan. Data pelatihan hanya bisa ditambah dan dihapus. Penambahan data pelatihan dilakukan dengan menekan tombol '+' di bagian atas dan akan menampilkan *form* penambahan data seperti pada Gambar 3.13. Penghapusan dilakukan dengan menekan data yang ingin dihapus di *listview*. Ketika data yang dipilih ditekan, maka akan muncul *dialog* berisi isi data serta tombol untuk hapus atau batal.

Tambah Data	SIMPAN
Target	
<input checked="" type="checkbox"/> Penyakit 1	
<input checked="" type="checkbox"/> Penyakit 2	
<input checked="" type="checkbox"/> Penyakit 3	
:	
:	
<input checked="" type="checkbox"/> Penyakit 8	
Kategori Usia	
<input checked="" type="radio"/> <5 Tahun	
<input type="radio"/> 5-13 Tahun	
<input type="radio"/> 14-30 Tahun	
<input type="radio"/> 30-40 Tahun	
<input type="radio"/> >41 Tahun	
Gejala	
<input checked="" type="checkbox"/> Gejala 1	
<input checked="" type="checkbox"/> Gejala 2	
<input checked="" type="checkbox"/> Gejala 3	
:	
:	
<input checked="" type="checkbox"/> Gejala N	

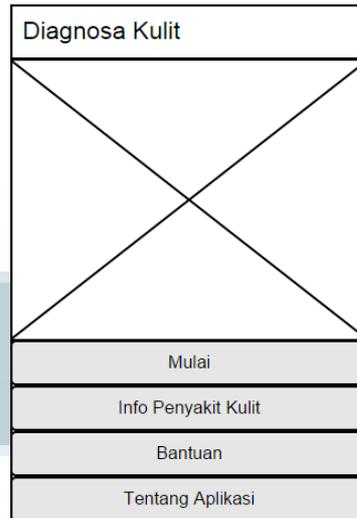
Gambar 3.13. Tampilan halaman *form* pelatihan

*Form* pelatihan terdiri dari beberapa *combo box* untuk memilih penyakit yang mungkin dialami, beberapa *radio button* untuk memilih kategori usia, dan beberapa *combo box* untuk memilih gejala yang dialami pada penyakit terpilih. Setelah *form* diisi, tombol menu di *action bar* dipilih untuk menyimpan data ke tabel. *Form* pelatihan dapat dilihat pada Gambar 3.13.

Bantuan
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec du. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.</p>

Gambar 3.14. Tampilan halaman bantuan

Di halaman bantuan hanya terdapat sebuah *textview* yang berisi cara penggunaan aplikasi secara singkat. Halaman bantuan ditunjukkan Gambar 3.14.



Gambar 3.15. Tampilan halaman utama sistem pakar

Halaman utama sistem pakar terdiri dari sebuah *imageview* dan empat buah tombol untuk memulai diagnosa, melihat info penyakit kulit, bantuan, dan tentang aplikasi. Tampilan halaman utama aplikasi sistem pakar ditunjukkan Gambar 3.15.

Gambar 3.16. Tampilan halaman *form* diagnosa

Tampilan dari halaman diagnosa ditunjukkan pada Gambar 3.16. Form diagnosa berisi sebuah *textbox* untuk diisi usia dari penderita dan sejumlah *radio button* untuk memilih gejala yang sesuai dengan penderita. Setelah halaman form diagnosa diisi kemudian ketika tombol Selesai ditekan, maka data akan dikirim ke *server* untuk diproses menggunakan hasil pelatihan yang telah dilakukan

sebelumnya. Jika berhasil, maka akan ditampilkan halaman hasil diagnosa berisi nama penyakit, keterangan, dan solusinya seperti pada Gambar 3.17.

Diagnosa Kulit	SELESAI
<p>Anda kemungkinan menderita</p> <p style="text-align: center;"><b>Penyakit N</b></p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec</p> <p><b>Solusi</b></p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec</p>	

Gambar 3.17. Tampilan halaman hasil diagnosa

Halaman info penyakit kulit ditunjukkan Gambar 3.18. Halaman ini memuat informasi singkat tentang penyakit yang dicakup dalam sistem pakar ini. Daftar penyakit dimuat dalam sebuah *listview* dan ketika salah satu penyakit dipilih akan menampilkan keterangan penyakit dalam suatu *dialog*. Halaman bantuan pada sistem pakar memiliki tampilan yang serupa dengan halaman bantuan pada sistem pelatihan di Gambar 3.14.

<table border="1"> <tr> <td><b>Info Penyakit Kulit</b></td> </tr> <tr> <td>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate</td> </tr> <tr> <td>Penyakit 1</td> </tr> <tr> <td>Penyakit 2</td> </tr> <tr> <td>Penyakit 3</td> </tr> <tr> <td>Penyakit 4</td> </tr> <tr> <td>Penyakit 5</td> </tr> <tr> <td>Penyakit 6</td> </tr> <tr> <td>Penyakit 7</td> </tr> <tr> <td>Penyakit 8</td> </tr> </table>	<b>Info Penyakit Kulit</b>	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate	Penyakit 1	Penyakit 2	Penyakit 3	Penyakit 4	Penyakit 5	Penyakit 6	Penyakit 7	Penyakit 8	<table border="1"> <tr> <td><b>Penyakit 1</b></td> </tr> <tr> <td>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique.</td> </tr> <tr> <td>Kembali</td> </tr> </table>	<b>Penyakit 1</b>	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique.	Kembali
<b>Info Penyakit Kulit</b>														
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate														
Penyakit 1														
Penyakit 2														
Penyakit 3														
Penyakit 4														
Penyakit 5														
Penyakit 6														
Penyakit 7														
Penyakit 8														
<b>Penyakit 1</b>														
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique.														
Kembali														

Gambar 3.18. Tampilan halaman info penyakit