

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

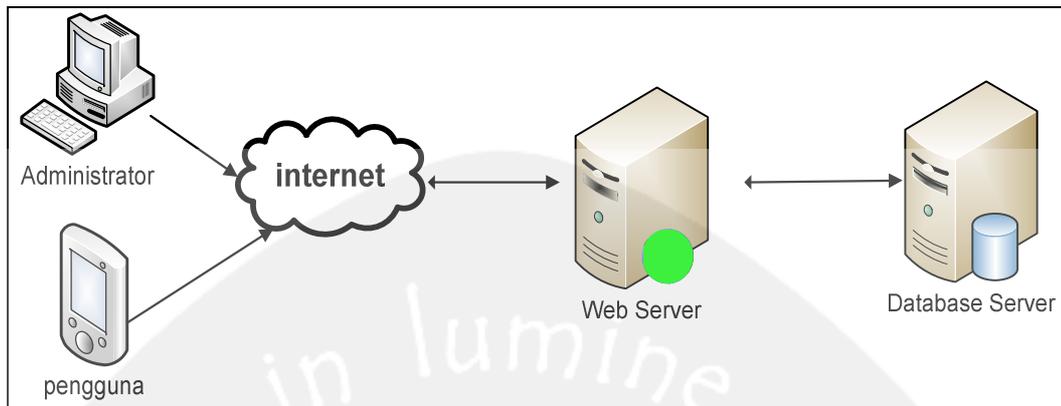
Pada bab ini akan dibahas mengenai analisis dan perancangan sistem aplikasi yang akan dibuat. Pokok bahasan yang terdapat dalam bab ini adalah deskripsi produk, kebutuhan khusus, kebutuhan fungsionalitas, dan perancangan arsitektur dari **SiPaPA**.

IV.1 Analisis Sistem

SiPaPA merupakan perangkat lunak yang dikembangkan untuk membantu pemilik anjing dalam mendiagnosa penyakit anjing berdasarkan gejala-gejala yang ada, serta mengetahui pencegahan yang dapat dilakukan atas penyakit yang diderita oleh anjing.

Perangkat lunak **SiPaPA** ini berjalan pada *platform* JAVA pada perangkat *mobile* dan dibuat menggunakan bahasa pemrograman J2ME (Java 2 Micro Edition) dengan lingkungan emulator NetBeans 6.9, MySQL sebagai penyimpanan data (*database*) serta untuk mendukung aplikasi ini, dibangun *website* menggunakan bahasa pemrograman PHP untuk memudahkan administrator dalam mengedit penyakit, mengelola informasi, dan mengedit user/pengguna sedangkan untuk lingkungan pemrogramannya menggunakan Macromedia Dreamweaver 8.

Pengguna akan berinteraksi dengan aplikasi melalui antarmuka GUI (*Graphical User Interface*). Pada aplikasi ini, seperti yang terlihat pada gambar 4.1, arsitektur perangkat lunak bersifat *Client-server*, dimana data tersimpan di server. User dapat mengakses data yang ada di server tersebut secara online dengan memanggil *web service* pada *web site* yang tersedia di *web server*.



Gambar 4.1 Arsitektur Perangkat lunak SiPaPA

IV.1.1 Lingkup Masalah

Aplikasi Sistem pakar untuk mendiagnosa penyakit anjing berbasis mobile (**SiPaPA**) dikembangkan dengan tujuan untuk :

1. Melakukan Identifikasi Penyakit, yaitu dengan memilih gejala-gejala yang dialami anjing melalui perangkat *mobile*.
2. Menampilkan hasil diagnosa jenis penyakit beserta informasi pencegahan yang dapat dilakukan secara dini melalui perangkat *mobile*.
3. Menampilkan jenis penyakit serta informasi terbaru mengenai anjing melalui perangkat *mobile*.
4. Pengelolaan Login yang akan digunakan oleh admin melalui website **SiPaPA**.
5. Melakukan pembaharuan terhadap jenis penyakit anjing melalui website **SiPaPA**.
6. Melakukan pengelolaan seperti penambahan, pembaharuan, penghapusan terhadap informasi terbaru anjing melalui website **SiPaPA**.
7. Melakukan pembaharuan terhadap data user melalui website **SiPaPA**.

IV.1.2 Fungsi Produk

Fungsi produk perangkat lunak **SiPaPA** adalah sebagai berikut:

A. Fungsi produk Administrator

1. Fungsi *Login* (SKPL-SiPaPA-001)

Merupakan fungsi yang digunakan oleh pengguna sistem agar dapat masuk ke dalam sistem **SiPaPA** yang akan digunakan.

2. Fungsi *Edit Penyakit* (SKPL-SiPaPA-002).

Digunakan untuk melakukan perubahan data penyakit yang telah ada pada tabel penyakit.

3. Fungsi *Kelola Informasi* (SKPL-SiPaPA-003).

Merupakan fungsi yang akan digunakan untuk mengelola data informasi terbaru seputar dunia anjing. Fungsi ini meliputi :

a. Fungsi *Tambah Informasi* (SKPL-SiPaPA-003-01).

Fungsi ini digunakan untuk menambah data informasi yang kemudian disimpan dalam tabel Informasi.

b. Fungsi *Ubah Informasi* (SKPL-SiPaPA-003-02).

Fungsi ini digunakan untuk melakukan perubahan data informasi yang telah ada dalam tabel Informasi.

c. Fungsi *Hapus Informasi* (SKPL-SiPaPA-003-03).

Fungsi ini digunakan untuk melakukan penghapusan data informasi dari tabel Informasi.

4. Fungsi *Edit User* (SKPL-SiPaPA-004).

Merupakan fungsi yang akan digunakan untuk mengelola data user yang menggunakan sistem. Fungsi ini digunakan untuk melakukan perubahan data user yang telah ada dalam tabel User.

B. Fungsi Produk Pemilik Anjing

1. Fungsi *Mengidentifikasi Penyakit* (SKPL-SiPaPA-005).

Pada fungsi ini, user akan diberi beberapa pertanyaan dan akan menjawab berdasarkan gejala yang dialami pada anjing. Kemudian sistem akan mengidentifikasi dan menampilkan hasil diagnosis penyakit beserta pencegahan secara dini atas penyakit tersebut.

2. Fungsi *Tampil Penyakit* (SKPL-SiPaPA-006)

Fungsi ini digunakan untuk menampilkan nama beserta deskripsi penyakit yang ada dalam basis data.

3. Fungsi *Tampil Informasi* (SKPL-SiPaPA-007)

Fungsi ini digunakan untuk menampilkan informasi mengenai dunia anjing yang ada dalam basis data.

IV.1.3 Kebutuhan khusus

IV.1.3.1 Kebutuhan Antarmuka Eksternal

Kebutuhan antar muka eksternal pada perangkat lunak **SiPaPA** meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

IV.1.3.2 Kebutuhan Antarmuka Pemakai

Pengguna pada aplikasi *mobile* dapat berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form, dan pada aplikasi web dapat berinteraksi dengan antarmuka halaman-halaman web.

IV.1.3.3 Kebutuhan Antarmuka Perangkat Keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak **SiPaPA** adalah:

1. Perangkat Komputer (PC, Laptop).
2. Perangkat *mobile*.

IV.1.3.4 Kebutuhan Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak **SiPaPA** adalah sebagai berikut :

1. Nama : MySQL
Sumber : OpenSource
Sebagai *database management sistem* (DBMS) yang digunakan untuk penyimpanan data di sisi *server*.
2. Nama : Windows
Sumber : Microsoft.
Sebagai sistem operasi untuk aplikasi *Web Base*.
 - a. Nama : Internet Explorer, Mozilla FireFox, Google Chrome
Sumber : Microsoft, Mozilla, Google, Apple.
Sebagai browser.
3. Nama : Apache
Sumber : Apache Software Foundation.
Sebagai *web server*.
4. Nama : MIDP&CLDC
Sumber : Sun Microsistem.
Sebagai *platform environment* pada perangkat *mobile*.
5. Nama : Symbian OS
Sumber : Symbian Software.
Sebagai sistem operasi untuk perangkat *mobile*.

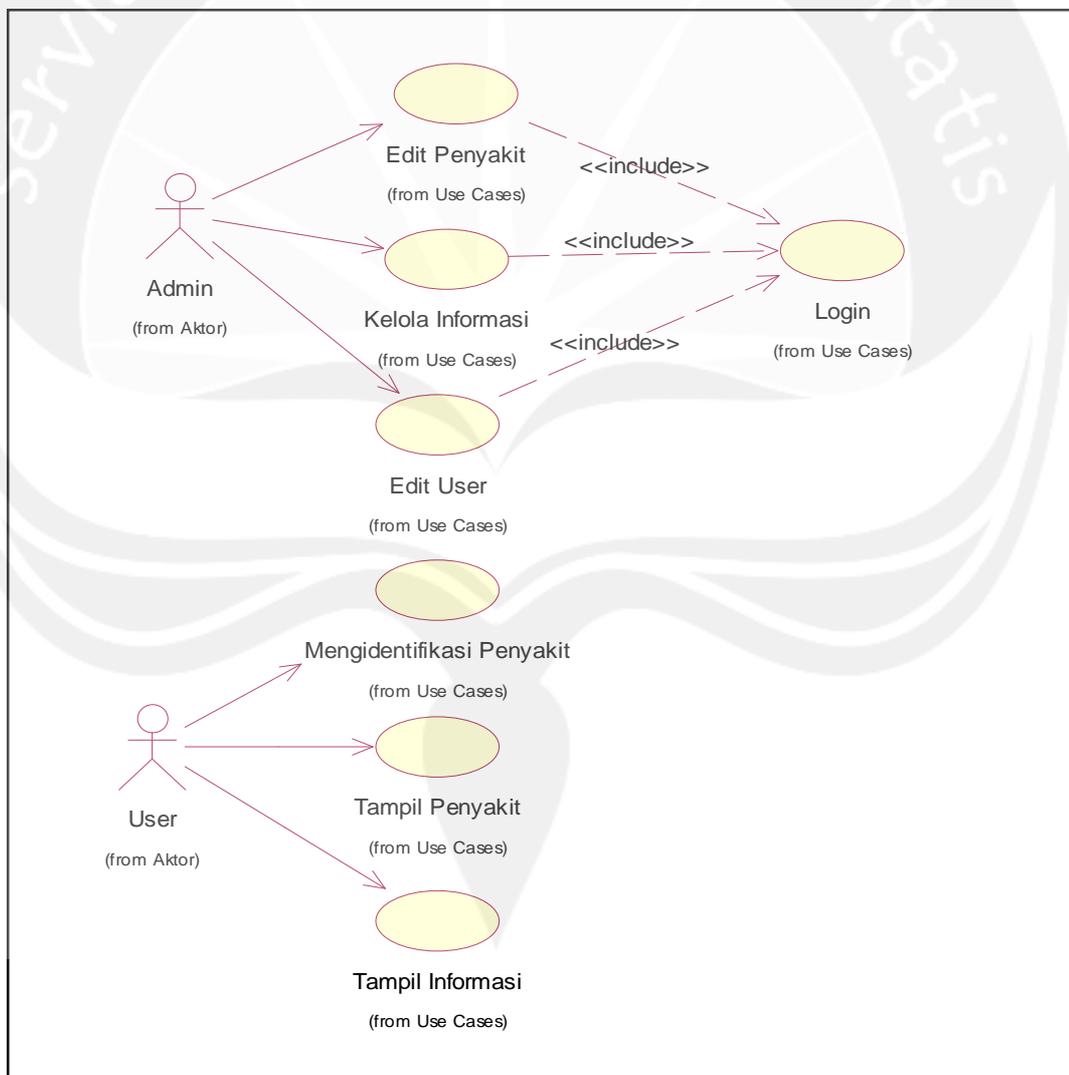
IV.1.3.5 Kebutuhan Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SiPaPA menggunakan protocol HTTP.

IV.1.4 Kebutuhan fungsionalitas Perangkat Lunak

IV.1.4.1 Use Case Diagram

Gambar 4.2 merupakan Use case diagram kebutuhan fungsionalitas yang dibutuhkan dalam perancangan perangkat lunak sistem pakar yang digunakan untuk mendiagnosa penyakit anjing.



Gambar 4.2 Use Case Diagram SiPaPA

IV.1.5 Spesifikasi Rinci Kebutuhan Fungsionalitas

IV.1.5.1 Use case Spesification : Login

1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah id unik yaitu *username* dan *password* yang berupa rangkaian karakter.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan *username* dan *password*.
4. Sistem memeriksa *username* dan *password* yang diinputkan aktor.
E-1 *username* atau *password* aktor tidak sesuai.
5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Username atau Password tidak sesuai.

1. Sistem menampilkan peringatan bahwa *username* atau *password* tidak sesuai.
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

none

8. PostConditions

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem

IV.1.5.2 Use case Spesification : Edit Penyakit

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan perubahan terhadap data-data penyakit anjing.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola penyakit.
2. Sistem menampilkan antarmuka pilihan untuk melakukan edit data penyakit.
3. Aktor memilih untuk melakukan edit data penyakit.
4. Aktor mengubah data penyakit seperti deskripsi penyakit dan deskripsi pencegahan.
5. Aktor meminta sistem untuk menyimpan data penyakit yang telah diubah.
6. Sistem mengecek data penyakit yang telah diubah.
E-1 Data penyakit yang diubah aktor salah.
7. Sistem menyimpan data penyakit ke database.
8. Use case selesai.

5. Alternative Flow

none

6. Error Flow

- E-1 Data penyakit yang diubah aktor salah.

1. Sistem menampilkan peringatan bahwa data penyakit yang diubah salah.
2. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data penyakit pada database telah *ter-update*.

IV.1.5.3 Use case Spesification : Kelola Informasi

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelola terhadap data-data informasi terbaru mengenai anjing. Aktor dapat melakukan tambah, ubah dan hapus data informasi.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola informasi.
2. Sistem menampilkan antarmuka pilihan untuk melakukan tambah, ubah, atau hapus data informasi.
3. Aktor memilih untuk melakukan tambah data informasi.
A-1 Aktor memilih untuk melakukan ubah data informasi.
A-2 Aktor memilih untuk melakukan hapus data informasi.

4. Aktor menginputkan data informasi, yaitu Judul serta isi/berita Informasi tersebut.
5. Aktor meminta sistem untuk menyimpan data informasi yang telah diinput.
6. Sistem menyimpan data informasi ke database.
7. Use case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan ubah data informasi.

1. Sistem menampilkan data informasi yang tersedia.
2. Aktor memilih informasi yang akan diubah.
3. Aktor mengubah data informasi.
4. Sistem menyimpan data informasi yang telah diubah.
5. Berlanjut ke Basic Flow langkah ke 7.

A-2 Aktor memilih untuk melakukan hapus data informasi.

1. Sistem menampilkan antarmuka untuk melakukan penghapusan data informasi.
2. Aktor memilih data informasi yang ingin dihapus.
3. Aktor meminta sistem untuk menghapus data informasi.
4. Sistem menghapus data informasi yang dipilih dari database.
5. Sistem menampilkan pesan bahwa data informasi telah dihapus.
6. Berlanjut ke basic Flow langkah ke 7.

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data informasi pada database telah ter-*update*.

IV.1.5.4 Use case Spesification : Edit User

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelola terhadap data user. Aktor dapat mengubah data user.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan ubah data user.
2. Sistem menampilkan antarmuka pilihan untuk melakukan ubah data user.
3. Aktor memilih untuk melakukan ubah data user.
4. Aktor mengubah data user dengan memasukan *password* lama, *password* baru, dan konfirmasi *password* baru.
5. Aktor meminta sistem untuk menyimpan data user yang telah diinput.
6. Sistem mengecek data user yang telah diinput.
E-1 *Password* lama yang diinput aktor salah.
7. Sistem menyimpan data user ke database.
8. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 *Password* lama yang diinput aktor salah.

1. Sistem menampilkan peringatan bahwa *Password* lama yang diinput aktor salah.
2. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data user pada database telah terupdate.

IV.1.5.5 Use case Spesification : Mengidentifikasi Penyakit.

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengidentifikasi penyakit anjing berdasarkan gejala yang timbul.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan identifikasi penyakit pada anjing berdasarkan gejala-gejala yang timbul.
2. Sistem menampilkan atau memberikan beberapa pertanyaan yang harus dijawab oleh aktor.
3. Aktor menjawab pertanyaan yang sesuai dengan gejala yang timbul pada anjing.

4. Sistem menampilkan hasil identifikasi penyakit yang didapat beserta pencegahan yang dapat dilakukan.

5. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Hasil identifikasi penyakit anjing dan cara pencegahan secara dini telah tertampil.

IV.1.5.6 Use case Spesification : Tampil Penyakit.

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat deskripsi penyakit anjing.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk memilih menu profile penyakit.

2. Sistem melakukan koneksi ke server untuk dapat mengambil data penyakit.

E-1 Koneksi gagal.

3. Sistem menampilkan list penyakit yang tersedia.

4. Aktor memilih penyakit yang akan dilihat deskripsinya.

5. Aktor meminta sistem menampilkan deskripsi dari penyakit.
6. Sistem menampilkan deskripsi dari penyakit yang dipilih oleh aktor.
7. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Koneksi gagal.

1. Sistem menampilkan pesan bahwa koneksi ke server gagal.

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Daftar penyakit anjing dan deskripsinya telah tampil.

IV.1.5.7 Use case Spesification : Tampil Informasi.

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat data informasi terbaru mengenai anjing.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk memilih menu Info Terbaru pada aplikias *mobile*.
2. Sistem melakukan koneksi ke server untuk dapat mengambil data informasi.

E-1 Koneksi gagal.

3. Sistem menampilkan Daftar Informasi yang tersedia.
4. Aktor memilih informasi yang akan di baca.
5. Aktor meminta sistem untuk menampilkan informasi.
6. Sistem menampilkan informasi yang telah dipilih oleh aktor.
7. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Koneksi gagal.

1. Sistem menampilkan pesan bahwa koneksi ke server gagal.

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Daftar informasi telah tampil.

IV.1.6 Analisis Mengidentifikasi Penyakit

Pada analisis identifikasi penyakit ini, sistem yang dibangun agar penerapan teori kedalam praktik program dapat sejalan. Sehingga hasil diagnosa dan perhitungan persentase kemungkinan jenis penyakit yang diderita tidak jauh berbeda jika dicek baik secara manual ataupun melalui program. Sistem pakar untuk mendiagnosa penyakit anjing berbasis mobile ini memberi solusi pencegahan berdasarkan penyakit yang didiagnosa serta persentase kemungkinan jenis penyakit dengan metode *forward chaining*.

Pada proses identifikasi penyakit, pengguna/user akan memilih jawaban ya atau tidak dari setiap

pertanyaan yang akan ditampilkan. Pada kasus ini akan diambil contoh proses identifikasi yang penyakit yang akan menampilkan pertanyaan oleh Sistem (P_1, P_2, \dots, P_n) dan jawaban *user/pengguna* (J_1, J_2, \dots, P_n) yaitu sebagai berikut :

P_1 : 'Apakah anjing anda tidak mau makan?'

J_1 : 'Ya'

Berdasarkan jawaban yang dipilih user 'Ya', maka sistem akan menampilkan pertanyaan atas penyakit yang gejalanya telah terpenuhi sesuai dengan pertanyaan P_1 . Dan jika jawaban yang dipilih user 'Tidak' maka sistem akan menampilkan pertanyaan yang tidak memiliki gejala tersebut. Dalam kasus ini, yang memiliki gejala 'tidak mau makan' yaitu penyakit distemper, pneumonia, parvovirus, leptospirosis, dan rabies. Dan yang tidak memiliki gejala tersebut yaitu hepatitis, cacangan dan dermatitis. Dengan demikian sistem akan menampilkan pertanyaan yang mengacu kepada jenis penyakit yang memenuhi gejala tersebut, yaitu sebagai berikut :

P_2 : 'Apakah keluar cairan kental dari hidung anjing?'

J_2 : 'Ya'

Berdasarkan jawaban yang dipilih user 'Ya', maka sistem akan menampilkan pertanyaan atas penyakit yang gejalanya telah terpenuhi sesuai dengan pertanyaan P_2 . Dan jika jawaban yang dipilih user 'Tidak' maka sistem akan menampilkan pertanyaan yang tidak memiliki gejala tersebut. Dalam kasus ini, yang memiliki gejala 'keluar cairan kental dari hidung' yaitu penyakit distemper dan pneumonia. Dengan demikian sistem akan menampilkan pertanyaan yang mengacu kepada jenis penyakit yang memenuhi gejala tersebut, yaitu sebagai berikut :

P3 : 'Apakah anjing anda sering kejang-kejang?'

J3 : 'Tidak'

Berdasarkan jawaban yang dipilih user 'Tidak' maka sistem akan menampilkan pertanyaan yang tidak memiliki gejala tersebut. Dalam kasus ini, yang tidak memiliki gejala 'kejang' yaitu penyakit pneumonia. Dengan demikian sistem akan menampilkan pertanyaan tentang penyakit pneumonia tersebut. Dengan demikian proses identifikasi penyakit telah berhasil dilakukan.

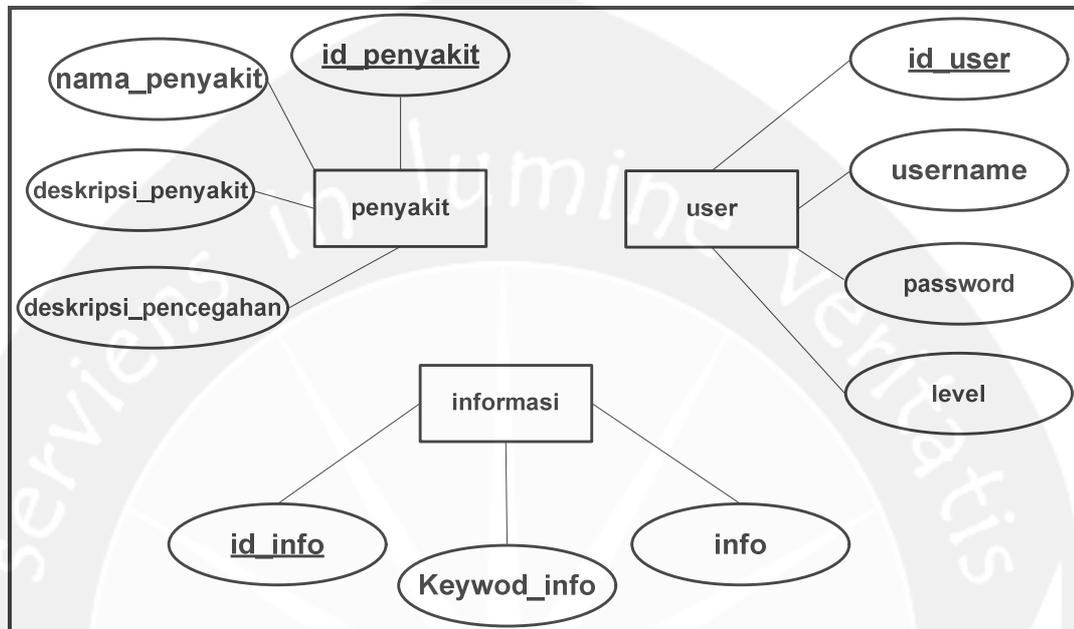
Dalam hal hasil presentase, diperoleh dengan menggunakan formula sebagai berikut :

$$= \frac{h}{h} \frac{h}{1} \times 100\%$$

Untuk penjelasan lebih lanjut, alur pertanyaan untuk melakukan proses identifikasi dapat dilihat pada lampiran.

IV.1.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram Sistem Pakar Penyakit Anjing dapat dilihat pada gambar 4.3.



Gambar 4.3 Entity Relationship Diagram (ERD)

Gambar 4.3 menggambarkan hubungan antar entitas yang dibutuhkan dalam memecahkan permasalahan pada perangkat lunak SiPaPA.

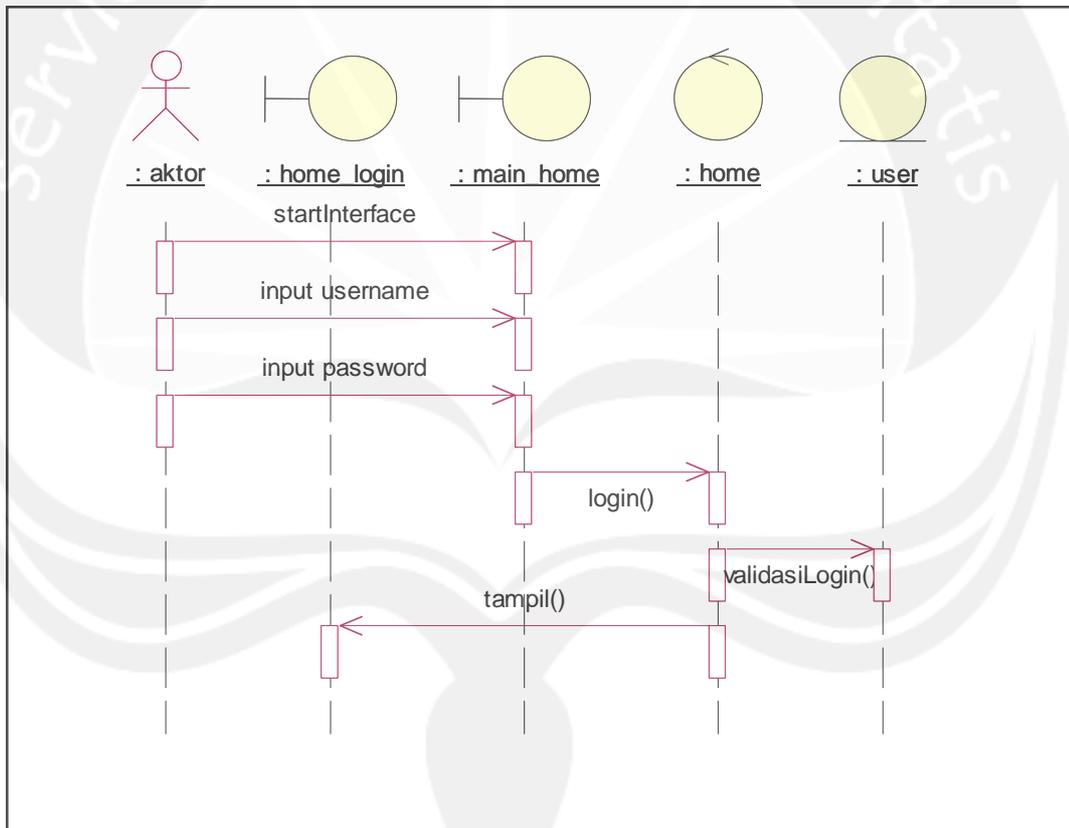
IV.2 Perancangan Perangkat Lunak

IV.2.1 Sequence Diagram

Untuk *Sequence Diagram* pada tahap perancangan dapat dilihat pada dokumen DPPL yang disertakan sebagai lampiran. Pada bab ini hanya akan ditampilkan *Sequence Diagram* untuk *Use Case* Login, Edit Penyakit, Mengidentifikasi Penyakit, dan Tampil penyakit.

IV.2.1.1 Login

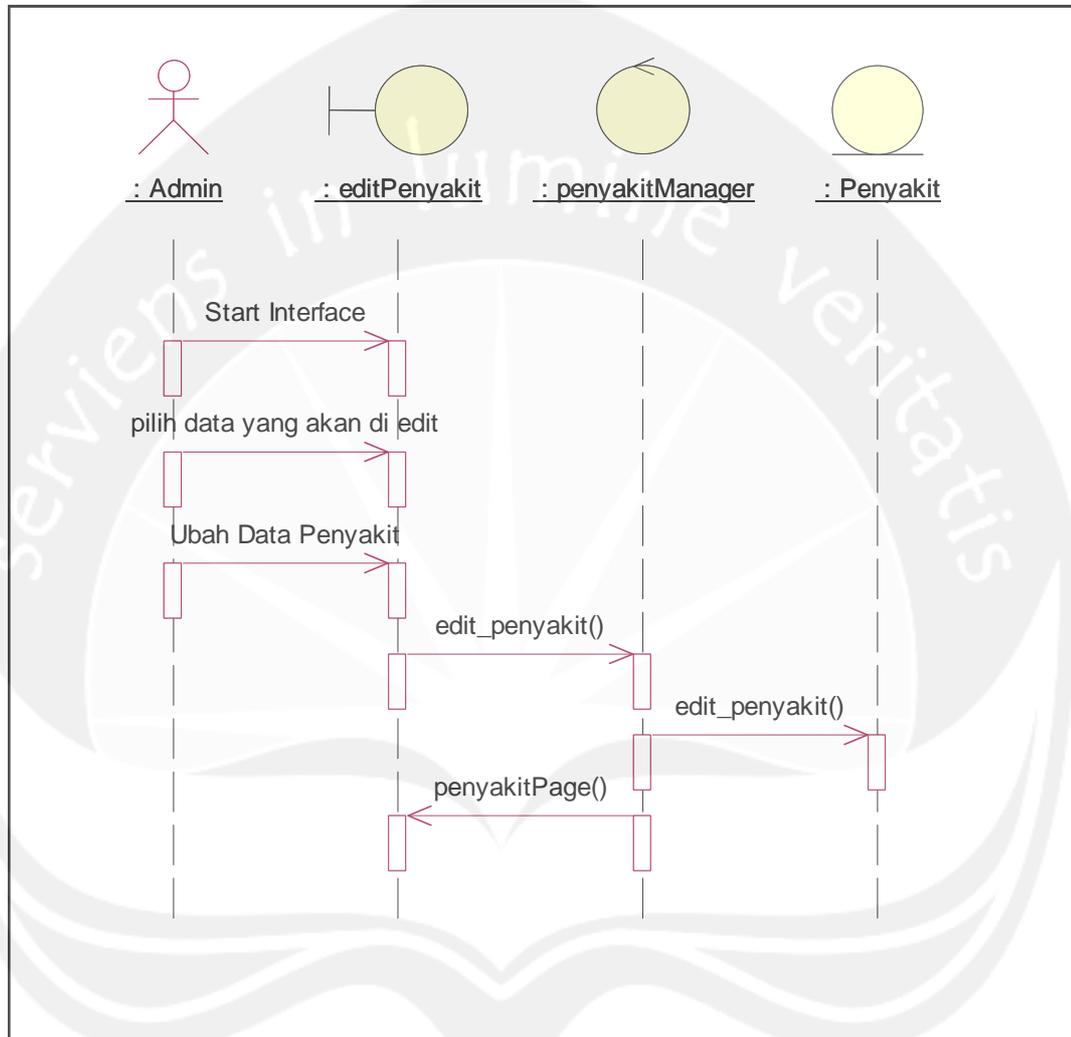
Sequence diagram untuk *use case login* dapat dilihat pada gambar 4.4.



Gambar 4.4 Sequence Diagram : Login

IV.2.1.2 Edit Penyakit

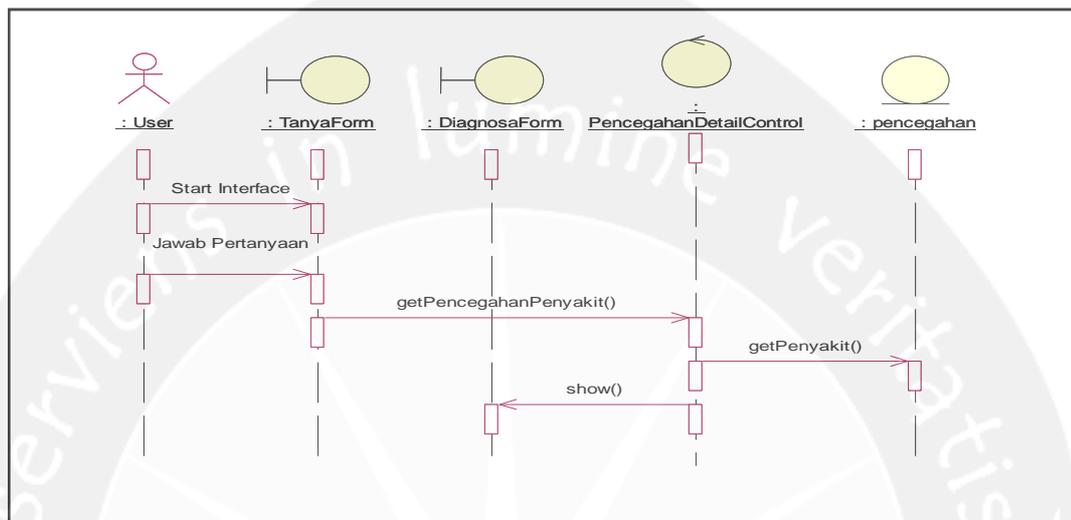
Sequence diagram untuk use case edit penyakit dapat dilihat pada gambar 4.5.



Gambar 4.5 Sequence Diagram : Edit Penyakit

IV.2.1.3 Mengidentifikasi Penyakit

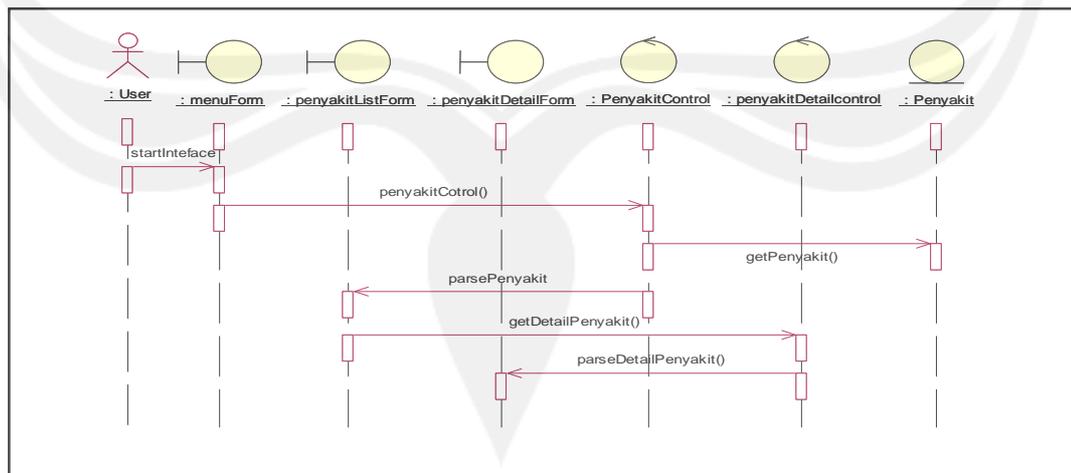
Sequence diagram untuk use case mengidentifikasi penyakit dapat dilihat pada gambar 4.6.



Gambar 4.6 Sequence Diagram : Mengidentifikasi Penyakit

IV.2.1.4 Tampil Penyakit

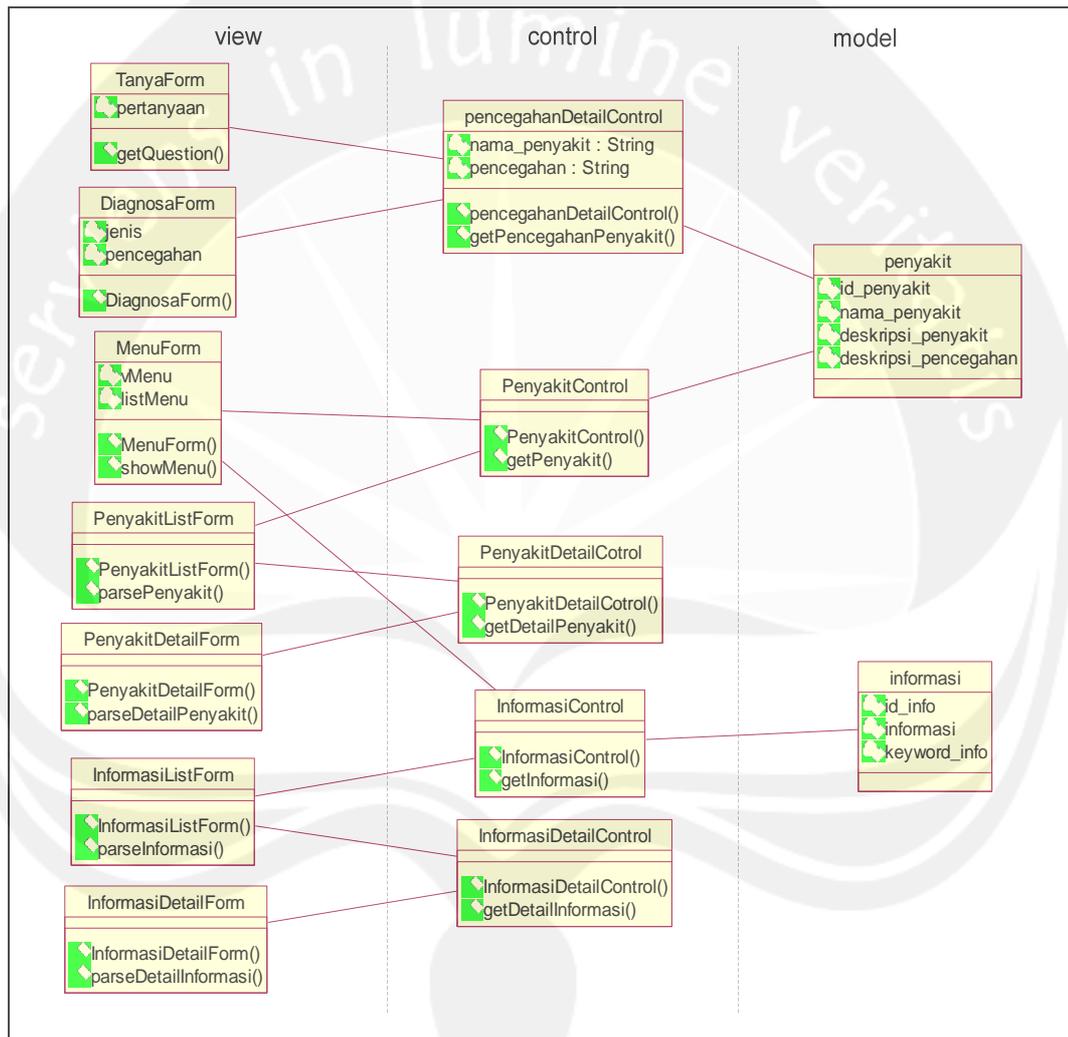
Sequence diagram untuk use case tampil penyakit dapat dilihat pada gambar 4.7.



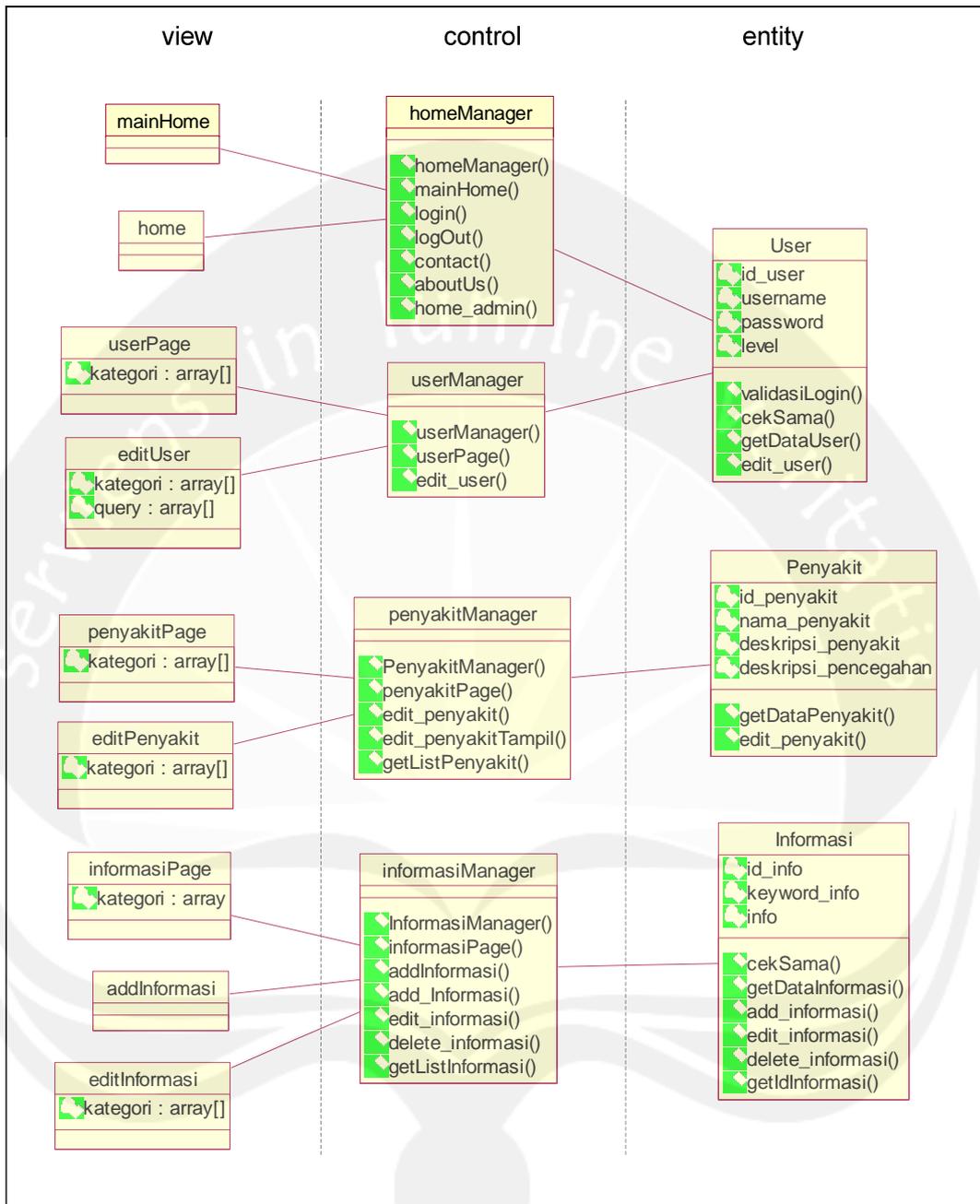
Gambar 4.7 Sequence Diagram : Tampil Penyakit

IV.2.2 Class Diagram

Gambar 4.8 merupakan gambar *class Diagram* pada aplikasi mobile **SiPaPA**. Sedangkan *class Diagram* untuk aplikasi web **SiPaPA** dapat dilihat pada gambar 4.9.



Gambar 4.8 Realisasi Class Diagram Aplikasi mobile SiPaPA



Gambar 4.9 Realisasi Class Diagram Aplikasi Web SiPaPA

IV.2.3 Class Diagram Spesific Description

Class Diagram Spesific Description perancangan perangkat lunak **SiPaPA** dapat dilihat pada dokumen DPPL yang disertakan sebagai lampiran. Pada bab ini akan ditampilkan beberapa contoh *Class Diagram* untuk penyakitPage, TanyaForm, PenyakitListForm dan PenyakitDetailForm.

IV.2.3.1 Spesific Design Class : TanyaForm

Tabel 4.1 merupakan spesifikasi antarmuka dari kelas TanyaForm yang digunakan untuk mengidentifikasi penyakit.

Tabel 4.1 Spesific design class TanyaForm

TanyaForm	<<boundary>>
+ TanyaForm() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+ getQuestion() : String Operasi ini digunakan untuk mendapatkan pertanyaan.	

IV.2.3.2 Spesific Design Class : PenyakitListForm

Tabel 4.2 merupakan spesifikasi antarmuka dari kelas PenyakitListForm.

Tabel 4.2 Spesific design class PenyakitListForm

PenyakitListForm	<<boundary>>
+ PenyakitListForm () Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	

+ parsePenyakit ()

Operasi ini digunakan untuk menampilkan data penyakit yang ada pada *database* server.

IV.2.3.3 *Specific Design Class* : PenyakitDetailForm

Tabel 4.3 merupakan spesifikasi antarmuka dari kelas PenyakitDetailForm.

Tabel 4.3 Specific design class PenyakitDetailForm

PenyakitDetailForm	<<boundary>>
+ PenyakitDetailForm () Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+ parseDetailPenyakit () Operasi ini digunakan untuk menampilkan deskripsi penyakit yang ada pada <i>database</i> server.	

IV.2.3.4 *Specific Design Class* : penyakitPage

Tabel 4.4 merupakan spesifikasi antarmuka dari kelas penyakitPage.

Tabel 4.4 Specific design class penyakitPage

penyakitPage	<<boundary>>
+penyakitPage() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+getDataPenyakit () Operasi ini digunakan untuk menampilkan semua data penyakit.	
+edit_penyakit ()	

Operasi ini digunakan untuk mengubah data penyakit lama menjadi data baru.

IV.2.4 Deskripsi Perancangan Antarmuka

Perancangan antarmuka ini memberikan gambaran bagaimana bentuk dari halaman antarmuka **SiPaPA** dan susunan dari kontrol yang dipakai sehingga dapat mudah digunakan dan dilihat oleh pengguna nantinya. Perancangan antarmuka ini disertai dengan penjelasan alur kegiatan. Untuk lebih lengkapnya, dapat melihat lampiran DPPL.

Gambar 4.10 merupakan rancangan antarmuka untuk halaman halaman menu pada aplikasi *mobile*. Pada antarmuka tersebut, diberi pilihan menu seperti identifikasi penyakit, profile penyakit, Informasi terbaru, tentang SiPaPA, bantuan, dan keluar. Jika memilih 'Identifikasi Penyakit' maka akan muncul antarmuka untuk mengidentifikasi penyakit. Jika memilih 'Profile Penyakit' maka akan muncul antarmuka untuk menampilkan penyakit. Jika memilih 'Info Terbaru' maka akan muncul antarmuka untuk menampilkan Informasi.



Gambar 4.10 Rancangan Antarmuka : Menu

Jika memilih 'Tentang SiPaPA' maka akan muncul antarmuka untuk menampilkan deskripsi mengenai perangkat lunak **SiPaPA**. Jika memilih 'Bantuan' maka

akan muncul antarmuka untuk menampilkan cara penggunaan perangkat lunak **SiPaPA**.

Antarmuka gambar 4.11 merupakan rancangan antarmuka dari TanyaForm yang digunakan untuk mengidentifikasi penyakit. Pada antarmuka ini, akan ditampilkan pertanyaan serta pilihan jawaban yang harus dijawab oleh user.

Pertanyaan No.pertanyaan	Pertanyaan No.pertanyaan
Isi pertanyaan yang akan di berikan	Isi pertanyaan yang akan di berikan
<input type="radio"/> Jawaban 1	<input type="radio"/> Jawaban 1
<input type="radio"/> Jawaban 2	<input type="radio"/> Jawaban 2
Lanjut	Lanjut
Menu	Hasil Diagnosa
	Kembali
	Menu Utama

Gambar 4.11 Rancangan Antarmuka : TanyaForm

Untuk memilih jawaban, maka dapat menekan salah satu *radiobutton*. Untuk ke pertanyaan selanjutnya maka tekan *commnad* 'Lanjut', tekan 'Hasil Diagnosa' untuk menampilkan hasil diagnosa, 'Kembali' untuk ke pertanyaan sebelumnya, serta 'Menu Utama' untuk kembali ke menu utama.

Antarmuka Gambar 4.12 merupakan rancangan antarmuka dari home_login pada aplikasi web. Antarmuka ini digunakan untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, user harus mengisikan username dan password dengan benar pada *textbox* yang telah disediakan. Pada saat tombol login diklik, sistem akan mengecek *username* dan *password* yang diisi dengan data *username* dan *password* yang telah tersimpan di *database*.

HEADER			
HOME	KONTAK	DOWNLOAD APLIKASI	TANTANG SiPaPA
Selamat Datang	Login		
<p>Selamat Datang di Situs Admin Sistem Pakar Identifikasi Penyakit Anjing.</p> <p>Sistem Pakar identifikasi Penyakit Anjing merupakan Sistem yang menangani</p>	<p>Login</p> <p>Username <input type="text"/></p> <p>Password <input type="password"/></p> <p>Login</p>		
FOOTER			

Gambar 4.12 Rancangan Antarmuka : home_login

Jika data *username* dan *password* benar maka *user* akan masuk ke dalam sistem, sebaliknya jika *username* dan *password* salah atau tidak cocok maka akan diberikan pesan kesalahan.

Antarmuka Gambar 4.13 merupakan rancangan antarmuka dari edit penyakit. Untuk melakukan perubahan data penyakit, admin dapat mengubah data penyakit seperti deskripsi penyakit dan deskripsi pencegahan, kemudian untuk menyimpan perubahan tersebut dengan mengklik button 'Ubah'.

HEADER	
HOME	EDIT PENYAKIT
KELOLA INFOMRASI	EDIT USER
KELUAR	
Ubah Penyakit	
Nama Penyakit	<input type="text"/>
Deskripsi Penyakit	<input type="text"/>
Deskripsi Pencegahan	<input type="text"/>
<input type="button" value="Ubah"/>	
FOOTER	

Gambar 4.13 Rancangan Antarmuka : editPenyakit

Antarmuka gambar 4.14 merupakan rancangan antarmuka dari addInformasi.

HEADER	
HOME	EDIT PENYAKIT
KELOLA INFORMASI	EDIT USER
KELUAR	
Tambah Informasi	
Judul	<input type="text"/>
Informasi	<input type="text"/>
<input type="button" value="Add"/>	
FOOTER	

Gambar 4.14 Rancangan Antarmuka : addInformasi

Untuk melakukan penambahan data informasi, admin harus mengisi data dari judul, serta detil informasi kemudian meng-klik *button Add*. Jika terjadi kesalahan, maka akan muncul pesan kesalahan, jika penambahan informasi berhasil maka akan muncul pesan berhasil.

Pada bab ini telah dijelaskan analisis perangkat lunak **SiPaPA**. Pada bab selanjutnya, akan menjelaskan pengkodean dan pengujian Perangkat Lunak yang akan mendeskripsikan bagian-bagian dalam aplikasi dan menganalisa apakah sistem aplikasi yang dibuat sudah memenuhi target yang ingin dicapai.