

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pembahasan dan dokumentasi dapat diambil kesimpulan yaitu :

1. Sistem pakar untuk mendiagnosa penyakit mata pada manusia berbasis web telah berhasil dibangun.
2. Sistem Pakar Mata (SIPMA) dapat membantu penderita penyakit mata mengetahui jenis penyakit dan penanganan awal dari penyakit tersebut.
3. Sistem Pakar Mata (SIPMA) membantu pasien melihat hasil rekam medis penyakit mata yang pernah dialami pasien.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan Sistem Pakar Mata (SIPMA) adalah dapat dikembangkan pada platform *mobile*. Dimana dapat digunakan aplikasi Sistem Pakar Mata (SIPMA) berbasis *mobile*.

DAFTAR PUSTAKA

Arhami, M. (2005). *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi.

Durki, J. (1994). *Expert Systems Design and Development*. New Jersey: Prentice Hall International Inc.

Data Penyakit,

<http://www.medicastore.com/med/subkategoripyk.php>

Haryanti. 2005. "Sistem Berbasis Cerdas untuk Mendiagnosa Penyakit Otak". Sekolah Tinggi Informatika dan Komputer (AKAKOM). Yogyakarta.

Kusrini. (2006). *Sistem Pakar (Teori dan Aplikasi)*. Yogyakarta: Andi Offset.

Kuwati. (2013). Sistem Pakar Pendeteksi Penyakit Saluran Pencernaan Pada Manusia Menggunakan Metode Forward Chaining dan Visual Basic 2010. *Sistem Informasi Jurnal Ilmiah USM*, 1, 11.

Nurfitria, F. (2013). SISTEM PAKAR PENDETEKSI KANKER PADA WANITA MENGGUNAKAN METODE FORWARD CHAINING. *Sistem Informasi Jurnal Ilmiah USM*, 2, 88-97.

Rachmawati, Damiri, J. D., & Susanto, A. (2012). Aplikasi Sistem Pakar Diagnosis Penyakit asam. *Online Jurnal STT-Garut*, 9.

Magiwa, Silva. 2013. "Pembangunan Sistem Pakar Diagnosa Jenis Pobia dengan Metode Bayes berbasis Web". Teknik Informatika Universitas Atma Jaya Yogyakarta. Yogyakarta.

Sidharta Lilyas, *Ilmu Penyakit Mata Edisi III*, Balai penerbit FKUI, Jakarta, 1997-2004.

Sri, K. (2003). *Artificial intelligence teknik dan aplikasinya*. Yogyakarta: Graha Ilmu.

Titis, A. (2009). *Sistem Pakar Untuk Mendiagnosa Penyakit Alopesia Pada Manusia*. Universitas Islam Malang: (<http://lib.uin-malang.ac.id/thesis/fullchapter/04550022-titis-astutik.ps35>, diakses 15 mei 2013).

Wicaksono. 2012. "Rancang Bangun Expert System Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining dan Backward Chaining". Teknik Informatika Universitas Siliwangi Tasikmalaya. Tasikmalaya.

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Sistem Pakar Mata (SIPMA)


Untuk:

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

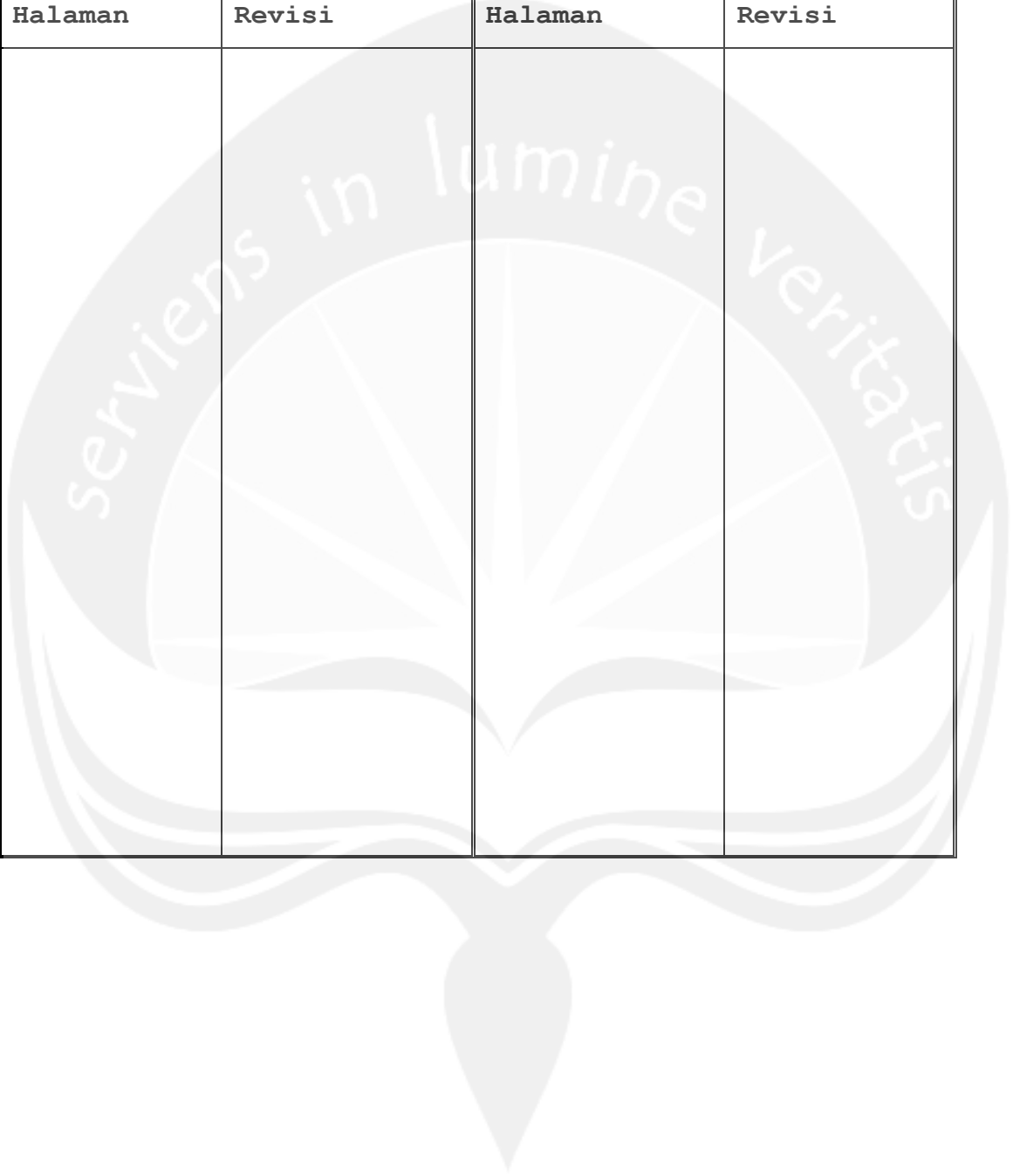
Raelda Rispadina Sitio /09 07 05965

Program Studi Teknik Informatika - Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL-SIPMA		1/35
		Revisi		

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi



DAFTAR ISI

1	Pendahuluan.....	166
1.1	Tujuan.....	166
1.2	Lingkup Masalah	166
1.3	Definisi, Akronim, dan Singkatan.....	166
Server	167
1.4	Referensi.....	167
1.5	Deskripsi umum (Overview).....	167
2	Deskripsi Kebutuhan.....	167
2.1	Perspektif produk.....	167
2.2	Fungsi Produk	168
2.3	Karakteristik Pasien	171
2.4	Batasan-batasan	171
2.5	Asumsi dan Ketergantungan.....	172
3	Kebutuhan Khusus.....	172
3.1	Kebutuhan antarmuka eksternal.....	172
3.2	Kebutuhan fungsionalitas perangkat lunak.....	174
4	Spesifikasi Rinci Kebutuhan	175
4.1	Spesifikasi Kebutuhan Fungsionalitas	175
5	Entity Relationship Diagram (ERD)	190

DAFTAR GAMBAR

1. Use Case Diagram	16
2. ERD	35



1. Pedahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIPMA (Sistem Pakar Mata) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara system dengan system lain perangkat lunak dan perangkat keras, dan Pasien), performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), atribut (feature-feature tambahan yang dimiliki system), serta mendefinisikan fungsi perangkat lunak. SKPL-SIPMA ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Tujuan Pembangunan perangkat lunak Sistem Pakar Mata (SIPMA) adalah untuk :

3. Membangun sistem pakar yang dapat Memberikan informasi jenis penyakit mata berdasarkan gejala yang diberikan dengan menggunakan metode *Forward Chaining*.
4. Membangun sistem pakar yang dapat memberikan solusi untuk penanganan penyakit mata.

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-SPF-XXX	Kode yang merepresentasikan kebutuhan pada SPF (Sistem Pakar Fobia) dimana XXX merupakan nomor fungsi produk.
SPF	Perangkat lunak sistem pakar diagnosa jenis penyakit mata
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai

	komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan Web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah :

1. Raelda Rispadina, *Spesifikasi Kebutuhan Perangkat Lunak Sistem Courier Service and Delivery (CSDS)*, 2012.
2. Junyathi Silpa, *Spesifikasi Kebutuhan Perangkat Lunak Fobia (SPF)*, 2013

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup 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 SIPMA yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik Pasien, batasan dalam Pasienan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SIPMA tersebut.

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

2. Deskripsi Kebutuhan

2.1 Perspektif produk

SIPMA merupakan perangkat lunak yang dibangun dengan tujuan membantu pasiennya mengetahui dan solusi dalam menangani masalah penyakit mata. Proses yang terjadi pada sistem ini adalah bagaimana menanamkan suatu ilmu kepada komputer sehingga komputer dapat berpikir seperti cara berpikir yang dipakai manusia untuk memecahkan masalah. Pasien menjawab pertanyaan

gejala yang dialami, lalu akan mengolah permasalahan tersebut sesuai dengan metode yang ditanamkan untuk kemudian memberikan solusi berdasarkan pengetahuan yang telah ditanamkan dibasis datanya. Cangkupan user sistem yaitu administrator dan Pasien. Administrator memiliki akses penuh terhadap sistem, pasien, serta mengelola proses yang terjadi pada sistem ini. Pasien memiliki hak akses untuk melakukan proses kerja pada sistem.

SIPMA dibangun dengan basis website menggunakan bahasa pemograman *PHP* dan database *MySQL*.

2.2 Fungsi Produk

Fungsi produk perangkat lunak SIPMA adalah sebagai berikut :

A. Administrator

1. Fungsi Login (SKPL-SIPMA-001)

Fungsi login merupakan fungsi yang digunakan oleh user (Administrator dan Pasien) untuk dapat masuk dalam sistem yang akan digunakan.

2. Fungsi Reset password pasien (SKPL-SIPMA-002)

mengelola data admin merupakan fungsi yang digunakan untuk mengelola reset password pasien.

3. Fungsi Mengelola Data Admin (SKPL-SIPMA-003)

mengelola data admin merupakan fungsi yang digunakan untuk mengelola data admin.

Fungsi mengelola data admin meliputi :

d. Fungsi Add data admin (SKPL-SIPMA-003-01)

Merupakan fungsi yang digunakan untuk menambahkan data admin.

e. Fungsi Delete data admin (SKPL-SIPMA-003-03)

Merupakan fungsi yang digunakan untuk menghapus data gejala.

f. Fungsi Display data admin (SKPL-SIPMA-003-04)

Merupakan fungsi yang digunakan untuk menampilkan data gejala.

4. Fungsi Mengelola Data Gejala (**SKPL-SIPMA-004**)

Fungsi mengelola data gejala merupakan fungsi yang digunakan untuk mengelola gejala penyakit mata.

Fungsi mengelola data gejala meliputi :

f. Fungsi Add data gejala (**SKPL-SIPMA-004-01**)

Merupakan fungsi yang digunakan untuk menambahkan data gejala.

g. Fungsi Edit data gejala (**SKPL-SIPMA-004-02**)

Merupakan fungsi yang digunakan untuk mengubah data gejala.

h. Fungsi Delete data gejala (**SKPL-SIPMA-004-03**)

Merupakan fungsi yang digunakan untuk menghapus data gejala.

i. Fungsi Display data gejala (**SKPL-SIPMA-004-04**)

Merupakan fungsi yang digunakan untuk menampilkan data gejala.

j. Fungsi Search data gejala (**SKPL-SIPMA-004-05**)

Merupakan fungsi yang digunakan untuk mencari data gejala berdasarkan kata kunci yang diinputkan.

5. Fungsi Mengelola Data Penyakit (**SKPL-SIPMA-005**)

Fungsi mengelola data penyakit merupakan fungsi yang digunakan untuk mengelola data penyakit mata.

Fungsi mengelola data penyakit meliputi :

a. Fungsi Add data penyakit (**SKPL-SIPMA-005-01**)

Merupakan fungsi yang digunakan untuk menambah data penyakit.

b. Fungsi Edit data penyakit (**SKPL-SIPMA-005-02**)

Merupakan fungsi yang digunakan untuk mengubah data penyakit.

c. Fungsi Delete data penyakit (**SKPL-SIPMA-005-03**)

Merupakan fungsi yang digunakan untuk menghapus data penyakit.

d. Fungsi Display data penyakit (**SKPL-SIPMA-005-04**)

Merupakan fungsi yang digunakan untuk menampilkan data penyakit.

e. Fungsi Search data penyakit **(SKPL-SIPMA-005-05)**

Merupakan fungsi yang digunakan untuk mencari data penyakit.

6. Fungsi Mengelola Data Aturan **(SKPL-SIPMA-006)**

Fungsi mengelola data aturan merupakan fungsi yang digunakan untuk mengelola data aturan.

Fungsi mengelola data aturan meliputi :

e. Fungsi Add data aturan **(SKPL-SIPMA-006-01)**

Merupakan fungsi yang digunakan untuk menambah data aturan.

f. Fungsi Delete data aturan **(SKPL-SIPMA-006-02)**

Merupakan fungsi yang digunakan untuk menghapus data aturan.

g. Fungsi Display data aturan **(SKPL-SIPMA-006-03)**

Merupakan fungsi yang digunakan untuk menampilkan data aturan.

h. Fungsi Search data aturan **(SKPL-SIPMA-006-04)**

Merupakan fungsi yang digunakan untuk mencari data aturan berdasarkan kata kunci yang diinputkan.

B. Pasien

9. Fungsi Menampilkan Home **(SKPL-SIPMA-007)**

Fungsi menampilkan home merupakan fungsi yang digunakan untuk menampilkan halaman utama.

10. Fungsi Menampilkan Kornea Mata **(SKPL-SIPMA-008)**

Fungsi menampilkan kornea mata merupakan fungsi yang digunakan untuk menampilkan informasi tentang kornea mata.

11. Fungsi Menampilkan Informasi Penyakit Mata **(SKPL-SIPMA-009)**

Fungsi menampilkan tentang informasi penyakit mata merupakan fungsi yang digunakan untuk menampilkan informasi tentang penyakit mata.

12. Fungsi Registrasi Pasien **(SKPL-SIPMA-010)**

Fungsi Registrasi pasien merupakan fungsi untuk mendaftarkan biodata pasien.

13. Fungsi Recam Medic **(SKPL-SIPMA-011)**

Fungsi recam medic merupakan fungsi yang digunakan untuk melihat hasil recam medic penyakit mata yang pernah dialami pasien.

14. Fungsi Konsultasi **(SKPL-SIPMA-012)**

Fungsi konsultasi merupakan fungsi untuk menjawab pertanyaan gejala yang dialami. Diberikan pilihan jawaban yaitu "Ya" dan "Tidak".

15. Fungsi Hasil Diagnosa **(SKPL-SIPMA-013)**

Fungsi mendiagnosa penyakit mata merupakan fungsi yang digunakan untuk hasil diagnosa jenis penyakit mata berdasarkan konsultasi, persentase penyakit dan cara penanganan awal.

16. Fungsi Change password **(SKPL-SIPMA-014)**

Fungsi change password merupakan fungsi untuk mengganti password yang telah direset oleh admin.

2.3 Karakteristik Pasien

Karakteristik dari Pasien perangkat lunak SIPMA adalah sebagai berikut :

1. Administrator

- a. Memahami Pasienan komputer dan internet.
- b. Memahami pengelolaan data administrator.
- c. Memahami proses pengelolaan sistem pakar.

2. Pasien

- a. Memahami Pasien computer dan internet.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIPMA tersebut adalah :

1. Kebijakan Umum

Bepedoman pada tujuan dari pengembangan perangkat lunak SIPMA.

2. Keterbatasan perangkat keras

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

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada web yang dapat diakses dengan menggunakan web server.

3. Kebutuhan Khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antarmuka eksternal pada perangkat lunak SIPMA meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

3.1.1. Antarmuka pemakai

Pasien berinteraksi dengan perangkat lunak SIPMA dengan antarmuka halaman web. Masukan dari perangkat lunak ini adalah pilihan menu yang dilakukan dengan memilih menu pilihan yang ada. Keluaran dari perangkat lunak SIPMA berupa kesimpulan medis dengan dasar pilihan pasien dari gejala-gejala yang diberikan oleh sistem.

3.1.2. Antarmuka perangkat keras

Perangkat keras yang digunakan dalam sistem ini antara lain:

1. *Monitor*, digunakan untuk menampilkan informasi kepada Pasien.
2. *Mouse*, digunakan untuk mengenali input yang dilakukan oleh Pasien berkaitan dengan *event click*.
3. *Keyboard*, digunakan untuk mengenali input yang digunakan oleh Pasien untuk menginputkan data berupa karakter, teks, ataupun menu *pull down*.
4. *CPU*, digunakan untuk memproses data.
5. *Hardisk*, digunakan untuk menyimpan data.

3.1.3. Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan dalam mengoperasikan perangkat lunak SIPMA adalah sebagai berikut :

1. Nama : XAMPP

Sebagai suatu web server yang mencakup :

- a. MySQL sebagai Database,
- b. Apache sebagai HTTP Server,

Aplikasi XAMPP telah terintegrasi dengan bahasa pemrograman PHP, dan Perl.

2. Nama : Windows 7

Sumber : Microsoft

Sebagai Sistem Operasi yang digunakan dalam SIPMA.

3. Nama : Mozilla Firefox

Sumber : Microsoft

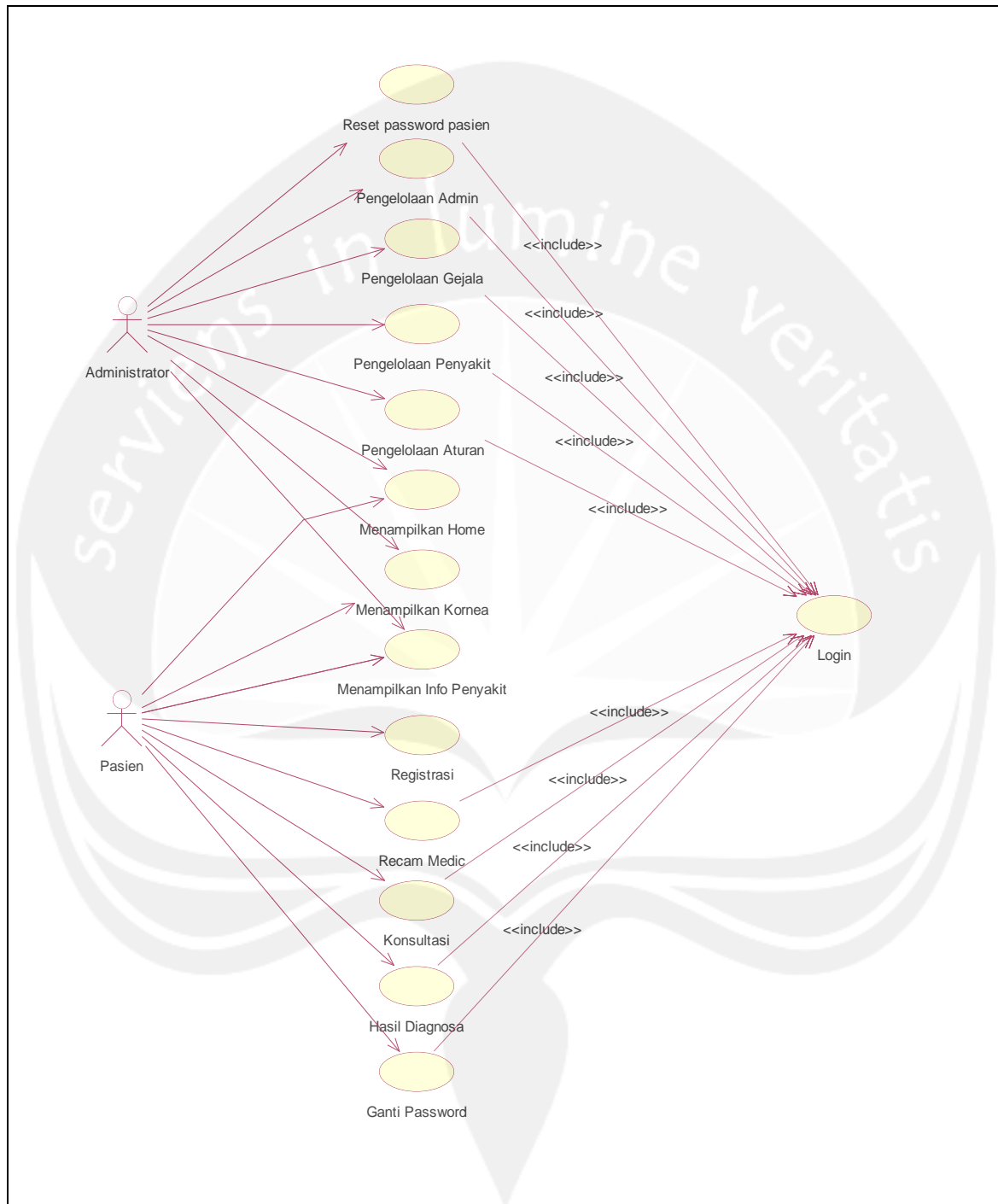
Sebagai web browser untuk menjalankan SIPMA.

3.1.4. Antarmuka perangkat komunikasi

Antarmuka komunikasi perangkat lunak SIPMA berbasis *website* menggunakan protocol HTTP.

Kebutuhan fungsionalitas perangkat lunak

3.2.1. Use Case Diagram



Gambar 3.1 Use Case Diagram

4. Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1. Use case Spesifikasi: Login

1. Brief Description

Use case ini digunakan oleh actor untuk memperoleh akses ke dalam sistem. Login berdasarkan pada sebuah *username* dan *password* .

2. Primary Actor

1. Administrator
2. Pasien

3. Supporting Actor

None

4. Basic Flow

1. Use case ini dimulai ketika actor memasuki sistem.
2. Sistem menampilkan antarmuka login.
3. Actor memasuki username dan password.
4. Sistem memeriksa username dan password yang diinputkan actor.
 - E-1. Username atau Password tidak sesuai
5. Sistem Memberikan akses ke actor.
6. Use case ini selesai.

5. Alternative Flow

None

6. Error Flow

- E-1 Username atau password tidak sesuai
1. Sistem menampilkan peringatan "Gagal Login"
 2. Actor memilih menginputkan ulang
 3. Kembali ke Basic Flow langkah ke-2.

7. PreConditions

None

8. PostConditions

1. Actor dapat menggunakan fungsi-fungsi pada sistem.

4.1.2. Use case Spesifikasi: Mengelola Reset Password Pasien

1. Brief Description

Use case ini digunakan oleh actor untuk mengelola data admin. Aktor dapat melakukan *reset password pasien*

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use case ini dimulai ketika actor memilih untuk mengelola data admin.
2. Sistem Memberikan pilihan untuk melakukan *reset password*
3. Aktor memilih untuk melakukan *reset password*
4. Aktor mereset password pasien.
5. Aktor meminta sistem untuk reset password
6. Sistem mereset password .
7. Sistem meng update data user ke database.
8. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

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

8. PostConditions

1. Data didatabase telah terupdate.

4.1.3. Use case Spesifikasi: Mengelola Data Admin

1. Brief Description

Use case ini digunakan oleh actor untuk mengelola data penyakit. Aktor dapat melakukan *add data admin, delete data admin, display data admin.*

2. Primary Actor

1. Administrator.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk mengelola data admin.
2. Sistem Memberikan pilihan untuk melakukan *add* data penyakit, *delete* data penyakit, *display* data penyakit.
3. Aktor memilih untuk melakukan *add* data admin.
A-1 Aktor memilih untuk melakukan *delete* data admin.
A-2 Aktor memilih untuk melakukan *display* data admin.
4. Aktor menginputkan data admin.
5. Aktor meminta sistem untuk menyimpan data admin yang telah diinputkan.
6. Sistem mengecek data admin yang telah diinputkan.
7. Sistem menyimpan data admin database.
8. Use case selesai.

5. Alternative Flow

A-1 Aktor memilih *delete* data admin.

1. Sistem menampilkan data admin.
2. Aktor memilih data admin yang mau dihapus.
3. Aktor meminta sistem untuk menghapus data admin yang telah dipilih.
4. Sistem menghapus data admin yang dipilih aktor dari database.
5. Berlanjut ke Basic Flow langkah ke-8.

A-2 Aktor memilih *display* data admin.

3. Sistem menampilkan data admin.
4. Berlanjut ke Basic Flow langkah ke-8.

6. Error Flow

None

7. PreConditions

1. Use case login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data admin di database telah terupdate.

4.1.4. Use case Spesifikasi: Mengelola Data Gejala

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola data gejala. Aktor dapat melakukan *add* data gejala, *edit* data gejala, *delete* data gejala, *display* data gejala, dan *search* data gejala.

2. Primary Actor

1. Administrator.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk mengelola data gejala.
2. Sistem Memberikan pilihan untuk melakukan *add* data gejala, *edit* data gejala, *delete* data gejala, *display* data gejala, atau *search* data gejala.
3. Aktor memilih untuk melakukan *add* data gejala.
 - A-1 Aktor memilih untuk melakukan *edit* data gejala.
 - A-2 Aktor memilih untuk melakukan *delete* data gejala.
 - A-3 Aktor memilih untuk melakukan *display* data gejala.
 - A-4 Aktor memilih untuk melakukan *search* data gejala.
4. Aktor menginputkan data gejala.
5. Aktor meminta sistem untuk menyimpan data gejala yang telah diinputkan.
6. Sistem mengecek data gejala yang telah diinputkan.
7. Sistem menyimpan data gejala database.
8. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih *edit* data gejala.
1. Sistem menampilkan data gejala.
 2. Aktor mengedit data gejala.
 3. Aktor meminta sistem untuk menyimpan data gejala yang telah diedit.
 4. Sistem melakukan pengecekan terhadap data gejala yang telah diedit.
 - E-2 Data gejala yang telah diedit salah.

5. Sistem menyimpan data gejala yang telah diedit ke database

6. Berlanjut ke Basic Flow langkah ke-8.

A-2 Aktor memilih *delete* data gejala.

1. Sistem menampilkan data gejala.

2. Aktor memilih data gejala yang mau dihapus.

3. Aktor meminta sistem untuk menghapus data gejala yang telah dipilih.

4. Sistem menghapus data gejala yang dipilih aktor dari database.

5. Berlanjut ke Basic Flow langkah ke-8.

A-3 Aktor memilih *display* data gejala.

1. Sistem menampilkan data gejala.

2. Berlanjut ke Basic Flow langkah ke-8.

A-4 Aktor memilih *search* data gejala.

1. Aktor menginputkan data gejala yang ingin dicari berdasarkan kata kunci.

2. Sistem menampilkan data gejala yang dicari

3. Berlanjut ke Basic Flow langkah ke-8.

6. Error Flow

E-1 Data gejala yang diinputkan aktor salah.

1. Sistem Memberikan pesan peringatan bahwa "data yang diinputkan tidak lengkap".

2. Kembali ke Basic Flow langkah ke-4.

E-2 Data gejala yang diinputkan aktor salah.

1. Sistem Memberikan pesan peringatan bahwa "data yang diedit tidak lengkap".

2. Kembali ke Alternative Flow A-1 langkah ke-3.

7. PreConditions

1. Use case login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data gejala didatabase telah terupdate.

4.1.5. Use case Spesifikasi: Mengelola Data Penyakit

1. Brief Description

Use case ini digunakan oleh actor untuk mengelola data penyakit. Actor dapat melakukan *add* data penyakit, *edit* data penyakit, *delete* data penyakit, *display* data penyakit atau *search* data penyakit.

2. Primary Actor

1. Administrator.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk mengelola data penyakit.
2. Sistem Memberikan pilihan untuk melakukan *add* data penyakit, *edit* data penyakit, *delete* data penyakit, *display* data penyakit, atau *search* data penyakit.
3. Aktor memilih untuk melakukan *add* data penyakit.
 - A-1 Aktor memilih untuk melakukan *edit* data penyakit.
 - A-2 Aktor memilih untuk melakukan *delete* data penyakit.
 - A-3 Aktor memilih untuk melakukan *display* data penyakit.
 - A-4 Aktor memilih untuk melakukan *search* data penyakit.
4. Aktor menginputkan data penyakit.
5. Aktor meminta sistem untuk menyimpan data penyakit yang telah diinputkan.
6. Sistem mengecek data penyakit yang telah diinputkan.
7. Sistem menyimpan data penyakit database.
8. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih *edit* data gejala.
1. Sistem menampilkan data gejala.
 2. Aktor mengedit data gejala.
 3. Aktor meminta sistem untuk menyimpan data gejala yang telah diedit.
 4. Sistem melakukan pengecekan terhadap data gejala yang telah diedit.
 - E-2 Data gejala yang telah diedit salah.

5. Sistem menyimpan data gejala yang telah diedit ke database

6. Berlanjut ke Basic Flow langkah ke-8.

A-2 Aktor memilih *delete* data gejala.

1. Sistem menampilkan data gejala.

2. Aktor memilih data gejala yang mau dihapus.

3. Aktor meminta sistem untuk menghapus data gejala yang telah dipilih.

4. Sistem menghapus data gejala yang dipilih aktor dari database.

5. Berlanjut ke Basic Flow langkah ke-8.

A-3 Aktor memilih *display* data gejala.

1. Sistem menampilkan data gejala.

2. Berlanjut ke Basic Flow langkah ke-8.

A-4 Aktor memilih *search* data gejala.

1. Aktor menginputkan data gejala yang ingin dicari berdasarkan kata kunci.

2. Sistem menampilkan data gejala yang dicari

3. Berlanjut ke Basic Flow langkah ke-8.

6. Error Flow

E-1 Data penyakit yang diinputkan aktor salah.

1. Sistem Memberikan pesan peringatan bahwa "data yang diinputkan tidak lengkap".

2. Kembali ke Basic Flow langkah ke-4.

E-2 Data penyakit yang diinputkan aktor salah.

1. Sistem Memberikan pesan peringatan bahwa "data yang diedit tidak lengkap".

2. Kembali ke Alternative Flow A-1 langkah ke-3.

7. PreConditions

1. Use case login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data penyakit di database telah terupdate.

4.1.6. Use case Spesifikasi: Mengelola Data Aturan

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola data aturan. Aktor dapat melakukan *add* data aturan, *delete* data aturan, *display* data aturan, dan *search* data aturan.

2. Primary Actor

2. Administrator.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk mengelola data aturan.
2. Sistem Memberikan pilihan untuk melakukan *add* data aturan, *delete* data aturan, *display* data aturan, atau *search* data aturan.
3. Aktor memilih untuk melakukan *add* data aturan.
 - A-1 Aktor memilih untuk melakukan *delete* data aturan.
 - A-2 Aktor memilih untuk melakukan *display* data aturan.
 - A-3 Aktor memilih untuk melakukan *search* data aturan.
4. Aktor menginputkan data aturan.
5. Aktor meminta sistem untuk menyimpan data aturan yang telah diinputkan.
6. Sistem mengecek data aturan yang telah diinputkan.
7. Sistem menyimpan data aturan database.
8. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih *delete* data aturan.
1. Sistem menampilkan data aturan.
 2. Aktor memilih data aturan yang mau dihapus.
 3. Aktor meminta sistem untuk menghapus data aturan yang telah dipilih.
 4. Sistem menghapus data aturan yang dipilih aktor dari database.
 5. Berlanjut ke Basic Flow langkah ke-8.
- A-2 Aktor memilih *display* data aturan.
1. Sistem menampilkan data aturan.

2. Berlanjut ke Basic Flow langkah ke-8.

A-3 Aktor memilih *search* data aturan.

1. Aktor menginputkan data aturan yang ingin dicari berdasarkan kata kunci.

2. Sistem menampilkan data aturan yang dicari

3. Berlanjut ke Basic Flow langkah ke-8.

6. Error Flow

E-1 Data aturan yang diinputkan aktor salah.

1. Sistem Memberikan pesan peringatan bahwa "data yang diinputkan tidak lengkap".

2. Kembali ke Basic Flow langkah ke-4.

E-2 Data aturan yang diinputkan aktor salah.

1. Sistem Memberikan pesan peringatan bahwa "data yang diedit tidak lengkap".

2. Kembali ke Alternative Flow A-1 langkah ke-3.

7. PreConditions

1. Use case login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data aturan didatabase telah terupdate.

4.1.1.7. Use case Spesifikasi: Menampilkan Home

1. Brief Description

Use case ini digunakan oleh aktor untuk menampilkan halaman home (halaman utama).

2. Primary Actor

1. User

3. Supporting Actor

None.

4. Basic Flow

3. Use case ini dimulai ketika aktor memilih menampilkan home.

4. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor berhasil menampilkan halaman home.

4.1.8. Use case Spesifikasi: Menampilkan Kornea Mata

1. Brief Description

Use case ini digunakan oleh aktor untuk menampilkan informasi kornea mata.

2. Primary Actor

1. User

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih menampilkan informasi kornea mata.
2. Sistem menampilkan halaman informasi kornea mata.
3. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor berhasil menampilkan halaman informasi kornea mata.

4.1.9. Use case Spesifikasi: Menampilkan Informasi Penyakit Mata

1. Brief Description

Use case ini digunakan oleh aktor untuk menampilkan informasi penyakit mata.

2. Primary Actor

User

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih manampilkan informasi penyakit mata.
2. Sistem menampilkan halaman informasi penyakit mata.
3. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor berhasil menampilkan halaman informasi penyakit mata.

4.1.10. Use case Spesifikasi: Registrasi Pasien

1. Brief Description

Use ini digunakan oleh aktor untuk menampilkan registrasi pasien.

2. Primary Actor

1. Pasien.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih registrasi pasien.
2. Sistem menampilkan form registrasi.
3. Aktor menginputkan data pasien.
4. Aktor meminta sistem untuk menyimpan data pasien yang telah diinputkan.
5. Sistem mengecek data yang telah diinputkan

6. Sistem menyimpan data user ke dalam database.

7. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

1. Use case registrasi pasien telah dilakukan
2. Aktor memasuki sistem.

8. PostConditions

1. Data pasien didatabase telah terupdate.

4.1.11. Use case Spesifikasi: Menampilkan Recam Medic

1. Brief Description

case ini digunakan oleh aktor untuk menampilkan recam medic penyakit mata.

2. Primary Actor

1. Pasien

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih manampilkan recam medic.
2. Sistem menampilkan halaman recam medic.
3. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor berhasil menampilkan halaman recam medic.

4.1.12. Use case Spesifikasi: Konsultasi

1. Brief Description

Use case ini digunakan oleh aktor untuk menampilkan halaman konsultasi.

2. Primary Actor

1. Pasien.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini digunakan ketika aktor memilih menu konsultasi.
2. Sistem menampilkan pertanyaan gejala yang dirasakan
3. Aktor menjawab pertanyaan gejala dengan menginputkan jawaban "Ya" atau "Tidak".

A-1 Aktor menjawab pertanyaan gejala yang dirasakan

4. Sistem melanjutkan pertanyaan hingga gejala terakhir.
5. Aktor selesai menjawab pertanyaan.
6. Sistem menghitung rumus proposisi gejala berdasarkan jumlah gejala yang diinputkan aktor dengan jumlah gejala dari penyakit.
7. Sistem menampilkan nama penyakit dan solusi penanganannya awal.
8. Use case selesai.

5. Alternative Flow

None

6. Error Flow

None

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. Aktor berhasil melakukan proses konsultasi.

4.1.13. Use case Spesifikasi: Hasil Diagnosa Penyakit

1. Brief Description

Use case ini digunakan oleh aktor untuk proses hasil diagnosa penyakit mata.

2. Primary Actor

1. Pasien.

3. Supporting Actor

None.

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih menampilkan menu diagnosa penyakit.
2. Sistem menampilkan data diagnosa pasien yang telah melakukan konsultasi.
3. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. aktor berhasil menampilkan diagnosa penyakit pasien.

4.1.14. Use case Spesifikasi: Change Password

1. Brief Description

Use case ini digunakan oleh actor untuk mengubah password lama menjadi password baru

2. Primary Actor

2. Pasien

3. Supporting Actor

None

4. Basic Flow

1. Use case ini dimulai ketika actor memilih untuk change password new.
2. Sistem Memberikan pilihan untuk melakukan *change password*
3. Aktor memilih untuk melakukan *change password*
4. Aktor mengganti password lama menjadi password baru.
5. Aktor meminta sistem untuk menyimpan password baru
6. Sistem menyimpan password .

7. Sistem meng update data user ke database.

8. Use case selesai.

5. Alternative Flow

None.

6. Error Flow

None.

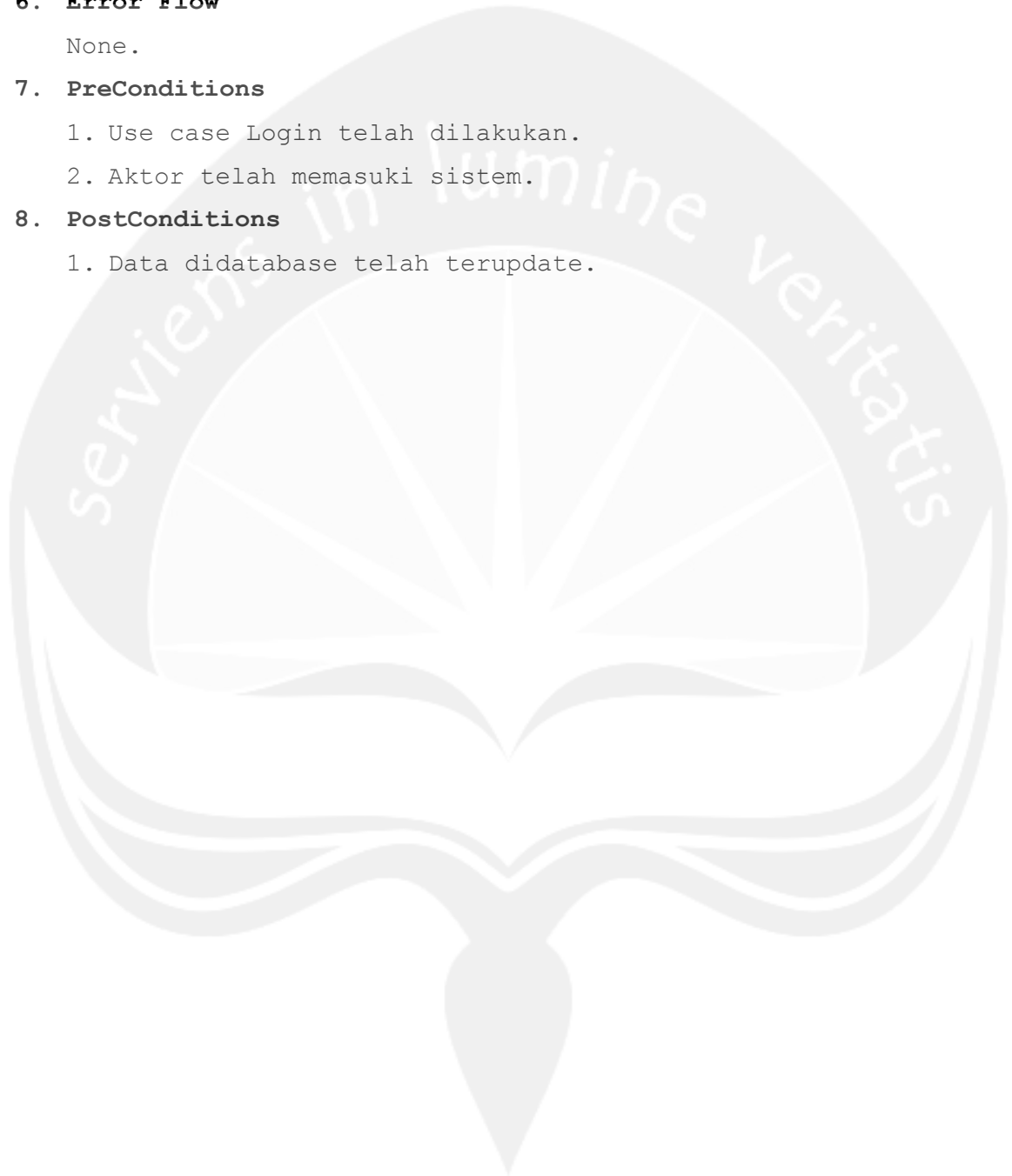
7. PreConditions

1. Use case Login telah dilakukan.

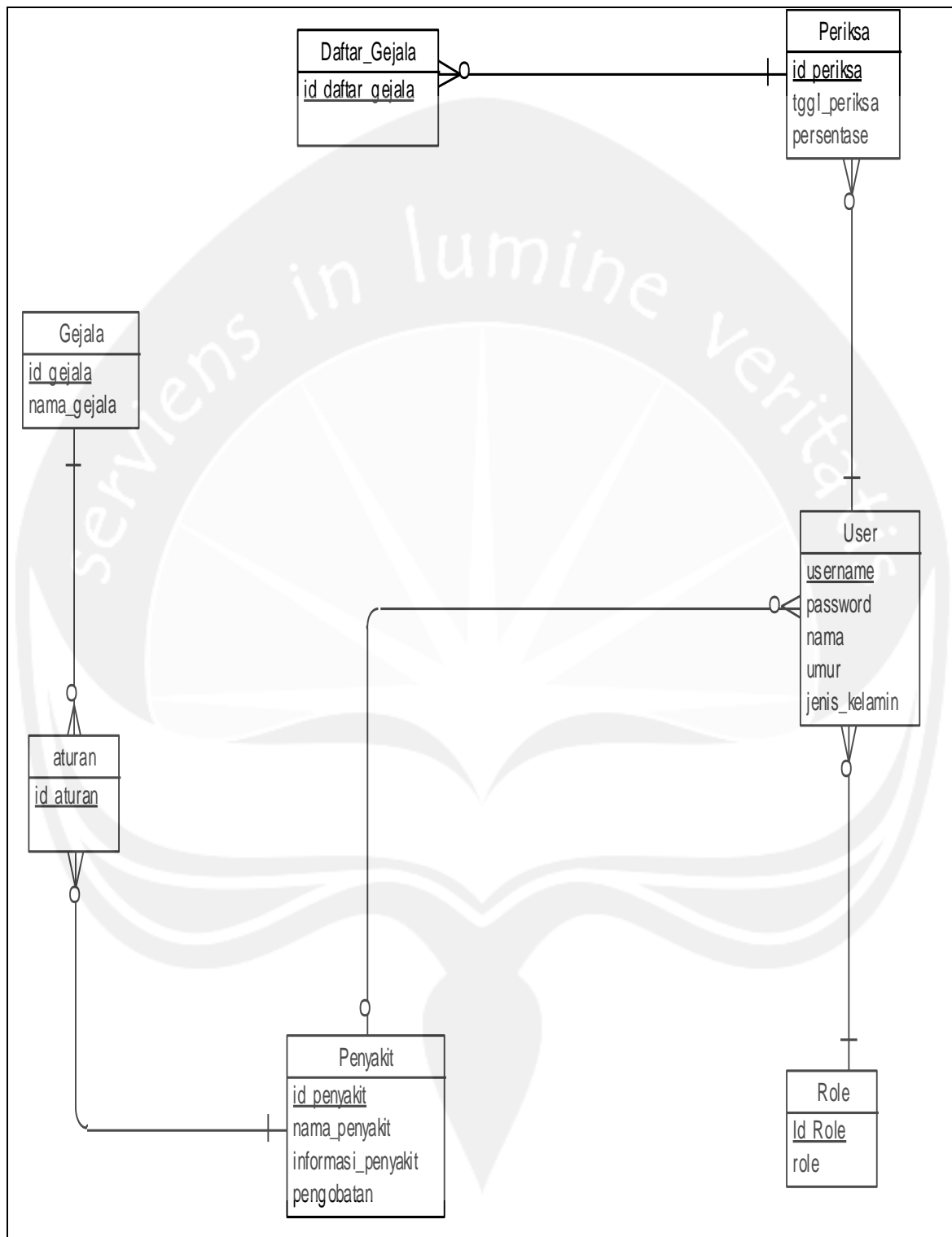
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data didatabase telah terupdate.



5. Entity Relationship Diagram (ERD)



Gambar 5.1 ERD (Entity Relationship Diagram)

DPPL


DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SIPMA (Sistem Pakar Mata)

Untuk:
Universitas Atma Jaya Yogyakarta


Dipersiapkan oleh:
Raelda Rispadina Sitio / 09.07.05965

Program Studi Teknik Informatika - Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		DPPL-SIPMA		1/46
		Revisi		

DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi



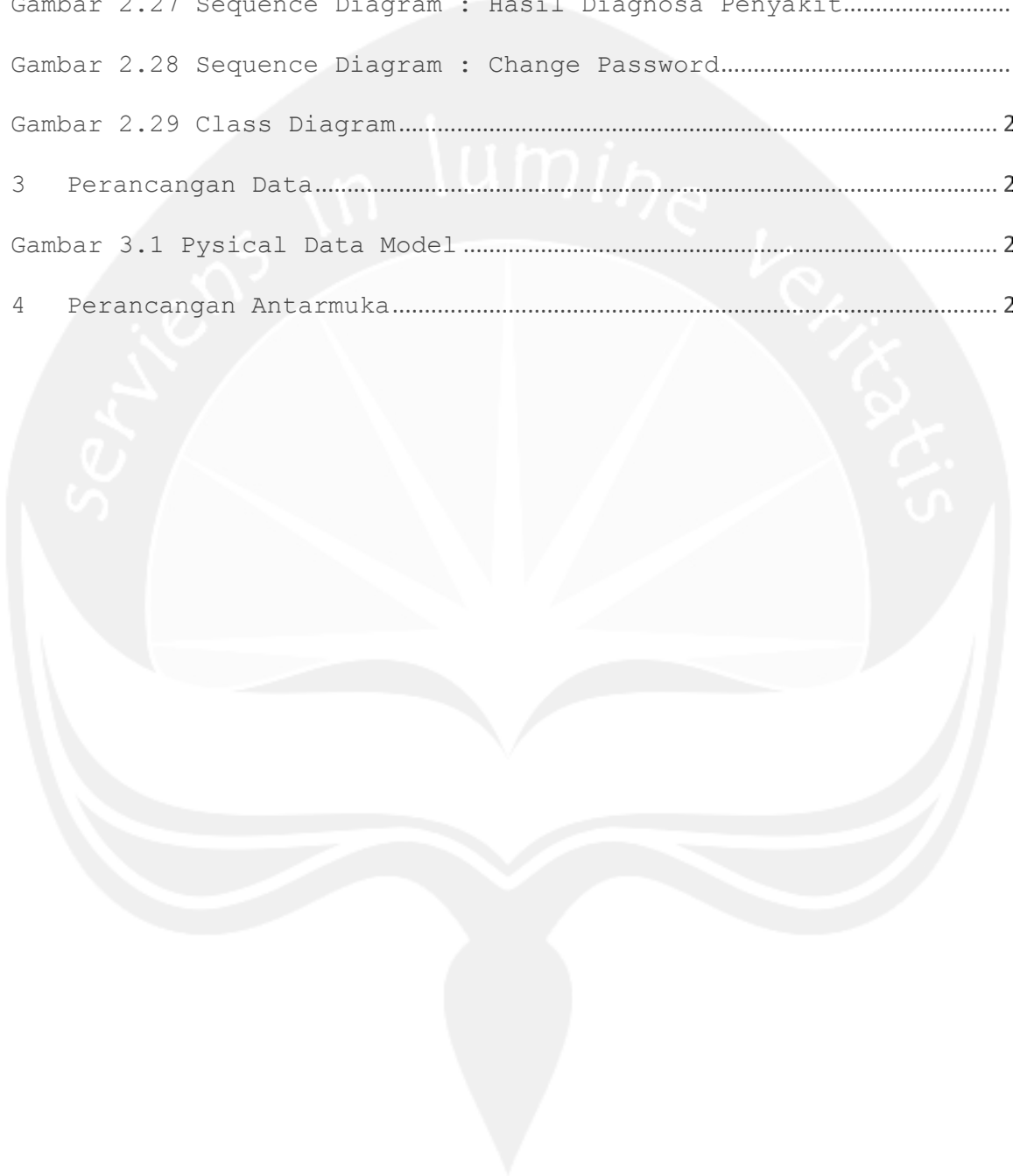
DAFTAR ISI

1	Pendahuluan.....	197
1.1	Tujuan	197
1.2	Ruang Lingkup.....	197
1.3	Definisi dan Akronim.....	197
1.4	Referensi	198
2	Perancangan Sistem.....	199
2.1.	Perancangan Arsitektur	199
2.2.	Perancangan Rinci	200
2.2.1.	Squence Diagram.....	200
2.2.2.	Class Diagram.....	214
2.2.3.	Spesifikasi Deskripsi Kelas Diagram.....	215
3	Perancangan Data.....	225
3.1	Dekomposisi Data.....	94
3.1.1.	Deskripsi Entitas Data Role	94
3.1.2.	Deskripsi Entitas Data User	94
3.1.3.	Deskripsi Entitas Data Gejala	95
3.1.4.	Deskripsi Entitas Data Penyakit.....	95
3.1.5.	Deskripsi Entitas Data Aturan	95
3.1.5.	Deskripsi Entitas Data Periksa.....	95
3.1.6.	Deskripsi Entitas Data Daftar gejala.....	96
3.2	Physical Data Model	227
4	Perancangan Antarmuka.....	228
4.1.	Sketsa UI dan Deskripsinya.....	228

DAFTAR GAMBAR

Gambar 2.1 Perancangan Arsitektur.....	199
Gambar 2.2 Sequence Diagram : Login.....	69
Gambar 2.3 Sequence Diagram : Reset password pasien.....	69
Gambar 2.4 Sequence Diagram : Add Data Admin.....	70
Gambar 2.5 Sequence Diagram : Delete Data Admin.....	70
Gambar 2.6 Sequence Diagram : Display Data Admin.....	71
Gambar 2.7 Sequence Diagram : Add Data Gejala.....	71
Gambar 2.8 Sequence Diagram : Edit Data Gejala.....	72
Gambar 2.9 Sequence Diagram : Delete Data Gejala.....	72
Gambar 2.10 Sequence Diagram : Display Data Gejala.....	73
Gambar 2.11 Sequence Diagram : Search Data Gejala.....	73
Gambar 2.12 Sequence Diagram : Add Data Penyakit.....	74
Gambar 2.13 Sequence Diagram : Edit Data Penyakit.....	74
Gambar 2.14 Sequence Diagram : Delete Data Penyakit.....	75
Gambar 2.15 Sequence Diagram : Display Data Penyakit.....	75
Gambar 2.16 Sequence Diagram : Search Data Penyakit.....	76
Gambar 2.17 Sequence Diagram : Add Data Aturan.....	76
Gambar 2.18 Sequence Diagram : Delete Data Aturan.....	77
Gambar 2.19 Sequence Diagram : Display Data Aturan.....	77
Gambar 2.20 Sequence Diagram : Search Data Aturan.....	78
Gambar 2.21 Sequence Diagram : Menampilkan Home.....	78
Gambar 2.22 Sequence Diagram : Informasi kornea mata.....	79
Gambar 2.23 Sequence Diagram : Menampilkan Informasi Penyakit Mata.....	79

Gambar 2.24 Sequence Diagram : Register Pasien.....	80
Gambar 2.25 Sequence Diagram : Recam Medic	80
Gambar 2.26 Sequence Diagram : Konsultasi.....	81
Gambar 2.27 Sequence Diagram : Hasil Diagnosa Penyakit.....	81
Gambar 2.28 Sequence Diagram : Change Password.....	82
Gambar 2.29 Class Diagram.....	214
3 Perancangan Data.....	225
Gambar 3.1 Pysical Data Model	227
4 Perancangan Antarmuka.....	228



1 Pendahuluan

1.1 Tujuan

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

1.2 Ruang Lingkup

Perangkat Lunak SIPMA dikembangkan dengan tujuan untuk :

1. Membangun sistem pakar yang dapat memberikan informasi jenis penyakit mata berdasarkan gejala yang diberikan dengan menggunakan metode *Forward Chaining*.
2. Membangun sistem pakar yang dapat memberikan solusi penanganan penyakit mata.

1.3 Definisi dan Akronim

Beberapa definisi, akronim, singkatan yang digunakan dalam DPPL SIPMA ini antara lain adalah sebagai berikut:

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
DPPL-SIPMA-XXX	Kode yang merepresentasikan kebutuhan pada SIPMA (Sistem Pakar Mata) dimana XXX merupakan nomor fungsi produk.
SIPMA	Perangkat lunak sistem pakar diagnosis penyakit mata.
GUI	<i>Graphical user Interface</i> , tampilan antarmuka program bagi pengguna.
Database	Kumpulan data terkait yang diorganisasikan dalam struktur tertentu dan dapat diakses

	dengan cepat.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk <i>e-mail</i> , <i>FTP</i> , dan <i>World Wide Web</i> .
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

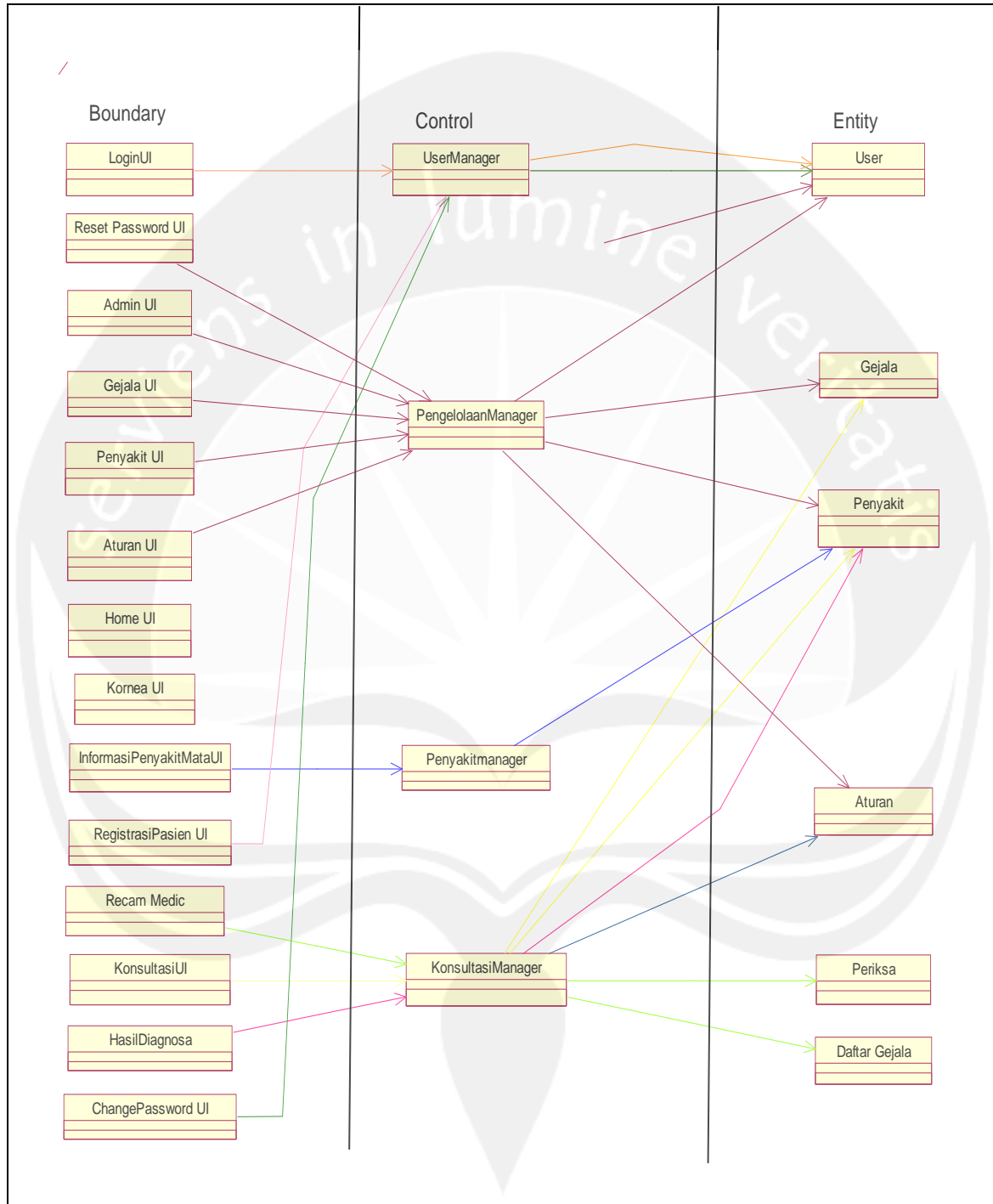
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah :

1. Bennet Simon, Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.
3. Deitel, *C# How to Program*, Prentice-Hall Inc, 2002.
4. Raelda, *Spesifikasi Kebutuhan Perangkat Lunak CSDS*, Universitas Atma Jaya Yogyakarta, 2013.

2. Perancangan Sistem

2.1 Perancangan Arsitektur

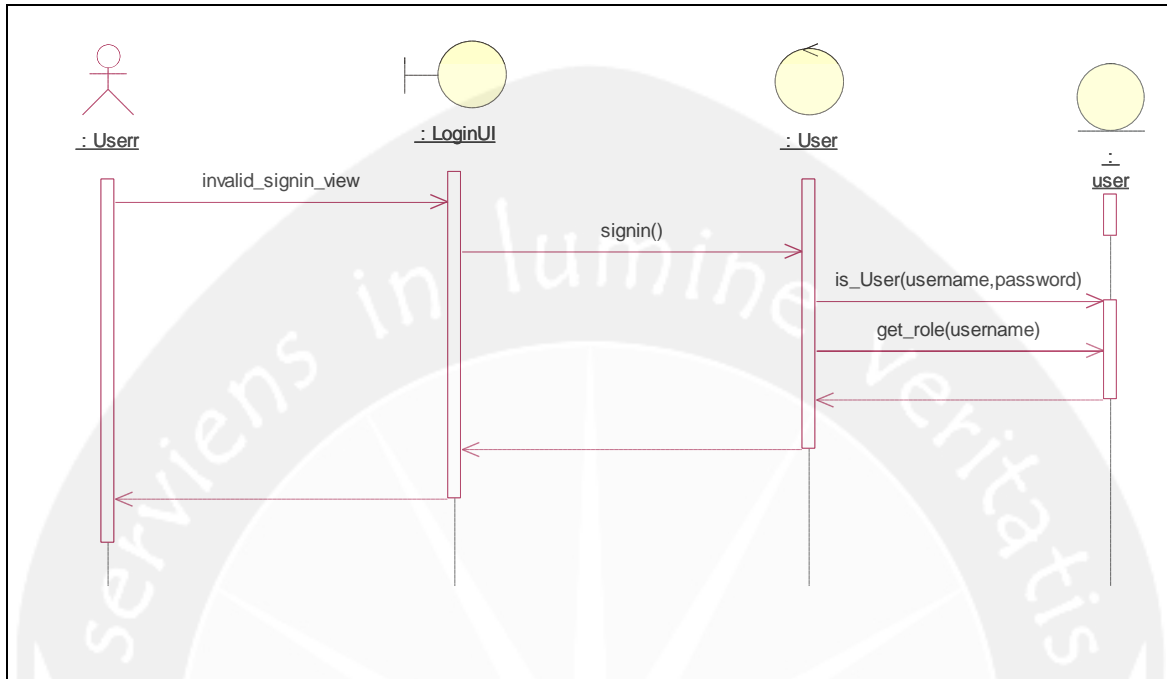


Gambar 2.1 Perancangan Arsitektur

2.2. Perancangan Rinci

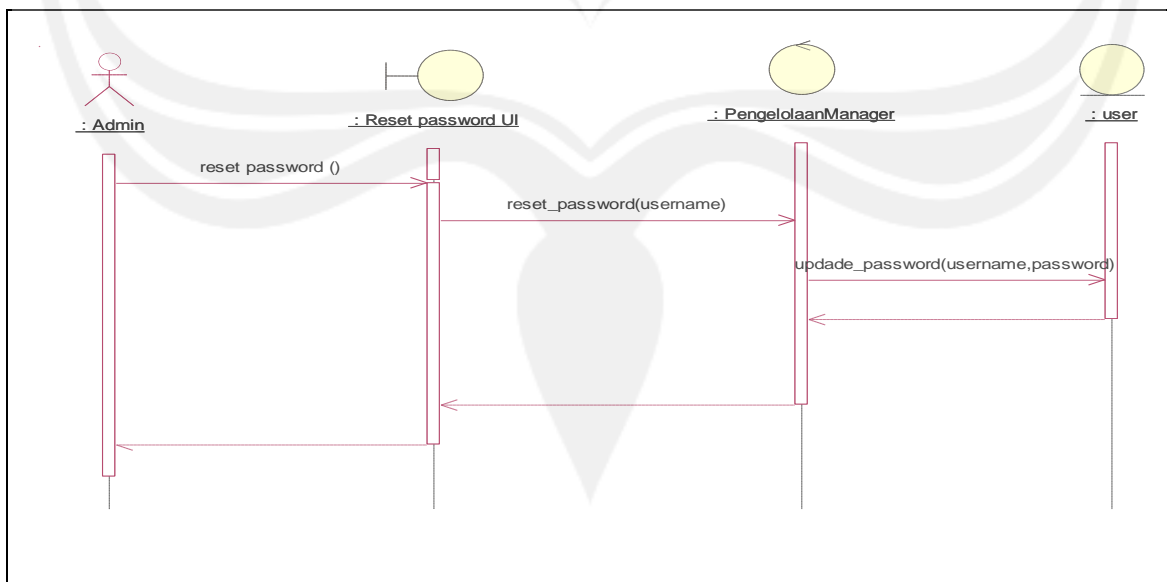
2.2.1. Squence Diagram

2.2.1.2. Login



Gambar 2.2 Sequence Diagram : Login

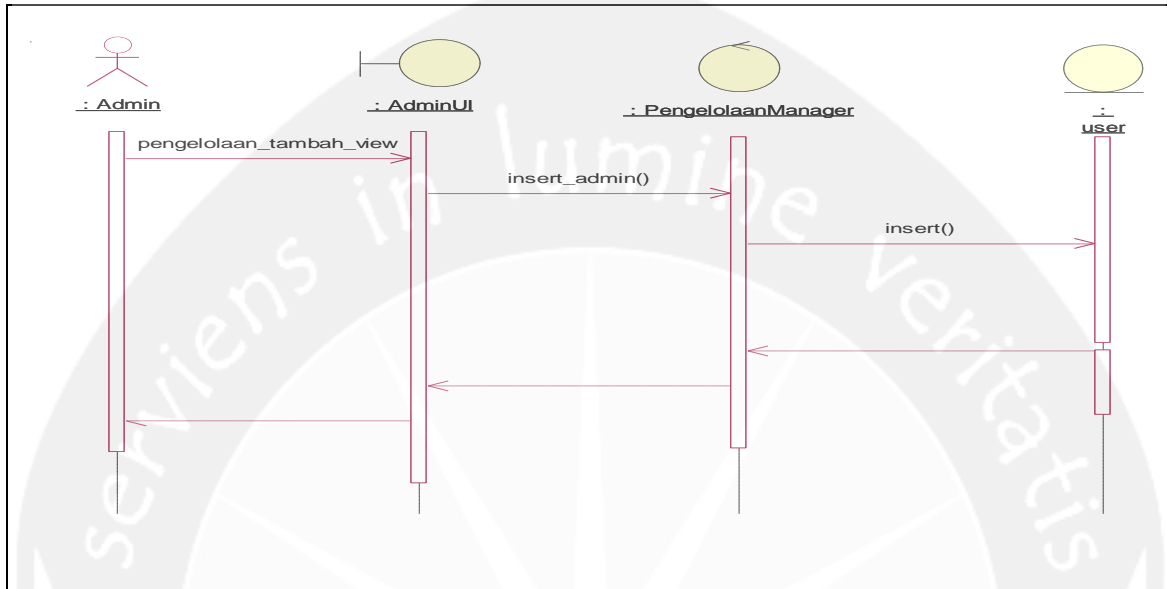
2.2.1.3. Reset Password Pasien



Gambar 2.3 Sequence Diagram : Reset password pasien

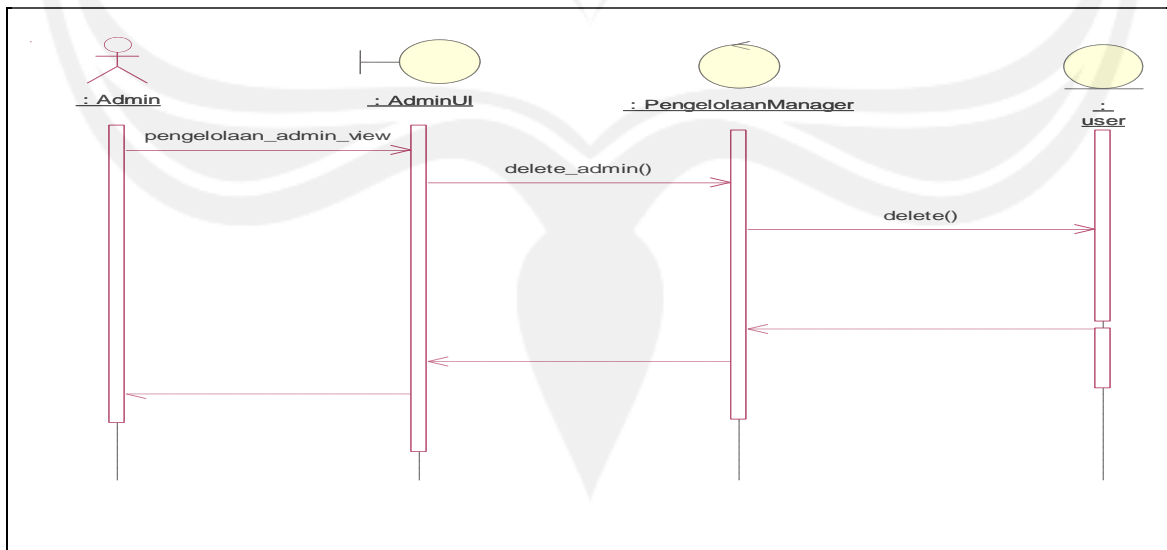
2.2.1.4. Mengelola Data Admin

2.2.1.4.1. Add Data Admin



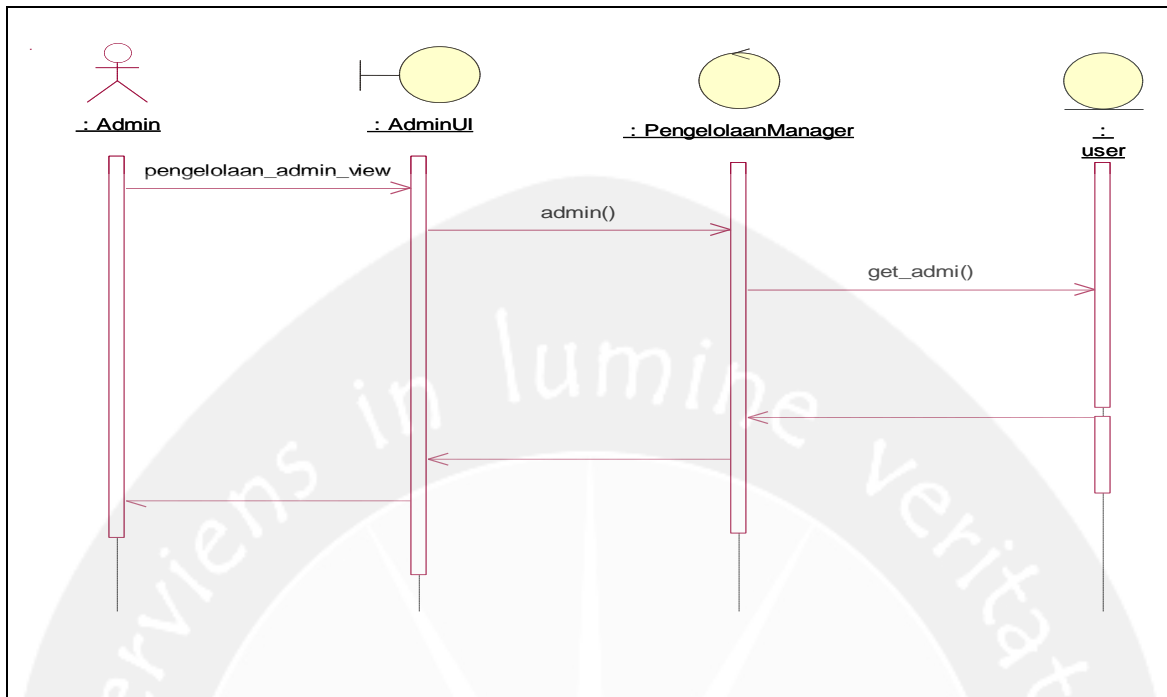
Gambar 2.4 Sequence Diagram : Add Data Admin

2.2.1.4.2. Delete Data Admin



Gambar 2.5 Sequence Diagram : Delete Data Admin

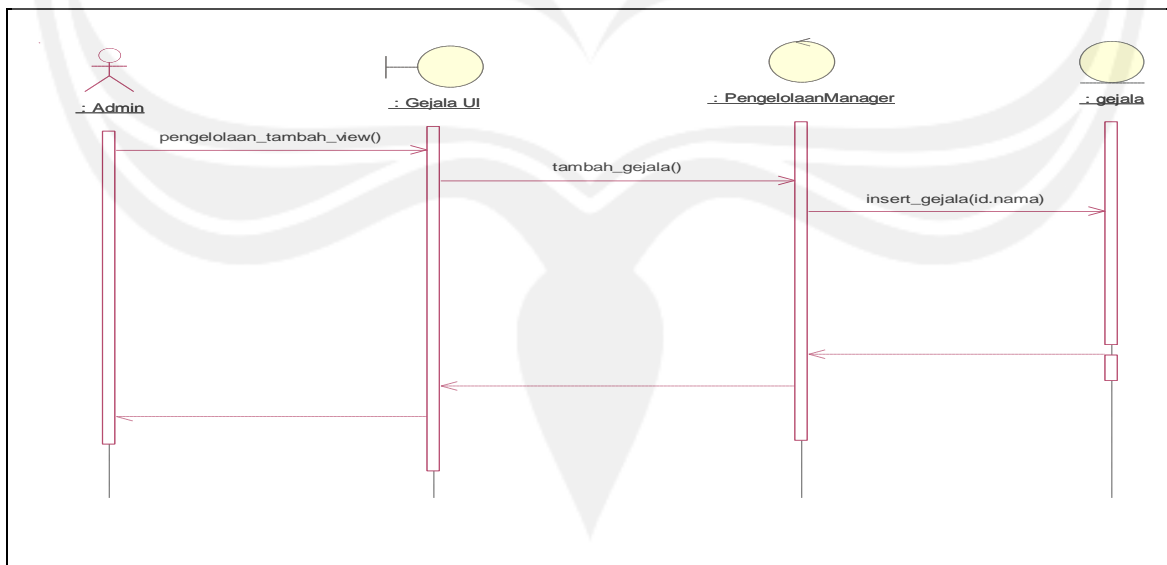
2.2.1.4.3. Display Data Admin



Gambar 2.6 Sequence Diagram : Display Data Admin

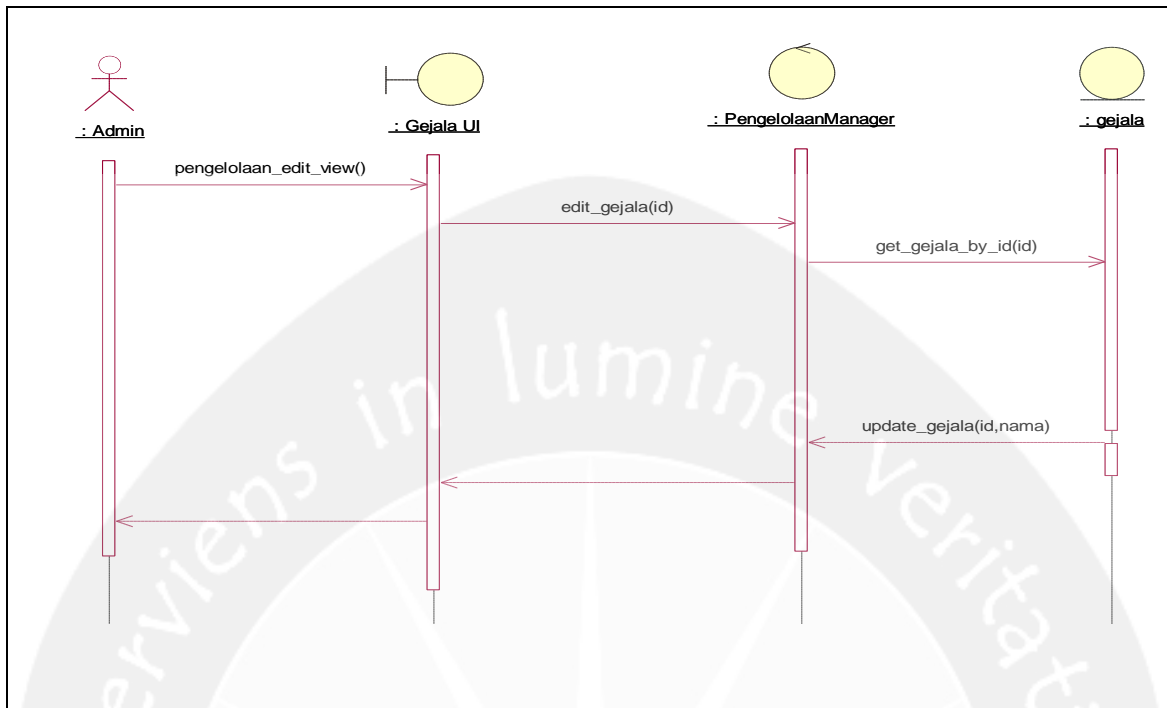
2.2.1.5. Mengelola Data Gejala

2.2.1.5.1. Add Data Gejala



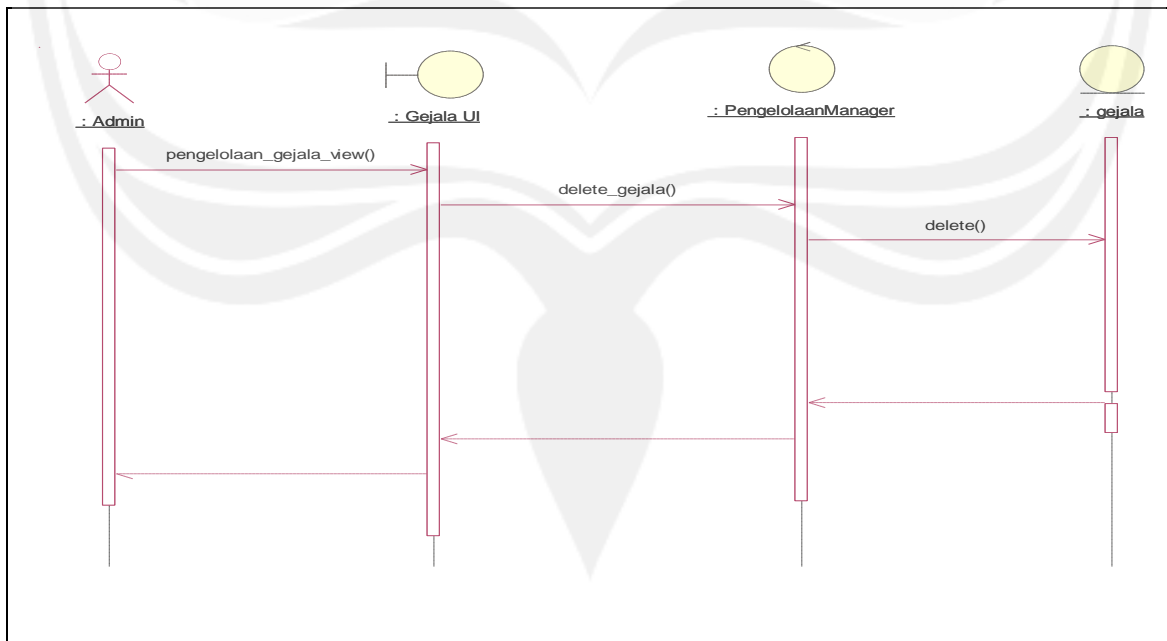
Gambar 2.7 Sequence Diagram : Add Data Gejala

2.2.1.5.2. Edit Data Gejala



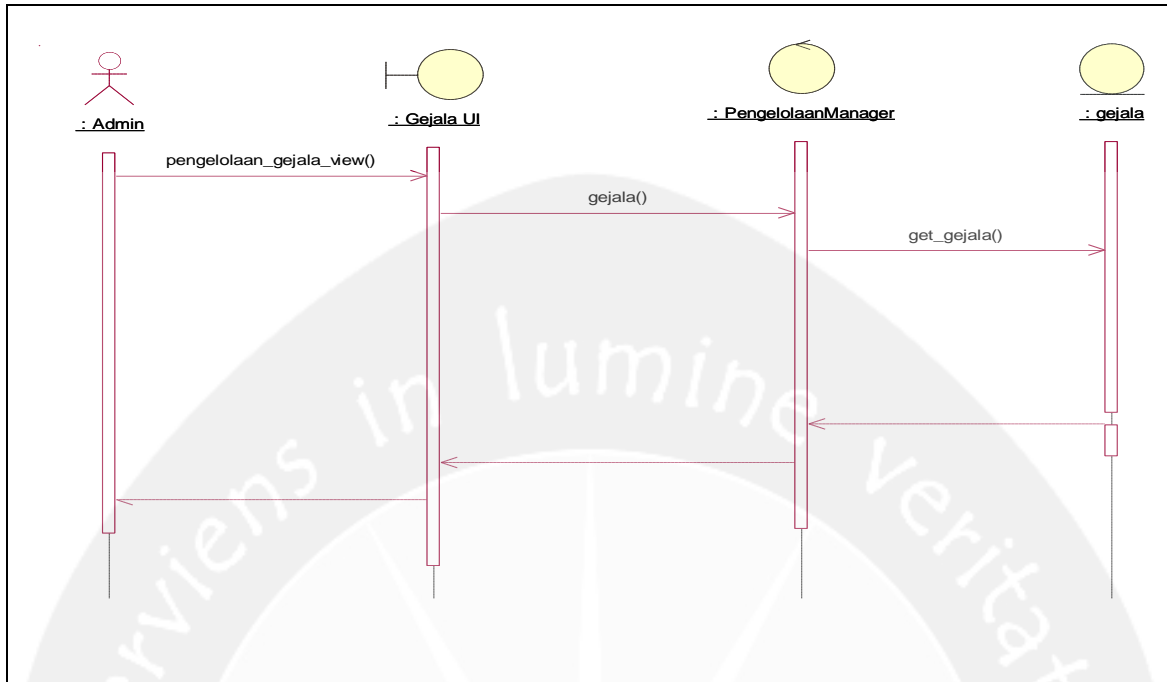
Gambar 2.8 Sequence Diagram : Edit Data Gejala

2.2.1.5.3. Delete Data Gejala



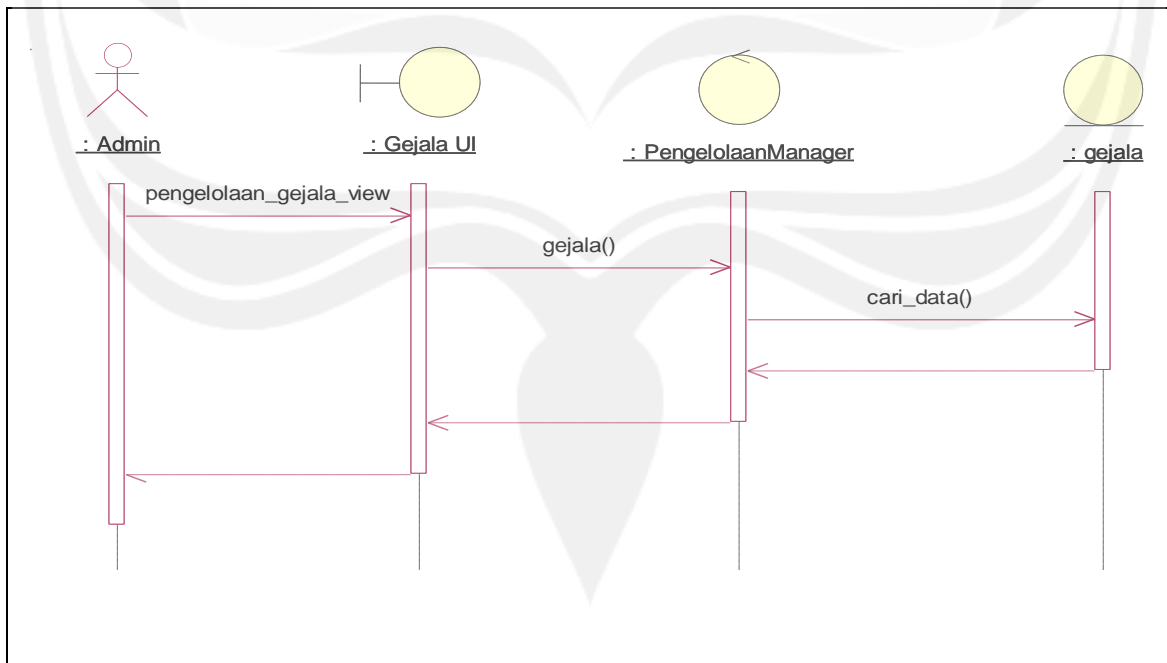
Gambar 2.9 Sequence Diagram : Delete Data Gejala

2.2.1.5.4. Display Data Gejala



Gambar 2.10 Sequence Diagram : Display Data Gejala

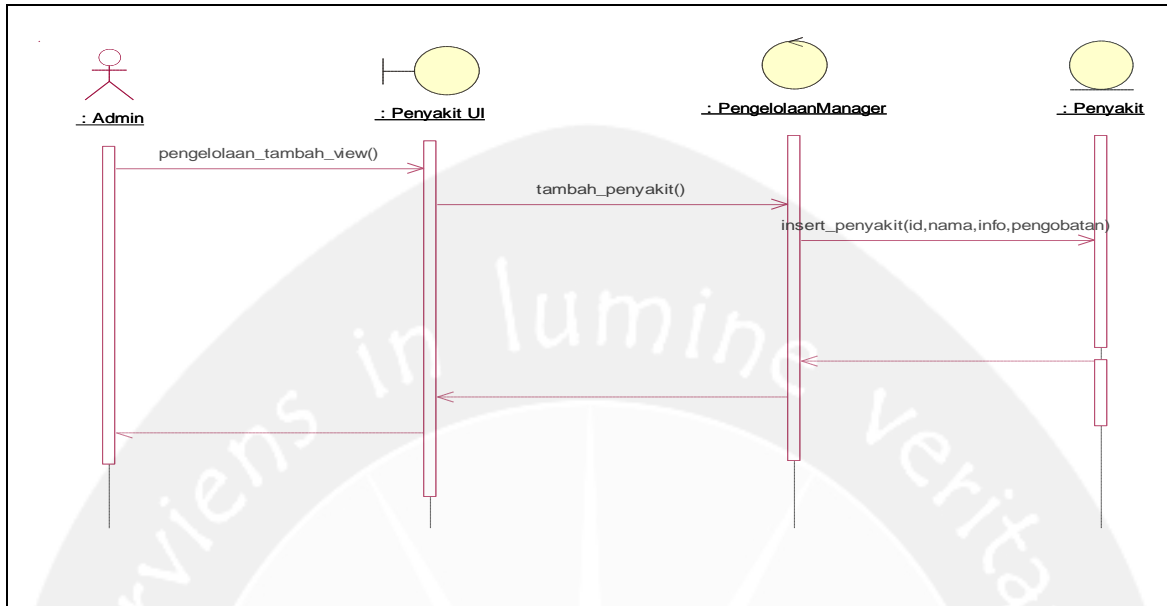
2.2.1.5.5. Search Data Gejala



Gambar 2.11 Sequence Diagram : Search Data Gejala

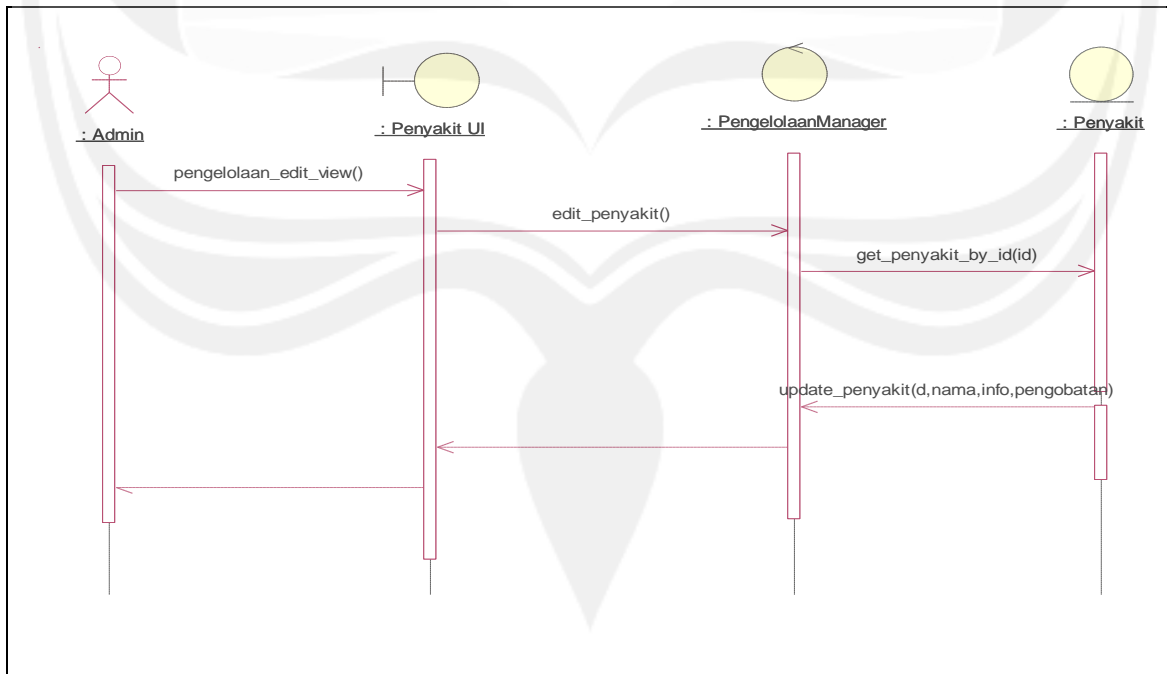
2.2.1.6. Mengelola Data Penyakit

2.2.1.6.1. Add Data Penyakit



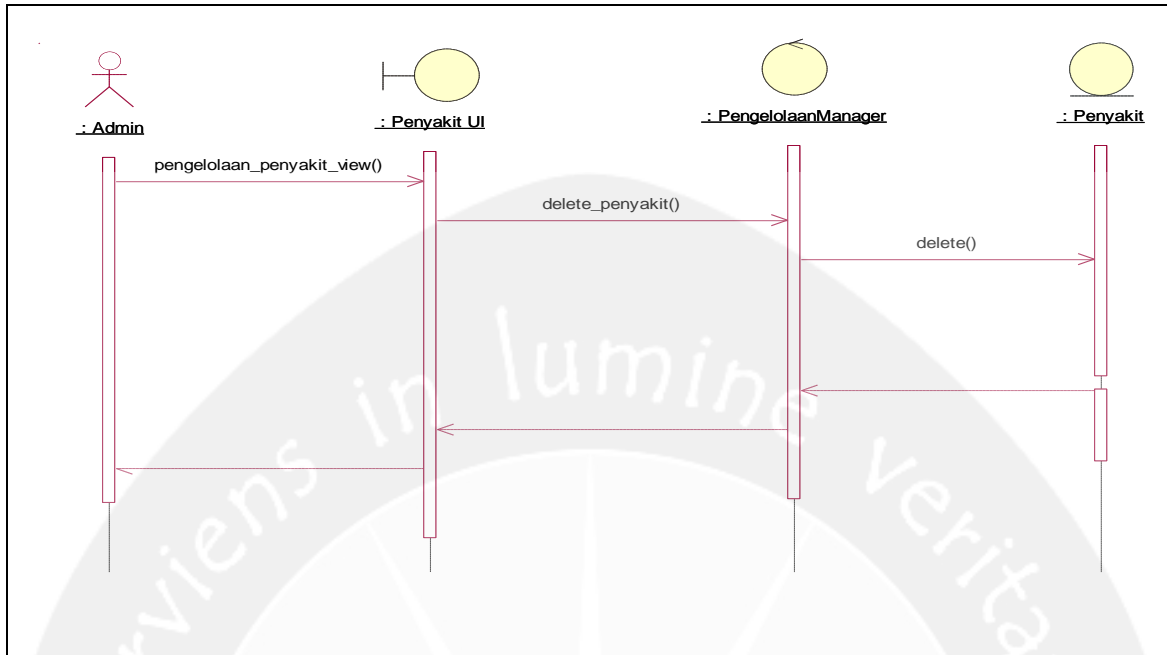
Gambar 2.12 Sequence Diagram : Add Data Penyakit

2.2.1.6.2. Edit Data Penyakit



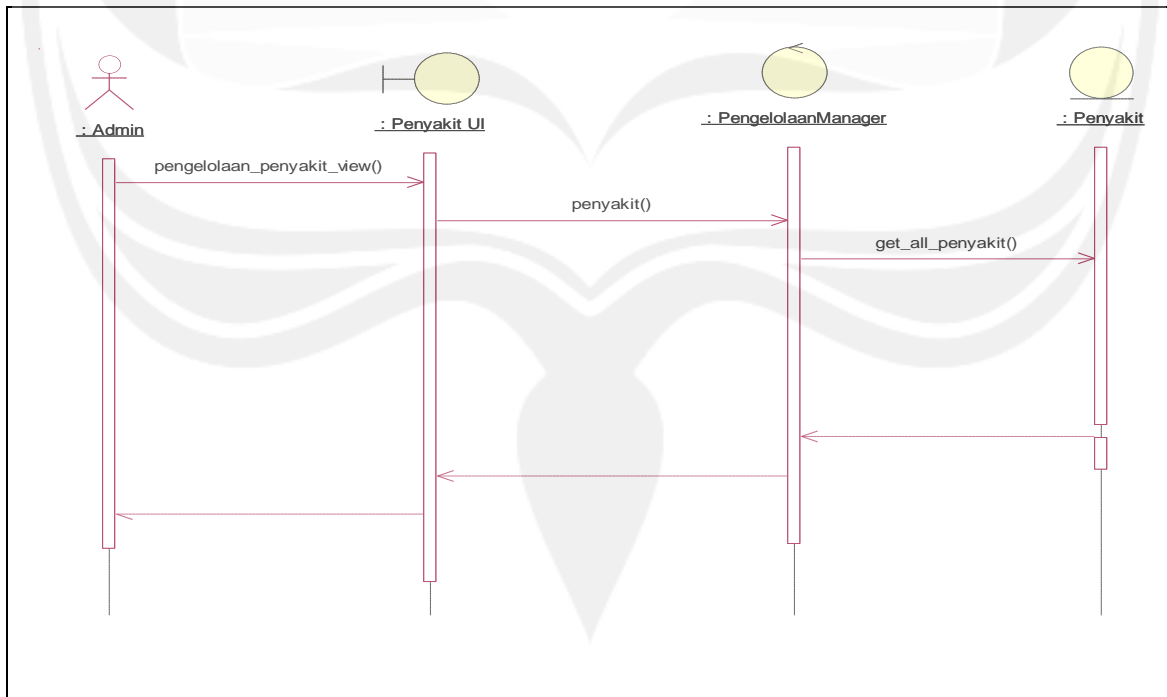
Gambar 2.13 Sequence Diagram : Edit Data Penyakit

2.2.1.6.3. Delete Data Penyakit



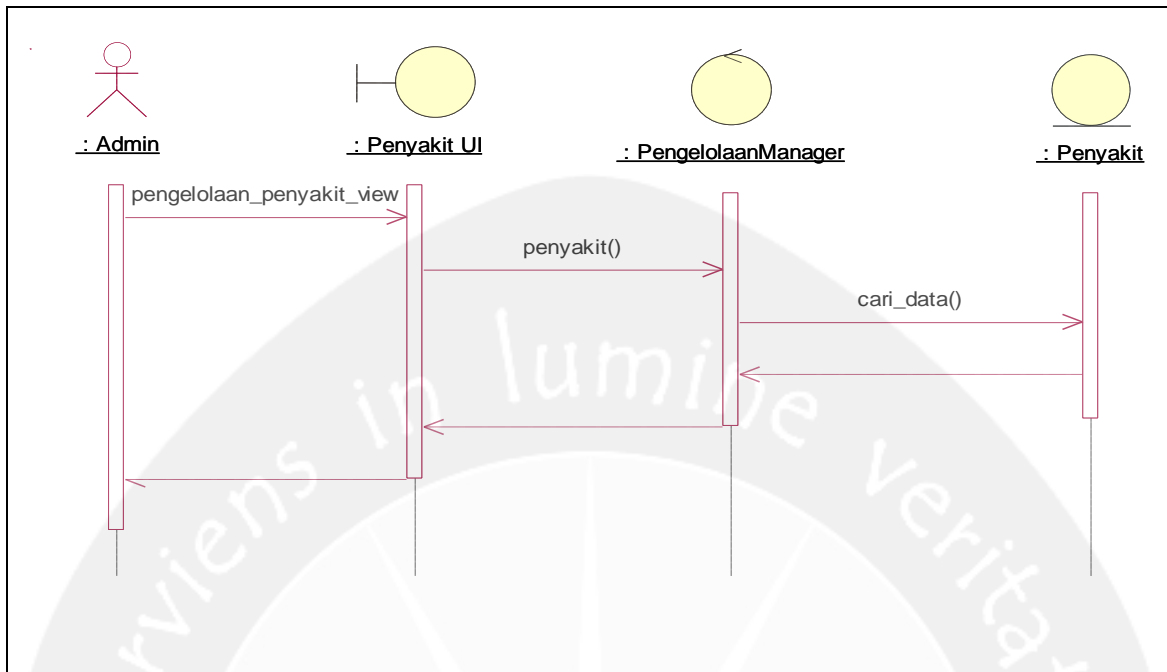
Gambar 2.14 Sequence Diagram : Delete Data Penyakit

2.2.1.6.4. Display Data Penyakit



Gambar 2.15 Sequence Diagram : Display Data Penyakit

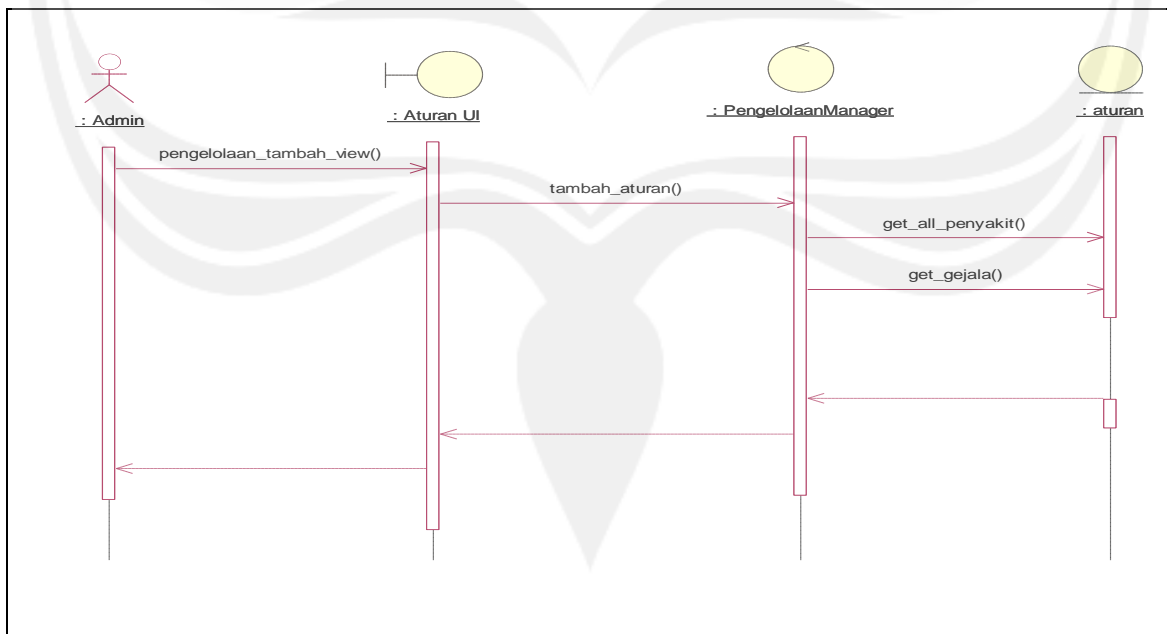
2.2.1.6.5. Search Data Penyakit



Gambar 2.16 Sequence Diagram : Search Data Penyakit

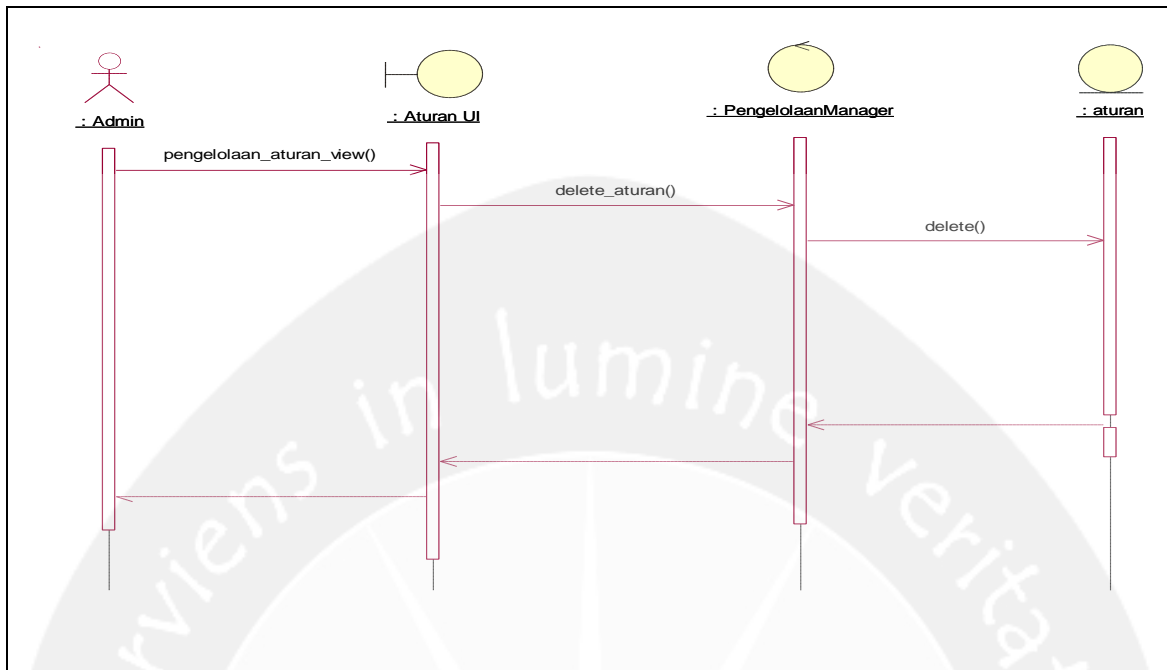
2.2.1.7. Mengelola Data Aturan

2.2.1.7.5. Add Data Aturan



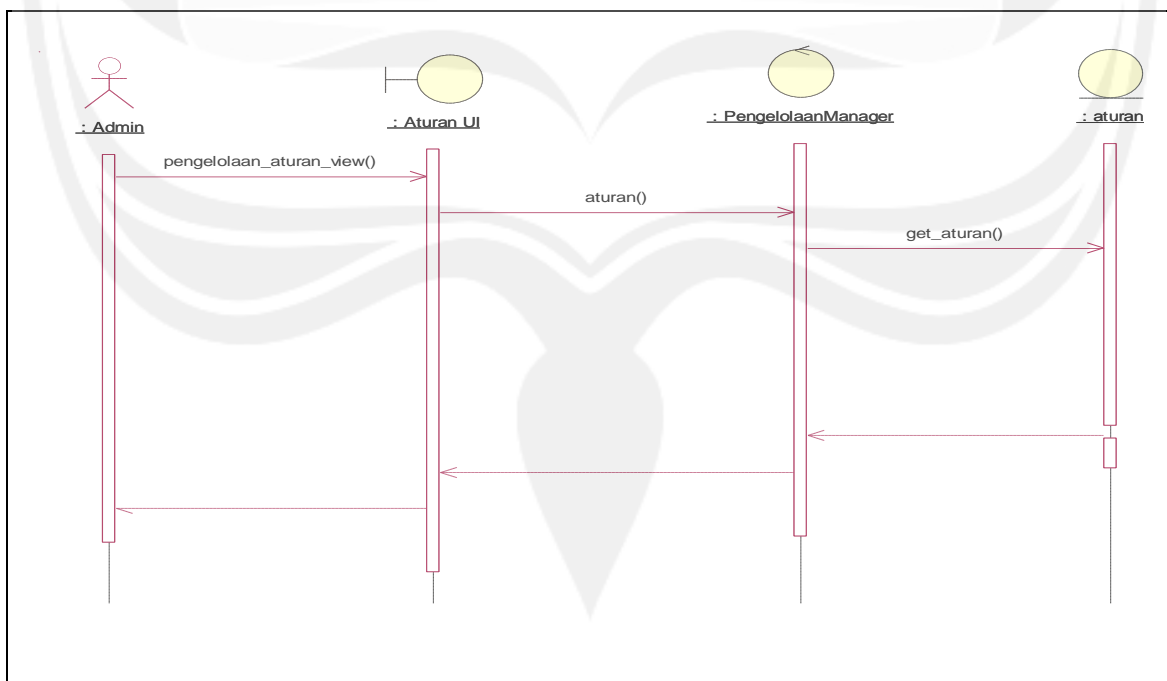
Gambar 2.17 Sequence Diagram : Add Data Aturan

2.2.1.7.6. Delete Data Aturan



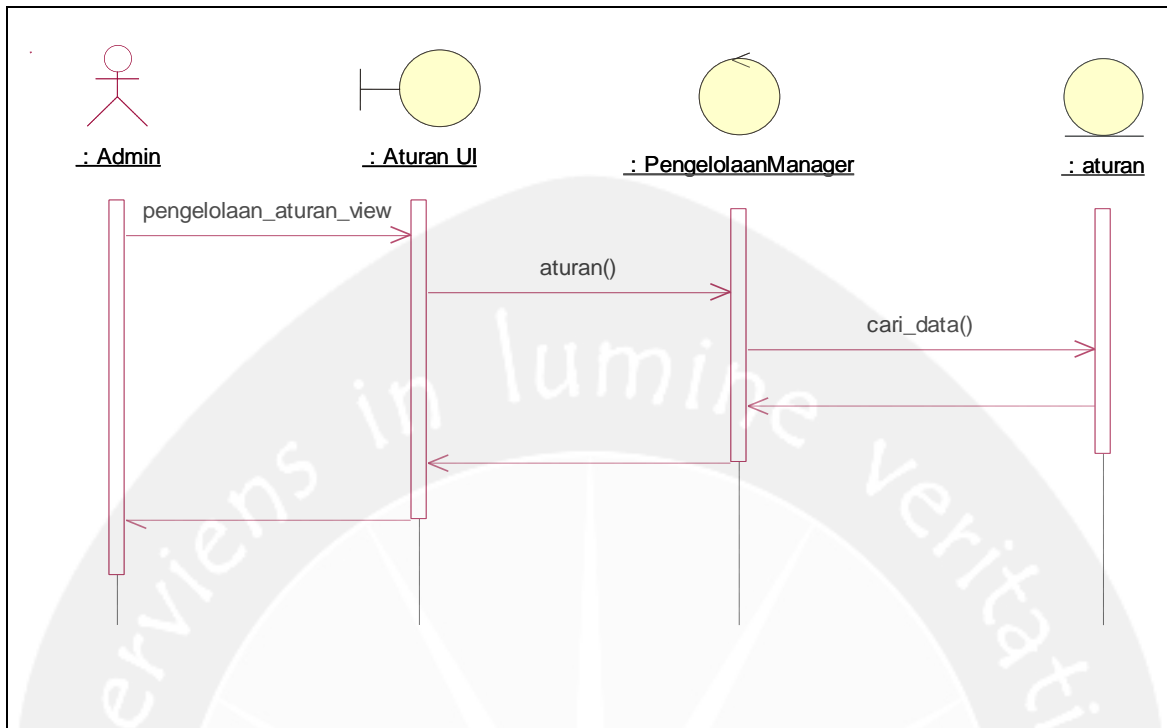
Gambar 2.18 Sequence Diagram : Delete Data Aturan

2.2.1.7.7. Display Data Aturan



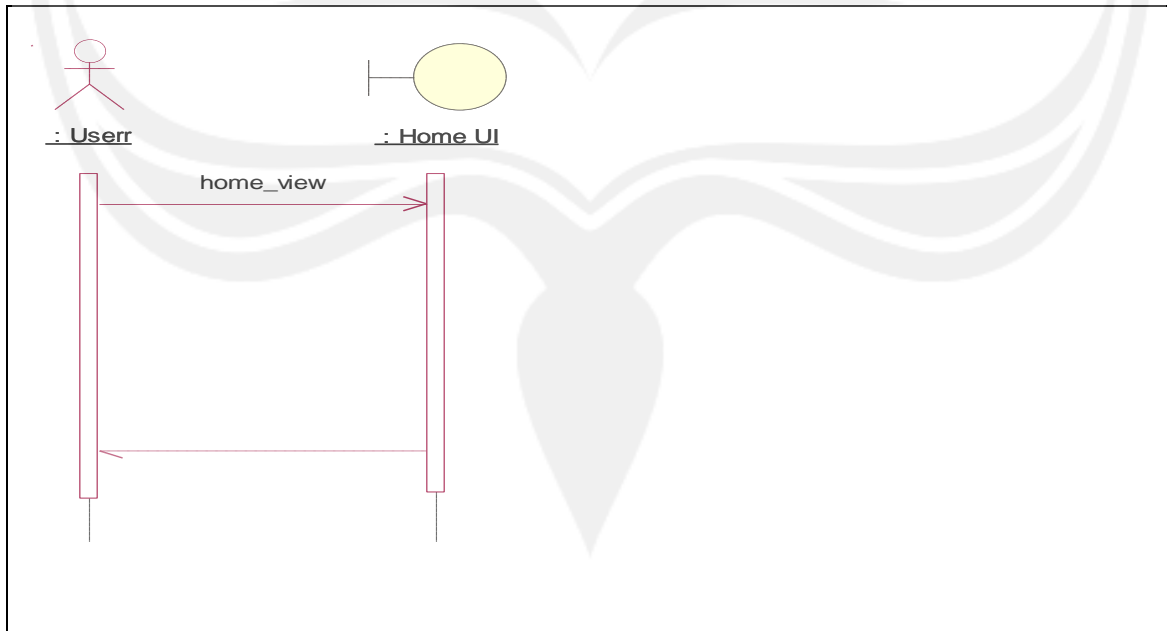
Gambar 2.19 Sequence Diagram : Display Data Aturan

2.2.1.7.8. Search Data Aturan



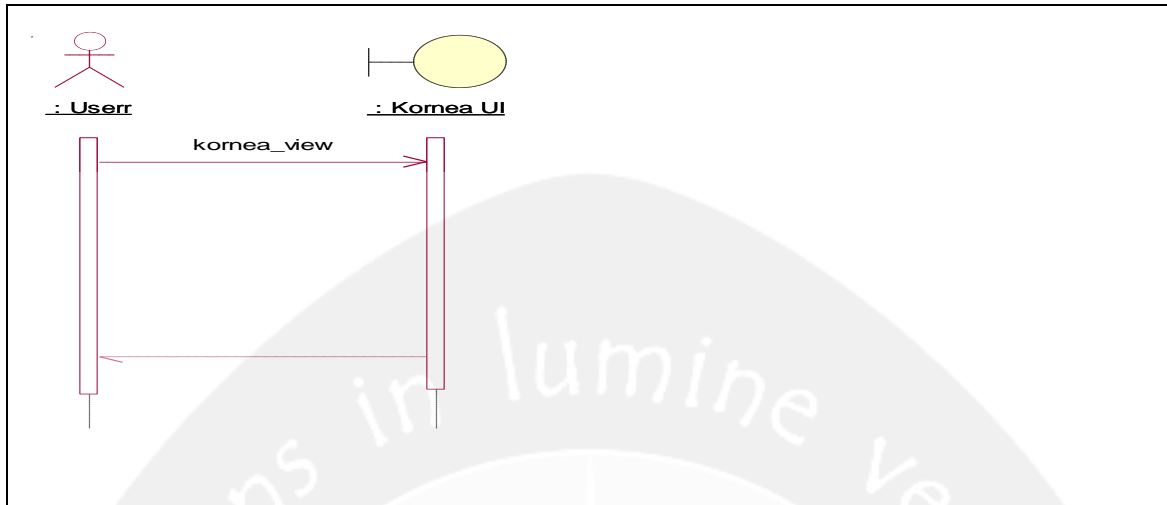
Gambar 2.20 Sequence Diagram : Search Data Aturan

2.2.1.8. Menampilkan Home



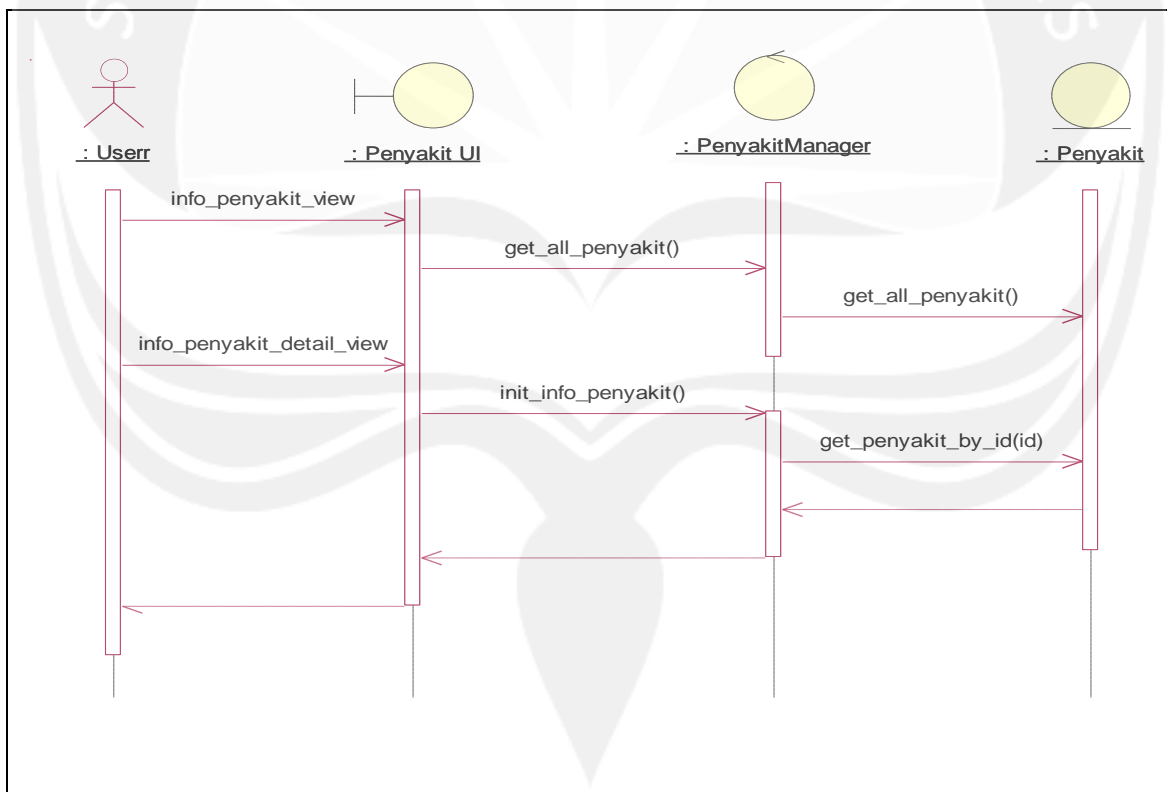
Gambar 2.21 Sequence Diagram : Menampilkan Home

2.2.1.9. Menampilkan Kornea mata



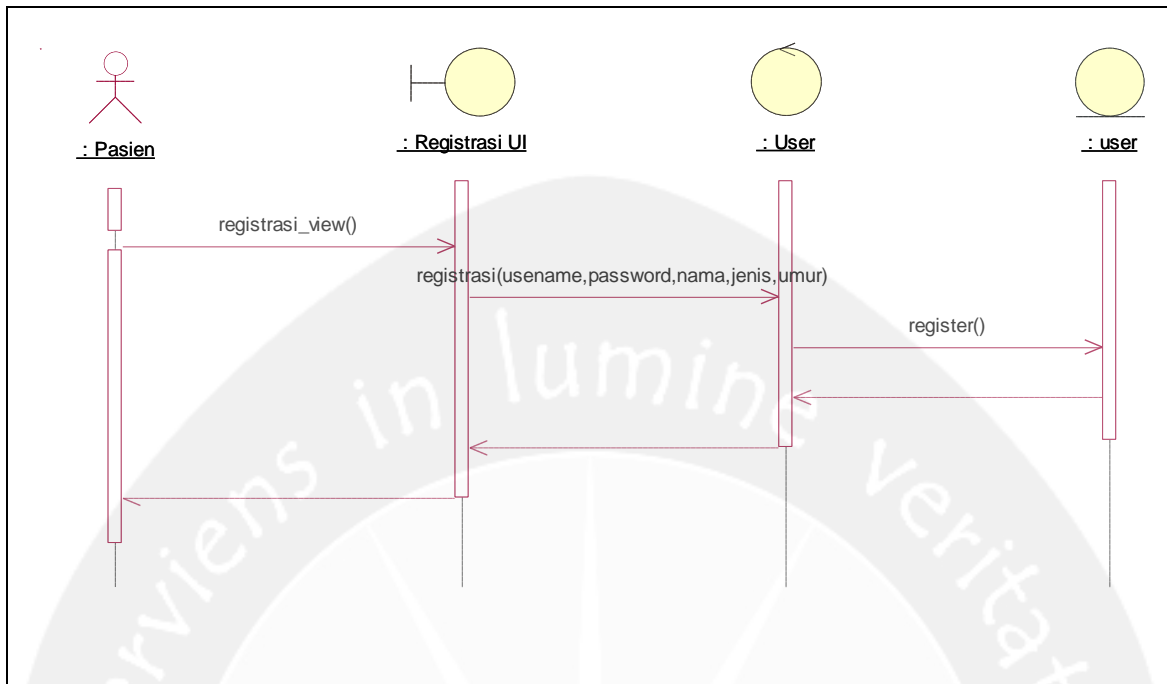
Gambar 2.22 Sequence Diagram : Informasi kornea mata

2.2.1.10. Menampilkan Informasi Penyakit Mata



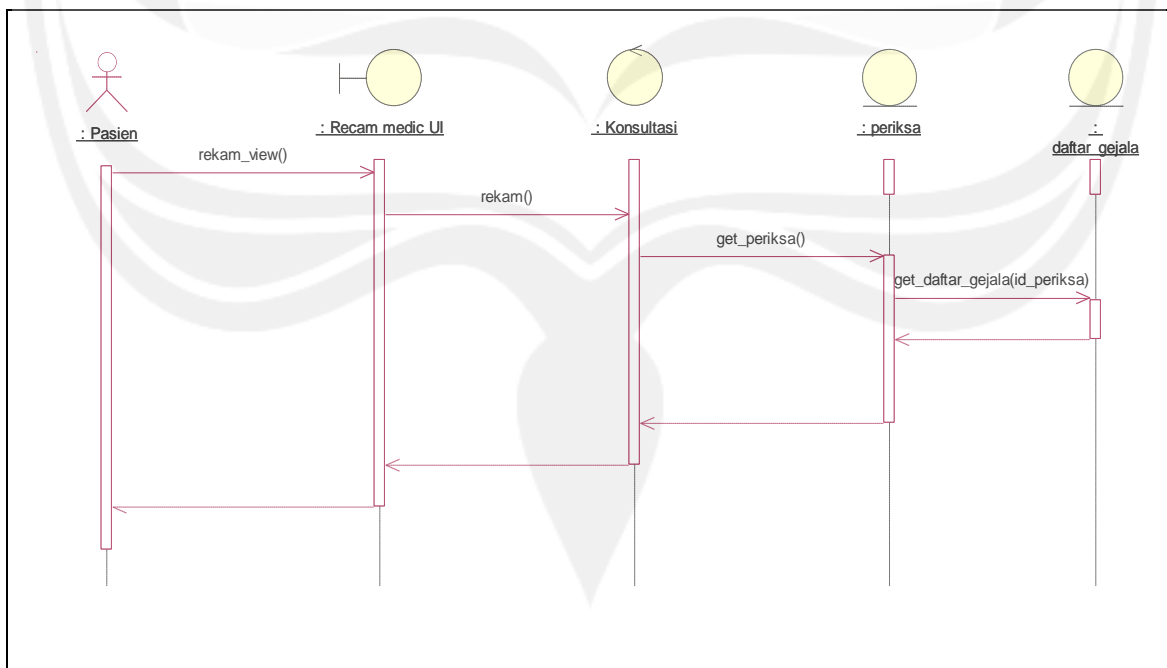
Gambar 2.23 Sequence Diagram : Menampilkan Informasi Penyakit Mata

2.2.1.11. Registrasi Pasien



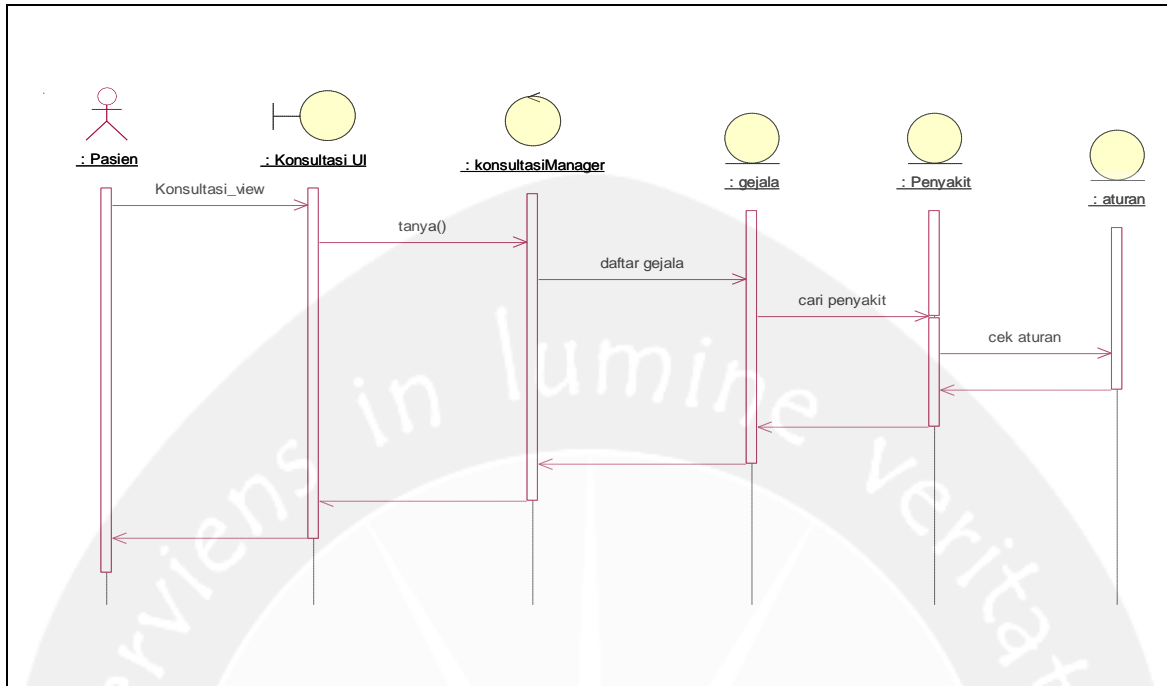
Gambar 2.24 Sequence Diagram : Register Pasien

2.2.1.12. Recam Medic



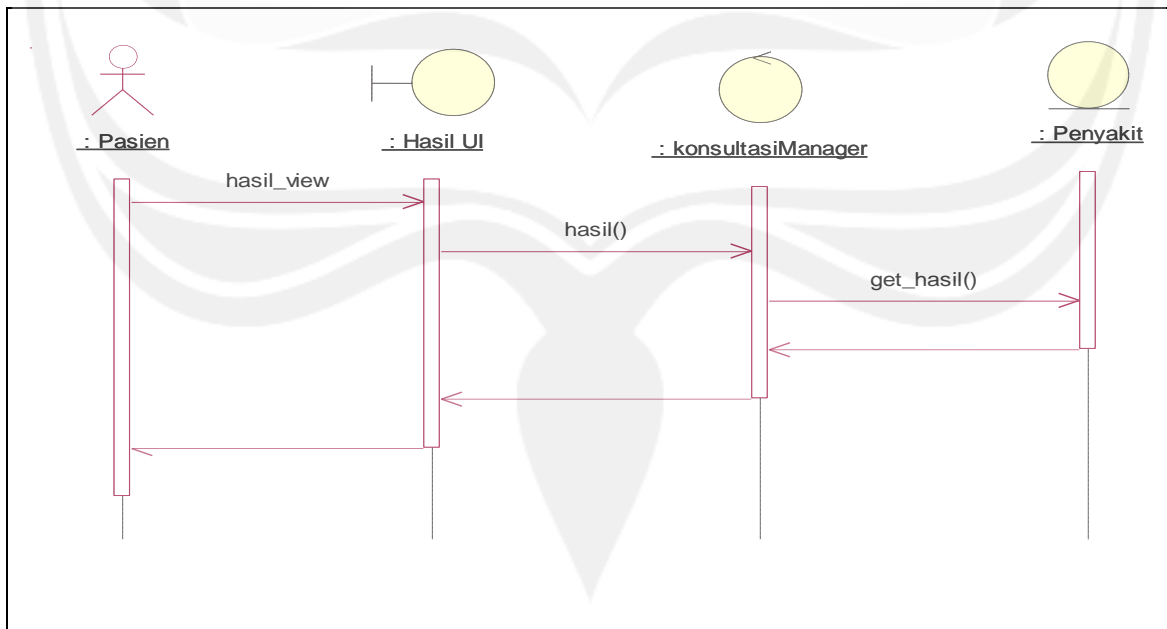
Gambar 2.25 Sequence Diagram : Recam Medic

2.2.1.13. Konsultasi



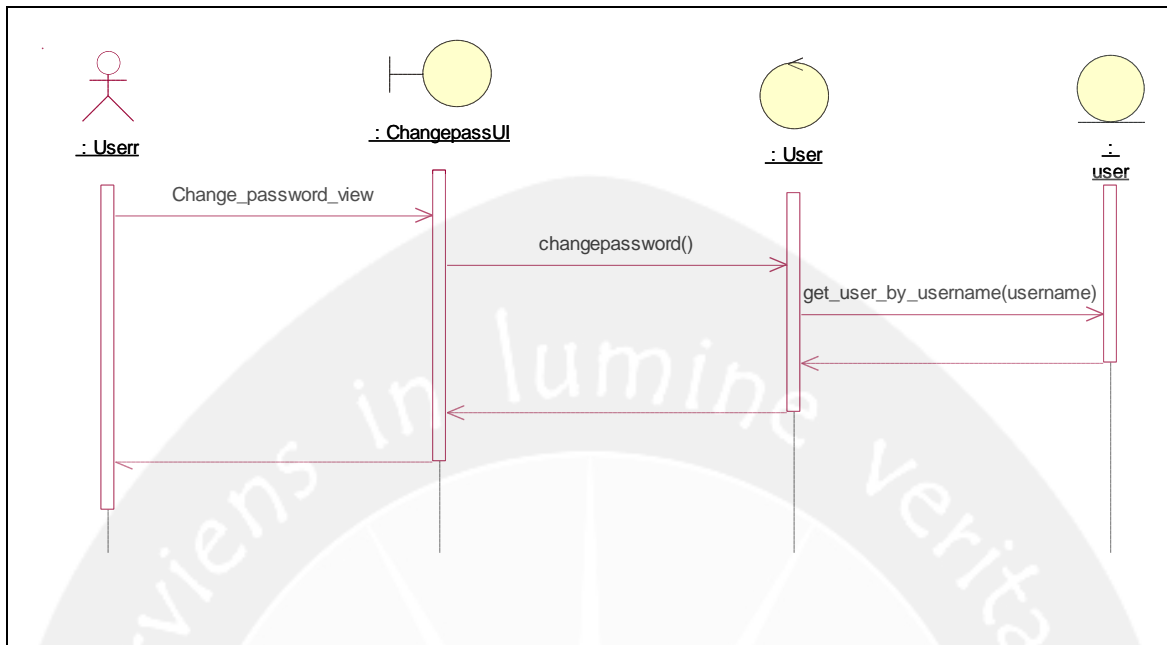
Gambar 2.26 Sequence Diagram : Konsultasi

2.2.1.14. Hasil Diagnosa



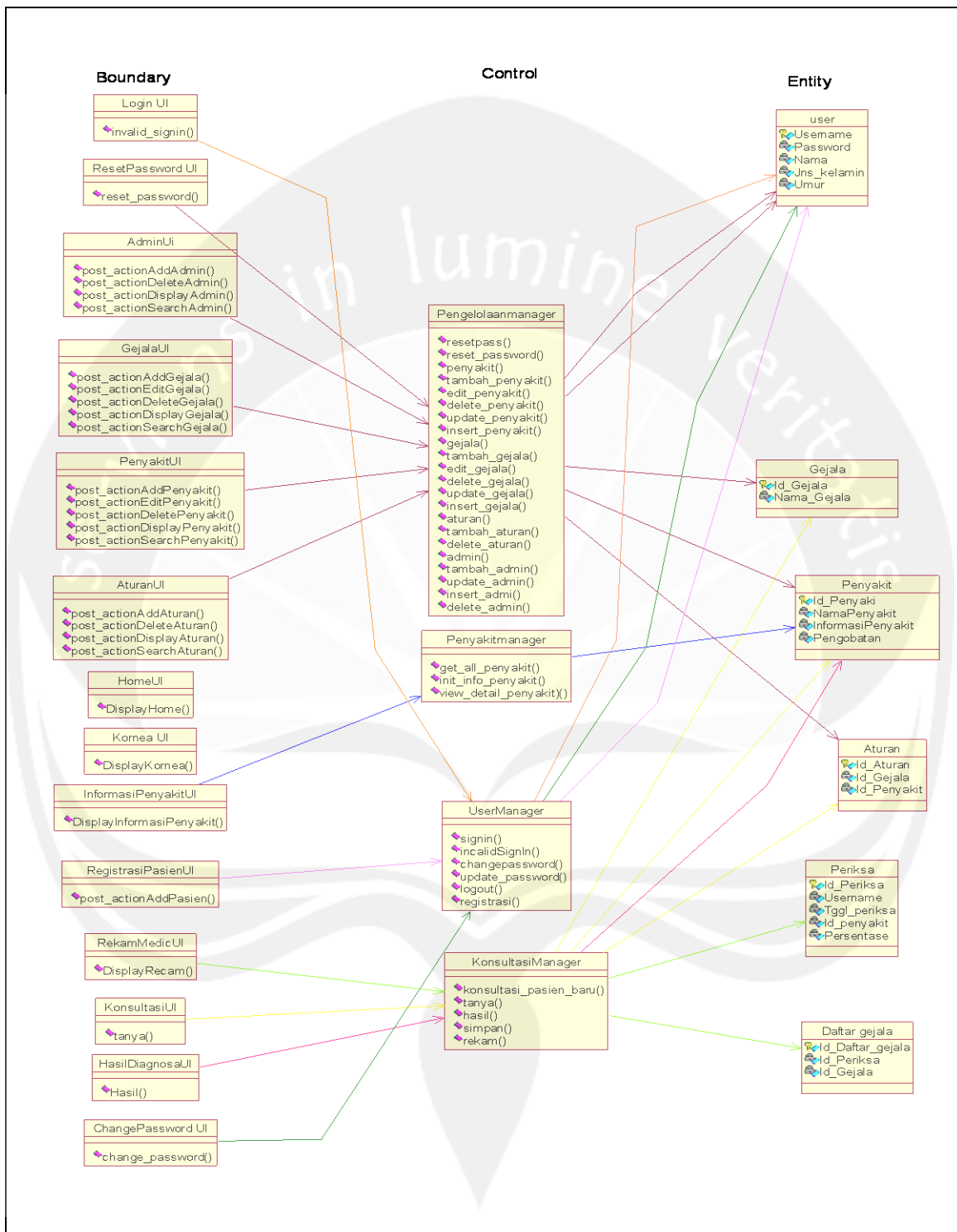
Gambar 2.27 Sequence Diagram : Hasil Diagnosa Penyakit

2.2.1.15. Change Password



Gambar 2.28 Sequence Diagram : Change Password

2.2.2. Class Diagram



Gambar 2.29 Class Diagram

2.2.3. Spesifikasi Deskripsi Kelas Diagram

2.2.3.1. Spesifikasi Design Kelas : LoginUI

LoginUI	<<boundary>>
+Invalid_signin() Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

2.2.3.2. Spesifikasi Design Kelas : ResetPasswordUI

ResetPasswordUI	<<boundary>>
+reset_password() Operasi ini digunakan untuk mengganti lama pasien menjadi password standar oleh admin	

2.2.3.3. Spesifikasi Design Kelas : AdmindUI

AdminUI	<<boundary>>
+post_actionAddAdmin() Operasi ini digunakan untuk mengirim data admin yang ditambahkan admin	
+post_actionEditAdmin() Operasi ini digunakan untuk mengirim data admin yang diedit admin	
+post_actionDeletelaAdmin() Operasi ini digunakan untuk mengirim data admin yang didelete admin	
+post_actionDisplayAdmin() Operasi ini digunakan untuk mengirim data admin untuk ditampilkan admin	
+post_actionSearchAdmin() Operasi ini digunakan untuk mengirim data admin yang dicari data	

admin

2.2.3.4. Spesifikasi Design Kelas : GejalaUI

GejalaUI	<<boundary>>
<pre>+post_actionAddGejala() Operasi ini digunakan untuk mengirim data gejala yang ditambahkan admin</pre>	
<pre>+post_actionEditGejala() Operasi ini digunakan untuk mengirim data gejala yang diedit admin</pre>	
<pre>+post_actionDeletelaGejala() Operasi ini digunakan untuk mengirim data gejala yang didelete admin</pre>	
<pre>+post_actionDisplayGejala() Operasi ini digunakan untuk mengirim data gejala untuk ditampilkan admin</pre>	
<pre>+post_actionSearchGejala() Operasi ini digunakan untuk mengirim data gejala yang dicari data admin</pre>	

2.2.3.5. Spesifikasi Design Kelas : PenyakitUI

PenyakitUI	<<boundary>>
<pre>+post_actionAddPenyakit() Operasi ini digunakan untuk mengirim data penyakit yang ditambahkan admin</pre>	
<pre>+post_actionEditPenyakit() Operasi ini digunakan untuk mengirim data penyakit yang diedit admin</pre>	
<pre>+post_actionDeletelaPenyakit() Operasi ini digunakan untuk mengirim data penyakit yang didelete</pre>	

<p>admin</p> <p>+post_actionDisplayPenyakit()</p> <p>Operasi ini digunakan untuk mengirim data penyakit untuk ditampilkan admin</p> <p>+post_actionSearchPenyakit()</p> <p>Operasi ini digunakan untuk mengirim data penyakit yang dicari data admin</p>
--

2.2.3.6. Spesifikasi Design Kelas : AturanUI

AturanUI	<<boundary>>
<p>+post_actionAddAturan()</p> <p>Operasi ini digunakan untuk mengirim data aturan yang ditambahkan admin</p> <p>+post_actionEditAturan()</p> <p>Operasi ini digunakan untuk mengirim data aturan yang diedit admin</p> <p>+post_actionDeletelaAturan()</p> <p>Operasi ini digunakan untuk mengirim data aturan yang didelete admin</p> <p>+post_actionDisplayAturan()</p> <p>Operasi ini digunakan untuk mengirim data aturan untuk ditampilkan admin</p> <p>+post_actionSearchAturan()</p> <p>Operasi ini digunakan untuk mengirim data aturan yang dicari data admin</p>	

2.2.3.7. Spesifikasi Design Kelas : HomeUI

HomeUI	<<boundary>>
<p>+DisplayHome()</p> <p>Operasi ini digunakan untuk menampilkan halaman home.</p>	

2.2.3.8. Spesifikasi Design Kelas : KorneaUI

KorneaUI	<<boundary>>
+DisplayKornea() Operasi ini digunakan untuk menampilkan halaman kornea.	

2.2.3.9. Spesifikasi Design Kelas : InformasiPenyakitUI

InformasiPenyakitUI	<<boundary>>
+DisplayInformasiPenyakit() Operasi ini digunakan untuk menampilkan halaman informasi.	

2.2.3.10. Spesifikasi Design Kelas : RegisterPasienUI

RegistrasiPasienUI	<<boundary>>
*post_actionAddPasien() Operasi ini digunakan untuk mengirim data registrasi yang ditambahkan pasien	

2.2.3.11. Spesifikasi Design Kelas : RekamMedicUI

RekamMedicUI	<<boundary>>
+DisplayRecam() Operasi ini digunakan untuk menampilkan halaman informasi.	

2.2.3.12. Spesifikasi Design Kelas : KonsultasiUI

KonsultasiUI	<<boundary>>
---------------------	---------------------------------

<p>+tanya ()</p> <p>Operasi ini digunakan untuk menjawab pertanyaan gejala yang dirasakan pasien.</p>

2.2.3.13. Spesifikasi Design Kelas : HasilDiagnosaUI

HasilDiagnosaUI	<<boundary>>
<p>+hasil ()</p> <p>Operasi ini digunakan untuk mengirim data hasil ke pasien</p>	

2.2.3.14. Spesifikasi Design Kelas : ChangePasswordUI

ChangePasswordUI	<<boundary>>
<p>+change_password ()</p> <p>Operasi ini digunakan untuk mengirim data change password yang diubah pasien</p>	

2.2.3.15. Spesifikasi Design Kelas : PengelolaanManager

PengelolaanManager	<<control>>
<p>+resetpass ()</p> <p>Operasi ini digunakan untuk menampilkan reset password</p> <p>+resetpassword ()</p> <p>Operasi ini digunakan untuk mereset password menjadi password standar '1234' oleh admin</p> <p>+penyakit ()</p> <p>Operasi ini digunakan untuk menampilkan data penyakit dan cari data penyakit didatabase</p>	

+tambah_penyakit()

Operasi ini digunakan untuk menampilkan tambah penyakit

+edit_penyakit()

Operasi ini digunakan untuk edit data penyakit didatabase

+delete_penyakit()

Operasi ini digunakan untuk delete data penyakit di database

+update_penyakit()

Operasi ini digunakan untuk update data penyakit yang diedit didatabase.

+insert_penyakit()

Operasi ini digunakan untuk menambahkan data penyakit didatabase

+gejala()

Operasi ini digunakan untuk menampilkan data gejala didatabase

+tambah_gejala()

Operasi ini digunakan untuk menampilkan tambah gejala

+edit_gejala()

Operasi ini digunakan untuk edit data gejala didatabase

+delete_gejala()

Operasi ini digunakan untuk delete data gejala didatabase

+update_gejala()

Operasi ini digunakan untuk update data gejala yang telah diedit

+insert_gejala()

Operasi ini digunakan untuk menambahkan data gejala didatabase

+aturan()

Operasi ini digunakan untuk menampilkan data aturan didatabase

+tambah_aturan()

Operasi ini digunakan untuk menampilkan tambah aturan

+delete_aturan()

Operasi ini digunakan untuk delete aturan didatabase

+insert_aturan()

Operasi ini digunakan untuk menambahkan data aturan didatabase

+admin()

Operasi ini digunakan untuk menampilkan data admin didatabase

+tambah_admin()

Operasi ini digunakan untuk menampilkan tambah admin

+delete_admin()

Operasi ini digunakan untuk delete data admin didatabase

+insert_admin()

Operasi ini digunakan untuk menambahkan data

2.2.3.16. Spesifikasi Design Kelas : PenyakitManager

PenyakitManager	<<control>>
+get_all_penyakit() Operasi ini digunakan untuk menampilkan semua data penyakit didatabase	
+init_info_penyakit() Operasi ini digunakan untuk menampilkan data penyakit berdasarkan id penyakit	
+view_detail_penyakit() Operasi ini digunakan untuk menampilkan data penyakit detail berdasarkan id	

2.2.3.17. Spesifikasi Design Kelas : UserManager

UserManager	<<control>>
+Signin() Operasi ini digunakan untuk memasukkan data username dan password	

untuk login

+invalidSigin()

Operasi ini digunakan untuk menampilkan data username dan password terjadi kesalahan dan meminta untuk diinputkan baru

+changepassword()

Operasi ini digunakan untuk menampilkan data untuk memasukkan password baru yang telah direset oleh admin

+update_password()

Operasi ini digunakan untuk update password yang telah diubah

+Logout()

Operasi ini digunakan untuk keluar dari menu untuk pasien

+Registrasi()

Operasi ini digunakan untuk memasukkan data registrasi oleh pasien

2.2.3.18. Spesifikasi Design Kelas : KonsultasiManager

KonsultasiManager	<<control>>
<p>+konsultasi_pasien_baru()</p> <p>Operasi ini digunakan untuk menampilkan konsultasi untuk pasien yang baru registrasi pertama kali.</p> <p>+Tanya()</p> <p>Operasi ini digunakan untuk menampilkan pertanyaan gejala yang dirasakan.</p> <p>+hasil()</p> <p>Operasi ini digunakan untuk menampilkan data hasil menu konsultasi</p> <p>+simpan()</p> <p>Operasi ini digunakan untuk menyimpan hasil diagnosa untuk di masukkan ke rekam medic</p> <p>+rekam()</p> <p>Operasi ini digunakan untuk menampilkan rekam medic hasil konsultasi pasien yang disimpan.</p>	

2.2.3.19. Spesifikasi Design Kelas : user

user	<<entty>>
-username= varchar(100) Atribut ini digunakan untuk menyimpan data username.	
-password=varchar(50) Atribut ini digunakan untuk menyimpan data password.	
-nama=varchar(100) Atribut ini digunakan untuk menyimpan data nama.	
-umur=varchar(5) Atribut ini digunakan untuk menyimpan data umur.	
-jenis_kelamin=varchar(100) Atribut ini digunakan untuk menyimpan data jenis kelamin	

2.2.3.20. Spesifikasi Design Kelas :gejala

gejala	<<entty>>
-id_gejala=int(11) Atribut ini digunakan untuk menyimpan id gejala.	
-nama_gejala=varchar(200) Atribut ini digunakan untuk menyimpan data nama gejala.	

2.2.3.21. Spesifikasi Design Kelas : penyakit

penyakit	<<entty>>
-Id_penyakit=int(11) Atribut ini digunakan untuk menyimpan data id penyakit	

-nama_penyakit=varchar(100)

Atribut ini digunakan untuk menyimpan data nama penyakit.

-informasi_penyakit=medium tex

Atribut ini digunakan untuk menyimpan data informasi penyakit

-pengobatan=medium tex

Atribut ini digunakan untuk menyimpan data pengobatan

2.2.3.22. Spesifikasi Design Kelas : aturan

aturan	<<entty>>
-id_aturan=int(11) Atribut ini digunakan untuk menyimpan data id aturan	
-id_gejala=int(11) Atribut ini digunakan untuk menyimpan data id gejala	
-id_penyakit=int(11) Atribut ini digunakan untuk menyimpan data id penyakit	

2.2.3.23. Spesifikasi Design Kelas : periksa

periksa	<<entty>>
-id_periksa=int(11) Atribut ini digunakan untuk menyimpan data id periksa	
-tggl_periksa=date Atribut ini digunakan untuk menyimpan data tanggal	
-persentase=double Atribut ini digunakan untuk menyimpan data persentase	

2.2.3.24. Spesifikasi Design Kelas : daftar gejala

Daftar gejala	<<entty>>
-id_daftar_gejala=int(11) Atribut ini digunakan untuk menyimpan data id daftar gejala	
-id_periksa=int(11) Atribut ini digunakan untuk menyimpan data id_periksa	

3. Perancangan Data

3.1 Dekomposisi Data

3.1.1. Deskripsi Entitas Data Role

Nama	Tipe	Panjang	Keterangan
Id_Role	int	5	Id_Role, primary key
Role	Varchar	5	Role

3.1.2. Deskripsi Entitas Data User

Nama	Tipe	Panjang	Keterangan
username	Varchar	20	Username , primary key
password	Varchar	50	Password
nama	Varchar	100	nama
Umur	Varchar	10	Umur
Jenis_kelamin	Varchar	20	Jenis kelamin

3.1.3. Deskripsi Entitas Data Gejala

Nama	Tipe	Panjang	Keterangan
id_gejala	int	5	Id gejala, primary key
nama_gejala	Varchar	200	Nama gejala

3.1.4. Deskripsi Entitas Data Penyakit

Nama	Tipe	Panjang	Keterangan
id_penyakit	int	5	Id penyakit, primary key
nama_penyakit	Varchar	100	Nama penyakit
Informasi_penyakit	Medium tex		Informasi penyakit
Pengobatan	Medium tex		Pengobatan

3.1.5. Deskripsi Entitas Data Aturan

Nama	Tipe	Panjang	Keterangan
id_aturan	int	5	Id aturan, primary key
id_penyakit	int	5	Id penyakit, foreign key
id_gejala	int	5	Id gejala, foreign key

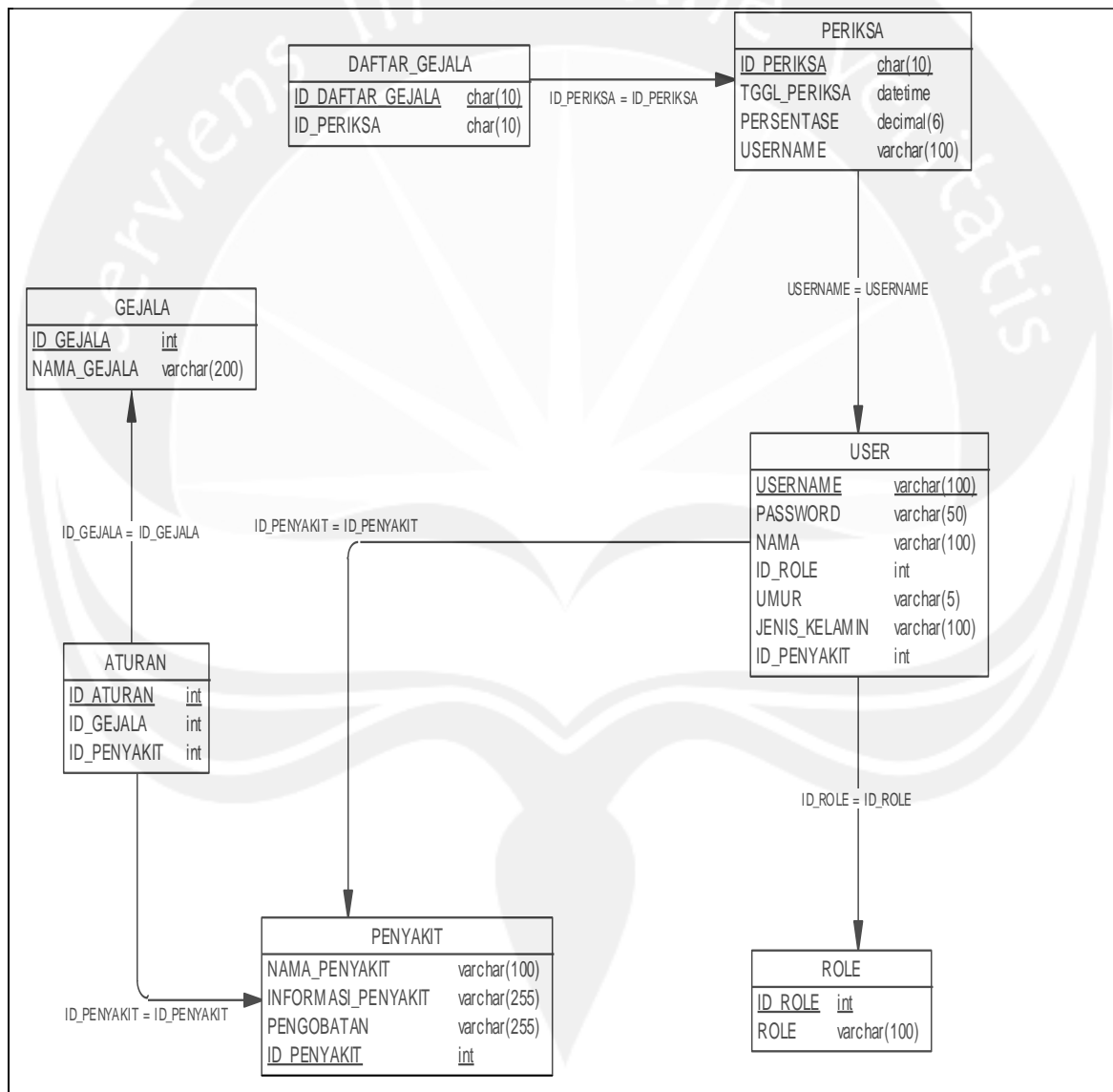
3.1.5. Deskripsi Entitas Data Periksa

Nama	Tipe	Panjang	Keterangan
id_periksa	int	5	Id periksa, primary key
Tggl_periksa	Date		Tggl_periksa
Persentase	Double		Persentase, foreign key

3.1.6. Deskripsi Entitas Data Daftar gejala

Nama	Tipe	Panjang	Keterangan
Id_daftar_gejala	int	5	Id daftar_gejala, primary key
id_periksa	int	5	Id periksa, foreign key

3.2 Physical Data Model



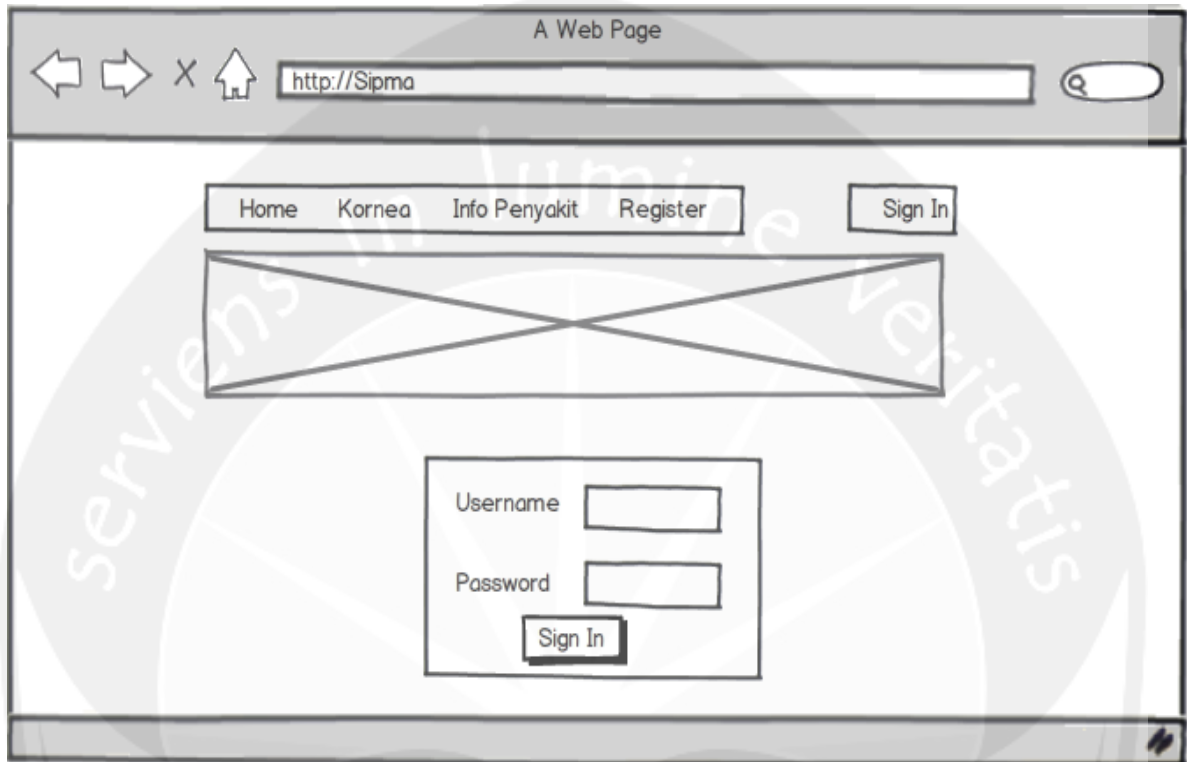
Gambar 3.1 Pysical Data Model

4. Perancangan Antarmuka

4.1. Sketsa UI dan Deskripsinya

4.1.1. Antarmuka Administrator

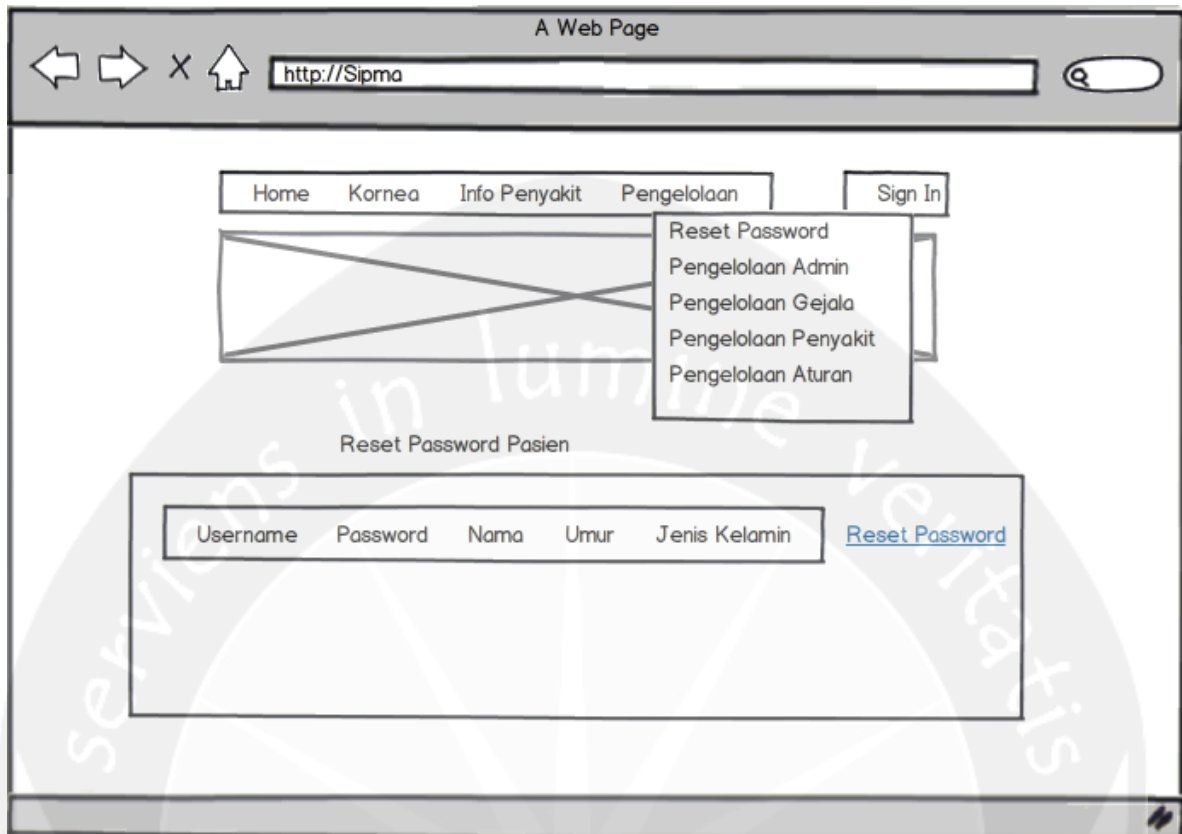
4.1.1.1. Antarmuka Login User



Gambar 4.1 Rancangan Antarmuka Login

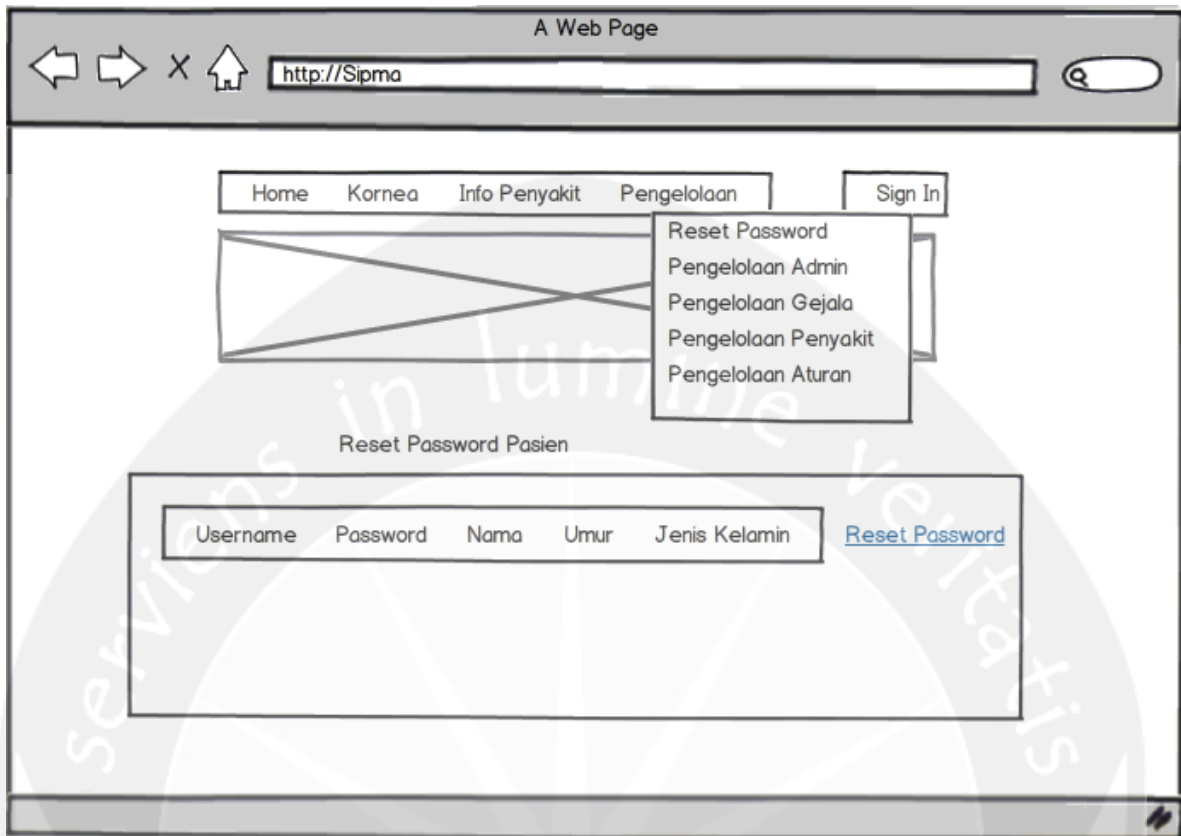
Antarmuka login (gambar 4.1) digunakan untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, *user* harus menginputkan *username* dan *password* dengan benar pada textbox yang telah disediakan. Pada saat tombol login ditekan, sistem akan mengecek *username* dan *password* yang diinputkan dengan data *username* dan *password* yang telah tersimpan di *database*. Jika data *username* dan *password* benar atau cocok maka *user* akan masuk ke dalam sistem, sebaliknya jika *username* dan *password* salah atau tidak cocok maka akan diberikan pesan peringatan.

4.1.1.2. Antarmuka Reset Password



Gambar 4.1 Rancangan Antarmuka Menu Reset Password
Antarmuka Menu Bar ini digunakan untuk proses admin
melakukan reset password.

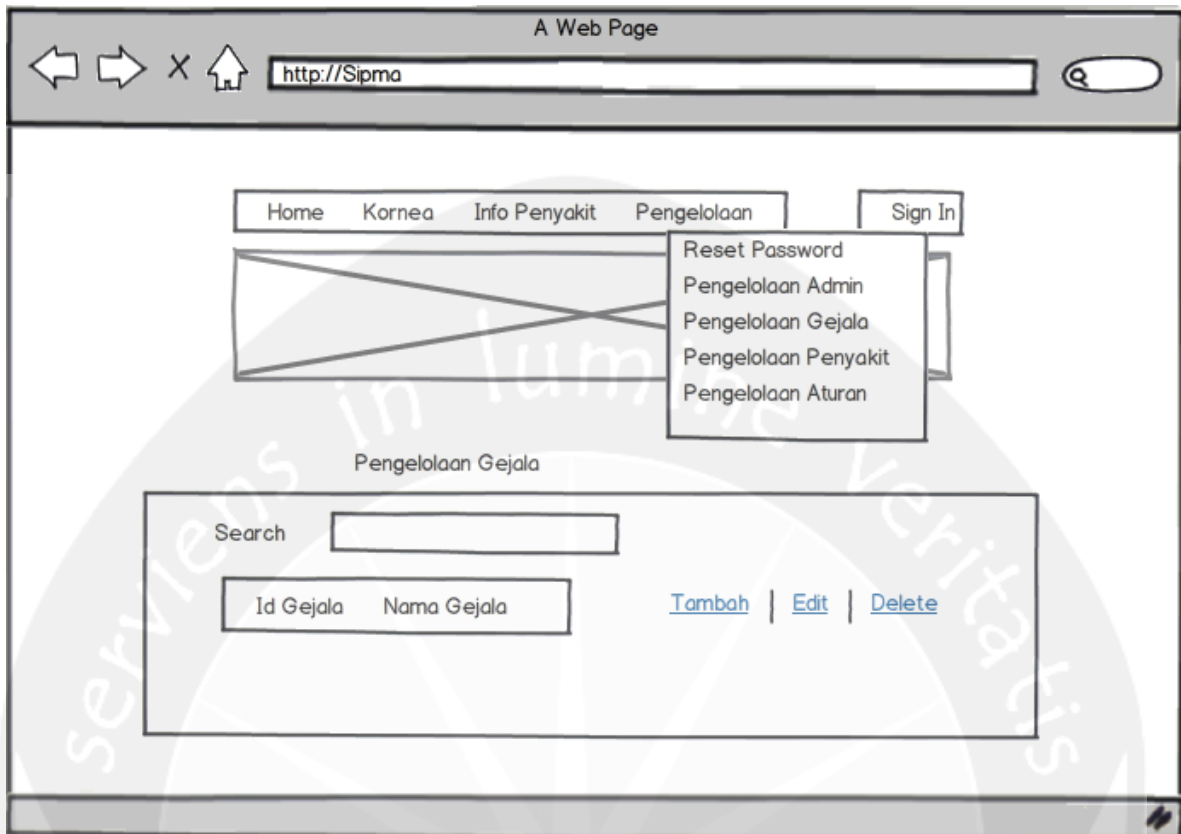
4.1.1.3. Antarmuka Pengelolaan Admin



Gambar 4.2 Rancangan Antarmuka Pengelolaan Admin

Antarmuka halaman admin (Gambar 4.2) digunakan oleh *user* untuk mengelola data admin. Sub menu tambah digunakan untuk menginputkan data admin ke *database*. Sub menu lihat untuk menampilkan data admin. *icon Delete* untuk menghapus data admin.

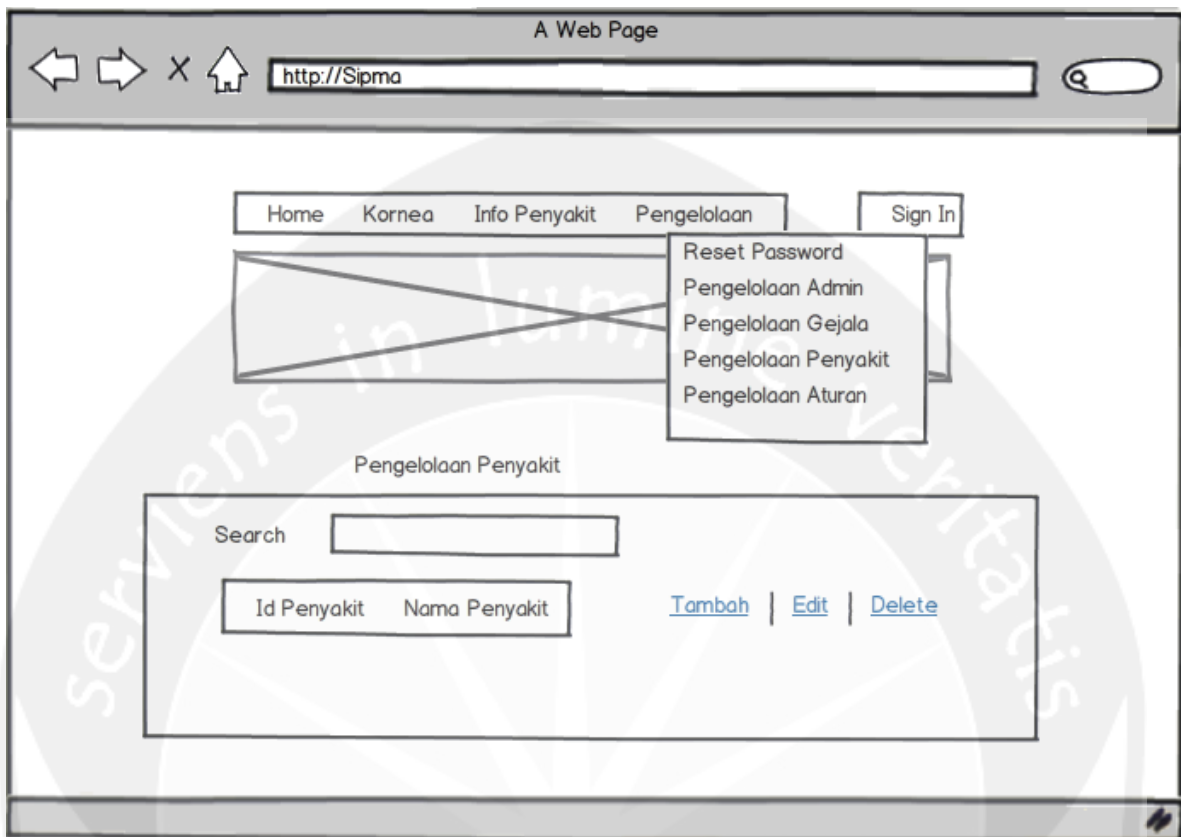
4.1.1.4. Antarmuka Pengelolaan Gejala



Gambar 4.3 Rancangan Antarmuka Pengelolaan Gejala

Antarmuka Halaman Gejala (Gambar 4.3) digunakan oleh *user* untuk mengelola data gejala. Sub menu tambah digunakan untuk menginputkan data gejala ke *database*. Sub menu lihat untuk menampilkan data gejala. *Icon Edit* untuk mengedit data gejala, *icon Delete* untuk menghapus data gejala. Sub menu search untuk mencari data gejala berdasarkan kata kunci yang diinputkan.

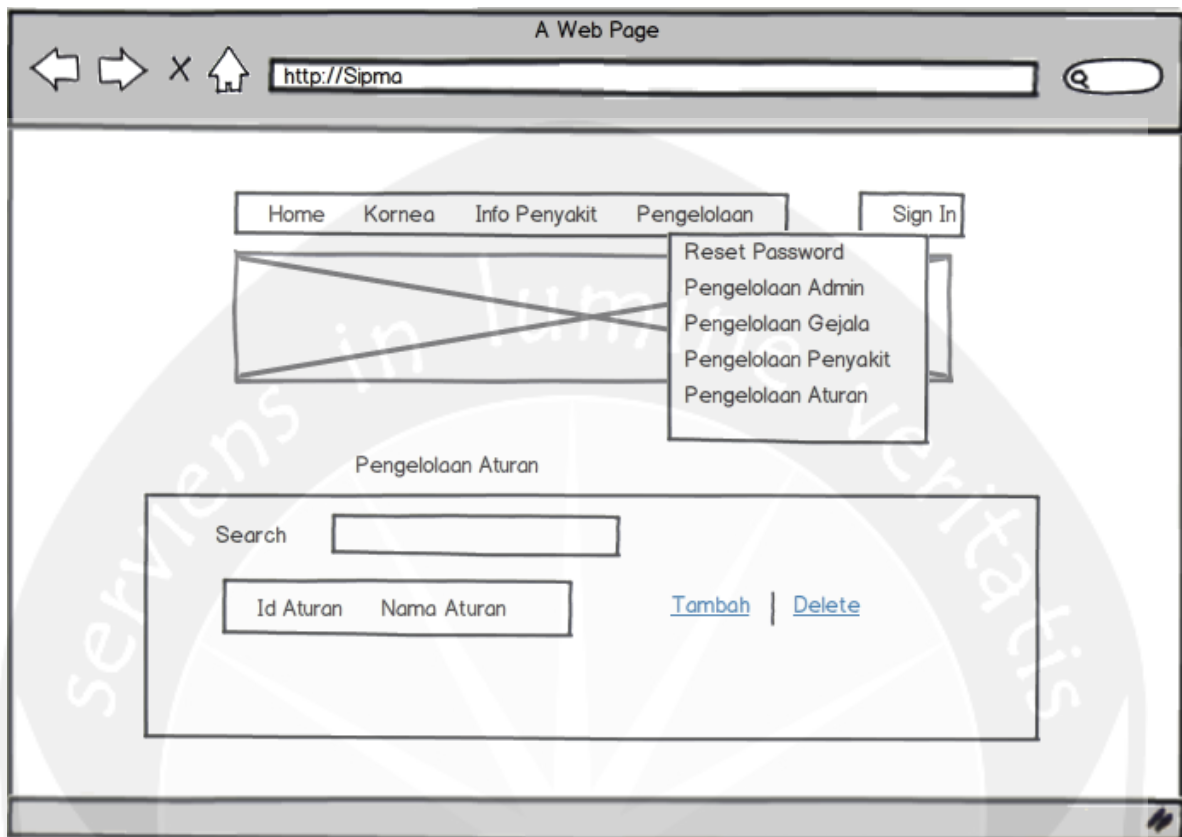
4.1.1.5. Antarmuka Pengelolaan Penyakit



Gambar 4.4 Rancangan Antarmuka Pengelolaan Data Penyakit

Antarmuka Halaman Penyakit (Gambar 4.4) digunakan oleh user untuk mengelola data penyakit. Sub menu tambah digunakan untuk menginputkan data penyakit ke *database*. Sub menu lihat untuk menampilkan data penyakit. *Icon Edit* untuk mengedit data penyakit, *icon Delete* untuk menghapus data penyakit. Sub menu cari untuk mencari data penyakit berdasarkan kata kunci yang diinputkan.

4.1.1.6. Antarmuka Pengelolaan Aturan

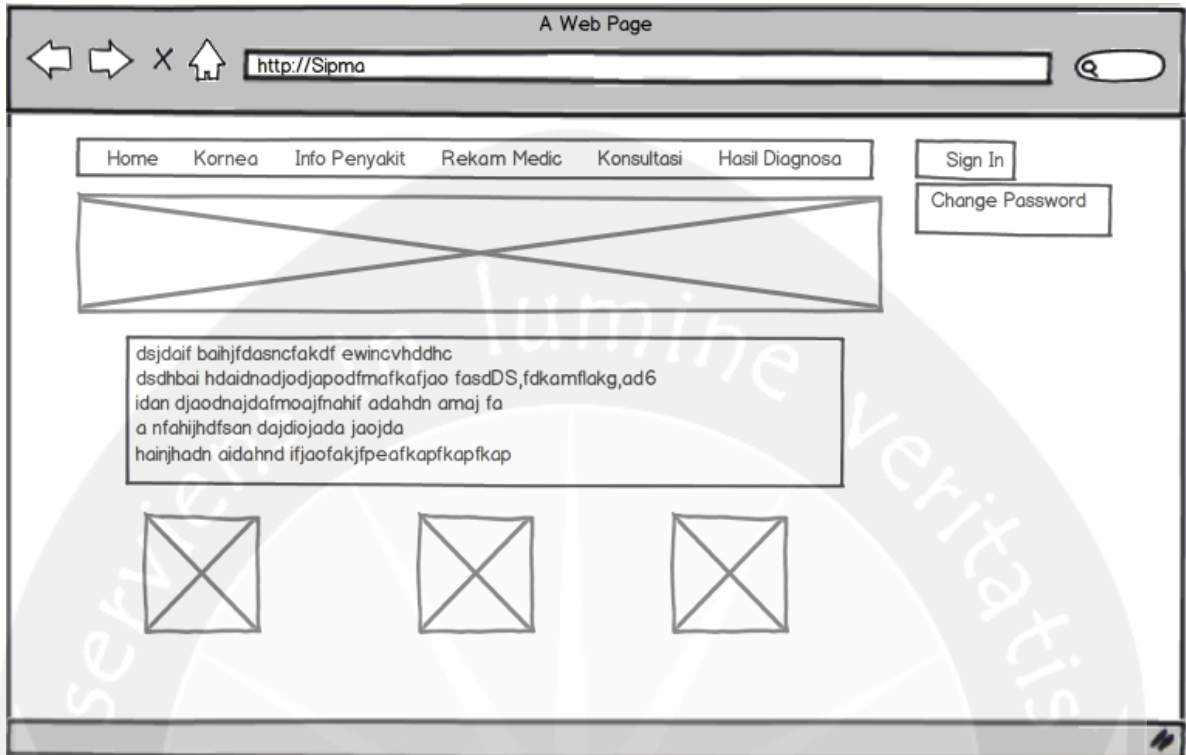


Gambar 4.5 Rancangan Antarmuka Pengelolaan Aturan

Antarmuka Halaman Aturan (Gambar 4.5) digunakan oleh user untuk mengelola data aturan. Sub menu tambah digunakan untuk menginputkan data aturan ke database. Sub menu lihat untuk menampilkan data aturan. Icon Delete untuk menghapus data aturan. Sub menu cari untuk mencari data aturan berdasarkan kata kunci yang diinputkan.

4.1.2. Antarmuka Pasien

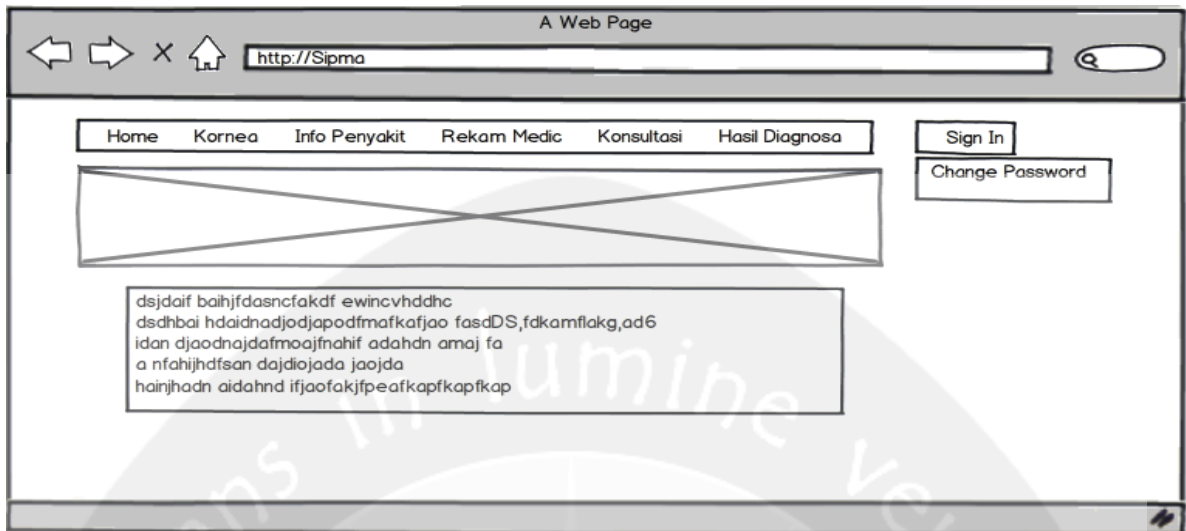
4.1.2.1. Antarmuka Halaman Home



Gambar 4.6 Rancangan Antarmuka Halaman Home

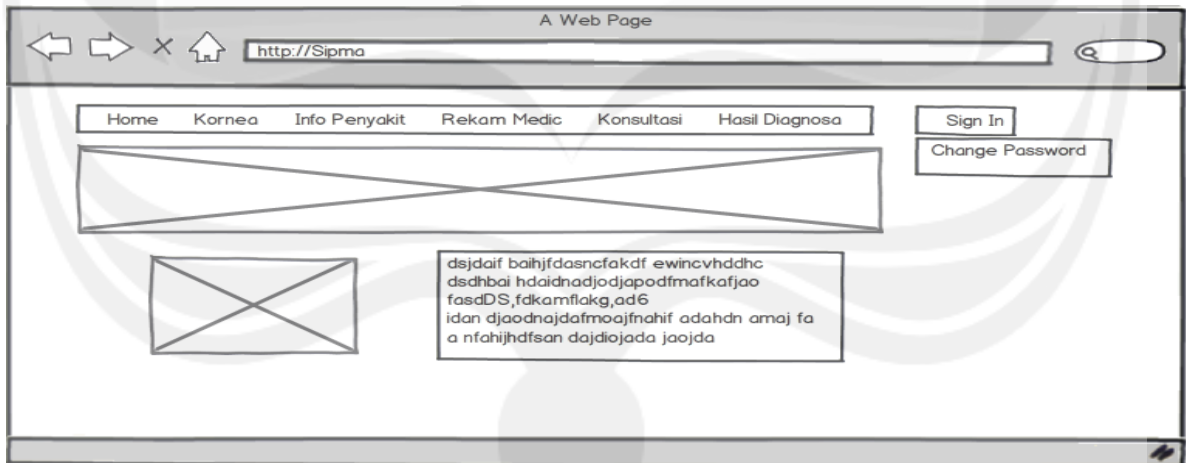
Halaman Home (Gambar 4.6) merupakan halaman awal website tentang mata, cara melindungi mata dan gejala umum penyakit mata.

4.1.2.2. Antarmuka Halaman Kornea



Gambar 4.7 Rancangan Antarmuka Halaman Kornea
Halaman cari data pasien (Gambar 4.7) merupakan halaman kornea untuk menampilkan informasi tentang kornea.

4.1.2.3. Antarmuka Halaman Informasi Penyakit Mata



Gambar 4.8 Rancangan Antarmuka Halaman Informasi Penyakit
Mata

Halaman informasi penyakit (Gambar 4.8) merupakan halaman menampilkan jenis-jenis penyakit mata dan beserta gambarnya kecuali gejala-gejalanya dan pengobatan.

4.1.2.4. Antarmuka Halaman Registrasi Pasien

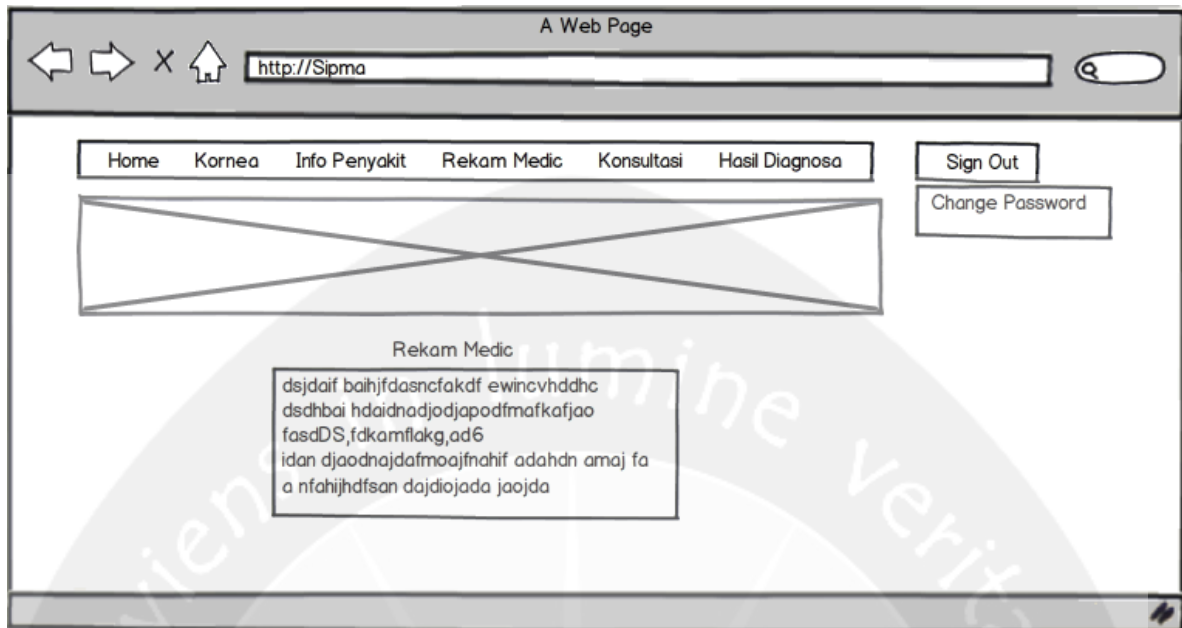
The image shows a web browser window titled "A Web Page" with the address bar containing "http://Sipma". The page features a navigation menu with buttons for "Home", "Kornea", "Info Penyakit", "Registrasi", and "Sign In". Below the menu is a large rectangular area with a diagonal cross, likely a placeholder for a logo or image. The main content area is titled "Registrasi" and contains the following form fields and elements:

- Username:
- Password:
- Re-Password:
- Nama:
- Umur:
- Jenis Kelamin: (dropdown menu)
- Simpan:

Gambar 4.9 Rancangan Antarmuka Halaman Registrasi Pasien

Halaman Registrasi Pasien (Gambar 4.9) merupakan halaman untuk melakukan registrasi atau pendaftaran pasien.

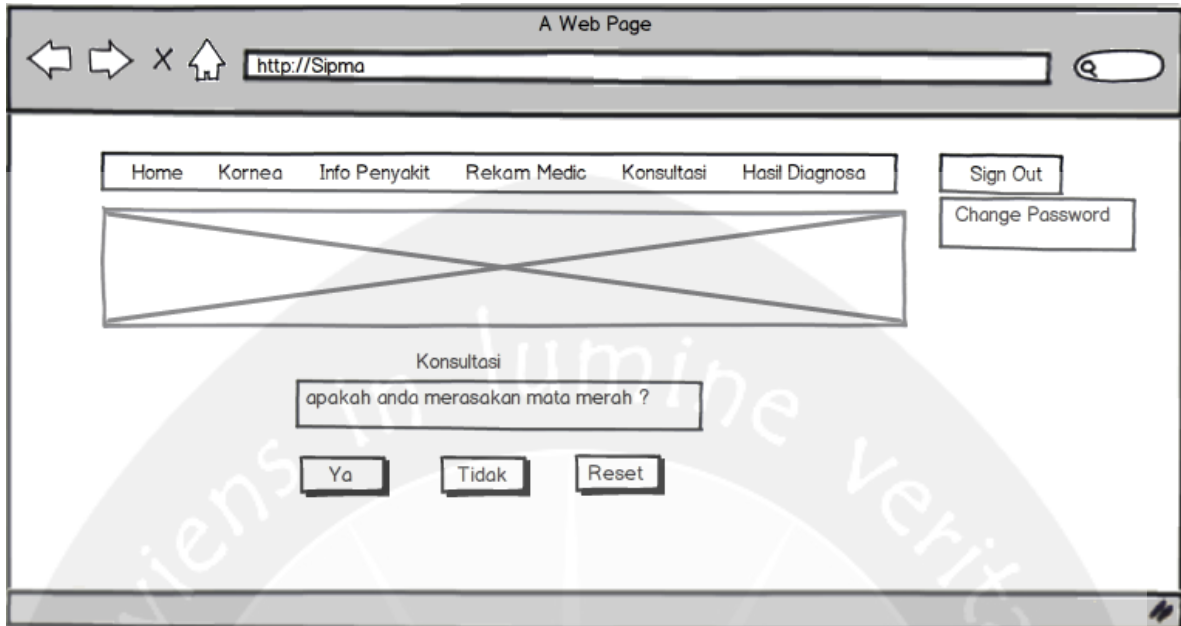
4.1.2.5. Antarmuka Halaman Rekam Medic



Gambar 4.10 Rancangan Antarmuka Halaman Rekam Medic

Halaman Registrasi Pasien (Gambar 4.9) merupakan halaman untuk melakukan Rekam medic.

4.1.2.6. Antarmuka Halaman Konsultasi



Gambar 4.11 Rancangan Antarmuka Halaman Konsultasi

Halaman konsultasi (Gambar 4.11) merupakan halaman pertanyaan yang diajukan kepada user berdasarkan gejala yang dirasakan. Pilihan jawaban yaitu "Ya" atau "Tidak". Jika pertanyaan selesai maka user menekan tombol simpan.

4.1.2.8. Antarmuka Change Password

The image shows a web browser window titled "A Web Page" with the address bar containing "http://Sipma". The page layout includes a navigation menu with links: Home, Kornea, Info Penyakit, Rekam Medic, Konsultasi, and Hasil Diagnosa. On the right side, there are buttons for "Sign Out" and "Change Password". Below the navigation menu is a large rectangular area with a diagonal cross, likely a placeholder for a logo or image. The main content area is titled "Change Password" and contains four input fields labeled "Username", "Password", "New Password", and "Re-Password". A "Simpan" button is located below the input fields. A large, faint watermark of a university logo is visible in the background of the page.

Gambar 4.13 Rancangan Antarmuka Halaman Change Password

Halaman change password (Gambar 4.13) berfungsi untuk menampilkan hasil change password untuk mengubah password lama menjadi password baru