

BAB 6

KESIMPULAN

6.1. Kesimpulan

Berdasarkan penelitian yang sudah dilaksanakan dan berdasarkan hasil pembahasan pada bab-bab sebelumnya dapat ditarik kesimpulan yaitu :

1. Aplikasi Sistem pakar pendeteksi penyakit kulit telah berhasil dibangun.
2. Dengan Menggunakan metode *certainty factor* memungkinkan pakar mengestimasi kepercayaannya dalam konklusi.
3. Aplikasi ini mendapat hasil pengujian dan penilaian responden yaitu 4,67% mengatakan sangat baik, 36% mengatakan baik, 53,33% mengatakan cukup, dan 6% mengatakan kurang.

6.2. Saran

Saran yang dapat diberikan untuk mengembangkan aplikasi ini adalah diharapkan aplikasi ini dibuat lebih kompleks lagi agar dapat mencakup informasi yang mendetail serta aplikasi ini dapat digunakan pada semua smartphone untuk platform yang lainnya seperti : android, IOS, dll.

Daftar Pustaka

- Agustine, A. (2012). Aplikasi SMS Gateway Pada Rukun Tetangga Berbasis Website Menggunakan Framework CodeIgniter.
- Ardianto, W., Anggraeni, W., & Mukhlason, A. (2012). Pembuatan Sistem Pakar untuk Pendeteksian dan Penanganan Dini Pada Penyakit Sapi Berbasis Mobile Android Dengan Kajian Kinerja Teknik Knowledge Representation. *Jurnal Teknik ITS*, 1, 2301-9271.
- Arhami, M. (2004). *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi Offset.
- Bumi, M. A. (2010). Sistem Pakar Untuk Mendeteksi Penyakit Kulit dan Kelamin dengan Metode Forward Chaining.
- Djuanda, A., Hamzah, M., & Aisah, S. (2011). *Ilmu Penyakit Kulit dan Kelamin* (6 ed.). Jakarta: Fakultas Kedokteran Universitas Indonesia.
- Fadhilah A. N., Destiani D. & Dhamiri D. J., 2012. Perancangan Aplikasi Sistem Pakar Penyakit Kulit pada anak dengan metode Expert System Development Life Cycle. Volume 9, pp. 2302-7339.
- Hamdani. (2010). Sistem Pakar Untuk Diagnosa Penyakit Mata Pada Manusia. *Jurnal Informatika Mulawarman*, V(2).
- Hartati, S. I. (2008). *Sistem Pakar dan Pengembangannya* (1 ed.). Yogyakarta: Graha Ilmu.
- Hendra, H. (2011). *Pemrograman Piranti Bergerak*.
- Juanda. (2006). *Torch Akibat Dan Solusinya*. Solo: PT. Wangsa Jatra Lestari.
- Kusrini. (2006). *Sistem Pakar dan Teori Aplikasi*. Yogyakarta: Andi Offset.

- Kusumadewi, S. (2003). *Artificial Intellegence*. Yogyakarta: Graha Ilmu.
- Mahardhika, G. P. (2013). Perancangan Sistem Pakar Medis untuk Kasus Dermatomikosis Superfisialis. *Seminar Nasional Informatika Medis, IV*, 39.
- McLeon. (2008). *Sistem Informasi Manajemen*. Jakarta: Salemba Empat.
- Paryati. (2011). Sistem Pakar Berbasis Web untuk Mendiagnosa Penyakit Kulit. *Telematika, VII(2)*, 95-102.
- Pramudya, P. (2012). *Membuat Aplikasi Untuk Windows Phone*. Yogyakarta: Andi.
- Sadeli, M. (2009). *Membuat Sistem Pakar Sederhana Menggunakan Visual Basic*. Yogyakarta: Andi Offset.
- Turban, Aronson, & Liang, T. P. (2005). *Decision Support Systems And Intelligent Systems (7 ed.)*. Yogyakarta: Andi Offset.
- Turban, E., & Aronson, J. E. (2005). *Decision Support Systems and Intelligent Systems (6 ed.)*.
- Wijaya, S. (2012). Penerapan Web Service Pada Aplikasi Sistem Akademik pada Platform Sistem Operasi Mobile Android.

Nama :

Pekerjaan :

Di bawah ini beberapa pertanyaan mengenai "Aplikasi Sistem Pakar Identifikasi Penyakit Kulit Berbasis Windows Phone" :

. Berilah jawaban dengan memberikan tanda silang (X) sesuai dengan pendapat Anda.

- Pilihan jawaban SB = " Sangat Baik "
- Pilihan jawaban B = " Baik "
- Pilihan jawaban C = " Cukup "
- Pilihan jawaban K = " Kurang "
- Pilihan jawaban TB = " Tidak Baik"

1. Bagaimana tanggapan Anda mengenai tampilan Aplikasi Sistem Pakar ?

Uraian	Penilaian				
	SB	B	C	K	TB
Bentuk Tampilan					
Bahasa yang digunakan					
Kemudahan Penggunaan					

2. Bagaimana tanggapan Anda mengenai isi Aplikasi Sistem Pakar?

Uraian	Penilaian				
	SB	B	C	K	TB
Informasinya berguna					
Outputnya sudah benar					

3. Bagaimana tanggapan (saran dan kritik) Anda secara umum mengenai Aplikasi Sistem Pakar Idnetifikasi Penyakit Kulit ?

.....
.....
.....
.....
.....

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

**Sistem Pakar Pendeteksi Penyakit Kulit
(SIPAKIT)**

Untuk :


Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Bobby Hermawan / 100706188

**Program Studi Teknik Informatika
Fakultas Teknologi Industri**

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		SKPL-SIPAKIT		1/25
		Revisi		

Program Studi Teknik Informatika	DPPL – SIPAKIT	91/165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	BH							
Diperiksa oleh	SYT							
Disetujui oleh	SYT							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	96
1.1	Tujuan.....	96
1.2	Lingkup Masalah.....	96
1.3	Definisi, Akronim dan Singkatan.....	97
1.4	Referensi.....	98
1.5	Deskripsi umum (Overview).....	98
2	Deskripsi Kebutuhan	99
2.1	Perspektif produk.....	99
2.2	Fungsi Produk.....	100
2.3	Karakteristik Pengguna.....	101
2.4	Batasan-batasan.....	101
2.5	Asumsi dan Ketergantungan.....	102
3.1	Kebutuhan antarmuka eksternal.....	102
3.1.1	Antarmuka pengguna.....	102
3.1.2	Antarmuka perangkat keras.....	102
3.1.3	Antarmuka perangkat lunak.....	103
3.1.4	Antarmuka Komunikasi.....	104
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	104
3.2.1	Use Case Diagram.....	104
4.	Spesifikasi Rinci Kebutuhan	104
4.1	Spesifikasi Kebutuhan Fungsional.....	104
4.1.2	Use case Spesification : Identifikasi Penyakit 104	
4.1.3.	Use case Spesification : Login	106
4.1.4.	Use case Spesification : Pengelolaan Data Jenis Penyakit	107
4.1.5.	Use case Spesification : Pengelolaan Data Admin.....	108
4.1.6.	Use case Spesification : Pengelolaan Data Gejala.....	110



Daftar Gambar

Gambar 2.1 Arsitektur Perangkat Lunak SIPAKIT.....	10
Gambar 3.1 Use Case Diagram	14
Gambar 5.1 Entity Relationship Diagram (ERD)	23

1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak Sistem Pakar Pendeteksi Penyakit Kulit (SIPAKIT) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SIPAKIT ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak Sistem Pakar Pendeteksi Penyakit Kulit (SIPAKIT) dikembangkan dengan tujuan untuk :

4. Membantu user untuk menentukan jenis penyakit kulit.

Program Studi Teknik Informatika	DPPL – SIPAKIT	96/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Membantu user untuk menemukan solusi dari penyakit kulit.
6. Membantu user dalam memberikan informasi mengenai penyakit kulit.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-SIPAKIT-XXX	Kode yang merepresentasikan kebutuhan pada SIPAKIT (Sistem Pakar Pendeteksi Penyakit Kulit) dimana XXX merupakan nomor fungsi produk.
SIPAKIT	Perangkat lunak untuk mendeteksi penyakit kulit dengan metode forward chaining dan certainty factor berbasis windows phone 8.
Windows Phone	keluarga sistem operasi perangkat bergerak yang dikembangkan oleh Microsoft dan merupakan pengganti platform Windows Mobile.

Sistem Pakar	Sistem Pakar Merupakan sistem informasi yang berisi dengan pengetahuan dari pakar sehingga dapat digunakan untuk konsultasi.
Certainty Factor	Certainty Factor Merupakan suatu metode untuk membuktikan apakah suatu fakta itu pasti ataukah tidak pasti yang biasanya digunakan dalam Sistem Pakar.
Web based	Menyatakan aplikasi yang dalam penggunaannya pada komputer client dapat diakses mempergunakan web browser tanpa memerlukan proses instalasi dahulu.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Goza Mauser, Pembangunan Sistem pakar Penanggulangan Hama dan Penyakit pada Tanaman Kentang menggunakan Metode Certainty Factor, Universitas Atma Jaya Yogyakarta, 2012.
2. Silvester Dian Handy Permana, Pengembangan Aplikasi Pengenalan Pola Penyakit Kulit menggunakan Backpropagataion Momentum, Universitas Atma Jaya Yogyakarta, 2012.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Ketiga bagian tersebut dijelaskan dalam uraian di bawah ini.

Bagian utama berisi penjelasan mengenai dokumen SKPL ini. Di dalamnya dijelaskan tujuan pembuatan SKPL, ruang

lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SIPAKIT yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SIPAKIT tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SIPAKIT yang akan dikembangkan.

2 Deskripsi Kebutuhan

2.1 Perspektif produk

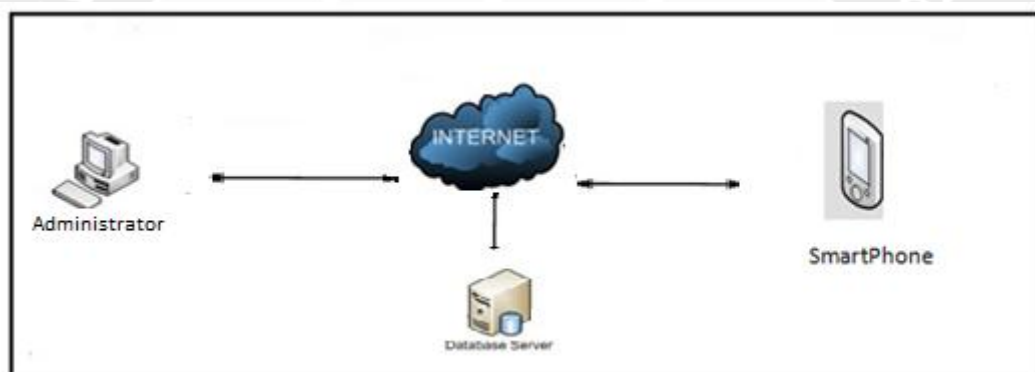
SIPAKIT merupakan perangkat lunak yang dikembangkan untuk membantu mendeteksi penyakit kulit dan memberikan informasi mengenai solusi dan penanganannya. Sistem ini membantu user dalam berkonsultasi mengenai penyakit kulit. Beberapa layanan yang disediakan sistem ini antara lain: mendeteksi penyakit kulit, memberikan solusi mengenai penyakit kulit, memberikan informasi mengenai penyakit kulit tersebut.

Perangkat lunak SIPAKIT dikembangkan dalam dua aplikasi yaitu berbasis *mobile* dan berbasis *web*. Aplikasi *mobile* perangkat lunak SIPAKIT berjalan pada platform *Windows Phone 8.0*. Sistem ini dibuat menggunakan bahasa pemrograman Microsoft Visual C#, dengan lingkungan pemrograman menggunakan *Microsoft Visual Studio 2012* dan Emulator *Windows Phone 8.0* sedangkan aplikasinya webnya diharapkan dapat berjalan di berbagai platform sistem operasi yang dilengkapi dengan browser sistem web ini dibuat

Program Studi Teknik Informatika	DPPL – SIPAKIT	99/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

dengan menggunakan codeigniter dan menggunakan bahasa pemrograman php dan html.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*) yang disediakan dalam aplikasi *mobile*. Arsitektur yang digunakan pada sistem ini, seperti terlihat pada gambar 2.1, berupa arsitektur Perangkat Lunak. Dimana semua data disimpan di server. User dapat mengakses data yang ada di server tersebut secara on-line dengan memanggil web service pada web site yang tersedia di web server.



Gambar 2.1 Arsitektur Perangkat lunak SIPAKIT

2.2 Fungsi Produk

Fungsi produk perangkat lunak SIPAKIT adalah sebagai berikut :

6. Fungsi Login (**SKPL-SIPAKIT-001**).

Fungsi ini merupakan fungsi yang digunakan login di website untuk melakukan pengolahan database.

7. Fungsi Pengelolaan Penyakit (**SKPL-SIPAKIT-002**).

Program Studi Teknik Informatika	DPPL – SIPAKIT	100/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Fungsi ini merupakan fungsi yang digunakan untuk melakukan pengelolaan database jenis penyakit kulit pada website.

8. Fungsi Pengelolaan Admin (**SKPL-SIPAKIT-003**).

Fungsi ini merupakan fungsi yang digunakan untuk melakukan pengelolaan database admin pada website.

9. Fungsi Pengelolaan Gejala (**SKPL-SIPAKIT-004**).

Fungsi ini merupakan fungsi yang digunakan untuk melakukan pengelolaan database Gejala pada website.

10. Fungsi Identifikasi Penyakit (**SKPL-SIPAKIT-005**).

Fungsi ini merupakan fungsi yang digunakan oleh aktor untuk mendeteksi penyakit kulit sesuai dengan inputan yang diberikan.

11. Fungsi Tampil Daftar Penyakit (**SKPL-SIPAKIT-006**).

Fungsi ini merupakan fungsi yang digunakan oleh aktor untuk melihat daftar penyakit kulit yang tersedia dalam sistem.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIPAKIT adalah sebagai berikut:

1. Memahami pengoperasian aplikasi SIPAKIT berbasis *Windows Phone 8.0*.
2. Mengerti pengoperasian *handphone windows phone*.
3. Dapat menggunakan internet.
4. Mengerti pengoperasian web browser.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIPAKIT tersebut adalah :

1. Kebijakanaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SIPAKIT.

2. Keterbatasan perangkat keras

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Perangkan lunak SIPAKIT dikembangkan sebagai aplikasi berbasis *mobile* serta berbasis web yang dapat dijalankan pada perangkat *smartphone* berbasis windows phone 8.0. Untuk dapat menggunakan semua fitur yang tersedia aplikasi ini membutuhkan koneksi internet, sedangkan untuk aplikasi berbasis web membutuhkan web browser yang tidak tergantung pada sistem operasi tertentu secara spesifik.

3. Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

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

3.1.1 Antarmuka pengguna

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form.

Program Studi Teknik Informatika	DPPL – SIPAKIT	102/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIPAKIT adalah:

3. *Smartphone* berbasis *Windows Phone 8.0*.
4. Perangkat komputer (PC desktop, laptop, netbook, dll).

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SIPAKIT adalah sebagai berikut :

7. Nama : Xampp
Sumber : Apache

Sebagai database management system (DBMS) yang digunakan untuk penyimpan data di web.

8. Nama : Windows Phone 8.0.
Sumber : Microsoft

Sebagai sistem operasi yang digunakan.

9. Nama : Microsoft Visual Studio 2012
Sumber : Microsoft

Sebagai tools pemrograman.

10. Nama : .NET Compact Framework 4.0
Sumber : Microsoft

Sebagai framework untuk menjalankan aplikasi.

11. Nama : Windows Phone SDK 8.0
Sumber : Microsoft

Sebagai emulator untuk menjalankan aplikasi.

12. Nama : Internet Explorer

Sumber : Microsoft

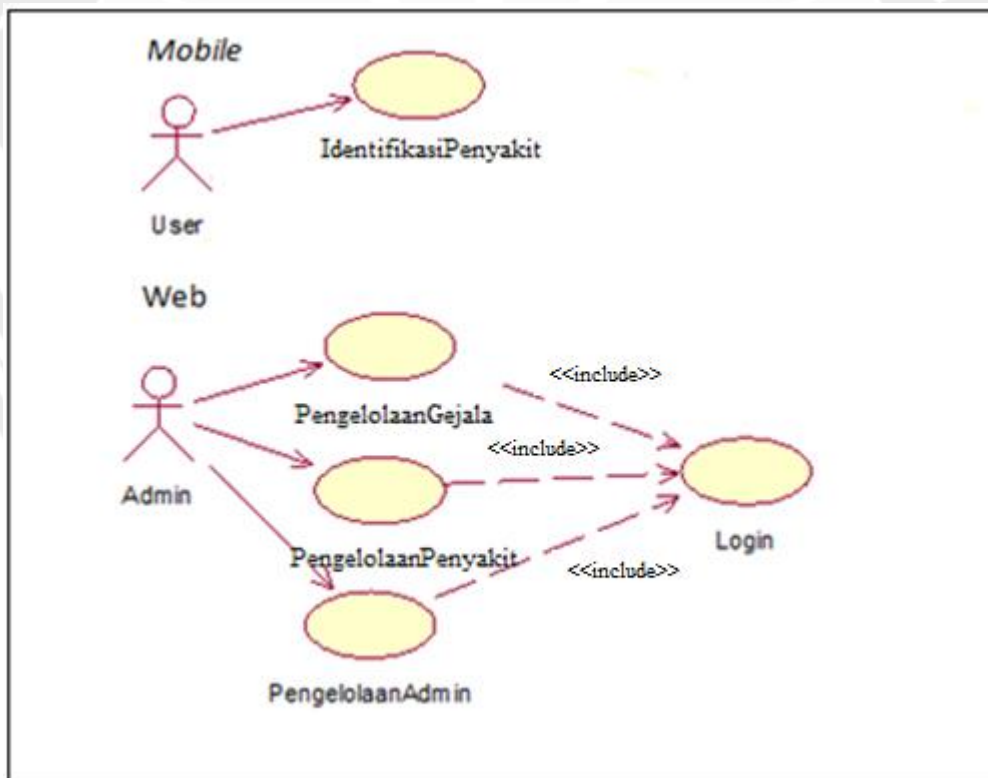
Sebagai web browser.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SIPAKIT menggunakan protocol TCP/IP.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 3.1 Use Case Diagram SIPAKIT

4. Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsional

4.1.2 Use case Specification : Identifikasi Penyakit

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengidentifikasi penyakit kulit sesuai dengan gejala yang dirasakan oleh aktor.

2. Primary Actor

User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan identifikasi penyakit kulit.
2. Sistem menampilkan antarmuka untuk identifikasi penyakit.
3. Sistem menampilkan Pertanyaan-pertanyaan kepada aktor.
4. Aktor menjawab pertanyaan yang diberikan oleh sistem.
5. Sistem memproses jawaban yang diberikan dari aktor.
6. Sistem menampilkan jenis penyakit berdasarkan jawaban dari aktor.
7. Aktor memilih informasi yang ingin diketahui lebih lanjut.

8. Sistem menampilkan informasi mengenai penyakit berdasarkan jawaban dari aktor.
9. Aktor memilih solusi untuk penyakit yang ditampilkan oleh sistem.
10. Sistem menampilkan solusi dari penyakit yang diderita oleh aktor.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

none

8. PostConditions

1. Aktor mengetahui jenis penyakit dan informasi penyakit yang sedang diderita.

Aplikasi Web

4.1.3. Use case Spesification : Login

1. Brief Description

Use case ini digunakan oleh aktor untuk mengakses sistem. Login berdasarkan id dan password yang unik.

2. Primary Actor

admin

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.

Program Studi Teknik Informatika	DPPL – SIPAKIT	106/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor memasukan id dan password.
4. Sistem memeriksa id dan password yang diinputkan aktor.

E-1 Password atau id admin tidak sesuai

5. Sistem memberikan akses ke aktor.
6. Use Case selesai.

5. Alternative Flow

None

6. Error Flow

E-1 Password atau nama admin tidak sesuai

1. Sistem menampilkan peringatan bahwa id admin atau password tidak sesuai.
2. Kembali ke basic flow langkah ke 3.
- 3.

7. PreConditions

None

8. PostConditions

1. Aktor telah memasuki sistem.

4.1.4. Use case Spesification : Pengelolaan Data Jenis Penyakit

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan data Jenis Penyakit.

2. Primary Actor

admin

3. Supporting Actor

None

4. Basic Flow

Program Studi Teknik Informatika	DPPL – SIPAKIT	107/ 165
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Jenis Penyakit.
2. Sistem menampilkan antarmuka untuk melakukan display data dan edit data Jenis Penyakit.
3. Aktor melakukan display data Jenis Penyakit.

A-1 Aktor memilih untuk melakukan edit data Jenis Penyakit.

4. Sistem menampilkan data Jenis Penyakit.
5. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data Jenis Penyakit.

1. Sistem menampilkan data Jenis Penyakit.
2. Aktor mengedit data Jenis Penyakit yang sudah ditampilkan.
3. Aktor meminta sistem untuk menyimpan data Jenis Penyakit yang telah diedit.
4. Sistem melakukan pengecekan terhadap data Jenis Penyakit yang telah diedit.
5. Sistem menyimpan data Jenis Penyakit yang telah diedit ke database.
6. Berlanjut ke Basic Flow langkah ke 5.

6. Error Flow

None

7. PreConditions

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

8. PostConditions

1. Data kategori di database telah terupdate.

4.1.5. Use case Spesification : Pengelolaan Data Admin

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan data admin.

2. Primary Actor

admin

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data admin.
2. Sistem menampilkan antarmuka untuk melakukan display data, insert, dan hapus data admin.
3. Aktor melakukan display data admin.
 - A-1 Aktor memilih untuk melakukan insert data admin.
 - A-2 Aktor memilih untuk melakukan hapus data admin.
4. Sistem menampilkan data admin.
5. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan insert data admin.
1. Sistem menampilkan data admin.
 2. Aktor menginputkan data admin.
 3. Aktor meminta sistem untuk menyimpan data admin yang telah diinput.
 4. Sistem melakukan pengecekan terhadap data admin yang telah diinput.
 - E-1 Data admin yang telah diinput salah.
 5. Sistem menyimpan data admin yang telah diinput ke database.

Program Studi Teknik Informatika	DPPL – SIPAKIT	109/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

6. Berlanjut ke Basic Flow langkah ke 5.

A-2 Aktor memilih untuk melakukan hapus data admin.

1. Sistem menampilkan data admin.
2. Aktor memilih untuk menghapus data admin.
3. Sistem menghapus data admin yang ada di database.
4. Berlanjut ke Basic Flow langkah ke 5.

6. Error Flow

E-1 Data Admin yang di inputkan salah

1. Sistem memberikan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Alternatif A-1 flow langkah ke 4.

7. PreConditions

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

8. PostConditions

1. Data admin di database telah terupdate.

4.1.6. Use case Spesification : Pengelolaan Data Gejala

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan data Gejala.

2. Primary Actor

admin

3. Supporting Actor

None

4. Basic Flow

Program Studi Teknik Informatika	DPPL – SIPAKIT	110/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Gejala.
2. Sistem menampilkan antarmuka untuk melakukan display data, insert, edit, dan hapus data Gejala.
3. Aktor melakukan display data Gejala.
 - A-1 Aktor memilih untuk melakukan insert data Gejala.
 - A-2 Aktor memilih untuk melakukan edit data Gejala.
 - A-3 Aktor memilih untuk melakukan hapus data Gejala.
4. Sistem menampilkan data Gejala.
5. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan insert data Gejala.
1. Sistem menampilkan data Gejala.
 2. Aktor menginputkan data Gejala yang sudah ditampilkan.
 3. Aktor meminta sistem untuk menyimpan data Gejala yang telah diinputkan.
 - E-1 Data Gejala yang telah diinput salah.
 4. Sistem menyimpan data Gejala yang telah diinputkan ke database.
 5. Berlanjut ke Basic Flow langkah ke 5.
- A-2 Aktor memilih untuk melakukan edit data Gejala.
1. Sistem menampilkan data Gejala.
 2. Aktor mengedit data Gejala.
 3. Aktor meminta sistem untuk menyimpan data Gejala yang telah diedit.

Program Studi Teknik Informatika	DPPL – SIPAKIT	111/165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Sistem melakukan pengecekan terhadap data Gejala yang telah diedit.
5. Sistem menyimpan data Gejala yang telah diedit ke database.
6. Berlanjut ke Basic Flow langkah ke 5.

A-3 Aktor memilih untuk melakukan hapus data Gejala.

1. Sistem menampilkan data Gejala.
2. Aktor memilih untuk menghapus data Gejala.
3. Sistem menyimpan data Gejala yang ada di database.
4. Berlanjut ke Basic Flow langkah ke 5.

6. Error Flow

E-1 Data Gejala yang di inputkan salah

1. Sistem memberikan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Alternatif A-2 flow langkah ke 4.

7. PreConditions

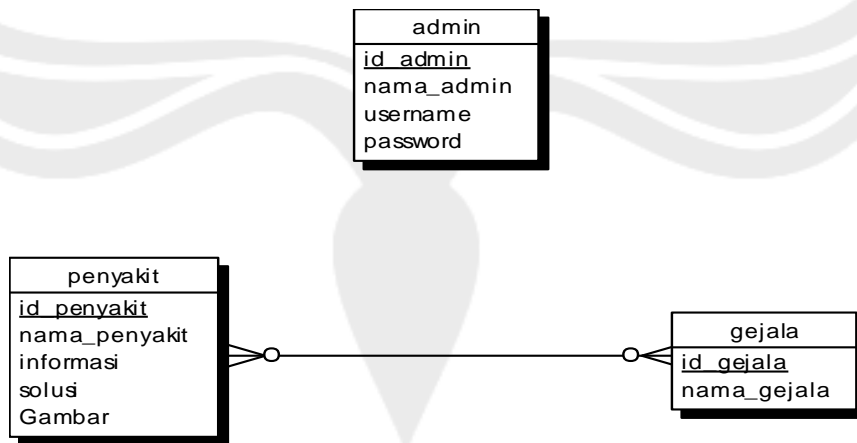
1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

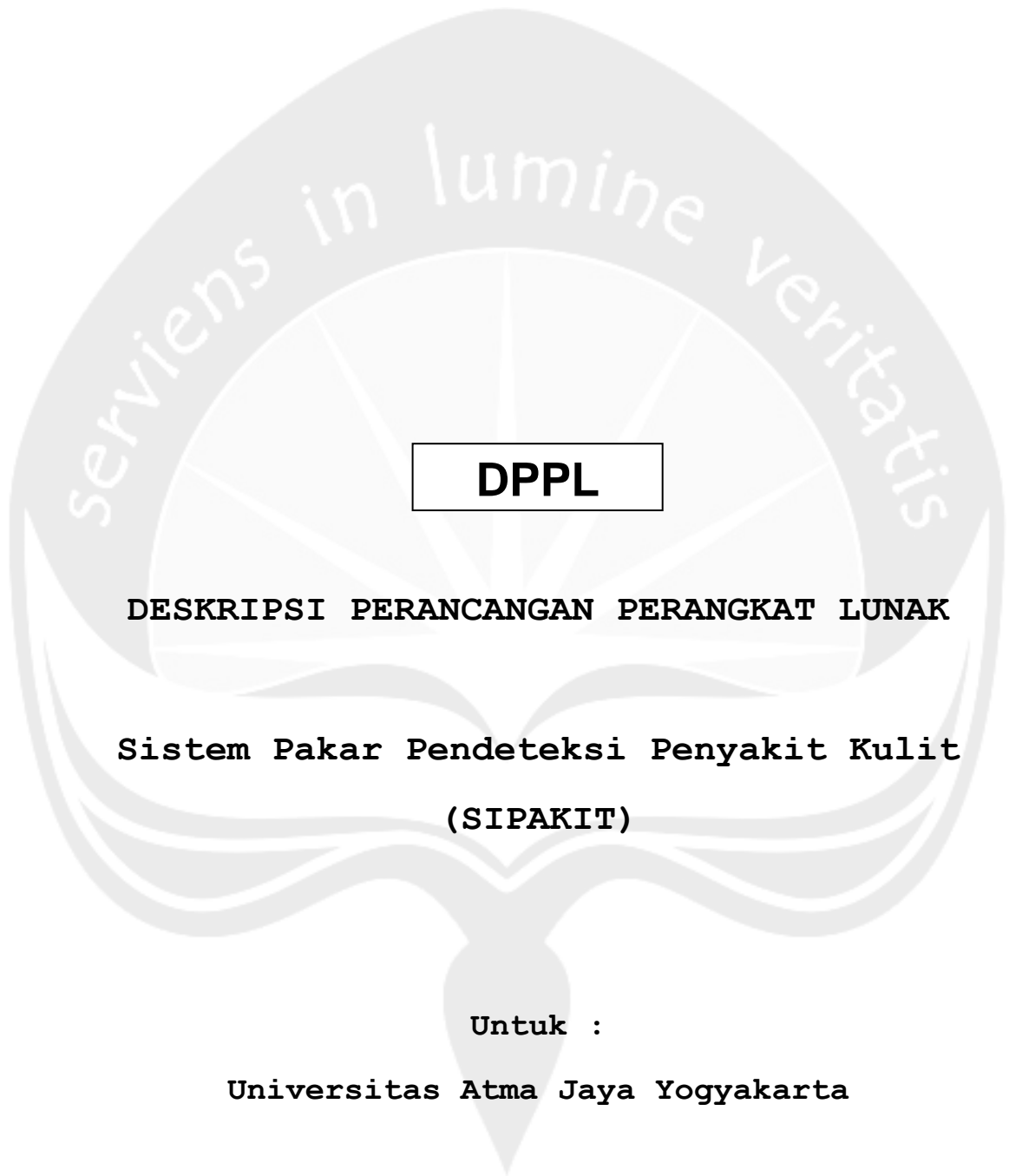
1. Data Gejala di database telah terinput.
2. Data Gejala di database telah terupdate.
3. Data Gejala di database telah terhapus.



5. Entity Relationship Diagram (ERD)



Gambar 5.1 Entity Relationship Diagram (ERD)



DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**Sistem Pakar Pendeteksi Penyakit Kulit
(SIPAKIT)**

Untuk :

Universitas Atma Jaya Yogyakarta


Dipersiapkan oleh:

Program Studi Teknik Informatika	DPPL – SIPAKIT	114/ 165
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Bobby Hermawan / 100706188

Program Studi Teknik Informatika
Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<i>DPPL-SIPAKIT</i>		
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX	-	A	B	C	D	E	F	G
TGL								

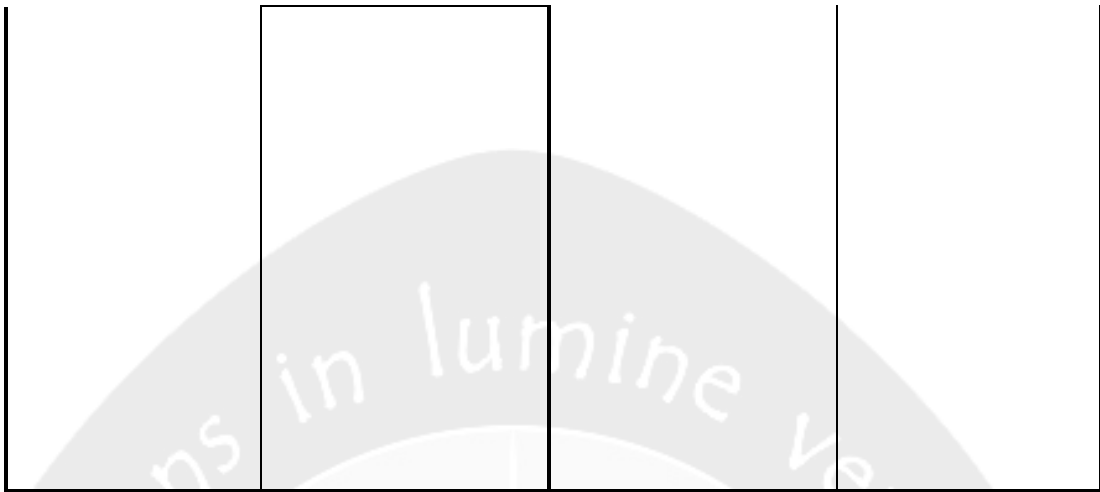
Program Studi Teknik Informatika	DPPL – SIPAKIT	115/ 165
----------------------------------	----------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

Ditulis oleh	BH							
Diperiksa oleh	SYT							
Disetujui oleh	SYT							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi



Daftar Isi

1	Pendahuluan	121
1.1	Tujuan	121
1.2	Ruang Lingkup	121
1.3	Definisi dan Akronim	121
1.4	Referensi	122
2	Analysis Model	123
2.1	Perancangan Arsitekur	123
2.2	Perancangan Rinci	124
2.2.1	Sequence Diagram	124
2.2.1.1	Identifikasi Penyakit	124
2.2.1.2	Login dan Ubah Password	125
2.2.1.2.1	Login	125
2.2.1.3	Pengelolaan Admin	125
2.2.1.3.1	Tambah Data Admin	125
2.2.1.3.2	Hapus Data Admin	126

2.2.1.4	Pengelolaan Penyakit	126
2.2.1.4.1	Ubah Data Penyakit	126
2.2.1.4.2	Tampil Data Penyakit	127
2.2.1.5	Pengelolaan Program Studi.....	127
2.2.1.5.1	Tambah Gejala	127
2.2.1.5.2	Ubah Data Gejala	127
2.2.1.5.3	Tampil Data Gejala	128
2.2.1.5.4	Hapus Data Gejala	129
2.2.2	Class Diagram	130
2.2.3	Class Diagram Specific Descriptions.....	80
2.2.3.1	Specific Design Class IdentifikasiUI	80
2.2.3.2	Specific Design Class PengelolaanAdminUI.....	80
2.2.3.3	Specific Design Class PengelolaanPenyakitUI	81
2.2.3.4	Specific Design Class PengelolaanGejalaUI	82
2.2.3.6	Specific Design Class IdentifikasiManager	82
2.2.3.7	Specific Design Class AdminManager.....	83
2.2.3.8	Specific Design Class PenyakitManager	84
2.2.3.9	Specific Design Class GejalaManager	84
2.2.3.11	Specific Design Class Identifikasi	85
2.2.3.12	Specific Design Class Admin	85
2.2.3.13	Specific Design Class Penyakit.....	86
2.2.3.14	Specific Design Class Gejala	86
3	Perancangan Data	87
3.1	Dekomposisi Data	87
3.1.1	Deskripsi Entitas Data Admin	87
3.1.2	Deskripsi Entitas Data Penyakit.....	87
3.1.3	Deskripsi Entitas Data Gejala	88

3.2	Physical Data Model	88
4.	Perancangan Antarmuka	89
4.1	Main Page	28
4.1.1	Menu	29
4.2	Identifikasi Penyakit	30
4.2.1	Informasi user	31
4.2.2	Deskripsi Pertanyaan	32
4.2.3	Aturan Informasi	33
4.2.4	Form Aturan	34
4.2.5	Tentang Kami	35
4.3	Pengelolaan Web	37
4.3.2	Pengelolaan Admin	38
4.3.3	Pengelolaan Penyakit	41
4.3.4	Pengelolaan Gejala	44

Daftar Gambar

Gambar 2.1 Arsitektur Aplikasi SIPAKIT	123
Gambar 2.2 Sequence Diagram : Identifikasi Penyakit	124
Gambar 2.3 Sequence Diagram : Login	125
Gambar 2.4 Sequence Diagram : Tambah Data Admin	125
Gambar 2.5 Sequence Diagram : Hapus Data Admin	126
Gambar 2.6 Sequence Diagram : Ubah Data Penyakit	126
Gambar 2.7 Sequence Diagram : Tampil Data Penyakit	127
Gambar 2.8 Sequence Diagram : Tambah Data Gejala	127
Gambar 2.9 Sequence Diagram : Ubah Data Gejala	128
Gambar 2.10 Sequence Diagram : Tampil Data Gejala	128
Gambar 2.11 Sequence Diagram : Hapus Data Gejala	129
Gambar 2.13 Class Diagram	131
Gambar 4.5 Rancangan Antarmuka Menu	89
Gambar 4.6 Rancangan Antarmuka Langkah Penggunaan Aplikasi	90
Gambar 4.7 Rancangan Antarmuka Bobot CF	91
Gambar 4.8 Rancangan Antarmuka Contoh Pengisian Gejala	92
Gambar 4.9 Rancangan Antarmuka Data user	93
Gambar 4.10 Rancangan Antarmuka Pengisian Gejala	94
Gambar 4.11 Antarmuka Hasil Identifikasi	95
Gambar 4.12 Rancangan Antarmuka Tampil Informasi	96
Gambar 4.13 Rancangan Antarmuka Daftar Penyakit	97
Gambar 4.14 Rancangan Antarmuka Login	98
Gambar 4.15 Rancangan Antarmuka Tampil Penyakit	99
Gambar 4.16 Rancangan Antarmuka Tambah Penyakit	100
Gambar 4.17 Rancangan Antarmuka Ubah Penyakit	101
Gambar 4.18 Rancangan Antarmuka Tampil Gejala	102
Gambar 4.19 Rancangan Antarmuka Tambah Gejala	103
Gambar 4.20 Rancangan Antarmuka Ubah Gejala	104
Gambar 4.21 Rancangan Antarmuka Tampil Admin	105
Gambar 4.22 Rancangan Antarmuka Tambah Admin	106
Gambar 4.23 Rancangan Antarmuka Ubah Admin	107

3 Pendahuluan

3.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

3.2 Ruang Lingkup

Perangkat Lunak Sistem Pakar Pendeteksi Penyakit Kulit (SIPAKIT) dikembangkan dengan tujuan untuk membantu orang dalam mendeteksi suatu penyakit kulit melalui inputan gejala dari user. Sistem ini juga memberikan informasi tentang penyakit kulit beserta dengan informasi Aturan yang harus dilakukan oleh user apabila terserang penyakit kulit dan juga pengolahan database berbasis website.

3.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

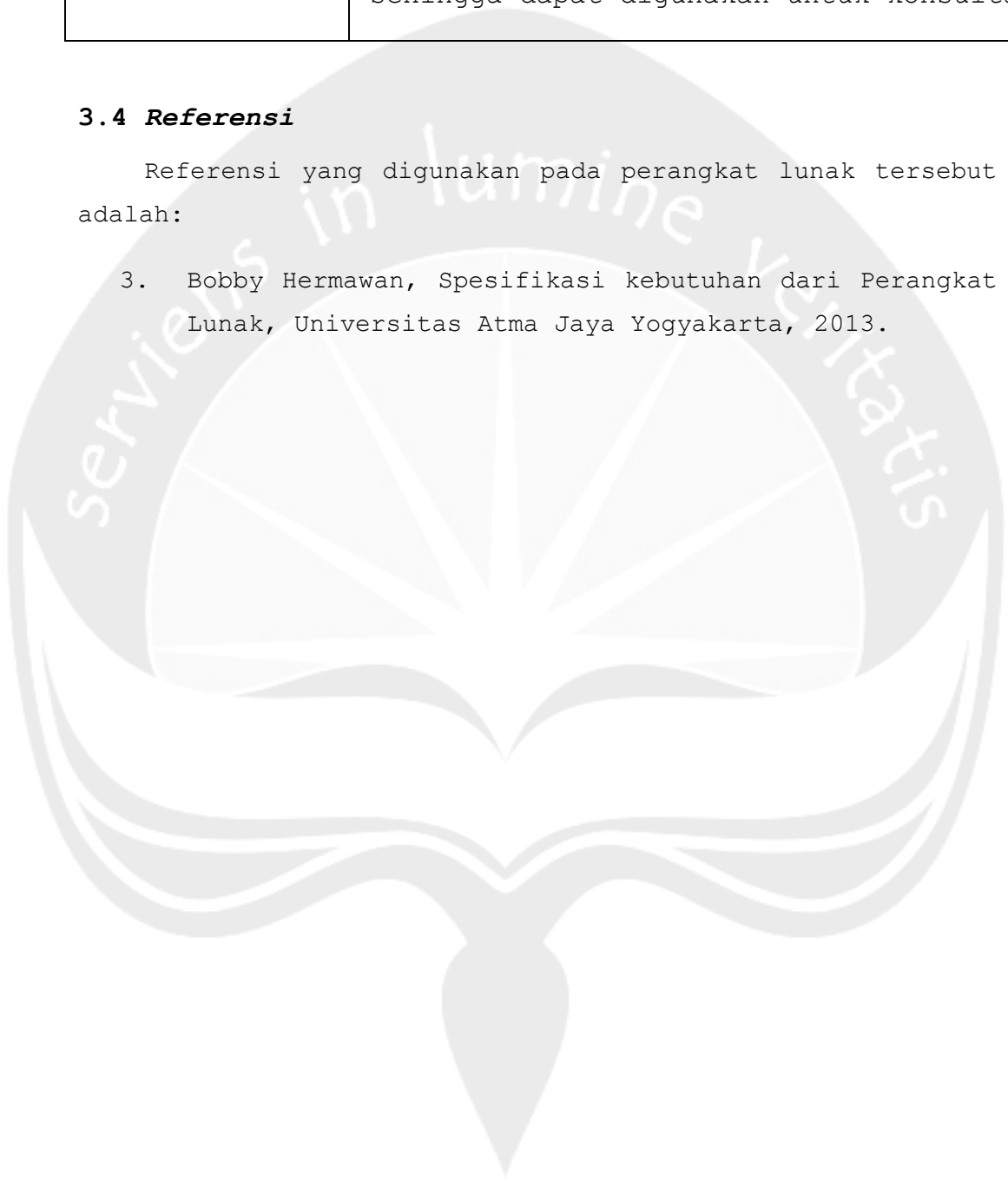
Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SIPAKIT	Sistem Pakar Pendeteksi Penyakit Kulit menggunakan metode Certainty Factor berbasis windows phone 8.
Windows Phone	keluarga sistem operasi perangkat bergerak yang dikembangkan oleh Microsoft dan merupakan pengganti platform Windows Mobile.

Sistem Pakar	Sistem Pakar Merupakan sistem informasi yang berisi dengan pengetahuan dari pakar sehingga dapat digunakan untuk konsultasi.
--------------	--

3.4 Referensi

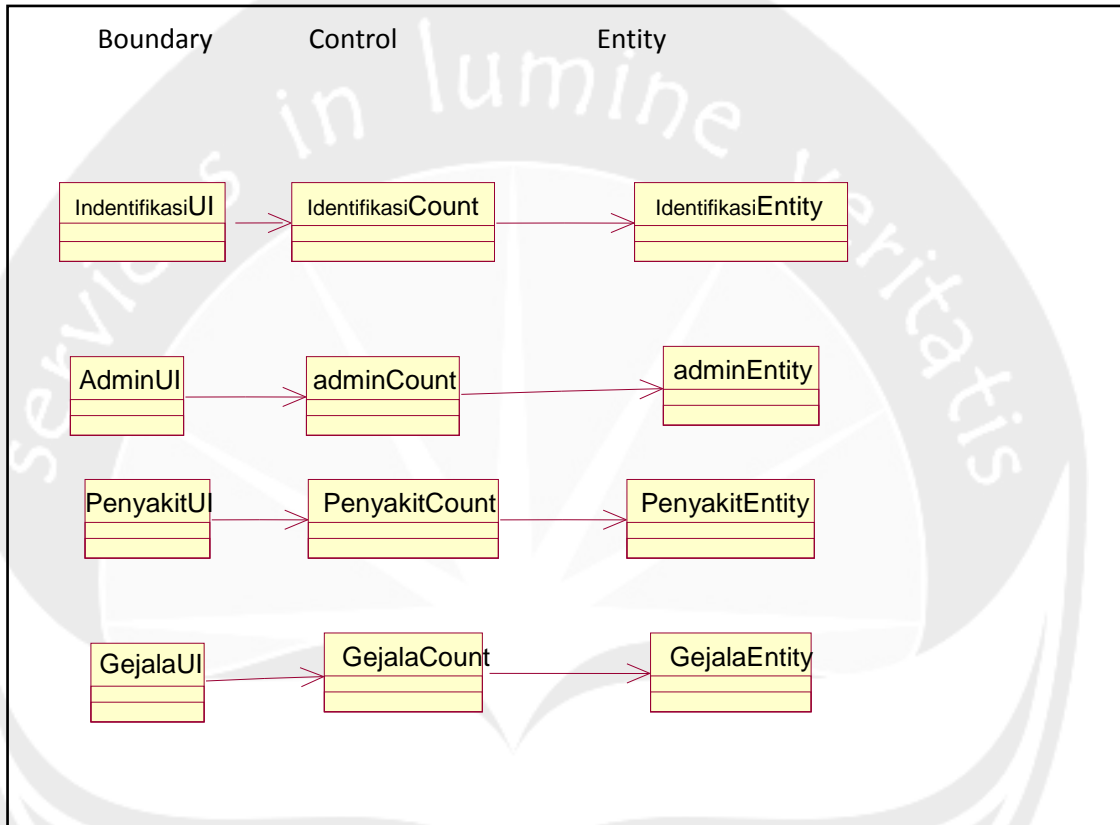
Referensi yang digunakan pada perangkat lunak tersebut adalah:

3. Bobby Hermawan, Spesifikasi kebutuhan dari Perangkat Lunak, Universitas Atma Jaya Yogyakarta, 2013.



4 Analysis Model

4.1 Perancangan Arsitekur

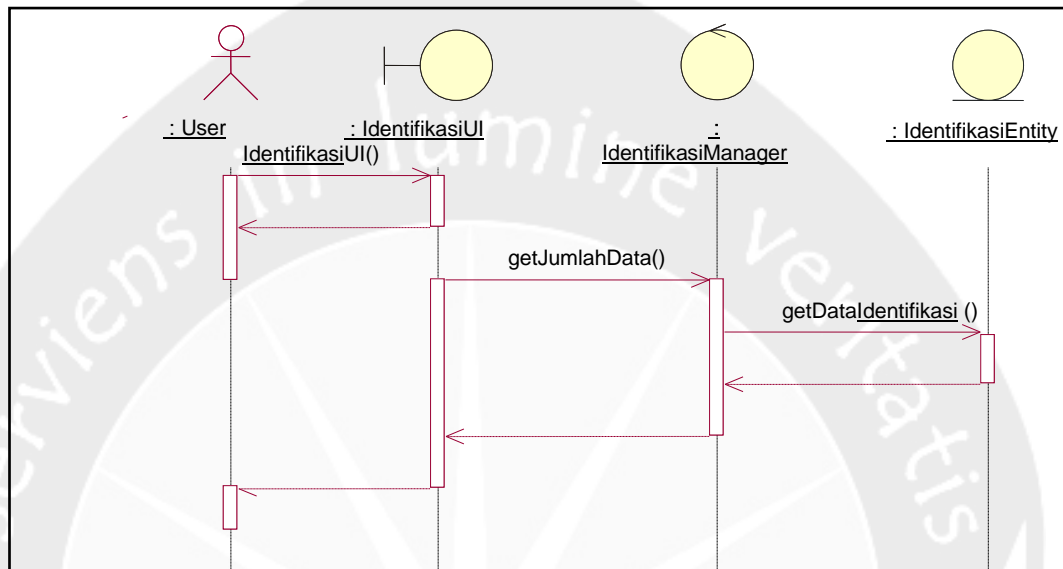


Gambar 2.1 Arsitektur Aplikasi SIPAKIT

4.2 Perancangan Rinci

4.2.1 Sequence Diagram

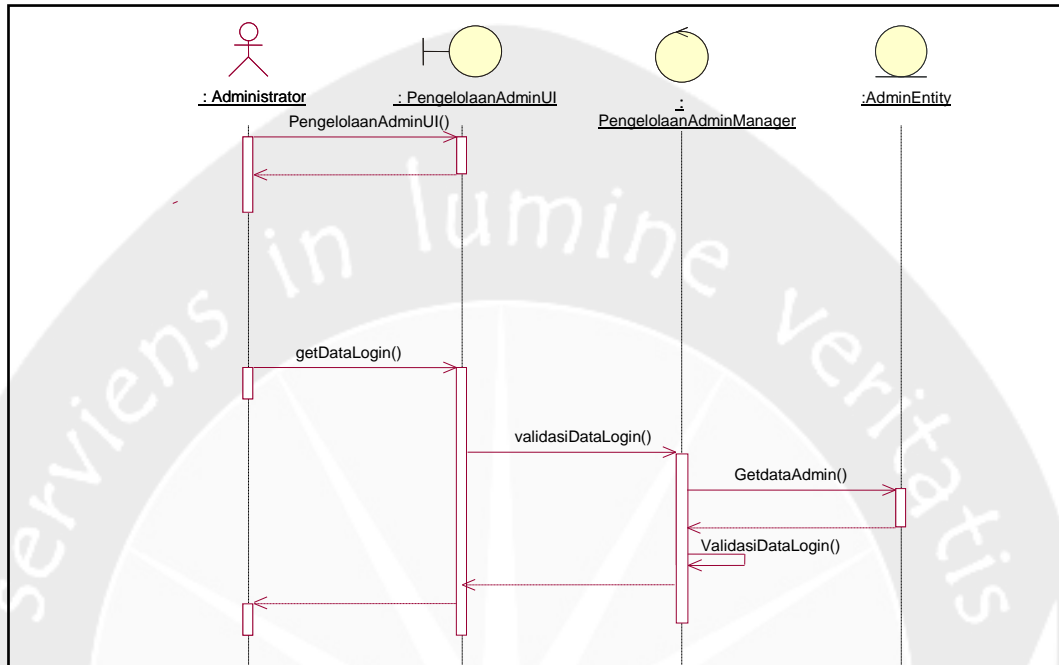
4.2.1.1 IdentifikasiPenyakit



Gambar 2.2 Sequence Diagram : Identifikasi Penyakit

4.2.1.2 Login

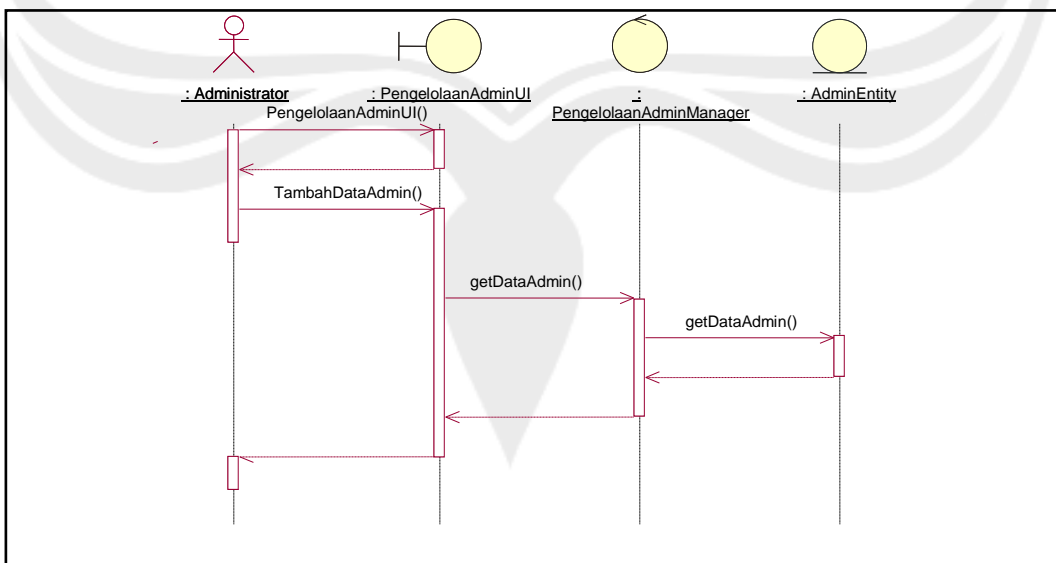
4.2.1.2.1 Login



Gambar 2.3 Sequence Diagram : Login

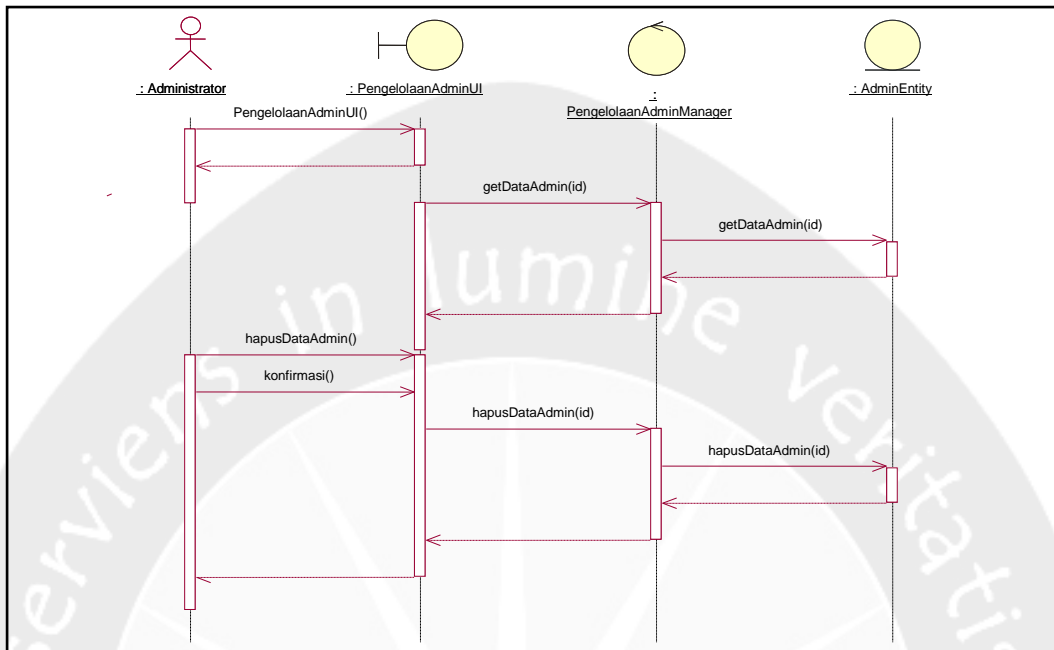
4.2.1.3 Pengelolaan Admin

4.2.1.3.1 Tambah Data Admin



Gambar 2.4 Sequence Diagram : Tambah Data Admin

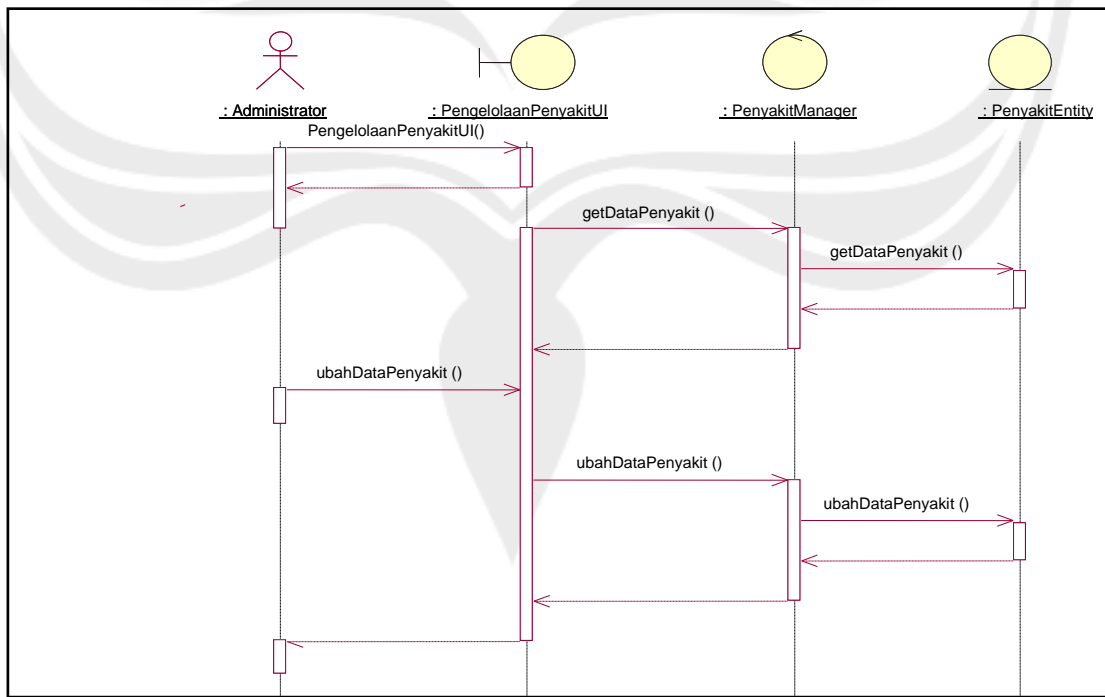
4.2.1.3.2 Hapus Data Admin



Gambar 2.5 Sequence Diagram : Hapus Data Admin

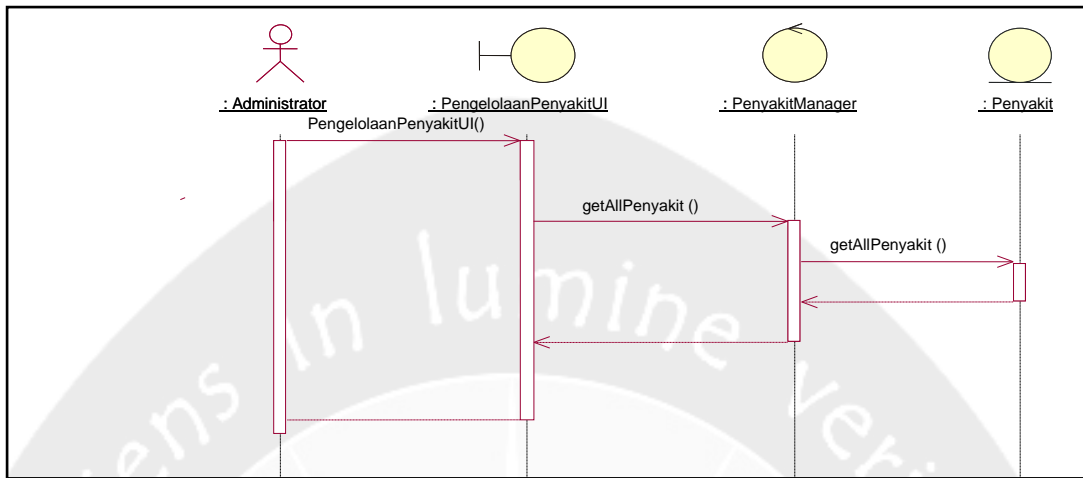
4.2.1.4 Pengelolaan Penyakit

4.2.1.4.1 Ubah Data Penyakit



Gambar 2.6 Sequence Diagram : Ubah Data Penyakit

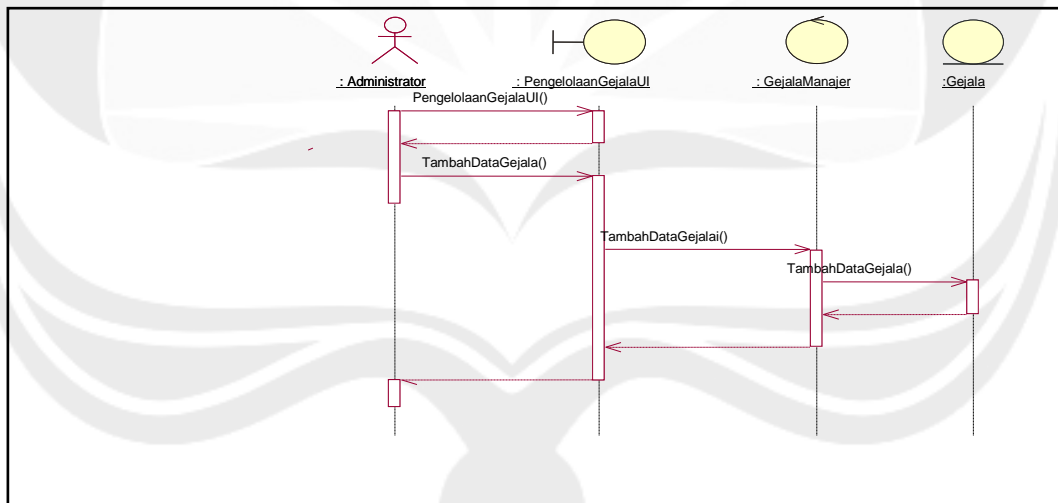
4.2.1.4.2 Tampil Data Penyakit



Gambar 2.7 Sequence Diagram : Tampil Data Penyakit

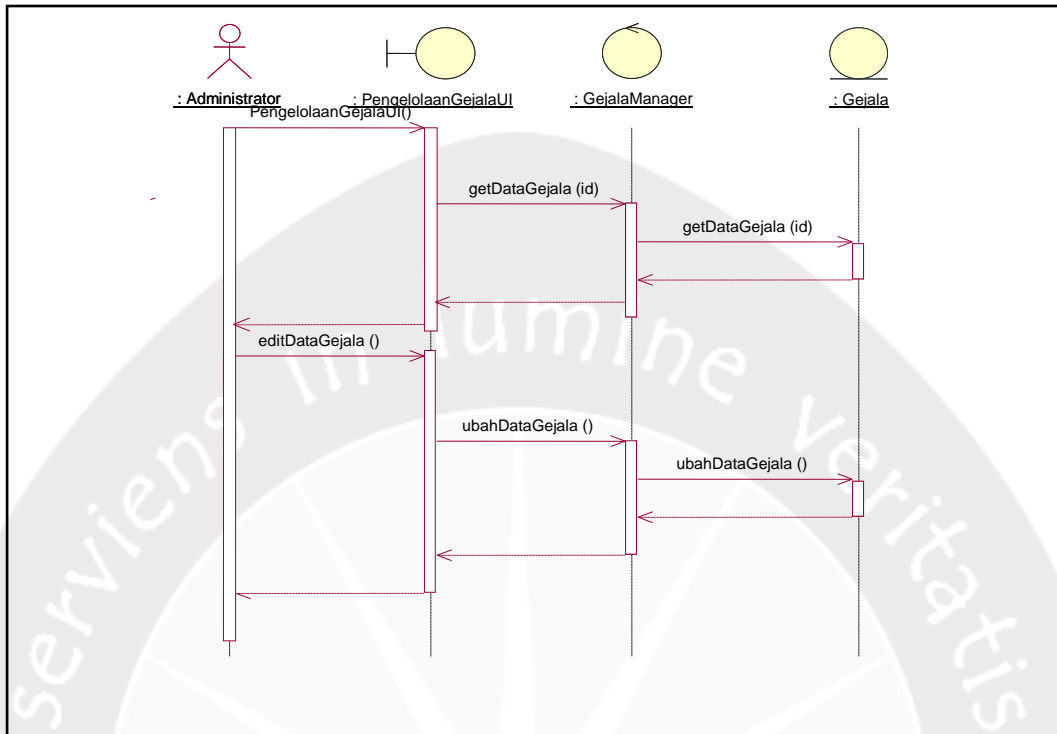
4.2.1.5 Pengelolaan Program Studi

4.2.1.5.1 Tambah Gejala



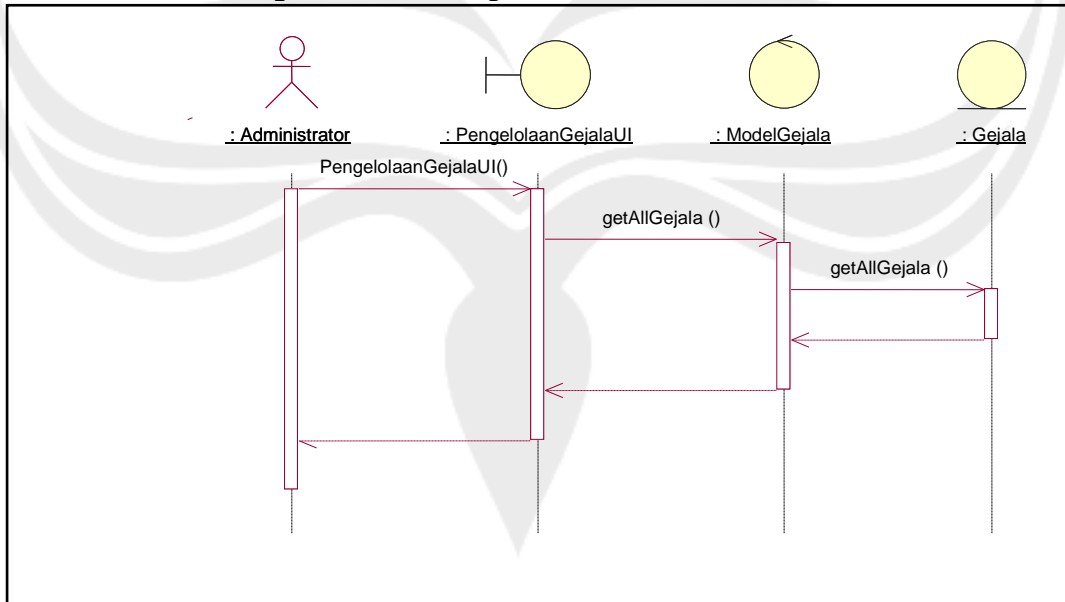
Gambar 2.8 Sequence Diagram : Tambah Data Gejala

4.2.1.5.2 Ubah Data Gejala



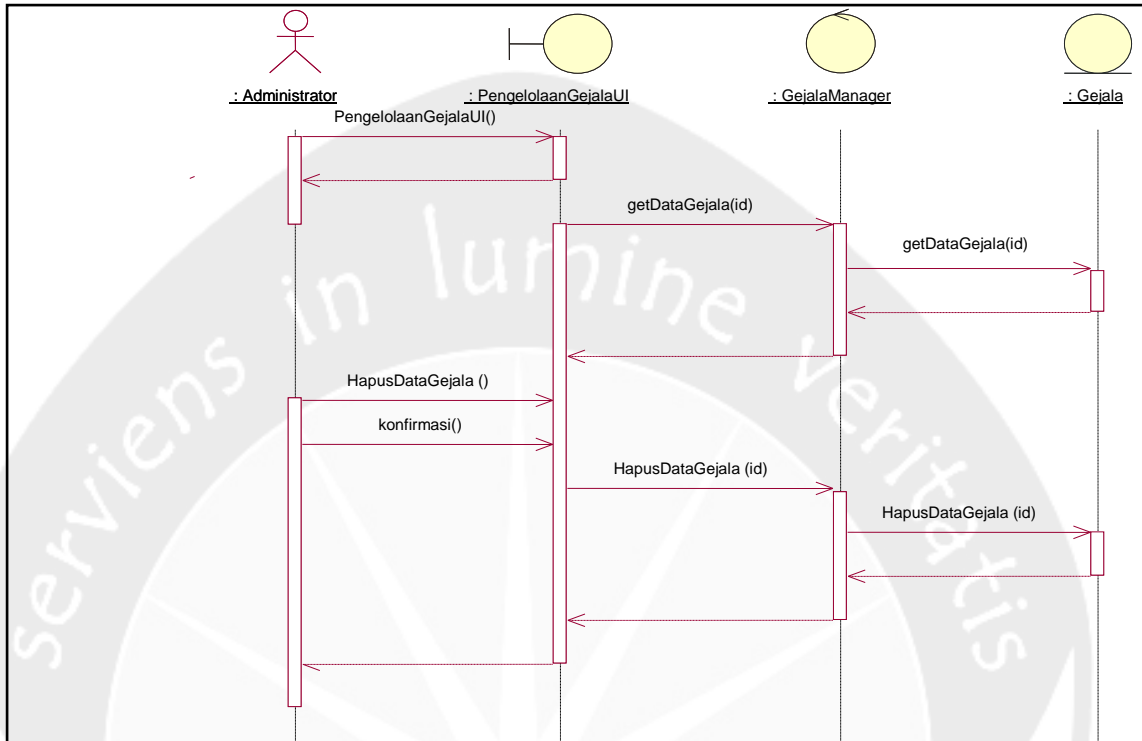
Gambar 2.9 Sequence Diagram : Ubah Data Gejala

4.2.1.5.3 Tampil Data Gejala



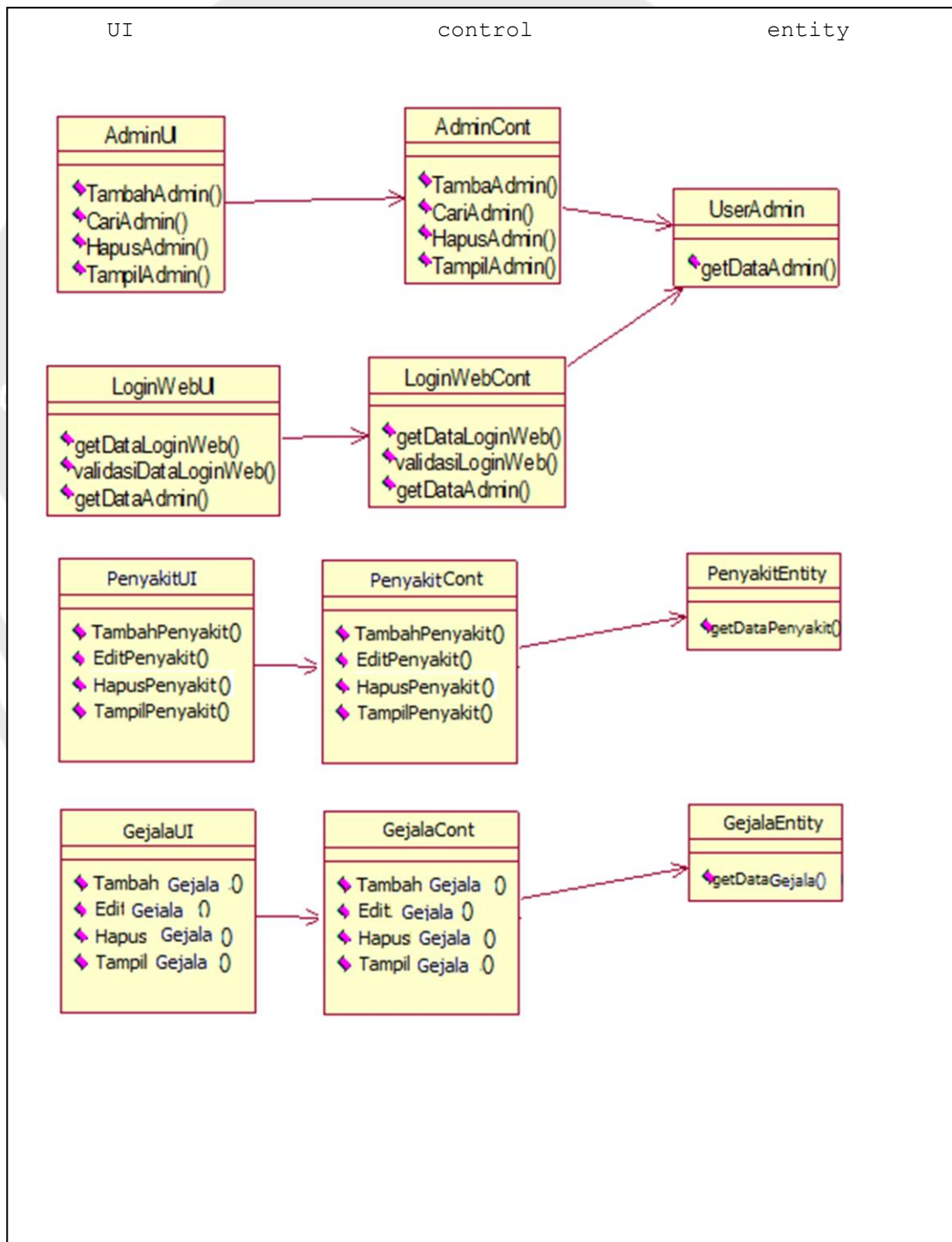
Gambar 2.10 Sequence Diagram : Tampil Data Gejala

4.2.1.5.4 Hapus Data Gejala



Gambar 2.11 Sequence Diagram : Hapus Data Gejala

4.2.2 Class Diagram





Gambar 2.13 Class Diagram

4.2.3 Class Diagram Specific Descriptions

4.2.3.1 Specific Design Class IdentifikasiUI

IdentifikasiUI()	<<boundary>>
+IdentifikasiUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+JumlahData() Operasi ini digunakan untuk menjumlah data Identifikasi yang telah di inputkan oleh user.	
+validasiDataIdentifikasi() Operasi ini digunakan untuk melakukan validasi terhadap input-input dari pengguna yang dimasukkan ke system.	
+getAllDataKategori():Kategori Operasi ini digunakan untuk mengembalikan semua data kategori yang ada dalam database.	
+getAllDataIdentifikasi():Identifikasi Operasi ini digunakan untuk mengembalikan semua data Identifikasi yang ada dalam database.	

4.2.3.2 Specific Design Class PengelolaanAdminUI

PengelolaanAdminUI()	<<boundary>>
+PengelolaanAdminUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+simpanDataAdmin() Operasi ini digunakan untuk mengambil data Admin yang telah tersimpan dalam data base.	
+validasiDataAdmin (nama):bool Operasi ini digunakan untuk melakukan validasi terhadap atribut nama dari Admin yang dimasukkan aktor, termasuk di	

dalamnya validasi terhadap field-field yang tidak dibolehkan kosong.

+hapusDataAdmin (id)

Operasi ini digunakan untuk menghapus data Admin yang ada dalam database.

+konfirmasi():bool

Operasi ini digunakan untuk meminta konfirmasi dari Admin sebelum melakukan penghapusan.

+getAllDataAdmin():User

Operasi ini digunakan untuk mengembalikan semua data Admin yang ada dalam database.

4.2.3.3 Specific Design Class PengelolaanPenyakitUI

PengelolaanPenyakitUI	<<boundary>>
<p>+PengelolaanPenyakitUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+simpanDataPenyakit()</p> <p>Operasi ini digunakan untuk untuk mengambil data model yang telah tersimpan dalam data base.</p> <p>+validasiDataPenyakit(nama):bool</p> <p>Operasi ini digunakan untuk melakukan validasi terhadap atribut nama dari penyakit yang dimasukkan aktor.</p> <p>+hapusDataPenyakit(id)</p> <p>Operasi ini digunakan untuk menghapus data penyakit yang ada dalam database.</p> <p>+konfirmasi():bool</p> <p>Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan.</p> <p>+getAllDataPenyakit():Penyakit</p> <p>Operasi ini mengembalikan data penyakit ke database.</p>	

4.2.3.4 Specific Design Class PengelolaanGejalaUI

PengelolaanGejalaUI	<<boundary>>
<p>+PengelolaanGejalaUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+simpanDataGejala() Operasi ini digunakan untuk mengambil data Gejala yang telah tersimpan dalam data base.</p> <p>+validasiDataGejala(nama):bool Operasi ini digunakan untuk melakukan validasi terhadap atribut nama dari gejala yang dimasukkan aktor.</p> <p>+hapusDataGejala(id) Operasi ini digunakan untuk menghapus data gejala yang ada dalam database.</p> <p>+cariDatagejala(id) Operasi ini digunakan untuk mencari data gejala yang ada dalam database.</p> <p>+konfirmasi():bool Operasi ini digunakan untuk meminta konfirmasi dari user sebelum melakukan penghapusan.</p> <p>+getAllDataGejala():Gejala Operasi ini digunakan untuk mengembalikan semua data gejala yang ada dalam database.</p>	

4.2.3.5 Specific Design Class IdentifikasiManager

IdentifikasiManager	<<control>>
<p>+ IdentifikasiManager() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getAllPenyakit(id) Operasi ini digunakan untuk mengembalikan semua data</p>	

penyakit yang ada dalam database.

+getAllGejala(id)

Operasi ini digunakan untuk mengembalikan semua data Gejala yang ada dalam database.

4.2.3.6 Specific Design Class AdminManager

AdminManager	<<control>>
+ AdminManager() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+TambahDataAdmin(admin) Operasi ini digunakan untuk memasukkan data Admin yang baru ke dalam database.	
+hapusDataAdmin(id) Operasi ini digunakan untuk menghapus satu data Admin dalam database.	
+getDataAdmin(id) Operasi ini digunakan untuk mengembalikan data Admin yang ada dalam database.	

4.2.3.7 Specific Design Class PenyakitManager

PenyakitManager	<<control>>
<pre>+ PenyakitManager() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +ubahDataPenyakit(id) Operasi ini digunakan untuk mengupdate data Penyakit ke dalam database. +getDataPenyakit(id) Operasi ini digunakan untuk mengembalikan data penyakit yang ada dalam database berdasarkan id penyakit. +getAllPenyakit(id) Operasi ini digunakan untuk mengembalikan semua data penyakit yang ada dalam database.</pre>	

4.2.3.8 Specific Design Class GejalaManager

GejalaManager	<<control>>
<pre>+ GejalaManager() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +TambahDataGejala (gejala) Operasi ini digunakan untuk memasukkan data model yang baru ke dalam database. +ubahDataGejala(id)</pre>	

Operasi ini digunakan untuk mengupdate data Gejala ke dalam database.

+hapusDataGejala(id)

Operasi ini digunakan untuk menghapus satu data Gejala dalam database.

+getDataGejala(id)

Operasi ini digunakan untuk mengembalikan data Gejala yang ada dalam database berdasarkan id Gejala.

+getAllGejala()

Operasi ini digunakan untuk mengembalikan semua data Gejala yang ada dalam database.

4.2.3.9 Specific Design Class Identifikasi

Identifikasi	<<entity>>
<p>+Identifikasi()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getpenyakit(id)</p> <p>Operasi ini digunakan untuk mengembalikan data pengguna yang ada dalam database.</p> <p>+getgejala(id)</p> <p>Operasi ini digunakan untuk mengembalikan data pengguna yang ada dalam database.</p>	

4.2.3.10 Specific Design Class Admin

Admin	<<entity>>
<p>+Admin()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+TambahDataAdmin(admin)</p>	

Operasi ini digunakan untuk memasukkan data admin yang baru ke dalam database.

+hapusDataAdmin(id)

Operasi ini digunakan untuk menghapus satu data admin dalam database.

+getDataAdmin(id)

Operasi ini digunakan untuk mengembalikan data admin yang ada dalam database.

4.2.3.11 Specific Design Class Penyakit

Kategori	<<entity>>
<p>+ Penyakit()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+ubahDataPenyakit(id)</p> <p>Operasi ini digunakan untuk mengupdate data penyakit ke dalam database.</p> <p>+getDataPenyakit(id)</p> <p>Operasi ini digunakan untuk mengembalikan data penyakit yang ada dalam database berdasarkan id penyakit.</p> <p>+getAllPenyakit(id)</p> <p>Operasi ini digunakan untuk mengembalikan semua data penyakit yang ada dalam database.</p>	

4.2.3.12 Specific Design Class Gejala

Gejala	<<entity>>
<p>+ Gejala()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+TambahDataGejala(Gejala)</p> <p>Operasi ini digunakan untuk memasukkan data Gejala yang baru ke dalam database.</p>	

+ubahDataGejala(id)

Operasi ini digunakan untuk mengupdate data Gejala ke dalam database.

+hapusDataGejala(id)

Operasi ini digunakan untuk menghapus satu data Gejala dalam database.

+getDataGejala(id)

Operasi ini digunakan untuk mengembalikan data Gejala yang ada dalam database berdasarkan id Gejala.

+getAllGejala()

Operasi ini digunakan untuk mengembalikan semua data Gejala yang ada dalam database.

5 Perancangan Data

5.1 Dekomposisi Data

5.1.1 Deskripsi Entitas Data Admin

Tabel 3.1 Deskripsi Entitas Data Admin

Nama	Tipe	Panjang	Keterangan
ID	Integer	-	Id Pengguna, Primary key
USERNAME	Varchar	25	Nama dari admin
PASSWORD	Varchar	15	Password dari admin
NAMA	Varchar	15	Nama dari admin

5.1.2 Deskripsi Entitas Data Penyakit

Tabel 3.2 Deskripsi Entitas Data Penyakit

Nama	Tipe	Panjang	Keterangan
ID_PENYAKIT	Char	4	Id Penyakit, Primary key

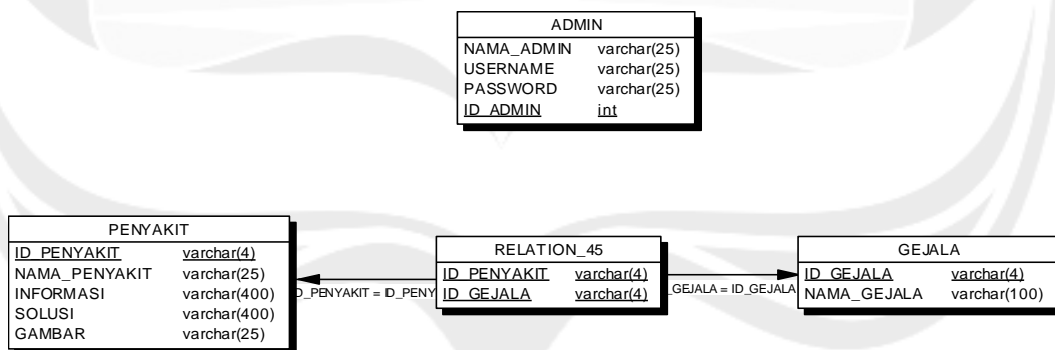
NAMA_PENYAKIT	Varchar	25	Nama dari Penyakit
INFORMASI	Varchar	400	Penjelasan Penyakit
SOLUSI	Varchar	400	Solusi dari Penyakit
GAMBAR	VARCHAR	25	Gambar dari Penyakit

5.1.3 Deskripsi Entitas Data Gejala

Tabel 3.3 Deskripsi Entitas Data Gejala

Nama	Tipe	Panjang	Keterangan
ID_GEJALA	Integer	-	Id Gejala, Primary key
JENIS_GEJALA	Varchar	255	Nama Gejala

5.2 Physical Data Model



4. Perancangan Antarmuka

4.1. Main Page

4.1.1. Menu

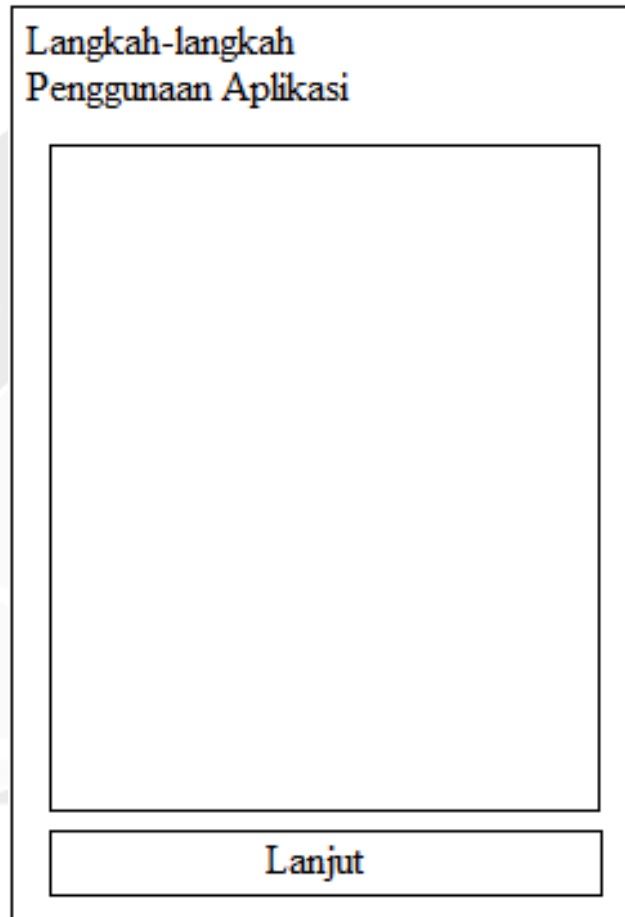


Gambar 4.5 Rancangan Antarmuka Menu

Antarmuka ini digunakan user untuk memilih fitur-fitur yang ada yang disediakan oleh sistem.

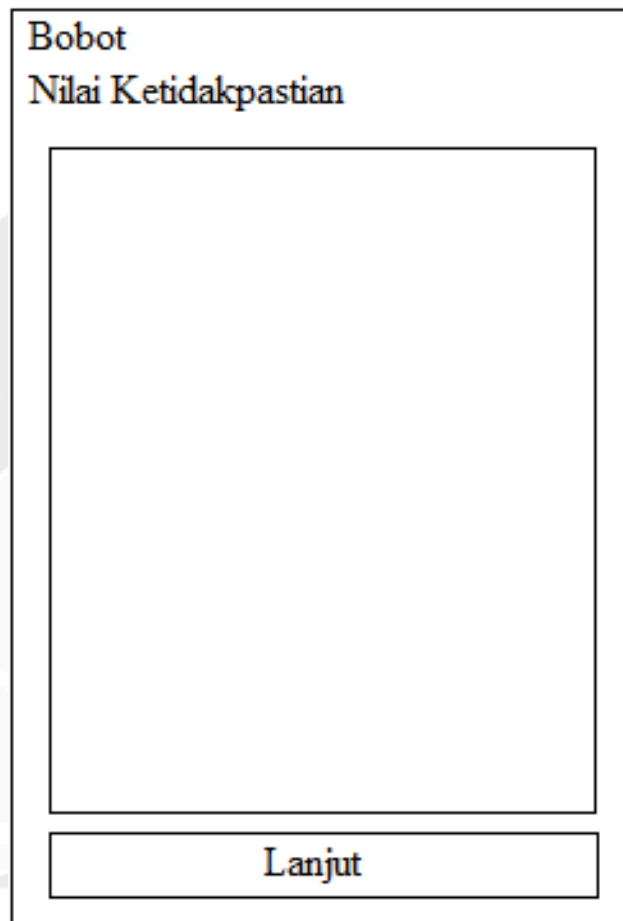
4.1.2. Identifikasi Penyakit

4.1.2.1. Langkah Penggunaan Aplikasi



Gambar 4.6 Rancangan Antarmuka Langkah Penggunaan Aplikasi
Antarmuka ini digunakan user untuk melihat dan mempelajari cara menggunakan sistem yang dibuat.

4.1.2.2. Bobot Nilai Ketidakpastian

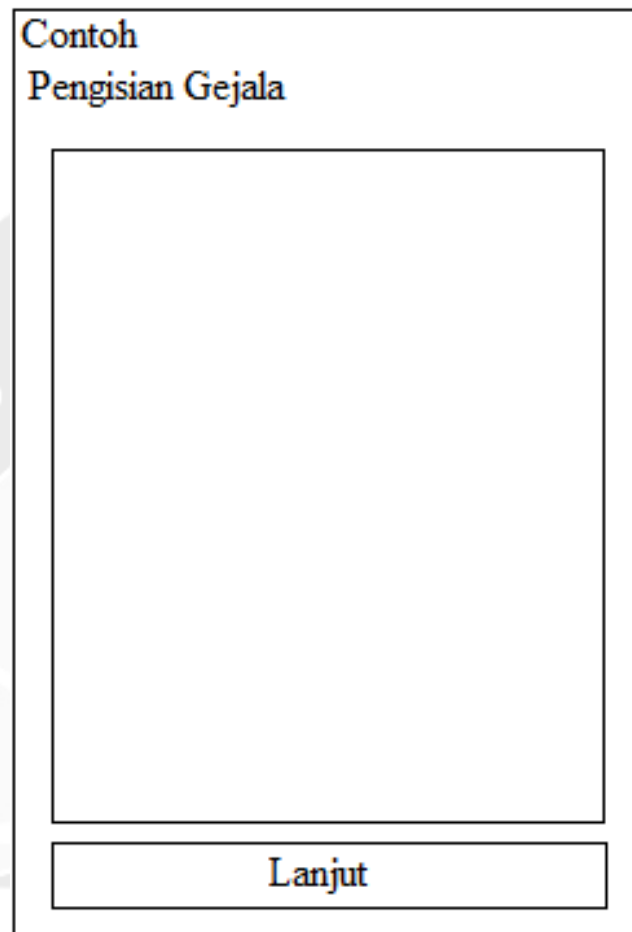


The image shows a software interface window with a title bar that reads "Bobot Nilai Ketidakpastian". Inside the window, there is a large, empty rectangular box, likely intended for displaying data or a calculation. Below this box is a button labeled "Lanjut" (Next).

Gambar 4.7 Rancangan Antarmuka Bobot CF

Antarmuka ini memberikan informasi kepada user tentang bobot dari nilai ketidakpastian (Certainty Factor).

4.1.2.3. Contoh Pengisian Gejala



Contoh
Pengisian Gejala

Lanjut

Gambar 4.8 Rancangan Antarmuka Contoh Pengisian Gejala
Antarmuka ini memberikan informasi kepada user tentang contoh pengisian gejala kepada user.

4.1.2.4. Pengisian Data User

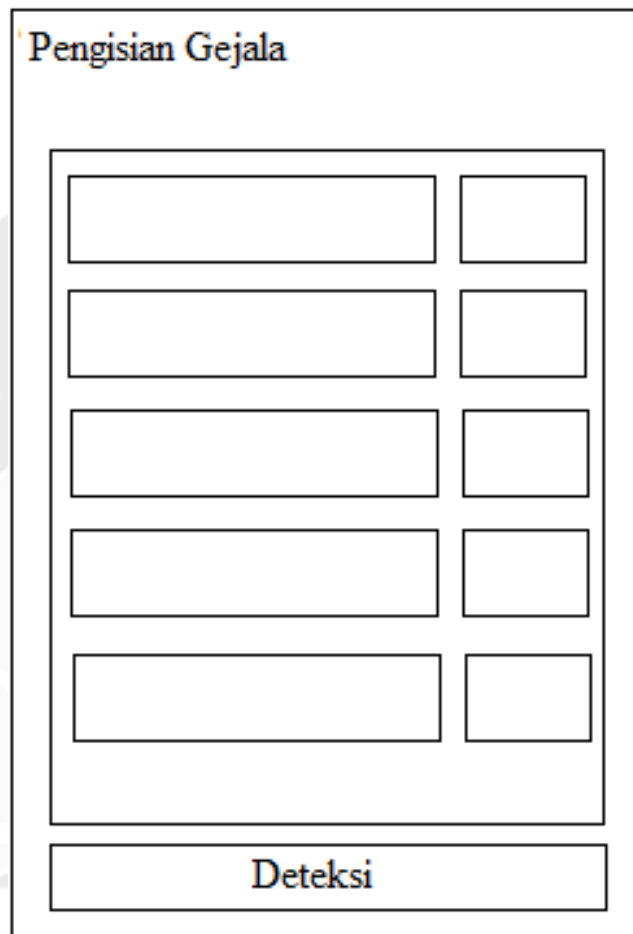
Data User

Mulai

Gambar 4.9 Rancangan Antarmuka Data user

Antarmuka ini digunakan user untuk memasukan data pribadi user ke dalam sistem.

4.1.2.5. Pengisian Gejala



The image shows a web form titled "Pengisian Gejala". It contains five rows of input fields, each consisting of a larger rectangular box on the left and a smaller square box on the right. Below these rows is a button labeled "Deteksi".

Gambar 4.10 Rancangan Antarmuka Pengisian Gejala
Antarmuka ini digunakan user untuk menginputkan gejala yang dirasakan ke dalam sistem.

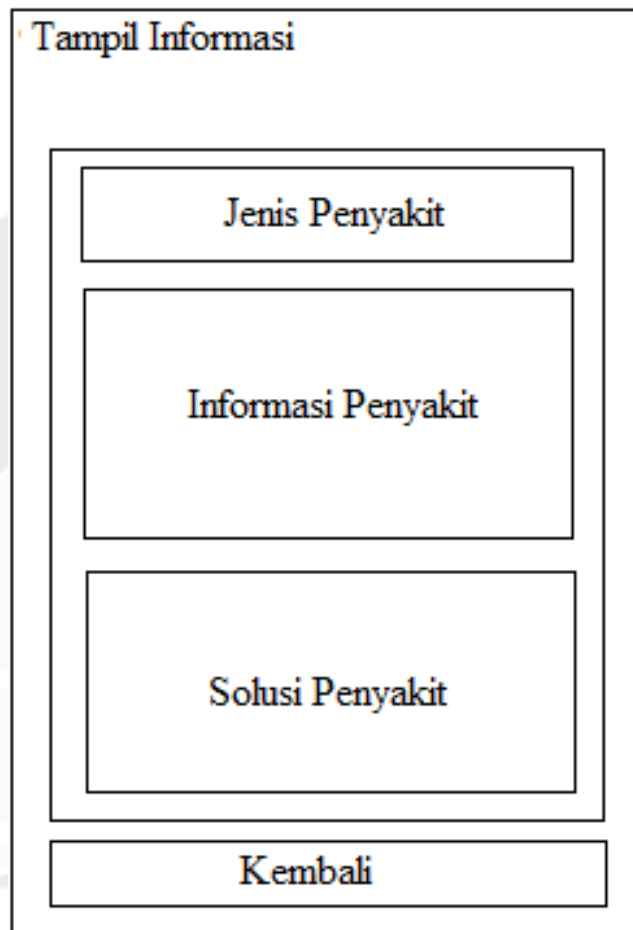
4.1.2.6. Hasil Identifikasi

The image shows a web interface titled "Hasil Identifikasi". It consists of three vertically stacked rectangular boxes. The top box is labeled "Data User". The middle box is labeled "Jenis Penyakit". The bottom box is labeled "Kembali". The interface is displayed on a mobile device, as indicated by the rounded corners and the presence of a watermark in the background.

Gambar 4.11 Antarmuka Hasil Identifikasi

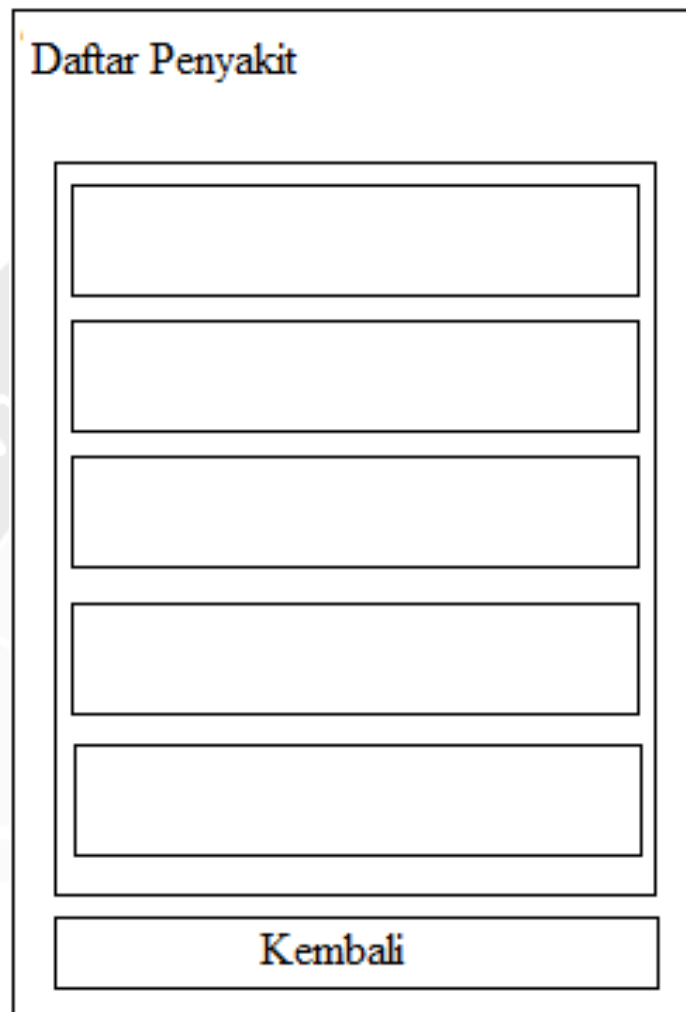
Antarmuka ini digunakan untuk memberikan informasi tentang hasil identifikasi penyakit.

4.1.2.7. Tampil Informasi Penyakit



Gambar 4.12 Rancangan Antarmuka Tampil Informasi
Antarmuka ini diberikan sistem agar user dapat mengerti informasi tentang penyakit kulit.

4.1.3. Daftar Penyakit



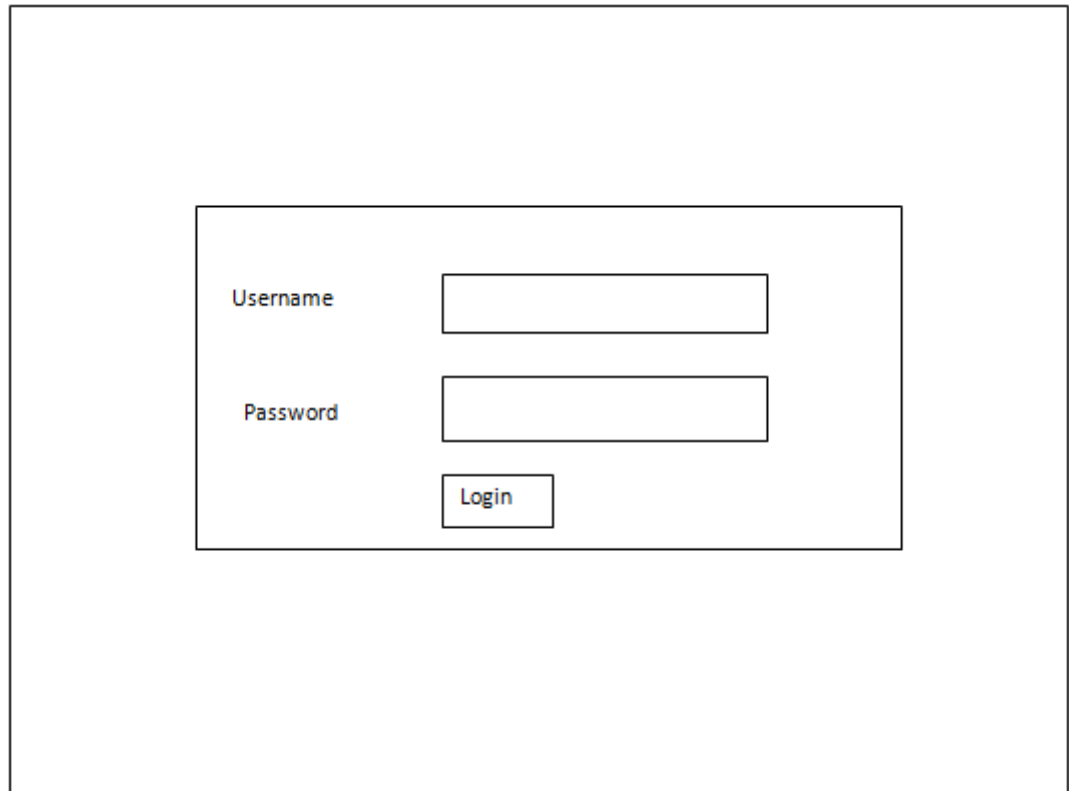
The image shows a web interface titled "Daftar Penyakit". It features a list of five empty rectangular input fields stacked vertically. Below these fields is a button labeled "Kembali". The interface is displayed on a mobile device, as indicated by the rounded corners and the presence of a watermark in the background.

Gambar 4.13 Rancangan Antarmuka Daftar Penyakit

Antarmuka ini diberikan sistem kepada user agar user dapat mengetahui jenis penyakit kulit apa saja yang ada dalam sistem.

4.1.4. Pengelolaan Web

4.1.4.1. Login



Gambar 4.14 Rancangan Antarmuka Login

Antarmuka ini digunakan untuk melakukan login ke dalam sistem. Untuk mendapat akses ke dalam sistem, user harus memasukkan username dan password pada textbox yang telah disediakan.

4.1.4.2. Tampil Penyakit

Selamat Datang, Admin [LogOut]	
Penyakit	<p style="text-align: center;">Halaman Tampil Penyakit</p>
Tampil Penyakit	
Tambah Penyakit	
Gejala	
Tampil Gejala	
Tambah Gejala	
Admin	
Tampil Admin	
Tambah Admin	

Gambar 4.15 Rancangan Antarmuka Tampil Penyakit
Antarmuka ini digunakan Admin untuk melihat jenis penyakit kulit yang ada dalam database.

4.1.4.3. Tambah Penyakit

Selamat Datang, Admin [LogOut]													
Penyakit	<table><tr><td>ID PENYAKIT</td><td><input type="text"/></td></tr><tr><td>NAMA PENYAKIT</td><td><input type="text"/></td></tr><tr><td>INFORMASI</td><td><input type="text"/></td></tr><tr><td>SOLUSI</td><td><input type="text"/></td></tr><tr><td colspan="2" style="text-align: center;"><input type="button" value="TAMBAH"/></td></tr><tr><td colspan="2" style="text-align: center;"><div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;">Halaman Tampil Penyakit</div></td></tr></table>	ID PENYAKIT	<input type="text"/>	NAMA PENYAKIT	<input type="text"/>	INFORMASI	<input type="text"/>	SOLUSI	<input type="text"/>	<input type="button" value="TAMBAH"/>		<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;">Halaman Tampil Penyakit</div>	
ID PENYAKIT		<input type="text"/>											
NAMA PENYAKIT		<input type="text"/>											
INFORMASI		<input type="text"/>											
SOLUSI		<input type="text"/>											
<input type="button" value="TAMBAH"/>													
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;">Halaman Tampil Penyakit</div>													
Tampil Penyakit													
Tambah Penyakit													
Gejala													
Tampil Gejala													
Tambah Gejala													
Admin													
Tampil Admin													
Tambah Admin													

Gambar 4.16 Rancangan Antarmuka Tambah Penyakit
Antarmuka ini digunakan oleh Admin untuk menambahkan jenis penyakit baru ke dalam database.

4.1.4.4. Ubah Penyakit

Selamat Datang, Admin [LogOut]	
Penyakit	ID PENYAKIT <input type="text"/>
Tampil Penyakit	
Tambah Penyakit	
Gejala	
Tampil Gejala	NAMA PENYAKIT <input type="text"/>
Tambah Gejala	INFORMASI <input type="text"/>
Admin	SOLUSI <input type="text"/>
Tampil Admin	<input type="button" value="Ubah"/>
Tambah Admin	<input type="text" value="Halaman Tampil Penyakit"/>

Gambar 4.17 Rancangan Antarmuka Ubah Penyakit
Antarmuka ini digunakan oleh Admin untuk mengubah jenis penyakit yang ada dalam database.

4.1.4.5. Tampil Gejala

Selamat Datang, Admin [LogOut]	
Penyakit	<p style="text-align: center;">Halaman Tampil Gejala</p>
Tampil Penyakit	
Tambah Penyakit	
Gejala	
Tampil Gejala	
Tambah Gejala	
Admin	
Tampil Admin	
Tambah Admin	

Gambar 4.18 Rancangan Antarmuka Tampil Gejala
Antarmuka ini digunakan Admin untuk melihat nama gejala yang ada dalam database.

4.1.4.6. Tambah Gejala

Selamat Datang, Admin [LogOut]	
Penyakit	ID GEJALA <input type="text"/>
Tampil Penyakit	
Tambah Penyakit	
Gejala	NAMA GEJALA <input type="text"/>
Tampil Gejala	
Tambah Gejala	
Admin	<input type="button" value="TAMBAH"/>
Tampil Admin	
Tambah Admin	
<input type="button" value="Halaman Tampil Gejala"/>	

Gambar 4.19 Rancangan Antarmuka Tambah Gejala
Antarmuka ini digunakan oleh Admin untuk menambahkan jenis gejala baru ke dalam database.

4.1.4.7. Ubah Gejala

Selamat Datang, Admin [LogOut]	
Penyakit	ID GEJALA <input type="text"/>
Tampil Penyakit	
Tambah Penyakit	
Gejala	NAMA GEJALA <input type="text"/>
Tampil Gejala	
Tambah Gejala	
Admin	<input type="button" value="UBAH"/>
Tampil Admin	
Tambah Admin	
<input type="text" value="Halaman Tampil Gejala"/>	

Gambar 4.20 Rancangan Antarmuka Ubah Gejala
Antarmuka ini digunakan oleh Admin untuk mengubah nama gejala yang ada dalam database.

4.1.4.8. Tampil Admin

Selamat Datang, Admin [LogOut]	
Penyakit	<p style="text-align: center;">Halaman Tampil Admin</p>
Tampil Penyakit	
Tambah Penyakit	
Gejala	
Tampil Gejala	
Tambah Gejala	
Admin	
Tampil Admin	
Tambah Admin	

Gambar 4.21 Rancangan Antarmuka Tampil Admin
Antarmuka ini digunakan Admin untuk melihat nama admin yang ada dalam database.

4.1.4.9. Tambah Admin

Selamat Datang, Admin [LogOut]	
Penyakit	ID <input type="text"/>
Tampil Penyakit	
Tambah Penyakit	
Gejala	
Tampil Gejala	
Tambah Gejala	
Admin	NAMA <input type="text"/>
Tampil Admin	USERNAME <input type="text"/>
Tambah Admin	PASSWORD <input type="text"/>
	<input type="button" value="TAMBAH"/>
	<div style="border: 1px solid black; padding: 10px; text-align: center;">Halaman Tampil Admin</div>

Gambar 4.22 Rancangan Antarmuka Tambah Admin

Antarmuka ini digunakan oleh Admin untuk menambahkan admin baru ke dalam database.

4.1.4.10. Ubah Admin

Selamat Datang, Admin [LogOut]													
Penyakit	<table><tr><td>ID</td><td><input type="text"/></td></tr><tr><td>NAMA</td><td><input type="text"/></td></tr><tr><td>USERNAME</td><td><input type="text"/></td></tr><tr><td>PASSWORD</td><td><input type="text"/></td></tr><tr><td colspan="2" style="text-align: center;"><input type="button" value="Ubah"/></td></tr><tr><td colspan="2" style="text-align: center;"><div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;">Halaman Tampil Admin</div></td></tr></table>	ID	<input type="text"/>	NAMA	<input type="text"/>	USERNAME	<input type="text"/>	PASSWORD	<input type="text"/>	<input type="button" value="Ubah"/>		<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;">Halaman Tampil Admin</div>	
ID		<input type="text"/>											
NAMA		<input type="text"/>											
USERNAME		<input type="text"/>											
PASSWORD		<input type="text"/>											
<input type="button" value="Ubah"/>													
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;">Halaman Tampil Admin</div>													
Tampil Penyakit													
Tambah Penyakit													
Gejala													
Tampil Gejala													
Tambah Gejala													
Admin													
Tampil Admin													
Tambah Admin													

Gambar 4.23 Rancangan Antarmuka Ubah Admin

Antarmuka ini digunakan oleh Admin untuk mengubah admin yang ada dalam database.