

**SISTEM PAKAR DIAGNOSA PENYAKIT MATA PADA  
MANUSIA MENGGUNAKAN METODE  
FORWARD CHAINING**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan**

**Jenjang Strata Satu (S1)**

**Pada program Studi Teknik Informatika**

**Oleh**

**Irsan Hamdani Prasetya**

**361601011**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI**

**BANDUNG**

**2021**

**LEMBAR PENGESAHAN**

**SISTEM PAKAR DIAGNOSA PENYAKIT MATA PADA  
MANUSIA MENGGUNAKAN METODE FORWARD  
CHAINING**

**Oleh:**

**Irsan Hamdani Prasetya  
361601011**

Tugas Akhir Ini Telah Diterima Dan Disahkan Untuk  
Memenuhi Persyaratan Mencapai Gelar  
Sarjana Teknik Informatika

Pada

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI

Bandung, 14 Februari 2021

Disetujui Oleh

Ketua Program Studi

Dosen Pembimbing

**Chalifa Chazar, S.T., M.T**  
NIDN: 0421098704

**Chairuddin, Ir., M.T., M.M., Dr.**  
NIDN: 437300016

**LEMBAR PERSETUJUAN REVISI**

**SISTEM PAKAR DIAGNOSA PENYAKIT KULIT**

**PADA MANUSIA MENGGUNAKAN**

**FORWARD CHAINING**

Oleh

Irsan Hamdani Prasetya  
361601011

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, 14 Februari 2021

Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1	<b>Chairuddin, Ir., M.T., M.M., Dr.</b>	Pembimbing	
2	<b>Pahlawan Sagala, Dr.</b>	Penguji 1	
3	<b>Moch Ali Ramdani., S.T., M.Kom.</b>	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika

**Chalifa Chazar, S.T., M.T.**  
**NIDN. 0421098704**

## **SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa:

1. Tugas akhir ini asli dan belum pernah diajukan untuk mendapat gelar akademik, baik di sekolah tinggi manajemen informatika dan komputer Indonesia Mandiri maupun perguruan tinggi lainnya
2. Tugas akhir ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
3. Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri

Bandung, 14 Februari 2021

Yang Membuat Pernyataan

**Irsan Hamdani Prasetya**  
**361601011**

## **ABSTRAK**

# **SISTEM PAKAR DIAGNOSA PENYAKIT MATA PADA MANUSIA MENGGUNAKAN METODE FORWARD CHAINING**

**Oleh:**

**IRSAN HAMDANI PRASETYA**  
**361601011**

Kesadaran masyarakat yang masih minim akan kesehatan mata, kebiasaan hidup dari masyarakat yang selalu ingin praktis, pengetahuan masyarakat yang sedikit akan gejala-gejala awal dari penyakit mata merupakan salah satu faktor penyebab penyakit mata menjadi parah ketika pasien ditangani oleh paramedis. Untuk mengantisipasi hal tersebut maka peneliti membangun sebuah sistem pakar untuk mendiagnosa penyakit mata berdasarkan dari gejala-gejala yang diderita oleh pasien. Sistem pakar yang akan dibangun peneliti menggunakan metode *Forward Chaining* sebagai metode pelacakannya. Dibangunnya sistem pakar diagnosa penyakit mata ini diharapkan dapat memberikan informasi kepada masyarakat dengan cara yang amat mudah.

**Kata Kunci: Penyakit, Mata, Diagnosa**

## **ABSTRACT**

### ***EXPERT SYSTEMS ON HUMAN EYE DISEASE DIAGNOSIS USING FORWARD CHAINING***

**Oleh:**

**IRSAN HAMDANI PRASETYA**

**361601011**

*Public awareness of eye health, the life habits of the people who always want to be practical, the public's little knowledge of the early symptoms of eye disease is one of the factors that cause eye disease to become severe when the patient is treated by paramedics. To anticipate this, the researchers built an expert system to diagnose eye diseases based on the symptoms suffered by the patient. The expert system that will be built by researches uses the Forward Chaining method as a tracking method. The construction of an expert system for diagnosing eye diseases is expected to provide information to the public in a very easy way*

*Keywords: Disease, Eye, Diagnose*

## KATA PENGANTAR

Puji syukur kepada Allah SWT berkat Rahmat, Hidayah, dan karunia – Nya sehingga penulis dapat menyelesaikan skripsi ini yang mengambil judul ***“Sistem Pakar Diagnosa Penyakit Mata Pada Manusia Menggunakan Metode Forward Chaining”***.

Tujuan penulis skripsi ini untuk memenuhi sebagian syarat memperoleh gelar Sarjana Teknik (S.T) bagi mahasiswa program Strata-1 di program studi Sistem Pakar Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri Bandung.

Akhir kata, Penulis mengucapkan rasa terima kasih kepada semua pihak dan apabila ada yang tidak disebutkan oleh Penulis mohon maaf, dengan besar harapan semoga skripsi yang ditulis oleh penulis ini dapat bermanfaat khususnya bagi Penulis sendiri dan umumnya bagi pembaca. Bagi para pihak yang telah membantu dalam penulisan skripsi ini semoga segala amal dan kebaikannya mendapat balasan yang berlimpah dari Allah SWT, Amin Ya Robal Alamin.

Bandung, 14 Februari 2021

Irsan Hamdani Prasetya

## UCAPAN TERIMAKASIH

Dengan mengucapkan syukur Allhamdulillah, penelitian ini dapat diselesaikan untuk memenuhi syarat tugas akhir. Laporan penelitian dalam tugas akhir ini diajukan untuk memenuhi dan melengkapi salah satu syarat akademik dalam kelulusan jenjang strata satu (S1) jurusan teknik informatika pada Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri Bandung.

Penyusunan tugas akhir ini tidak lepas dari dukungan dan bimbingan dari berbagai pihak, maka pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Chairuddin, M.T., M.M selaku Dosen Pembimbing dan Rektor Sekolah Tinggi Manajemen Informatika dan Komputer & Sekolah Tinggi Ilmu Ekonomi STAN Indonesia Mandiri Bandung.
2. Ibu Chalifa Chazar, S.T., M.T selaku Ketua Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri Bandung.
3. Bapak/Ibu dosen dan staff di lingkungan STMIK-IM Bandung, khususnya program studi teknik informatika yang telah banyak membantu dalam penulisan ini.
4. Teristimewa untuk kedua orang tua tercinta bapak Syamsu Rachman dan ibu Iis Sri Iriani yang sangat penulis sayangi. Terimakasih selalu memberikan nasihat, dukungan, bantuan moril maupun materi dalam menyelesaikan skripsi ini.

5. Teman-teman dari “Geng Gunting” Ananda Giri Sukmawijaya, Alvin Surya Ramadhan, Rezki Dwi, Muhammad Ramdha Desapta, Adnan Hafizh, Aswin Winangga, dan Bobby Rahman yang selalu memberikan dukungan, semangat dalam menyelesaikan skripsi ini.
6. Keluarga besar “FOLKS” Abrian Saputra, Reza Lavira, Farhan Taufiq, Oki Iqbal, Fahri Fatur, Fachrur Dwi, Rofil, Hildan Juniantara, Om Jason, Om Kusuma, Rian Fauzi, Pringgo Wicaksono, Ari Ramadhan, Rofil Rusmeina dan seluruh teman yang berkaitan dalam keluarga ini untuk masukan ide, tenaga, dan waktunya dalam penyelesaian skripsi ini.
7. Teman-teman mahasiswa teknik informatika angkatan 2016 yaitu Ananda Giri, Agriani Mega Astari, Danu Syafrizal, Rezki Dwi, Alvin Surya Ramadhan, Adila Rizky, Arief Febrianto, Nanda Mochammad Rizal, Bobby Rahman, Muhammad Ramdha, Adnan Hafizh, Agung Nur Haliim, Andrian, Aswin Winangga, Firman Maulana yang selalu memberikan motivasi dan semangat dalam menyelesaikan skripsi ini.
8. Teruntuk Tri Puji Wulandari saya berterimakasih untuk semua wejangan, motivasi, tenaga, dan fikirannya yang telah banyak membantu dalam penyelesaian skripsi ini.
9. Terimakasih kepada kakak sepupu saya selaku perawat dari rumah sakit yang telah membantu dalam penelitian ini.
10. Terima kasih juga kepada semua pihak yang telah membantu dalam penyelesaian skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih banyak kekurangan yang mendasar pada laporan penelitian skripsi ini. Oleh karena itu penulis memohon kritik dan saran nya dari pembaca untuk menjadikan skripsi ini lebih baik lagi.

Akhir kata saya, berharap semoga dengan selesainya laporan skripsi ini dapat memberikan manfaat bagi semua pihak serta menambah wawasan bagi pemikiran kita semua. Terimakasih.

## DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERSETUJUAN REVISI.....	ii
SURAT PERNYATAAN.....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR.....	vi
UCAPAN TERIMAKASIH.....	vii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR .....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah .....	4
1.3 Tujuan Penulisan .....	4
1.4 Batasan Masalah.....	5
1.5 Metode Penelitian.....	6
1.5.1 Teknik Pengumpulan Data .....	6
1.5.2 Metode Perancangan .....	7

1.6	Sistematika Penulisan.....	9
BAB II LANDASAN TEORI .....		12
2.1	Pengertian Aplikasi .....	12
2.1.1	Aplikasi Web .....	14
2.2	Pengertian Sistem .....	15
2.3	Konsep Dasar MySQL .....	15
2.4	Pengertian PHP .....	16
2.5	Pengertian HTML.....	18
2.6	Pengertian phpMyAdmin .....	19
2.7	Pengertian <i>Xampp</i> .....	19
2.8	Metode Pengembangan .....	21
2.9	Flowchart.....	23
2.10	Pengertian <i>Database</i> .....	26
2.11	Pengertian UML (Unified Modeling Language) .....	26
2.11.1	<i>Use Case</i> Diagram.....	27
2.11.2	<i>Activity</i> Diagram.....	30
2.11.3	<i>Class</i> Diagram.....	32
2.11.4	<i>Sequence</i> Diagram.....	33
2.12	Pengertian Penyakit.....	35

2.13	Pengertian Diagnosa .....	36
2.14	Pengertian Mata .....	36
2.15	Forward Chaining .....	37
2.16	Sistem Pakar.....	41
BAB III ANALISIS MASALAH DAN PERANCANGAN PROGRAM.....		43
3.1	Communication .....	43
3.1.1	Pengumpulan Data .....	43
3.1.2	Analisis Sistem Pakar.....	45
3.1.3	Analisi Identifikasi Masalah.....	46
3.1.4	Analisis Sistem Yang Sedang Berjalan.....	49
3.1.5	Analisis Sistem Diusulkan .....	50
3.1.6	Analisis Pengguna.....	52
3.2	Planning .....	54
3.3	Modeling.....	57
3.3.1	Arsitektur <i>Forward Chaining</i> Dengan <i>Tree</i> .....	62
3.3.2	Definisi Aktor.....	62
3.3.3	Perancangan Sistem.....	63
3.3.4	<i>Use Case</i> Diagram.....	63
3.3.5	<i>Activity</i> Diagram.....	74

3.3.6	<i>Class Diagram</i> .....	91
3.3.7	<i>Sequence Diagram</i> .....	92
3.3.8	ERD ( <i>Entity Relationship Diagram</i> ).....	102
3.4	Design Interface.....	104
BAB IV IMPLEMENTASI DAN UJI COBA .....		127
4.1	Construction (Code & Test) .....	127
4.1.1	Implementasi <i>Hardware &amp; Software</i> .....	127
4.1.2	Implementasi Antarmuka .....	128
4.1.3	Testing.....	133
4.1.4	Kesimpulan <i>Testing</i> .....	153
BAB V PENUTUP .....		154
5.1	Kesimpulan.....	154
5.2	Saran .....	154
DAFTAR PUSTAKA .....		155
LAMPIRAN .....		158

## DAFTAR TABEL

TABEL: 2.1 Simbol <i>Flowchart</i> .....	24
TABEL: 2.2 Simbol <i>Use Case</i> Diagram (Sukamto dan Shalahuddin, 2015) .....	28
TABEL: 2.3 Simbol <i>Activity</i> Diagram (Sukamto & Shalahuddin, 2015) .....	31
TABEL: 2.4 <i>Class</i> Diagram (Sukamto & Shalahuddin, 2015) .....	32
TABEL: 2.5 Notasi <i>Sequence</i> Diagram .....	33
TABEL: 3.1 Referensi Penelitian.....	44
TABEL: 3.2 Tabel Penjadwalan Penelitian .....	56
TABEL: 3.3 Tabel Daftar Penyakit Mata .....	57
TABEL: 3.5 Tabel Basis Pengetahuan.....	60
TABEL: 3.6 Skenario <i>Use Case</i> Mengakses Diagnosa.....	65
TABEL: 3.7 Skenario <i>Use Case</i> Mengakses Tentang Mata .....	66
TABEL: 3.8 Skenario <i>Use Case</i> Mengakses Daftar Penyakit .....	67
TABEL: 3.9 Skenario <i>Use Case Login</i> .....	68
TABEL: 3.10 Skenario <i>Use Case</i> Mengelola Gejala.....	69
TABEL: 3.11 Skenario <i>Use Case</i> Mengelola Penyakit .....	70
TABEL: 3.12 Skenario <i>Use Case</i> Mengelola Pencegahan .....	71
TABEL: 3.13 Skenario <i>Use Case</i> Mengelola Rule.....	72
TABEL: 3.14 Skenario <i>Use Case</i> Mengelola Solusi .....	73

TABEL: 3.15 Tabel Gejala .....	103
TABEL: 3.16 Tabel Penyakit .....	103
TABEL: 3.17 Tabel Pencegahan.....	103
TABEL: 3.18 Tabel Solusi.....	104
TABEL: 3.19 Tabel Rule .....	104
TABEL: 4.1 Tabel Rencana Pengujian .....	134
TABEL: 4.2 Tabel Hasil Pengujian Login Admin.....	135
TABEL: 4.3 Tabel Hasil Pengujian Tambah Data Gejala .....	136
TABEL: 4.4 Tabel Hasil Pengujian Ubah Data Gejala.....	137
TABEL: 4.5 Tabel Hasil Pengujian Hapus Data Gejala .....	138
TABEL: 4.6 Tabel Hasil Pengujian Tambah Data Penyakit.....	139
TABEL: 4.7 Tabel Hasil Pengujian Ubah Data Penyakit .....	140
TABEL: 4.8 Tabel Hasil Pengujian Hapus Data Penyakit.....	141
TABEL: 4.9 Tabel Hasil Pengujian Tambah Data Pencegahan.....	142
TABEL: 4.10 Tabel Hasil Pengujian Ubah Data Pencegahan .....	143
TABEL: 4.11 Tabel Hasil Pengujian Hapus Data Pencegahan .....	144
TABEL: 4.12 Tabel Hasil Pengujian Tambah Data Solusi.....	145
TABEL: 4.13 Tabel Hasil Pengujian Ubah Data Solusi .....	146
TABEL: 4.14 Tabel Hasil Pengujian Hapus Data Solusi.....	147

TABEL: 4.15 Tabel Hasil Pengujian Tambah Data Rule .....	148
TABEL: 4.16 Tabel Hasil Pengujian Ubah Data Rule .....	149
TABEL: 4.17 Tabel Hasil Pengujian Hapus Data Rule .....	150
TABEL: 4.18 Tabel Hasil Pengujian Diagnosa Gejala.....	151
TABEL: 4.19 Tabel Kelas Uji.....	152

## DAFTAR GAMBAR

GAMBAR: 1.1 <i>Waterfall</i> Pressman (Pressman, 2015).....	7
GAMBAR: 2.1 <i>Waterfall</i> Pressman (Pressman, 2015).....	21
GAMBAR: 3.1 <i>Flowmap</i> Sistem Yang Sedang Berjalan .....	50
GAMBAR: 3.2 <i>Flowmap</i> Sistem Yang Diusulkan .....	51
GAMBAR: 3.3 Arsitektur <i>Forward Chaining</i> Dengan <i>Tree</i> .....	62
GAMBAR: 3.4 <i>Use Case</i> Diagram Sistem Aplikasi.....	64
GAMBAR 3.5 <i>Activity</i> Diagram Mengakses Diagnosa .....	74
GAMBAR: 3.6 <i>Activity</i> Diagram Mengakses Tentang Mata.....	75
GAMBAR: 3.7 <i>Activity</i> Diagram Mengakses Daftar Penyakit.....	75
GAMBAR: 3.8 <i>Activity</i> Diagram <i>Login</i> .....	76
GAMBAR: 3.9 <i>Activity</i> Diagram Menambahkan Gejala.....	77
GAMBAR: 3.10 <i>Activity</i> Diagram Mengubah Gejala .....	78
GAMBAR: 3.11 <i>Activity</i> Diagram Menghapus Gejala .....	79
GAMBAR: 3.12 <i>Activity</i> Diagram Menambahkan Penyakit .....	80
GAMBAR: 3.15 <i>Activity</i> Diagram Menambahkan Pencegahan .....	82
GAMBAR: 3.16 <i>Activity</i> Diagram Mengubah Pencegahan.....	83
GAMBAR: 3.17 <i>Activity</i> Diagram Menghapus Pencegahan .....	84
GAMBAR: 3.18 <i>Activity</i> Diagram Menambahkan Solusi .....	85

GAMBAR: 3.19 <i>Activity</i> Diagram Mengubah Solusi.....	86
GAMBAR: 3.20 <i>Activity</i> Diagram Menghapus Solusi .....	87
GAMBAR: 3.21 <i>Activity</i> Diagram Menambahkan Rule.....	88
GAMBAR: 3.22 <i>Activity</i> Diagram Mengubah Rule .....	89
GAMBAR: 3.23 <i>Activity</i> Diagram Menghapus Rule.....	90
GAMBAR: 3.24 <i>Class</i> Diagram .....	91
GAMBAR: 3.25 <i>Sequence</i> Diagram <i>Login</i> .....	92
GAMBAR: 3.26 <i>Sequence</i> Diagram Menambahkan Gejala .....	93
GAMBAR: 3.27 <i>Sequence</i> Diagram Mengubah Gejala.....	93
GAMBAR: 3.28 <i>Sequence</i> Diagram Menghapus Gejala .....	94
GAMBAR: 3.29 <i>Sequence</i> Diagram Menambahkan Penyakit .....	94
GAMBAR: 3.30 <i>Sequence</i> Diagram Mengubah Penyakit.....	95
GAMBAR: 3.31 <i>Sequence</i> Diagram Menghapus Penyakit .....	95
GAMBAR: 3.32 <i>Sequence</i> Diagram Menambahkan Pencegahan .....	96
GAMBAR: 3.33 <i>Sequence</i> Diagram Mengubah Pencegahan.....	96
GAMBAR: 3.34 <i>Sequence</i> Diagram Menghapus Pencegahan .....	97
GAMBAR: 3.35 <i>Sequence</i> Diagram Menambahkan Solusi .....	97
GAMBAR: 3.36 <i>Sequence</i> Diagram Mengubah Solusi.....	98
GAMBAR: 3.37 <i>Sequence</i> Diagram Menghapus Solusi .....	98

GAMBAR: 3.38 <i>Sequence</i> Diagram Menambahkan Rule.....	99
GAMBAR: 3.39 <i>Sequence</i> Diagram Mengubah Rule .....	99
GAMBAR: 3.40 <i>Sequence</i> Diagram Menghapus Rule .....	100
GAMBAR: 3.41 <i>Sequence</i> Diagram Mengakses Diagnosa.....	100
GAMBAR: 3.42 <i>Sequence</i> Diagram Mengakses Daftar Penyakit.....	101
GAMBAR 3.43 <i>Sequence</i> Diagram Mengakses Tentang Mata.....	101
GAMBAR: 3.44 ERD Sistem Pakar Diagnosa Penyakit Mata .....	102
GAMBAR: 3.45 Rancangan <i>Interface</i> Menu Awal.....	105
GAMBAR: 3.46 Rancangan <i>Interface</i> Menu Diagnosa.....	106
GAMBAR: 3.47 Rancangan <i>Interface</i> Hasil Diagnosa Penyakit .....	107
GAMBAR: 3.48 Rancangan <i>Interface</i> Daftar Penyakit.....	108
GAMBAR: 3.49 Rancangan <i>Interface</i> Tentang Mata.....	109
GAMBAR: 3.50 Rancangan <i>Interface</i> Halaman <i>Login</i> .....	110
GAMBAR: 3.51 Rancangan <i>Interface</i> Halaman Awal Admin.....	111
GAMBAR: 3.52 Rancangan <i>Interface</i> Halaman Gejala Admin .....	112
GAMBAR: 3.53 Rancangan <i>Interface</i> Halaman Penyakit Admin .....	113
GAMBAR: 3.54 Rancangan <i>Interface</i> Halaman Pencegahan Admin .....	114
GAMBAR: 3.55 Rancangan <i>Interface</i> Halaman Solusi Admin .....	115
GAMBAR: 3.56 Rancangan <i>Interface</i> Halaman Rule Admin.....	116

GAMBAR: 3.57 Rancangan <i>Interface</i> Tambah Gejala Admin .....	117
GAMBAR: 3.58 Rancangan <i>Interface</i> Tambah Penyakit Admin.....	118
GAMBAR: 3.59 Rancangan <i>Interface</i> Tambah Pencegahan Admin.....	119
GAMBAR: 3.60 Rancangan <i>Interface</i> Tambah Solusi Admin.....	120
GAMBAR: 3.61 Rancangan <i>Interface</i> Tambah Rule Admin .....	121
GAMBAR: 3.62 Rancangan <i>Interface</i> Ubah Gejala Admin.....	122
GAMBAR: 3.63 Rancangan <i>Interface</i> Ubah Penyakit Admin .....	123
GAMBAR: 3.64 Rancangan <i>Interface</i> Ubah Penyakit Admin .....	124
GAMBAR: 3.65 Rancangan <i>Interface</i> Ubah Solusi Admin .....	125
GAMBAR: 3.66 Rancangan <i>Interface</i> Ubah Rule Admin.....	126
GAMBAR: 4.1 Tampilan Memilih Gejala Penyakit Mata .....	128
GAMBAR: 4.2 Tampilan Hasil Diagnosa Penyakit Mata .....	129
GAMBAR: 4.3 Tampilan Menu Dari Daftar Penyakit Mata .....	129
GAMBAR: 4.4 Tampilan Menu Tentang Mata Dari Aplikasi.....	130
GAMBAR: 4.5 Tampilan Menu Login Admin .....	130
GAMBAR: 4.6 Tampilan Menu Dari Data Gejala Penyakit Mata .....	131
GAMBAR: 4.7 Tampilan Menu Dari Data Penyakit Mata .....	131
GAMBAR: 4.8 Tampilan Menu Dari Data Pencegahan Penyakit Mata .....	132
GAMBAR: 4.9 Tampilan Menu Dari Data Solusi Penyakit Mata.....	132

GAMBAR: 4.10 Tampilan Menu Dari Data Rule Penyakit Mata ..... 133

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Mata adalah bagian penting yang merupakan salah satu dari panca indera manusia. Mata penting karena berguna untuk mengetahui dan mendeteksi apa saja yang dapat dilihat oleh manusia. Mata manusia memiliki beberapa struktur yang berfungsi untuk memfokuskan cahaya retina. Komponen yang masuk lewat cahaya ke retina hampir berwarna hitam atau gelap karena itu untuk mengurangi bayangan gelap dari cahaya, kornea dan lensa sebagai pengumpul cahaya yang akan difokuskan ke retina mata dan menyebabkan perubahan kimiawi pada sel fotosintesis sebelum melanjutkannya ke otak. Mata merupakan salah satu organ tubuh vital manusia. Oleh karena itu, kita harus selalu menjaga dan mencegah hal-hal yang dapat merusak mata (Murtopo & Sarimurni, 2005). Mata merupakan indera penglihatan pada manusia. Mata dibentuk untuk menerima rangsangan berkas-berkas cahaya pada retina selanjutnya dengan perantara serabut-serabut nervus optikus, mengalihkan rangsangan ini ke pusat penglihatan pada otak untuk ditafsirkan (Evelin, 1999). Mata terdiri dari dua bagian organ, yaitu organ dalam dan luar. Bagian luar mata diantaranya, alis mata, kelopak mata, kelenjar air mata, bulu mata, sedangkan bagian dalam mata diantaranya, kornea, sklera, konjungtiva, pupil dan iris, badan siliar, koroid, lensa mata, retina, saraf optik. Dengan perkembangan sistem pakar dapat dibuat aplikasi sistem pakar yang dapat

mendiagnosa penyakit mata manusia berdasarkan gejala-gejala yang dialami, jadi dengan menggunakan sistem ini kita bisa menemukan fakta dalam memecahkan masalah.

Sistem ini mendiagnosa penyakit mata berdasarkan gejala-gejala yang dialami. Gejala penyakit mata yang ada dalam sistem. Dengan kriteria yang dilakukan berdasarkan identifikasi penyakit yang dialami oleh pasien. Sistem diagnosa ini dibangun menggunakan *forward chaining*. *Forward chaining* yaitu pelacakan ke depan yang dimulai dari sekumpulan fakta-fakta dengan mencari kaidah yang cocok dengan dugaan atau hipotesa yang ada menuju kesimpulan. Pada sistem diagnosa, terdapat banyak sekali metode yang digunakan untuk menangani suatu masalah. *Teorema bayes* dan teori *Dampster-Shafer* merupakan bagian dari teknik probabilitas. Pada penelitian ini *forward chaining* dengan proses yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Pendekatan ini diawali dengan mengumpulkan fakta-fakta dilapangan, yang kemudian di proses untuk mencapai sebuah kesimpulan akhir (Hananto, Sasanko, Sugiharto, 2012). Proses *forward chaining* dimulai dengan memasukkan variasi IF informasi masukan dan berlanjut ke THEN konklusi (Silitonga Budiharto, 2015).

1. Sistem dipresentasikan dengan satu atau lebih dari kondisi.
2. Untuk setiap kondisi sistem akan mencari *rule knowledge base* untuk *rule* tersebut yang cocok dengan kondisi pada bagian IF.
3. Setiap *rule* dapat merubah suatu kondisi baru dari konklusi dari bagian THEN. Kondisi baru ini selanjutnya akan ditambahkan. Ada beberapa kondisi yang telah ditambahkan pada sistem yang akan diproses. Jika ada suatu kondisi, maka sistem akan kembali pada langkah ke-2 dan akan mencari *rule* pada *database* lagi. Jika tidak ada kondisi baru, maka sesi akan berakhir.

Dari hasil pada aplikasi ini diharapkan mampu memberikan gambaran berupa diagnosa awal mengenai penyakit mata pada manusia. Aplikasi ini juga memberikan pengetahuan tentang hal-hal yang berhubungan dengan penyakit mata pada manusia.

Penyakit mata terjadi karena disebabkan oleh alergi, iritasi dan infeksi. Seseorang yang mengidap infeksi mata dapat mengalami gejala yang bermacam, mulai dari ringan hingga berat. Infeksi mata yang ringan biasanya dapat diatasi hanya menggunakan obat-obatan yang dijual bebas diapotik biasa. Namun, pada kasus infeksi mata yang berat, sangat diperlukan penanganan medis oleh para dokter ahli. Maka judul penelitian yang akan diambil adalah **“SISTEM PAKAR DIAGNOSA PENYAKIT MATA PADA MANUSIA MENGGUNAKAN METODE FORWARD CHAINING”**.

## 1.2 Identifikasi Masalah

Beberapa identifikasi masalah yang akan dibahas dalam penelitian diagnosa penyakit mata pada manusia menggunakan metode *forward chaining* sebagai berikut:

1. Bagaimana menerapkan metode *forward chaining* dalam mendiagnosa penyakit mata pada manusia berdasarkan gejala-gejala umum yang terlihat?
2. Bagaimana implementasi metode *forward chaining* pada sistem diagnosa penyakit mata?
3. Bagaimana mengetahui suatu gejala termasuk kedalam penyakit mata?

## 1.3 Tujuan Penulisan

Tujuan dilakukannya penelitian ini adalah menganalisis diagnosa penyakit dengan metode *forward chaining* berbasis *web* yang berisi pengetahuan dari seorang pakar atau dokter yang diyakini kebenarannya untuk mendiagnosa penyakit dari gejala-gejala yang ada secara cepat dan tepat seperti para pakar yang menggunakan metode *forward chaining* untuk menganalisis hasil dari diagnosa yang ada. Adapun tujuan dari penelitian ini adalah:

1. Dapat merancang sistem diagnosa awal penyakit mata pada manusia menggunakan *forward chaining*.
2. Dapat mengimplementasikan metode *forward chaining* pada sistem diagnosa gejala-gejala pada mata.

3. Mengetahui hasil dari penelitian mengenai diagnosa penyakit dengan metode *forward chaining*.

#### **1.4 Batasan Masalah**

Agar pembahasan masalah pada skripsi tidak meluas dari permasalahan yang ada, maka diperlukan batasan masalah terhadap penelitian ini.

1. Aplikasi ini dibuat menggunakan bahasa PHP dan *database MySQL*.
2. Permasalahan penyakit ini yang dibahas merupakan gejala penyakit mata yang dicocokkan sumber sebenarnya.
3. Penelitian ini dilakukan hanya sampai proses *construction* (tidak melakukan proses *deployment*).
4. Sistem yang dibuat tidak bisa memproses penyakit mata lebih dari 1 penyakit.
5. Metode yang digunakan adalah *forward chaining* untuk proses diagnosa.
6. Gejala dan penyakit yang didapat hanya berdasarkan pengetahuan pakar.

## **1.5 Metode Penelitian**

### **1.5.1 Teknik Pengumpulan Data**

Berikut teknik pengumpulan data yang dilakukan diantaranya:

#### **1. Studi literatur**

Peneliti melakukan studi literatur dengan mengumpulkan data-data dengan membaca, dan memahami referensi teoritis yang berasal dari buku-buku, jurnal-jurnal penelitian, dan sumber pustaka lainnya yang berkaitan dengan penelitian. Metode yang digunakan *forward chaining* serta informasi lainnya berkaitan dengan penelitian ini.

#### **2. Pengumpulan data**

Mengumpulkan data dan informasi yang terkait dengan informasi jenis, gejala dan pengobatan penyakit tersebut.

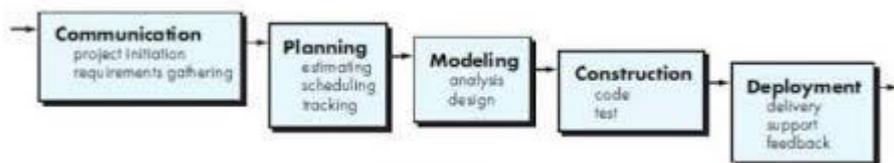
#### **3. Metode wawancara**

Wawancara untuk mendapatkan data-data yang berkaitan dengan penelitian tentang penyakit mata.

### 1.5.2 Metode Perancangan

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh *Winston Royce* sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pressman, 2015).

Fase-fase dalam *Waterfall Model* menurut referensi Pressman:



**GAMBAR: 1.1** *Waterfall Pressman* (Pressman, 2015)

**1. *Communication (Project Initiation & Requirements Gathering)***

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*.

**2. *Planning (Estimating, Scheduling, Tracking)***

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

**3. *Modeling (Analysis & Design)***

Tahapan ini adalah tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan

#### 4. *Construction (Code & Test)*

Tahapan *construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

#### 5. *Deployment (Delivery, Support, Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

### 1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam skripsi ini terbagi dalam beberapa bagian bahasan sebagai berikut:

## **BAB I : PENDAHULUAN**

Berisi pembahasan masalah secara umum, terdiri dari latar belakang, identifikasi masalah, tujuan penelitian, batasan masalah, metode penelitian, dan sistematika penulisan tugas akhir.

**BAB II : LANDASAN TEORI**

Berisi dasar-dasar teori yang berkaitan dengan pemecahan masalah serta pembahasannya yang diperoleh dari buku-buku referensi dan catatan-catatan yang dikembangkan “Sistem pakar diagnosa penyakit mata pada manusia menggunakan metode *forward chaining* beserta pengembangan perancangan perangkat lunak untuk diagnosa penyakit mata pada manusia”.

**BAB III : ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

Bab ini berisikan tentang proses perancangan program sistem diagnosa penentu penyakit kulit berdasarkan analisa permasalahan terlebih dahulu dan disesuaikan dengan metode penelitian yang telah dipilih.

**BAB IV : IMPLEMENTASI DAN UJI COBA**

Bab ini berisikan tentang penelitian yang dilakukan sehingga dapat dibuat suatu perancangan pada program yang dibuat. Disini juga ada tahapan implementasi program, termasuk juga kelebihan dan kekurangan pada program. Terdapat juga evaluasi pada program dengan menggunakan metode kualitatif, kuantitatif, atau metode lainnya yang bisa dan sesuai digunakan.

**BAB V : PENUTUP**

Bab ini menjelaskan mengenai kesimpulan dan saran yang dibutuhkan terhadap hasil akhir dari penelitian yang telah dibuat dan untuk peningkatan topik skripsi yang dapat digunakan di masa mendatang.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Aplikasi**

Aplikasi adalah sebuah perangkat lunak atau program yang diciptakan dan dikembangkan untuk melakukan tugas-tugas tertentu pada perangkat komputer, laptop, atau *smartphone*. Aplikasi berasal dari bahasa Inggris *application* yang artinya penerapan atau penggunaan. Secara teknis *back-end* aplikasi dibuat oleh para programmer atau *developer* dengan menggunakan bahasa pemrograman tertentu. *front-end* aplikasi dapat digunakan oleh para pengguna dengan menginstalnya pada perangkat elektronik agar dapat digunakan untuk berbagai keperluan, seperti mengolah dokumen, komunikasi, desain grafis, manajemen perangkat keras, bermain *game*, dan lain sebagainya. Aplikasi sendiri terbagi menjadi 3 jenis yakni, Aplikasi *desktop* yang dijalankan pada komputer atau *pc*, Aplikasi *web* yang dijalankan menggunakan komputer dengan koneksi *internet* dan Aplikasi *mobile* yang dapat dijalankan di perangkat *mobile*.

“Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*” (Jogiyanto, 1999).

“Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah

suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna” (Kamus Besar Bahasa Indonesia, 1998).

“Aplikasi mobile adalah aplikasi yang telah dirancang khusus untuk *platform mobile* (misalnya *iOS*, *android*, atau *windows mobile*). Dalam banyak kasus, aplikasi *mobile* memiliki *user interface* dengan mekanisme interaksi unik yang disediakan oleh *platform mobile*, *interoperabilitas*, dengan sumber daya berbasis *web* yang menyediakan akses ke beragam informasi yang relevan dengan aplikasi, dan kemampuan pemrosesan *local* untuk pengumpulan, analisis, dan format informasi dengan cara yang paling cocok untuk *platform mobile*. Selain itu aplikasi *mobile* menyediakan kemampuan penyimpanan persisten dalam platform” (Pressman dan Bruce, 2014).

Aplikasi merupakan suatu program atau perangkat lunak yang memanfaatkan kemampuan komputer untuk melakukan suatu tugas yang diinginkan oleh *user*. Sedangkan *mobile* bisa diartikan sebagai perpindahan suatu tempat ke tempat lain. *Mobile* juga bisa diartikan sebagai ringan tangan atau mudah dibawa kemana-mana. Jadi aplikasi *mobile* sendiri berarti aplikasi yang dapat digunakan meskipun penggunanya berpindah tempat tanpa memutuskan suatu komunikasi. Aplikasi *mobile* memiliki karakteristik tersendiri jika dibandingkan dengan aplikasi *web*. Aplikasi *mobile* memiliki karakteristik diantaranya:

1. Mudah diakses, aplikasi *mobile* dapat diakses lebih mudah, karena dilihat dari fisiknya saja yang sangat fleksibel dan mudah dibawa kemana-mana,

aplikasi *mobile* dilengkapi dengan perangkat portable yang memungkinkan *user* dapat terkoneksi internet serta dapat terhubung dengan pengguna lain. Perangkat-perangkat tersebut bisa berupa *smartphone*, tablet PC, *notebook*, atau *mobile device* lainnya. Jadi untuk mengakses aplikasi apapun yang ada dalam perangkat tersebut sangatlah mudah.

2. Bersifat produktif, aplikasi *mobile* bersifat produktif, terbukti dengan sangat seringnya aplikasi *mobile* memberikan notifikasi jika terdapat pembaruan aplikasi. Dengan sangat produktifnya aplikasi *mobile* ini membuat aplikasi *mobile* lebih digemari karena bisa saja semua fitur yang terdapat pada aplikasi *web* sudah terpenuhi bahkan lebih bagus dari aplikasi *web*.
3. Tampilan responsif pada aplikasi *mobile*, layer disesuaikan dengan *gadget* dari pengguna. Dibanding dengan aplikasi *web*, memang aplikasi *web* lebih bagus secara grafis. Dan kebanyakan pengguna pun lebih nyaman menggunakan aplikasi *web*. Tetapi aplikasis *web* juga memiliki kekurangan salah satunya tidak semua aplikasi *web* bersifat responsif.

### **2.1.1 Aplikasi Web**

Aplikasi *web* adalah aplikasi yang disimpan dan dieksekusikan di lingkungan *web server*. Setiap permintaan yang dilakukan oleh *user* melalui aplikasi klien (*web browser*) akan direspon oleh aplikasi *web* dan hasilnya akan dikembalikan lagi ke hadapan *user*. Dengan aplikasi *web*, halaman yang tampil di layar *web browser* dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dimasukkan oleh *user* (Budi Raharjo, 2010).

## 2.2 Pengertian Sistem

“Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu” (Subtari, 2012).

“Sistem dapat didefinisikan sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan” (Nugroho, 2010).

Hal pertama yang perlu diperhatikan dalam suatu sistem adalah elemen-elemennya. Tentunya setiap sistem memiliki elemennya sendiri, yang kombinasinya berbeda antara sistem yang satu dengan sistem yang lain. Namun demikian, susunan dasarnya tetap sama. Maka dapat disimpulkan bahwa sistem adalah sekelompok unsur atau elemen yang berhubungan satu dengan yang lain untuk mencapai suatu tujuan.

## 2.3 Konsep Dasar MySQL

“MySQL adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel” (Kustiyaningsih, 2010).

“MySQL adalah *database server open source* yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat *software database* ini banyak digunakan oleh praktisi untuk membangun suatu *project*. Adanya fasilitas API (*Application Programming Interface* yang dimiliki oleh

*MySQL*, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data *MySQL*” (Komputer, Wahana, 2010).

“Tipe data *MySQL* adalah data yang terdapat dalam sebuah tabel berupa *field-field* yang berisi nilai dari data tersebut. Nilai data dalam *field* memiliki tipe sendiri-sendiri” (Kustiyahningsih, 2016).

#### **2.4 Pengertian PHP**

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari.

PHP merupakan bahasa *scripting server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server*lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

Adapun pengertian lain PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”.

“PHP (atau resminya PHP: *Hypertext Preprocessor*) adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML”. Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server* (Kustiyahningsih, 2010).

Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*.

Selanjutnya *webserver* akan mencari berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh *user* adalah halaman yang mengandung *script* PHP. Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe *file* yang diminta *user*. Jika tipe *file* yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut.

Apabila dalam *file* tersebut tidak mengandung *script* PHP, permintaan *user* akan langsung ditampilkan ke *browser*, namun jika dalam *file* tersebut mengandung *script* PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script-script* PHP dan mengolah *script* tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke *browser user*.

## 2.5 Pengertian HTML

“HTML adalah suatu bahasa yang dikendalikan oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*)” (Oktavian, 2013).

HTML adalah bahasa *Markup* untuk menyebarkan informasi pada *web*. Ketika merancang HTML, ide ini diambil dari *Standard Generalized Markup Language* (SGML). SGML adalah cara yang terstandarisasi dari pengorganisasian dan informasi yang terstruktur dalam dokumen atau sekumpulan dokumen. Walaupun HTML tidak dengan mudah dapat dipahami kebanyakan orang, ketika diterbitkan penggunaannya menjadi jelas.

“HTML adalah bahasas pemrograman yang fleksibel dimana kita bisa meletakkan *script* dari bahasa pemrograman lain seperti *JAVA*, *Visual Basic*, *C* dan lain-lain. Jika HTML tersebut tidak dapat mendukung suatu perintah pemrograman tertentu *browser* tidak akan menampilkan kotak dialog “*Syntax Error*” jika terdapat penulisan kode yang keliru pada *script* HTML sepanjang kode-kode yang kita tuliskan merupakan kode-kode HTML tanpa penambahan kode-kode dari luar seperti java. Oleh karena itu, jika terjadi *syntax error* pada *script* HTML, efek yang paling jelas adalah HTML tersebut tak akan ditampilkan pada halaman jendela *browser*” (Rio, 2005).

## 2.6 Pengertian phpMyAdmin

“*PhpMyAdmin* adalah *tools* yang dapat digunakan dengan mudah untuk memanajemen database *MySQL* secara visual dan *Server MySQL*, sehingga kita tidak perlu lagi harus menulis *query SQL* setiap akan melakukan perintah operasi *database*. *Tools* ini cukup populer, anda dapat mendapatkan fasilitas ini ketika menginstal paket triad *phpMyAdmin*, karena termasuk dalam *Xampp* yang sudah diinstal” (Nugroho, 2013).

“*PhpMyAdmin* merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan *command SQL*” (Hikmah, 2015).

## 2.7 Pengertian Xampp

“*Xampp* (*X*(*Windows/Linux*) *Apache MySQL PHP dan Perl*) merupakan paket *server web PHP* dan *database MySQL* yang paling populer dikalangan pengembang *web* dengan menggunakan *PHP* dan *MySQL* sebagai databasenya” (Betha Sidik, 2014).

“*XAMPP* merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket, adapun paketnya sudah terdapat *Apache* (*Web Server*), *MySQL* (*Database*), *PHP* (*ServerSide Scripting*), *Perl*, *FTP Server*, *PhpMyAdmin*, dan berbagai pustka bantu lainnya. Dengan menginstal *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*” (Kartini, 2016).

Adapun pengertian dari *XAMPP* adalah sebagai berikut:

1. X yang berarti program ini dapat dijalankan di banyak sistem operasi
2. A yang berarti *Apache* merupakan suatu aplikasi *web server*
3. M yang berarti *MySQL* merupakan untuk penyimpanan *database server*
4. P yang berarti *PHP* merupakan bahasa pemrograman yang dipakai untuk membangun *website* dinamis.
5. P yang berarti *Perl* bahasa pemrograman untuk segala keperluan. *Perl* merupakan penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum, *perl* sangat populer digunakan dalam pemrograman CGI (*Common Gateway Interface*).

Bagian penting dari *XAMPP* yang biasa digunakan:

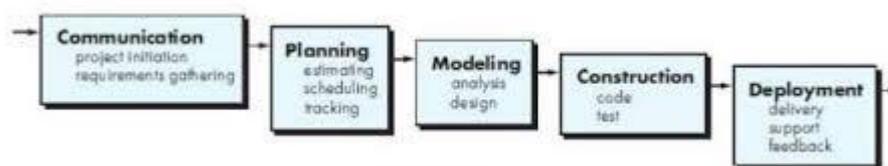
1. *Hitdoc* adalah *folder* tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas *PHP*, *HTML*, dan *script* lainnya.
2. *PhpMyAdmin* merupakan bagian untuk mengelola basis data *MySQL* yang ada di komputer. Untuk membukanya, buka *browser* lalu ketikkan alamat <http://localhost/phpMyAdmin>. maka akan muncul halaman *phpMyAdmin*.
3. Kontrol panel yang berfungsi untuk mengelola layanan (*service*) *XAMPP*, seperti menghentikan layanan (*stop*) atau memulai layanan (*start*).

## 2.8 Metode Pengembangan

“Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*” (Pressman, 2015).

Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh *Winston Royce* sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Fase-fase dalam *Waterfall Model* menurut Pressman:



**GAMBAR: 2.1** *Waterfall Pressman* (Pressman, 2015)

### 1. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek,

seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*.

## **2. *Planning (Estimating, Scheduling, Tracking)***

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem

## **3. *Modeling (Analysis & Design)***

Tahapan ini adalah tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan

## **4. *Construction (Code & Test)***

Tahapan *construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

## 5. *Deployment (Delivery, Support, Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

## 2.9 Flowchart

“*Flowchart* atau bagan alir merupakan metode untuk menggambarkan tahap-tahap penyelesaian masalah (prosedur) beserta aliran data dengan simbol-simbol standar yang mudah dipahami” (Soeherman, Bonnie dan Marion Pinontoan, 2008).

*Flowchart* dibedakan dalam tiga kategori, yaitu:

### 1. *Flowchart* Dokumen

*Flowchart* dokumen menggambarkan aliran data dan informasi antar-area dalam sebuah organisasi.

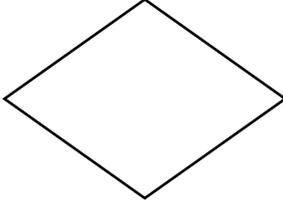
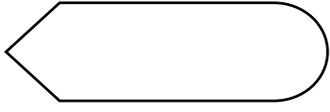
### 2. *Flowchart* Sistem

*Flowchart* sistem lebih menekankan pada gambaran tentang aliran input, prosedur pemrosesan, dan output yang dihasilkan sistem.

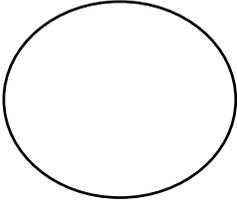
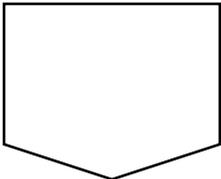
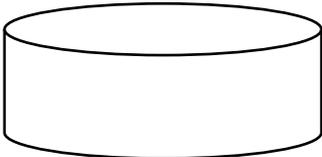
### 3. *Flowchart* Program

*Flowchart* program lebih banyak digunakan untuk menggambarkan logika suatu prosedur penyelesaian masalah dalam pemrograman komputer.

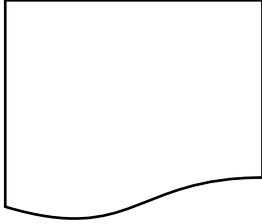
**TABEL: 2.1 Simbol *Flowchart***

NO	SIMBOL	FUNGSI
1		Terminal untuk memulai dan mengakhiri suatu proses/kegiatan
2		Proses sesuatu yang menunjukkan semua pengolahan yang dilakukan komputer
3		Input untuk memasukkan hasil dari suatu proses
4		Decision, suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5		Display, output yang ditampilkan di layar terminal

Lanjutan TABEL: 2.1 Simbol *Flowchart*

6		Connector, simbol untuk keluar/masuk prosedur atau proses dalam lembar yang sama
7		Off Page Connector, simbol untuk keluar/masuk proses dalam lembar yang berbeda
8		Hard Disk Storage, input/output yang menggunakan hardisk
9		Predefied Process, untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
10		Arus simbol yang digunakan untuk menghubungkan antara simbol satu dengan yang lain

**Lanjutan TABEL: 2.1 Simbol *Flowchart***

11		<p>Printer, simbol ini digunakan untuk menggambarkan suatu dokumen atau kegiatan mencetak suatu informasi dengan mesin printer</p>
----	---	--

### **2.10 Pengertian *Database***

“Basis Data (*Database*) adalah sekumpulan informasi yang berkaitan dengan subjek yang diorganisasikan dengan cara tertentu sebagai basis atau pondasi untuk prosedur seperti mengakses informasi, menarik kesimpulan dan membuat keputusan” (Sanjaya, 2013).

“Basis data merupakan koleksi dari data-data yang terorganisir dengan cara sedemikian rupa sehingga data tersebut mudah disimpan dan dimanipulasi” (Muhammad Luqman, 2012).

### **2.11 Pengertian UML (Unified Modeling Language)**

“*Unified Modeling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan piranti lunak” (Risma, 2015).

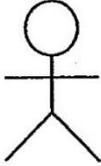
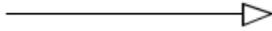
“Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif, UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan” (Fowler,2004).

### **2.11.1 Use Case Diagram**

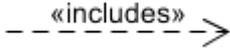
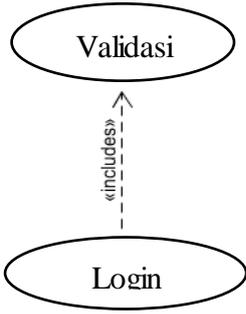
*Use case* atau *Diagram Use case* merupakan pemodelan untuk melakukan (*behaviour*) sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

“*Use Case Diagram* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat secara kasar. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu” (Sukanto dan Shalaluddin, 2015).

**TABEL: 2.2 Simbol Use Case Diagram (Sukamto dan Shalahuddin, 2015)**

Simbol	Deskripsi
<p data-bbox="518 562 635 595"><i>Use case</i></p> 	<p data-bbox="930 562 1343 846">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p data-bbox="467 860 644 893"><i>Aktor/ Actor</i></p> 	<p data-bbox="930 860 1343 1272">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="435 1285 719 1319"><i>Asosiasi / association</i></p> 	<p data-bbox="930 1285 1343 1444">Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan</p>
<p data-bbox="435 1458 660 1491"><i>Ekstensi/ extend</i></p> 	<p data-bbox="930 1458 1343 1653">Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.</p>
<p data-bbox="395 1666 770 1700"><i>Generalisasi / generalitation</i></p> 	<p data-bbox="930 1666 1343 1912">Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.</p>

Lanjutan TABEL: 2.2 Simbol *Use Case Diagram* (Sukamto dan Shalahuddin, 2015)

Simbol	Deskripsi
<p data-bbox="411 638 592 674"><i>Include / uses</i></p> 	<p data-bbox="959 638 1342 842">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat yang dijalankan <i>use case</i> ini.</p> <p data-bbox="959 891 1337 1010">Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul data-bbox="914 1061 1337 1265" style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul> 

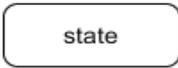
### 2.11.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

**TABEL: 2.3 Simbol *Activity Diagram* (Sukamto & Shalahuddin, 2015)**

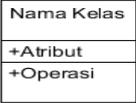
Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### 2.11.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

**TABEL: 2.4 Class Diagram (Sukamto & Shalahuddin, 2015)**

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi / <i>Association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum – khusus)
<p>Kebergantungan/ <i>dependency</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.

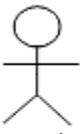
Lanjutan TABEL: 2.4 Simbol *Class Diagram* (Sukamto & Shalahuddin, 2015)

Simbol	Deskripsi
 Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna semua – bagian ( <i>whole – part</i> )

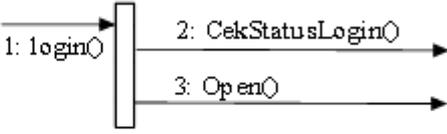
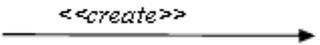
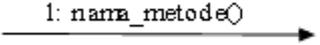
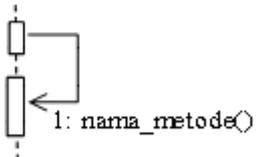
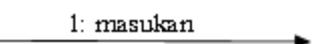
#### 2.11.4 *Sequence Diagram*

“Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek yang terjadi pada titik tertentu dalam eksekusi sistem” (Fowler, 2014).

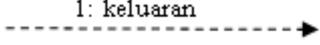
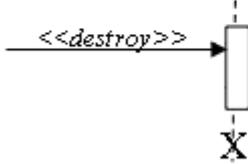
TABEL: 2.5 Notasi *Sequence Diagram*

Simbol	Deskripsi
Aktor/ <i>actor</i>  Nama aktor atau <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama aktor</div> tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang, biasanya di nyatakan menggunakan kata benda di awal frase nama aktor.

Lanjutan TABEL: 2.5 Notasi *Sequence Diagram*

<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">nama objek : nama kelas</div>	<p>Menyatakan objek yang berinteraksi dengan pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>maka <code>CekStatusLogin()</code> dan <code>Open()</code> dilakukan didalam metode <code>login()</code> Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain. arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada kelas objek yang berinteraksi.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, Arah panah mengarah pada objek yang dikirim pesan.</p>

Lanjutan TABEL: 2.5 Notasi *Sequence Diagram*

Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

## 2.12 Pengertian Penyakit

Penyakit adalah suatu keadaan abnormal dari tubuh atau pikiran yang menyebabkan ketidaknyamanan, *disfungsi* atau kesukaran terhadap orang yang dipengaruhinya. Dari pengertian penyakit kita bisa mengambil kesimpulan, bahwa adanya suatu keadaan yang tidak normal, dan gangguan pada tubuh maupun pikiran seseorang, yang menyebabkan rasa tidak nyaman, bisa disebut sebagai sebuah penyakit.

“Penyakit adalah gangguan pada bagian tubuh atau gangguan kesehatan”  
 (Ramli, 1990).

### 2.13 Pengertian Diagnosa

Diagnosa dapat diartikan sebagai berikut:

1. Upaya atau proses menemukan kelemahan atau penyakit (*weakness disease*) apa yang dialami seseorang dengan melalui pengujian dan studi yang seksama mengenai gejala-gejalanya (*symptoms*).
2. Studi yang seksama terhadap fakta tentang suatu hal untuk menemukan karakteristik atau kesalahan-kesalahan dan sebagainya yang *esensial*.
3. Keputusan yang dicapai setelah dilakukan suatu studi yang seksama atas gejala-gejala atau fakta tentang suatu hal (Thorndike & Hagen, 2011).

### 2.14 Pengertian Mata

“Mata adalah salah satu organ tubuh vital manusia. Oleh karena itu, kita harus selalu menjaga dan mencegah hal-hal yang dapat merusak mata” (Murtopo & Sarimurni, 2005).

“Mata merupakan indera penglihatan pada manusia. Mata dibentuk untuk menerima rangsangan berkas-berkas cahaya pada retina selanjutnya dengan perantara serabut-serabut *nervus optikus*, mengalihkan rangsangan ini ke pusat penglihatan pada otak untuk ditafsirkan. Mata itu tidak melihat, pola cahaya mengubahnya menjadi *impuls* saraf dan mengirimkannya di sepanjang saraf optik ke otak. Saraf *neuron optikus* yang berfungsi untuk penglihatan” (Irianto, 2010).

Mata terdiri atas 6 bagian, antara lain:

1. Kelopak mata yang berfungsi untuk melindungi bola mata terhadap sinar dan pengeringan bola mata
2. Sistem sekresi air mata berfungsi untuk menjaga kornea agar tetap bersih, lembab, dan bebas kuman.
3. *Konjungtiva* adalah membrane yang menutupi *sklera* dan kelopak mata bagian belakang.
4. Bola mata terdiri atas 3 lapisan jaringan yaitu, *Sklera*, *Uvea*, dan *Retina*.
5. Rongga *orbita* yaitu rongga tempat bola mata.
6. Otot penggerak mata yaitu otot yang berfungsi untuk menggerakkan bola mata (Ilyas, 2003).

### **2.15 Forward Chaining**

“*Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rules* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian THEN) informasi masukan dapat berupa data, bukti temuan atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan, atau diagnosa. Sehingga jalannya *forward chaining* maju dapat dimulai dari data menuju tujuan dan bukti menuju hipotesa, dari temuan menuju penjelasan, atau dari pengamatan menuju diagnose” (Hartati, 2008).

Ditambahkan ke dalam *database* langkah-langkah membuat sistem diagnosa dengan menggunakan metode *forward chaining* yaitu:

1. Pendefinisian masalah dimulai dengan pemilihan *domain* masalah dan akuisi pengetahuan.
2. Pendefinisian data input untuk memulai *inferensi* karena diperlukan oleh sistem *forward chaining*.
3. Pendefinisian struktur pengendalian data untuk membantu mengendalikan pengaktifan suatu aturan.
4. Penulisan kode awal dalam *domain* pengetahuan.
5. Pengujian sistem agar dapat mengetahui sejauh mana sistem berjalan.
6. Perancangan antarmuka dengan basis pengetahuan
7. Pengembangan sistem
8. Evaluasi sistem

Contoh aturan menggunakan *forward chaining*

R1     IF A & B THEN C

R2     IF C THEN D

R3     IF A & E THEN F

R4     IF A THEN G

R5     IF F & G THEN D

R6     IF G & E THEN H

R7 IF C & H THEN I

R8 IF I & A THEN J

R9 IF G THEN J

R10 IF J THEN K

Contoh aturan menggunakan *forward chaining* terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Jika fakta awal yang diberikan hanya: A dan F (artinya A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesa: K).

Langkah-langkah inferensi adalah sebagai berikut:

1. Dimulai dari R1, A merupakan fakta sehingga bernilai benar, sedangkan B belum bisa diketahui kebenarannya, sehingga C pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun pada R1 ini, sehingga kita menuju ke R2
2. Pada R2 tidak diketahui informasi apapun tentang C, sehingga tidak bisa dipastikan kebenaran D. Oleh karena itu tidak didapatkan informasi apapun pada R1 ini, sehingga harus menuju ke R3.
3. Pada R3, baik A maupun E adalah fakta sehingga jelas benar. Dengan demikian F sebagai konsekuen juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu F. karena F bukan hipotesis yang hendak dibuktikan (=K) maka penelusuran dilanjutkan ke R4.

4. Pada R4, A adalah fakta sehingga jelas benar, dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang didapatkan fakta baru yaitu G. karena G bukan hipotesis yang hendak dibuktikan ( $=K$ ), maka penelusuran dilanjutkan ke R5.
5. Pada R5, baik F maupun G bernilai benar berdasarkan aturan R3 dan R4, dengan demikian G sebagai konsekuen juga ikut benar. Sehingga 11 sekarang terdapat fakta baru yaitu D. karena D bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R6.
6. Pada R6, baik A maupun G adalah ebnar berdasarkan fakta dari R4, dengan demikian H sebagai konsekuen juga ikut benar. Sehingga sekarang terdapat fakta baru yaitu H. karena H bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R7.
7. Pada R7, meskipun H benar berdasarkan R6, namun tidak diketahui kebenaran C sehingga I pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun pada R7 ini sehingga dilanjutkan ke R8.
8. Pada R8, meskipun A benar karena fakta, namun tidak diketahui kebenaran I, sehingga J pun juga belum bisa diketahui kebenarannya. Oleh karena itu tidak didapatkan informasi apapun ada R8 ini sehingga dilanjutkan menuju ke R9.
9. Pada R9, J bernilai benar karena G benar berdasarkan R4. Karena J bukan hipotesis yang hendak dibuktikan, maka penelusuran dilanjutkan ke R10.

10. Pada R10, K bernilai benar karena J benar berdasarkan R9, karena K sudah merupakan hipotesis yang hendak dibuktikan maka terbukti bahwa K adalah benar.

## 2.16 Sistem Pakar

Sistem pakar (*expert system*) adalah salah satu teknik kecerdasan buatan yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli. (Kusumadewi, 2003). Pada dasarnya sistem pakar diterapkan untuk mendukung aktifitas pemecahan masalah. Beberapa aktifitas pemecahan masalah yang dimaksud seperti (Lestari, 2012):

- a. Interpretasi

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. Pengambilan keputusan dari hasil observasi, termasuk pengelolaan ucapan, analisis citra, interpretasi sinyal, dll.

- b. Prediksi

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu. Contoh: prediksi demografi, prediksi ekonomi, dll.

- c. Diagnosa

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis, dll.

d. Perancangan

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu.

Contoh: perancangan *layout* sirkuit, bangunan.

e. Perencanaan

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer, dll.

f. *Monitoring*

Membandingkan hasil pengamatan dengan kondisi yang diharapkan.

Contoh: *computer aided monitoring system*.

g. *Debugging*

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.

Contoh: memberikan resep obat terhadap kegagalan.

h. Intruksi

Mendeteksi dan mengkoreksi defisiensi dalam pemahaman domain subjek.

Contoh: melakukan instruksi untuk diagnosis dan *debugging*.

i. Kontrol

Mengatur tingkah laku suatu *environment* yang kompleks. Contoh: melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan *monitoring* sistem.

## **BAB III**

### **ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

#### **3.1 Communication**

Pada tahap ini dilakukan proses pengumpulan informasi berupa data yang berkaitan dengan penelitian.

##### **3.1.1 Pengumpulan Data**

Pengumpulan data merupakan langkah yang dilakukan peneliti untuk mendapatkan data-data yang berkaitan dengan penyakit mata untuk mendukung penelitian ini yang sedang dilakukan peneliti. Teknik pengumpulan data yang dilakukan dalam penelitian ini adalah:

###### **3.1.1.1 Metode Wawancara**

Wawancara untuk mendapatkan data-data yang berkaitan dengan penelitian tentang penyakit mata.

###### **3.1.1.2 Studi Literatur**

Peneliti melakukan studi literatur dengan mengumpulkan data-data dengan membaca, dan memahami referensi teoritis yang berasal dari buku-buku, jurnal-jurnal penelitian, dan sumber pustaka lainnya yang berkaitan dengan penelitian. Metode yang digunakan *Forward Chaining* dan informasi lainnya berkaitan dengan penelitian, berikut adalah tabel literatur yang menjadi referensi penelitian:

**TABEL: 3.1 Referensi Penelitian**

<b>NO</b>	<b><i>Literature</i></b>	<b>Pembahasan</b>
1	Akil, 2017 “Analisa Efektifitas Metode Forward Chaining Dan Backward Chaining Pada Sistem Pakar” Jurnal Pilar Nusa Mandiri Volume 13 No.1 Maret	Menganalisa sejauh mana efektifitas kedua metode tersebut dalam penggunaannya pada sistem pakar.
2	Putra, dkk, 2018 “Sistem Pakar Diagnosa Penyakit Mata dengan Fuzzy Logic dan Naïve Bayes” Jurnal Merpati Volume 6 No.1 April	Penelitian ini dibuat untuk membuktikan kemiripan antara sistem pakar dengan pakar aslinya sebesar 81%.
3	Purba, 2017 “Perancangan Sistem Pakar Diagnosa Penyakit Mata Katarak Menggunakan Konsep Metode Runut Mundur” JUSIKOM PRIMA Vol 1 No.1 Juni	Penelitian ini dibuat untuk mendiagnosa penyaki katarak dengan memberikan beberapa pertanyaan berupa gejala yang dirasakan.

Lanjutan TABEL: 3.1 Referensi Penelitian

	<i>Literature</i>	<b>Pembahasan</b>
4	Yanto, dkk, 2017 “Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Bawah Lima Tahun Menggunakan Metode <i>Forward Chaining</i> ” <i>Journal of Information System Engineering and Business Intelligence</i> Vol.3 No.1 April	Penelitian ini dibuat untuk menunjukkan tingkat akurasi metode <i>Forward Chaining</i> dan sistem yang telah dibuat sebesar 82% dengan menggunakan 50 data uji coba.
5	Rahmatullah, dkk, 2018 “Sistem Pakar Untuk Mendiagnosa Penyakit Mata Dengan Metode <i>Forward Chaining</i> ” <i>Journal Speed Sentra Penelitian Engineering dan Edukasi</i> Volume 10 No.2	Penelitian ini dibuat untuk mengetahui atau mendiagnosa penyakit mata dari gejala-gejala yang diderita oleh pasien menggunakan metode <i>Forward Chaining</i> .

### 3.1.2 Analisis Sistem Pakar

Dalam membangun sebuah perangkat lunak sistem pakar untuk mendiagnosa penyakit mata berbasis *web* dilakukan beberapa tahap yaitu:

1. Menentukan masalah yang akan dibangun untuk sebuah perangkat lunak sistem pakar. Sistem pakar yang akan dibangun merupakan sebuah

perangkat lunak sistem pakar untuk mendiagnosa penyakit mata berbasis *web*.

2. Mengumpulkan data yang diperlukan untuk membangun sistem, yaitu berupa informasi tentang pengertian penyakit, gejala, jenis penyakit dan cara pengobatannya melalui studi literatur dan observasi yang digunakan sebagai *base knowledge*.
3. Mempresentasikan pengetahuan ke dalam tabel gejala yang telah dianalisis, aturan produksi serta pohon pelacakan dan penelusuran gejala dan jenis penyakit.
4. Usulan sistem yang akan dibangun.

### **3.1.3 Analisis Identifikasi Masalah**

Analisis masalah merupakan langkah dimana mengidentifikasi masalah yang akan dikaji, dalam hal ini dengan mengidentifikasi masalah yang akan dibuat terlebih dahulu, adapun masalah-masalah yang akan diambil dalam pembangunan sistem pakar untuk mendiagnosa penyakit mata beserta solusi penanganannya.

Jenis penyakit dan ciri-ciri penyakit mata sebagai berikut:

1. Katarak

Katarak adalah suatu penyakit ketika lensa mata menjadi keruh dan berawan. Pada umumnya, katarak berkembang perlahan dan awalnya tidak terasa mengganggu. Namun, lama-kelamaan, katarak akan mengganggu penglihatan dan membuat pengidap merasa seperti melihat jendela

berkabut, sulit menyetir, membaca, serta melakukan aktifitas sehari-hari.

Penyakit ini penyebab kebutaan di dunia yang dapat diobati.

## 2. *Ablasio*

*Ablatio* retina atau ablasi retina adalah penyakit mata terlepasnya retina dari jaringan penyokongnya. Retina merupakan lapisan tipis didalam mata yang kaya akan sel-sel yang peka terhadap cahaya. Retina berfungsi memproses cahaya yang ditangkap dan diubah menjadi sinyal listrik kemudian diteruskan oleh saraf optic dan diinterpretasi oleh otak.

## 3. *Astigmatisme*

*Astigmatisme* merupakan penyakit gangguan penglihatan yang disebabkan oleh cacat pada lengkungan lensa atau kornea. Kondisi ini dapat mengakibatkan pandangan menjadi terdistorsi atau kabur. *Astigmatisme* umumnya muncul saat lahir, tetapi bisa juga disebabkan oleh cedera yang dialami oleh mata dikemudian hari atau sebagai komplikasi dari operasi mata. Ketajaman penglihatan pada berbagai jangkauan jarak akan berkurang. Hal tersebut dapat dialami oleh pengidap rabun jauh atau disebut juga *myopia* dan rabun dekat atau disebut *hioermetropi*.

## 4. Glaukoma

Glaukoma adalah jenis gangguan penglihatan yang ditandai dengan terjadinya kerusakan saraf mata. Kerusakan ini sering disebabkan oleh tekanan tinggi pada mata.

#### 5. *Bupthalmus*

*Bupthalmus* adalah kondisi terjadinya pembesaran bola mata. Kondisi ini sering kali ditemukan pada anak-anak sejak lahir hingga usia tiga tahun. Istilah lain yang dapat digunakan untuk menggambarkan keadaan ini adalah *ox eye* atau mata sapi. *Bupthalmus* dapat ditemukan pada salah satu atau kedua bola mata.

#### 6. *Konjungtivitis*

*Konjungtivitis* atau yang dikenal juga sebagai *pinkeye*, adalah peradangan *konjungtiva*, yaitu selaput bening yang menutupi bagian putih mata (*sklera*) dan bagian dalam kelopak mata. *Konjungtivitis* paling sering menyerang anak-anak, karena penyakit mata ini bisa menyebar dengan cepat.

#### 7. *Pterygium*

*Pterygium* adalah penyakit mata, dimana terjadi pertumbuhan selaput jaringan berbentuk segitiga berwarna merah muda dan biasanya muncul pada bagian putih bola mata. *Pterygium* juga dikenal sebagai mata peselancar atau *surfer's eye*. Kondisi ini dimulai pada kornea di dekat hidung dan dapat bertumbuh hingga pupil.

#### 8. *Blefaritis*

*Blefaritis* adalah penyakit pada mata yang timbul karena peradangan pada tepi kelopak mata. Kondisi ini membuat bagian tersebut jadi bengkak dan memerah. Umumnya *blefaritis* membuat salah satu mata terlihat lebih meradang, ketimbang mata yang satunya.

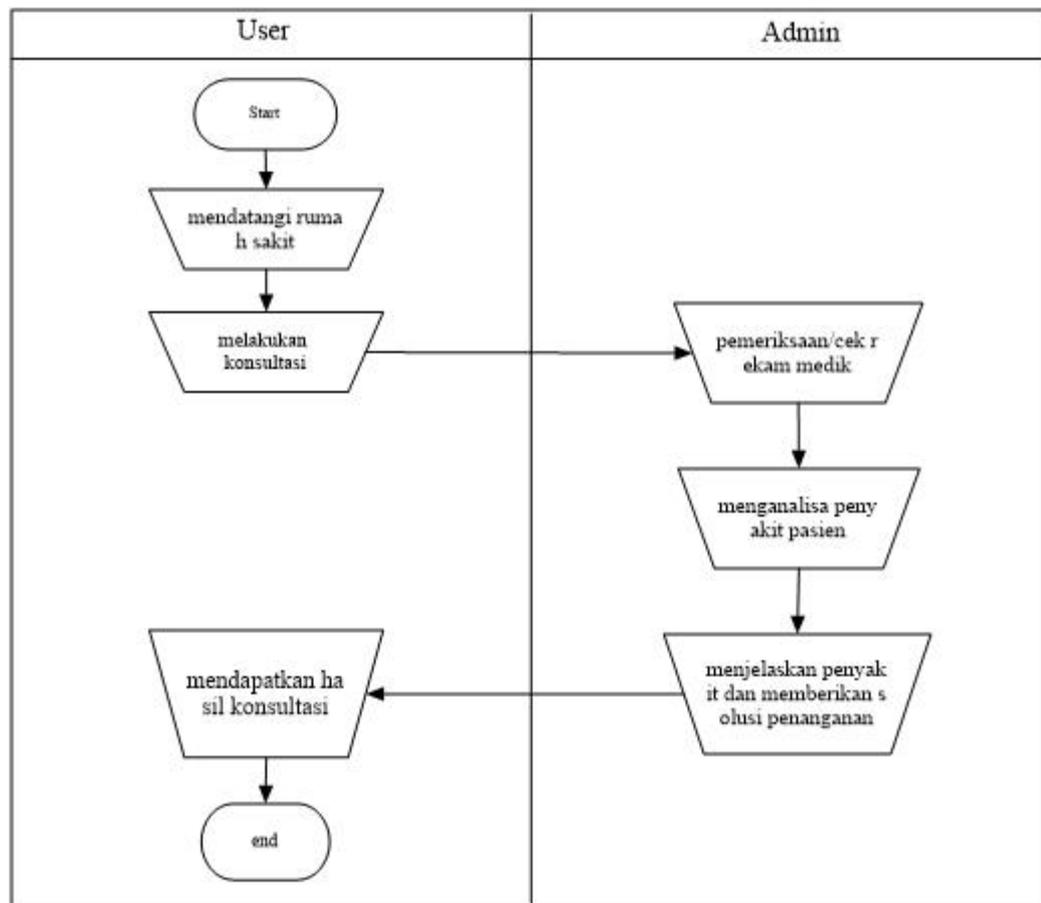
#### 9. *Dakriosistitis*

*Dakriositis* adalah penyakit infeksi kelenjar air mata. Mata yang mengalami kondisi ini akan mengalami luka dan pembengkakan serta tampak kemerahan pada bagian mata yang berbatasan dengan hidung. Hal ini disebabkan oleh proses peradangan yang dapat berlangsung secara akut maupun kronis.

#### **3.1.4 Analisis Sistem Yang Sedang Berjalan**

Sistem yang sekarang yang sedang berjalan pada diagnosa penyakit mata yaitu sebagai berikut:

1. Orang yang akan melakukan konsultasi harus datang langsung ke dokter spesialis mata.
2. Orang tersebut melakukan interaksi dengan dokter spesialis mata.
3. Setelah melakukan interaksi, orang tersebut harus melakukan tes mata.
4. Hasil interaksi dan tes mata tersebut di diagnosa oleh dokter spesialis kemudian dokter spesialis menyimpulkan penyakit mata apa yang diderita oleh orang tersebut.
5. Orang yang berkonsultasi akan mendapatkan hasil dari apa yang telah dikonsultasi dengan dokter spesialis tersebut.



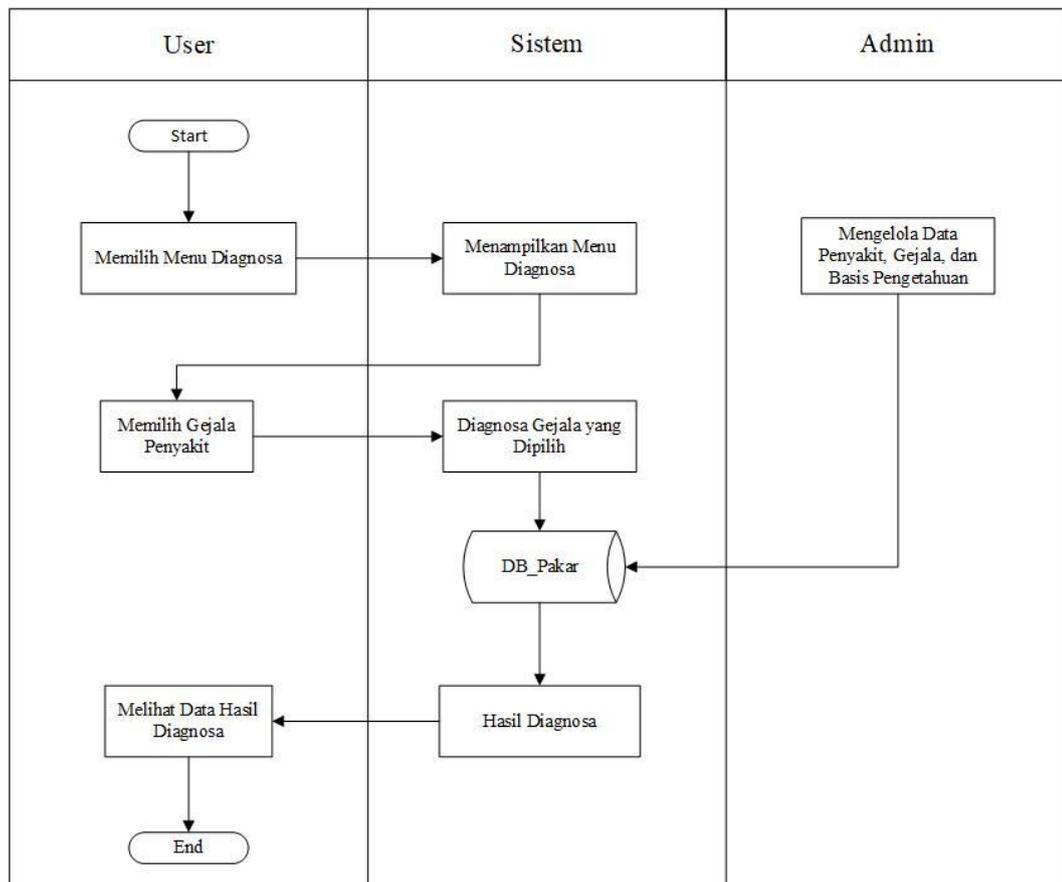
**GAMBAR: 3.1** *Flowmap* Sistem Yang Sedang Berjalan

### 3.1.5 Analisis Sistem Diusulkan

Sistem pakar diagnosa penyakit mata menggunakan metode *forward chaining* sebagai berikut dan diperlihatkan pada gambar 3.2 *flowmap* sistem yang diusulkan

1. Peserta konsultasi masuk ke halaman *web* diagnosa penyakit mata dan klik menu diagnosa.
2. Peserta menjawab pertanyaan yang ditampilkan.

3. Peserta konsultasi akan diberikan informasi dari hasil jawaban yang telah dilakukan sebelumnya. Jawaban dari konsultasi sebelumnya berupa gejala dari penyakit mata. Peserta akan mendapatkan hasil berupa penyakit mata apa yang didiagnosa yang didapat dari gejala penyakit mata.
4. Peserta konsultasi akan diberikan informasi penyebab dan solusi penyakit yang di diagnosa oleh sistem.



**GAMBAR: 3.2 Flowmap Sistem Yang Diusulkan**

Analisis terhadap sistem yang berjalan bertujuan untuk mengetahui bagaimana cara kerja suatu sistem dan mengetahui masalah yang dihadapi sistem

untuk dapat dijadikan sebagai landasan usulan perancangan. Tahap analisis diperlukan untuk mengetahui bagaimana proses yang bisa dijalankan pada sistem pakar mendiagnosa penyakit mata. Penulis menggambarkan sistem yang berjalan dalam bentuk *flowchart*, observasi, proses mendiagnosa penyakit mata pada sistem pakar ini dilakukan sebagai berikut:

1. Peserta konsultasi melakukan konsultasi dengan cara menjawab pertanyaan yang terdapat pada sistem pakar.
2. Setelah melakukan konsultasi peserta konsultasi akan mendapatkan hasil dari penyakit mata yang di diagnosa berdasarkan gejala penyakit mata.

### **3.1.6 Analisis Pengguna**

Agar penggunaan aplikasi dapat berjalan secara optimal dan sesuai dengan yang diharapkan, maka dibutuhkan kemampuan untuk bisa menjalankan aplikasi sistem pakar yang dirancang oleh peneliti. Dalam hal ini, kemampuan tersebut terutama adalah dapat mengoperasikan sistem pakar dengan struktur pada aplikasi mudah untuk dioperasikan oleh peserta konsultasi seperti masyarakat yang mengalami gejala dan ciri-ciri penyakit mata.

#### **3.1.6.1 Peserta Konsultasi**

Peserta konsultasi disini bertujuan untuk melakukan konsultasi terhadap perilaku keseharian peserta. Sistem pakar ini akan menampilkan hasil konsultasi yang telah dilakukan oleh peserta konsultasi.

### 3.1.6.2 Pakar

Pakar disini bertugas mengelola sistem, menambahkan, mengedit, dan menghapus gejala penyakit dan penyakit untuk sistem pakar yang dirancang

### 3.1.6.3 Analisis Perangkat Keras

Analisis kebutuhan perangkat keras dimaksudkan untuk mengetahui spesifikasi perangkat keras yang sedang digunakan pada sistem pakar ini adalah sebagai berikut:

1. *Processor* dengan kecepatan minimal 1GHz + *Motherboard*
2. RAM minimal 2 GB
3. *Hard Disk* minimal 1TB
4. Monitor
5. Mouse
6. Keyboard

Secara keseluruhan spesifikasi perangkat keras semua yang ada sudah memenuhi syarat untuk kebutuhan yang akan diaplikasikan.

### 3.1.6.4 Analisis Perangkat Lunak

Perangkat lunak yang digunakan peserta konsultasi dan dokter spesialis mata, adalah sebagai berikut:

1. Perangkat penelusuran seperti *Google Chrome, Mozilla Firefox, Microsoft Edge* dan *Internet Explorer*.

Perangkat lunak yang digunakan dalam perancangan aplikasi ini adalah sebagai berikut:

1. Sistem Operasi *Windows 10*
2. *MySQLi* sebagai DBMS
3. *PHP*
4. *Apache*
5. *Jquery*

Berdasarkan analisis perangkat lunak yang telah dijelaskan diatas sebagian besar sudah mencukupi untuk dapat menjalankan aplikasi yang akan dirancang.

#### **3.1.6.5 Sumber Informasi**

Data mengenai penyakit mata aplikasi sistem pakar diagnosa penyakit mata, terkait gejala penyakit, jenis penyakit, dan solusi penyakit tersebut didapatkan dari jurnal, website serta rumah sakit.

### **3.2 Planning**

Dalam tahap penelitian ini memfokuskan pada penjadwalan pengerjaan penelitian. Pada penelitian ini terdapat beberapa proses yang harus dilakukan dari tahap *communication* hingga *implementasi* dan *testing* maka dari itu diperlukan penjadwalan yang tepat agar penelitian ini dapat selesai tepat pada waktunya. Berikut adalah penjadwalan penelitian berdasarkan aktifitas yang dilakukan dengan skala waktu.

Tabel 3.2 dibawah ini akan menjelaskan penjadwalan penelitian:



### 3.3 Modeling

Peneliti telah melakukan proses akuisisi pengetahuan dengan mengumpulkan pengetahuan dan fakta dari sumber-sumber yang tersedia. Sumber pengetahuan dan fakta diperoleh melalui wawancara dengan pakar atau konselor dan studi literatur tentang materi yang berkaitan dengan penyakit mata. Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan penyakit mata, gejala penyakit mata didapat berdasarkan penelitian sebelumnya, *web* resmi dan juga rumah sakit.

**TABEL: 3.3 Tabel Daftar Penyakit Mata**

<b>Kode Penyakit</b>	<b>Penyakit</b>
P01	Penyakit Katarak
P02	Penyakit Ablasio
P03	Penyakit Astigmatisme
P04	Penyakit Glaukoma
P05	Penyakit Bufthalmus
P06	Penyakit Konjungtivitis
P07	Penyakit Pterygium
P08	Penyakit Blefaritis
P09	Penyakit Dakriosistitis

**TABEL: 3.4 Tabel Daftar Gejala Penyakit**

<b>Kode Gejala Penyakit</b>	<b>Gejala Penyakit</b>
<b>G01</b>	Pandangan kabur seperti berkabut
<b>G02</b>	Pandangan ganda
<b>G03</b>	Warna di sekitar memudar
<b>G04</b>	Melihat kilatan cahaya (Fotopsis)
<b>G05</b>	Terdapat bintik hitam yang selalu bergerak/melayang
<b>G06</b>	Lapangan pandang yang menyempit
<b>G07</b>	Pandangan yang samar atau tidak fokus
<b>G08</b>	Kesulitan membedakan warna-warna yang bersebelahan
<b>G09</b>	Sensitif terhadap sorotan cahaya
<b>G10</b>	Melihat bayangan lingkarang di sekeliling cahaya
<b>G11</b>	Mata memerah
<b>G12</b>	Penglihatan yang semakin menyempit hingga akhirnya tidak dapat melihat objek sama sekali
<b>G13</b>	Gangguan penglihatan
<b>G14</b>	Pembesaran kornea
<b>G15</b>	Kemerahan pada konjungtiva mata
<b>G16</b>	Mata sering terasa gatal dan seperti ada debu

Lanjutan TABEL: 3.4 Tabel Daftar Gejala Penyakit

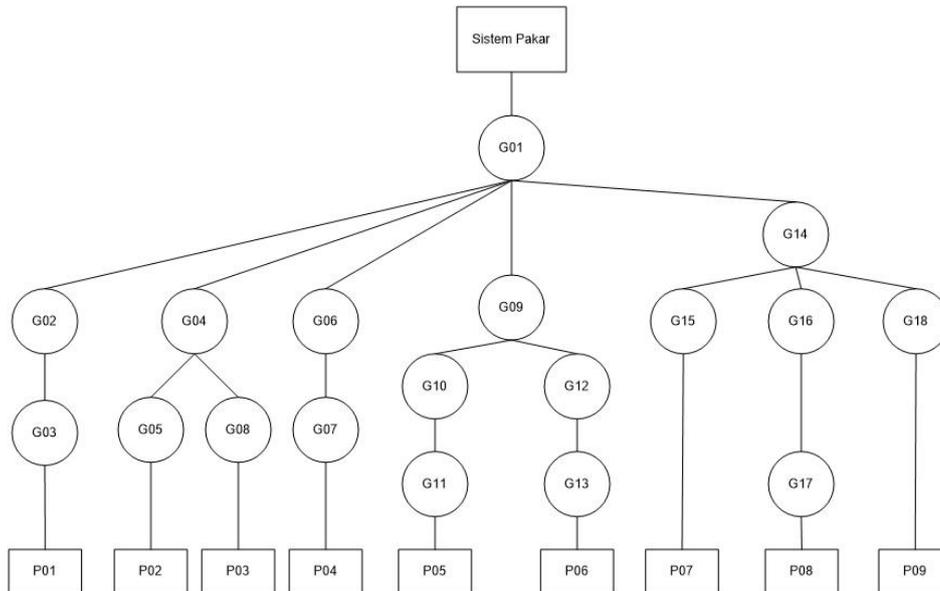
Kode Gejala Penyakit	Gejala Penyakit
<b>G17</b>	Mata mengeluarkan cairan kental yang membentuk kerak pada malam hari
<b>G18</b>	Mata dapat mengeluarkan cairan kental yang membentuk kerak pada malam hari, sehingga menyulitkan kamu membuka mata di pagi hari.
<b>G19</b>	Terkadang mata mengeluarkan air
<b>G20</b>	Iritasi dan perih pada mata
<b>G21</b>	Pertumbuhan selaput berwarna putih dengan pembuluh darah yang terlihat pada sudut mata bagian dalam atau luar
<b>G22</b>	Kelopak mata terasa gatal
<b>G23</b>	Kelopak mata menjadi lengket
<b>G24</b>	Tepi kelopak mata terlihat bengkak
<b>G25</b>	Mata yang terlalu berair
<b>G26</b>	Munculnya nanah yang keluar dari sudut mata
<b>G27</b>	Pembengkakan pada saluran air mata dikelopak bagian bawah

**TABEL: 3.5 Tabel Basis Pengetahuan**

ID	Penyakit								
	P01	P02	P03	P04	P05	P06	P07	P08	P09
G01	√								
G02	√								
G03	√								
G04		√							
G05		√							
G06		√							
G07			√						
G08			√						
G09			√						
G10				√					
G11				√					
G12				√					
G13					√				
G14					√				
G15					√				
G16						√			
G17						√			
G18						√			



### 3.3.1 Arsitektur *Forward Chaining* Dengan *Tree*



**GAMBAR: 3.3** Arsitektur *Forward Chaining* Dengan *Tree*

### 3.3.2 Definisi Aktor

Sistem pakar ini digunakan oleh dua pengguna, yaitu pakar dan *user*, dimana kedua bagian tersebut memiliki wewenang dan hak akses yang berbeda.

#### a. Pakar

Pakar disini adalah seseorang dokter spesialis mata yang memiliki keahlian dalam ilmu tentang mata.

#### b. Pengguna

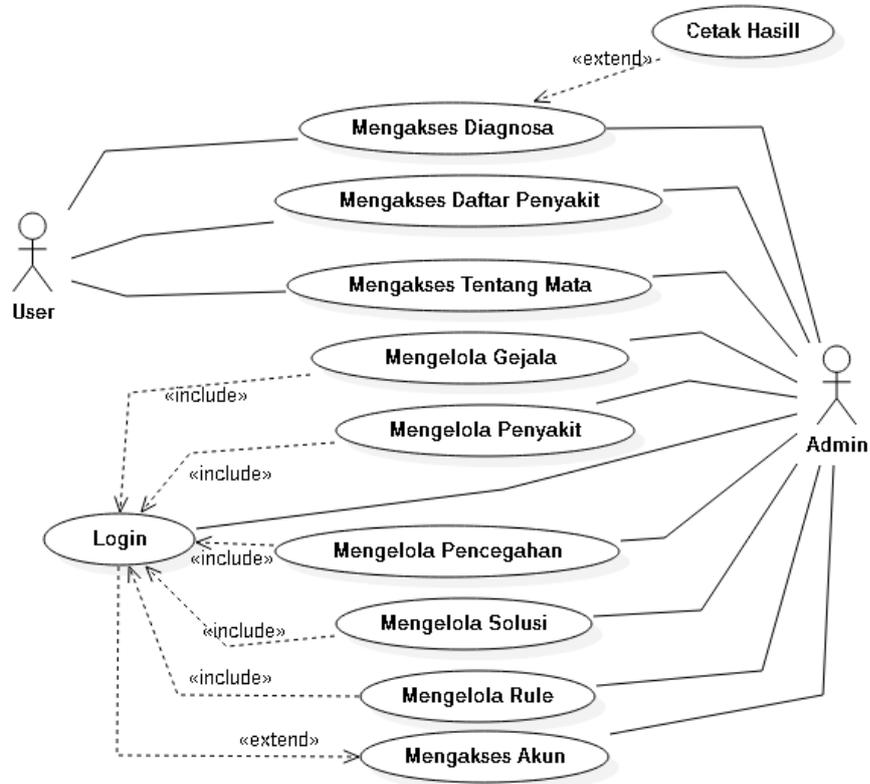
Peserta konsultasi disini adalah seorang pengguna yang akan melakukan konsultasi tentang penyakit mata. Peserta konsultasi memiliki hak akses hanya untuk melakukan konsultasi, dan melihat hasil konsultasi.

### 3.3.3 Perancangan Sistem

Perancangan ini bertujuan untuk memenuhi kebutuhan *user* mengenai gambaran yang jelas tentang perancangan sistem yang akan dirancang serta diimplementasikan. Dalam perancangan sistem yang digunakan penulis yaitu menggunakan *Unified Modeling Language* (UML).

### 3.3.4 Use Case Diagram

*Use Case Diagram* merupakan diagram yang menggambarkan hubungan antara actor dengan sistem. *Use Case* diagram bisa mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. *Use Case* diagram juga bisa digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem. Komponen tersebut kemudian menjelaskan komunikasi antara aktor, dengan sistem yang ada. Dengan demikian, *use case* dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen. Manfaat dari *use case* sendiri adalah untuk memudahkan komunikasi dengan menggunakan domain *expert* dan jagan *end user*, memberikan kepastian pemahaman yang pas tentang requirement atau juga kebutuhan sebuah sistem.



**GAMBAR: 3.4 Use Case Diagram Sistem Aplikasi**

Berdasarkan gambar diatas, maka dibuatlah deskripsi dengan skenario dari setiap proses yang ada pada *use case* diagram tersebut ke dalam tabel dibawah ini.

Interaksi *user* dengan *use case* mengakses diagnosa dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.6 Skenario Use Case Mengakses Diagnosa**

<b>Identifikasi</b>	
<b>Nama</b>	Mengakses Diagnosa dan Cetak Hasil
<b>Tujuan</b>	Agar user dapat mengakses menu diagnosa
<b>Deskripsi</b>	Menu ini akan menampilkan isi dari menu diagnosa
<b>Aktor</b>	User
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
User memilih menu diagnosa	Sistem akan menampilkan halaman dari menu diagnose
User memilih gejala penyakit dan klik cek diagnosa	Menampilkan hasil diagnosa dari beberapa gejala yang sudah dipilih
<b>Kondisi Akhir</b>	User dapat memilih gejala dari menu diagnosa

Interaksi *user* dengan *use case* mengakses tentang mata dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.7 Skenario Use Case Mengakses Tentang Mata**

<b>Identifikasi</b>	
<b>Nama</b>	Mengakses Tentang Mata
<b>Tujuan</b>	Agar user dapat mengakses menu tentang mata dan melihat isi dari menu tersebut
<b>Deskripsi</b>	Menu ini berisi tentang penjelasan struktur keseluruhan dari mata
<b>Aktor</b>	User
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
User memilih menu tentang mata	Sistem akan menampilkan halaman dari menu tentang mata
<b>Kondisi Akhir</b>	User dapat membaca struktur keseluruhan dari mata

Interaksi *user* dengan *use case* mengakses daftar penyakit dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.8 Skenario *Use Case* Mengakses Daftar Penyakit**

<b>Identifikasi</b>	
<b>Nama</b>	Mengakses Daftar Penyakit
<b>Tujuan</b>	Agar user dapat mengakses menu daftar penyakit
<b>Deskripsi</b>	Menu ini akan menampilkan penyakit-penyakit yang terkait dengan mata
<b>Aktor</b>	User
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
User memilih menu daftar penyakit	Sistem akan menampilkan halaman dari menu daftar penyakit
<b>Kondisi Akhir</b>	User dapat melihat dan mengetahui tentang penyakit-penyakit yang berhubungan dengan mata.

Interaksi admin dengan *use case Login* dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.9 Skenario *Use Case Login***

<b>Identifikasi</b>	
<b>Nama</b>	Login
<b>Tujuan</b>	Memastikan admin agar bisa mengakses secara keseluruhan agar dapat menambahkan, mengedit, dan menghapus data-data
<b>Deskripsi</b>	Menu ini diperuntukan untuk admin agar dapat mengelola aplikasi
<b>Aktor</b>	Admin
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih menu akun	Sistem akan menampilkan halaman keseluruhan dari aplikasi
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih kembali menu akun	Admin dapat memasukkan kembali id dan password agar sesuai dengan data pada aplikasi
<b>Kondisi Akhir</b>	Admin dapat melihat dan memilih menu yang ada pada aplikasi

Interaksi antara admin dengan *use case* mengelola gejala dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.10 Skenario *Use Case* Mengelola Gejala**

<b>Identifikasi</b>	
<b>Nama</b>	Mengelola Gejala
<b>Tujuan</b>	Agar admin dapat menambahkan, merubah, dan menghapus gejala pada database
<b>Deskripsi</b>	Menu ini dapat menambahkan, merubah, dan menghapus gejala
<b>Aktor</b>	Admin
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih menu gejala setelah <i>login</i>	Sistem akan menampilkan halaman dari menu gejala
<b>Kondisi Akhir</b>	Admin dapat merubah, menghapus, atau menambahkan gejala baru

Interaksi antara admin dengan *use case* mengelola penyakit dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.11 Skenario *Use Case* Mengelola Penyakit**

<b>Identifikasi</b>	
<b>Nama</b>	Mengelola Penyakit
<b>Tujuan</b>	Agar admin dapat menambahkan, merubah, dan menghapus penyakit pada database
<b>Deskripsi</b>	Menu ini akan menambahkan, merubah, dan menghapus penyakit
<b>Aktor</b>	Admin
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih menu penyakit setelah <i>login</i>	Sistem akan menampilkan halaman dari menu penyakit
<b>Kondisi Akhir</b>	Admin dapat merubah, menghapus, atau menambahkan penyakit baru

Interaksi antara admin dengan *use case* mengelola pencegahan dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.12 Skenario Use Case Mengelola Pencegahan**

<b>Identifikasi</b>	
<b>Nama</b>	Mengelola Pencegahan
<b>Tujuan</b>	Agar admin dapat menambahkan, merubah, dan menghapus pencegahan pada database
<b>Deskripsi</b>	Menu ini akan menambahkan, merubah, dan menghapus pencegahan
<b>Aktor</b>	Admin
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih menu pencegahan setelah <i>login</i>	Sistem akan menampilkan halaman dari menu pencegahan
<b>Kondisi Akhir</b>	Admin dapat merubah, menghapus atau menambahkan pencegahan baru

Interaksi antara admin dengan *use case* mengelola rule dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.13 Skenario *Use Case* Mengelola Rule**

<b>Identifikasi</b>	
<b>Nama</b>	Mengelola Rule
<b>Tujuan</b>	Agar admin dapat menambahkan, merubah, dan menghapus rule pada database
<b>Deskripsi</b>	Menu ini dapat menambahkan, merubah, menghapus rule
<b>Aktor</b>	Admin
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih menu rule setelah <i>login</i>	Sistem akan menampilkan halaman dari menu rule
<b>Kondisi Akhir</b>	Admin dapat merubah, menghapus, atau menambahkan rule baru

Interaksi antara admin dengan *use case* mengelola solusi dijelaskan dalam *use case* skenario sebagai berikut:

**TABEL: 3.14 Skenario Use Case Mengelola Solusi**

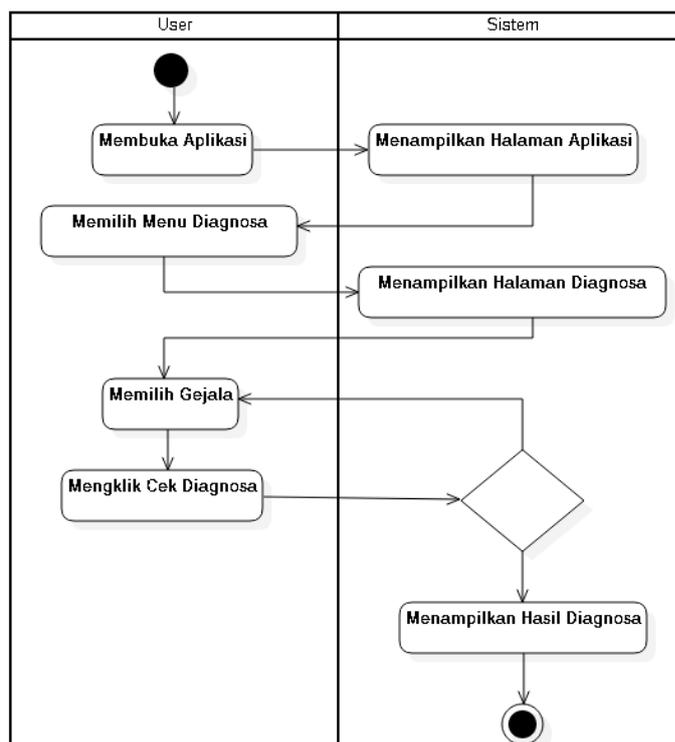
<b>Identifikasi</b>	
<b>Nama</b>	Mengelola Solusi
<b>Tujuan</b>	Agar admin dapat menambahkan, merubah, dan menghapus solusi pada database
<b>Deskripsi</b>	Menu ini dapat menambahkan, merubah, dan menghapus solusi
<b>Aktor</b>	Admin
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
Admin memilih menu solusi setelah <i>login</i>	Sistem akan menampilkan halaman dari menu solusi
<b>Kondisi Akhir</b>	Admin dapat merubah, menghapus, atau menambahkan solusi baru

### 3.3.5 Activity Diagram

Activity diagram merupakan diagram yang memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *use case* diagram yang ada.

#### 1. Activity Diagram Mengakses Diagnosa

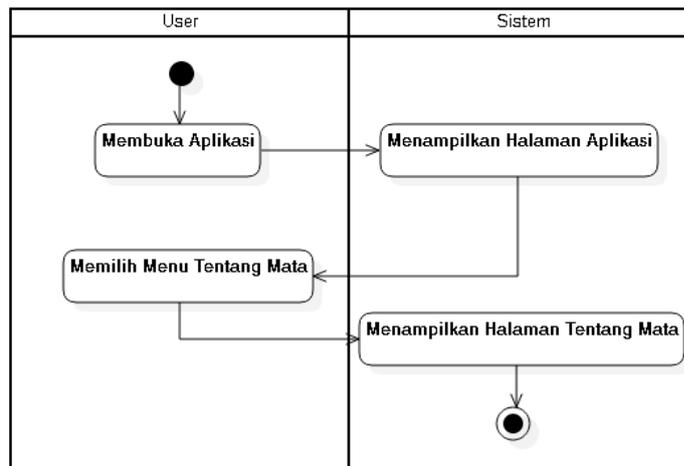
Activity diagram pada gambar 3.5 menggambarkan proses user dalam mengakses menu diagnosa.



**GAMBAR 3.5 Activity Diagram Mengakses Diagnosa**

## 2. Activity Diagram Mengakses Tentang Mata

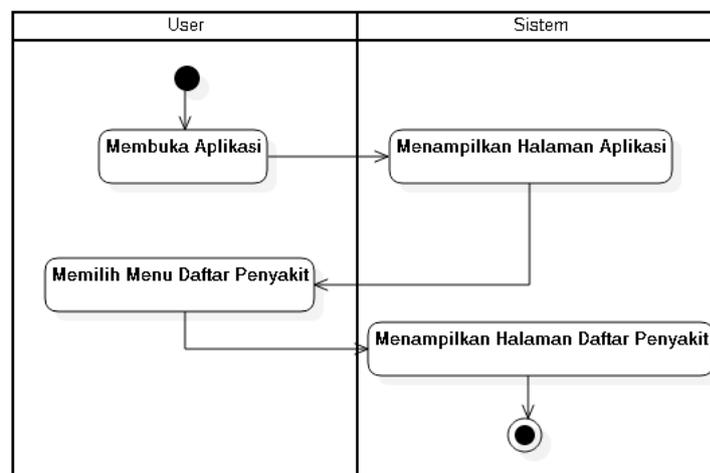
Activity diagram pada gambar 3.6 menggambarkan proses user dalam mengakses menu tentang mata.



**GAMBAR: 3.6 Activity Diagram Mengakses Tentang Mata**

## 3. Activity Diagram Mengakses Daftar Penyakit

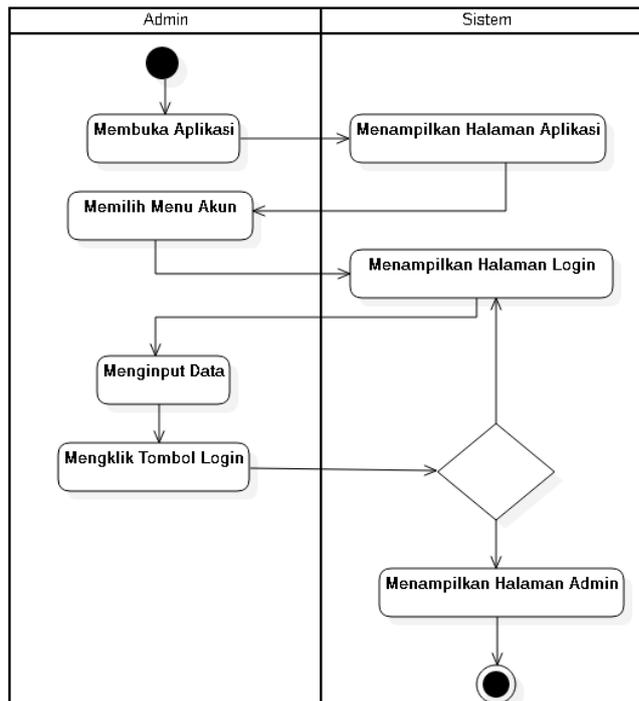
Activity diagram pada gambar 3.7 menggambarkan proses user dalam mengakses menu daftar penyakit.



**GAMBAR: 3.7 Activity Diagram Mengakses Daftar Penyakit**

#### 4. Activity Diagram Login

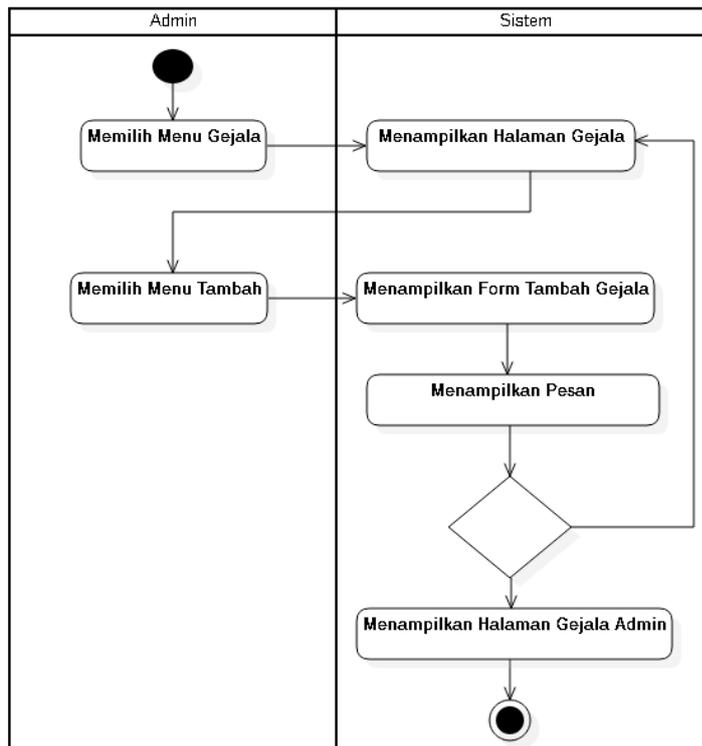
Activity diagram pada gambar 3.8 menggambarkan admin dalam melakukan proses login



**GAMBAR: 3.8 Activity Diagram Login**

### 5. Activity Diagram Menambahkan Gejala

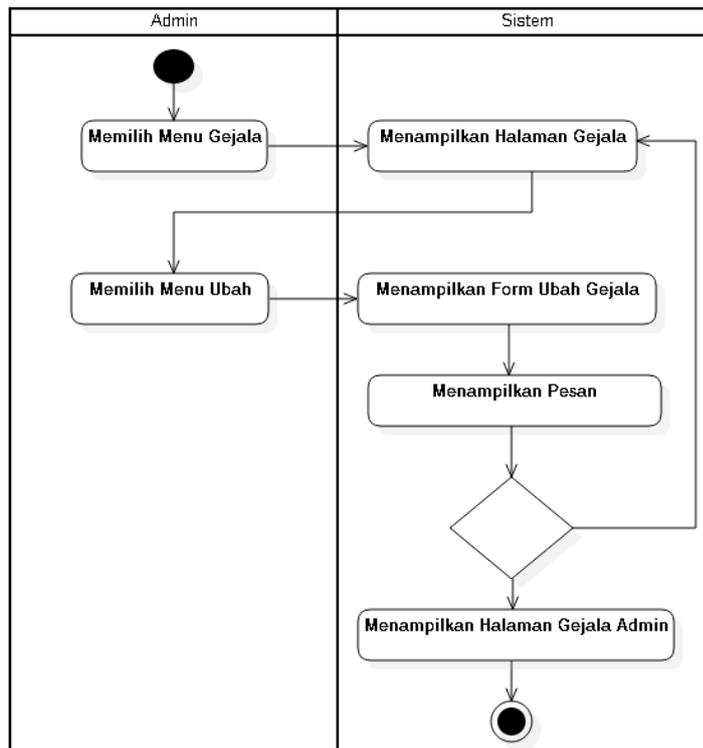
Activity diagram pada gambar 3.9 menggambarkan admin dalam melakukan proses menambahkan gejala



**GAMBAR: 3.9 Activity Diagram Menambahkan Gejala**

## 6. Activity Diagram Mengubah Gejala

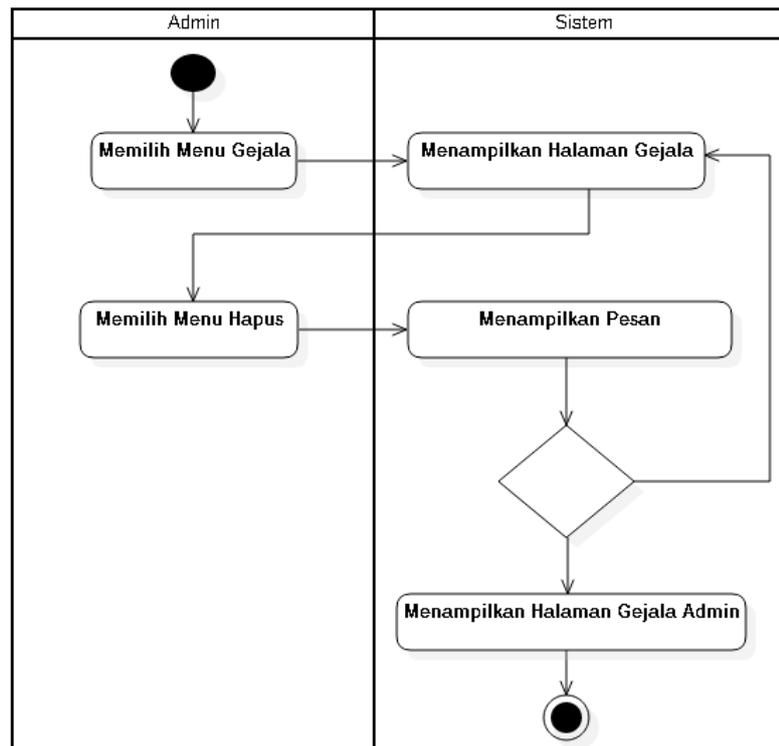
Activity diagram pada gambar 3.10 menggambarkan admin dalam melakukan proses mengubah gejala



**GAMBAR: 3.10** Activity Diagram Mengubah Gejala

### 7. Activity Diagram Menghapus Gejala

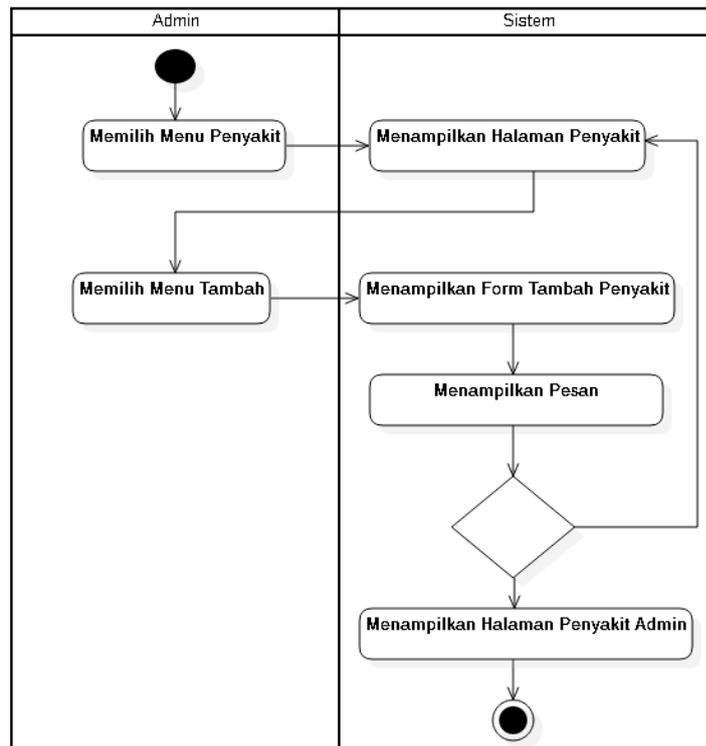
Activity diagram pada gambar 3.11 menggambarkan admin dalam melakukan proses menghapus gejala



**GAMBAR: 3.11** Activity Diagram Menghapus Gejala

### 8. Activity Diagram Menambahkan Penyakit

Activity diagram pada gambar 3.12 menggambarkan admin dalam melakukan proses menambahkan penyakit



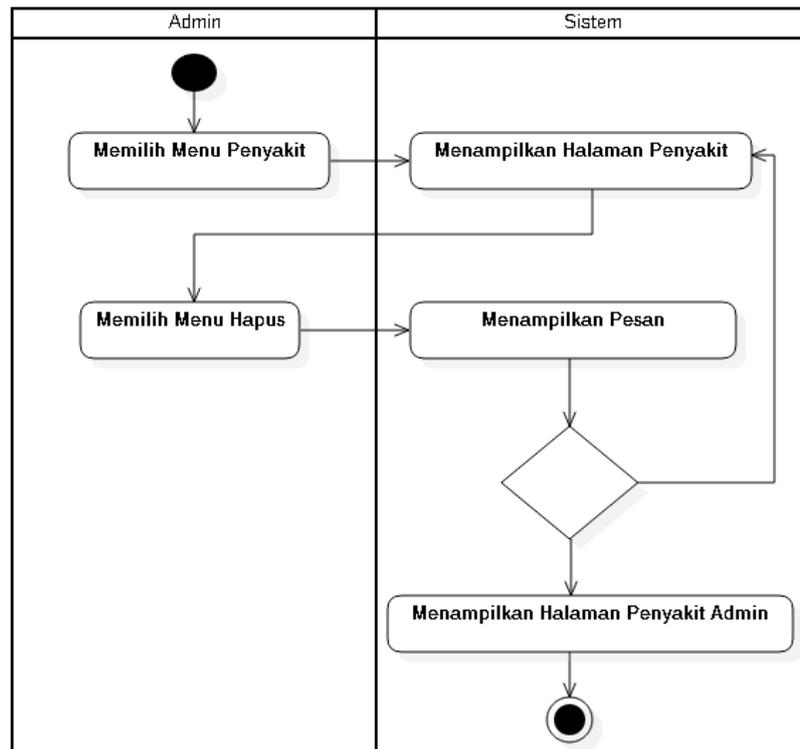
**GAMBAR: 3.12 Activity Diagram Menambahkan Penyakit**

### 9. Activity Diagram Mengubah Penyakit

Activity diagram pada gambar 3.13 menggambarkan admin dalam melakukan proses mengubah penyakit

## 10. Activity Diagram Menghapus Penyakit

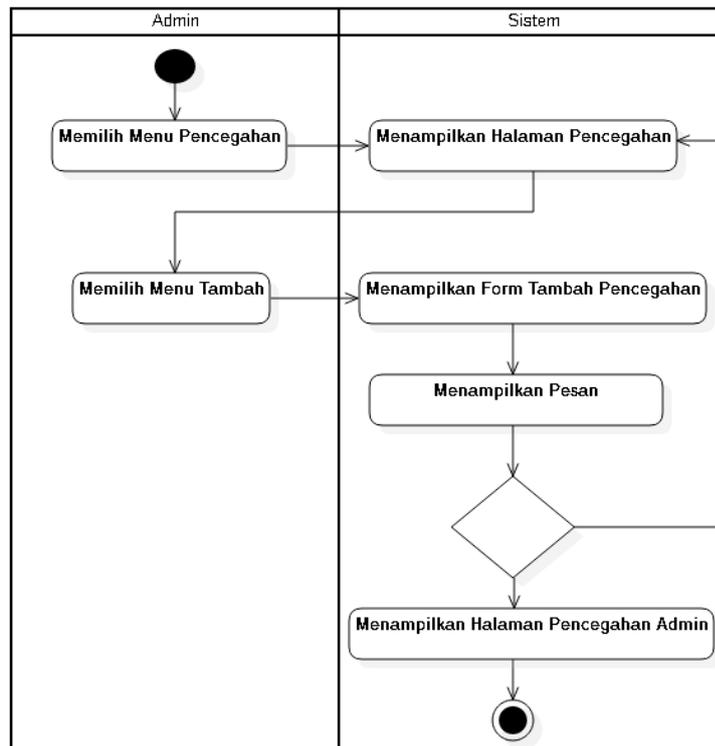
Activity diagram pada gambar 3.14 menggambarkan admin dalam melakukan proses menghapus penyakit



**GAMBAR: 3.14** Activity Diagram Menghapus Penyakit

### 11. Activity Diagram Menambahkan Pencegahan

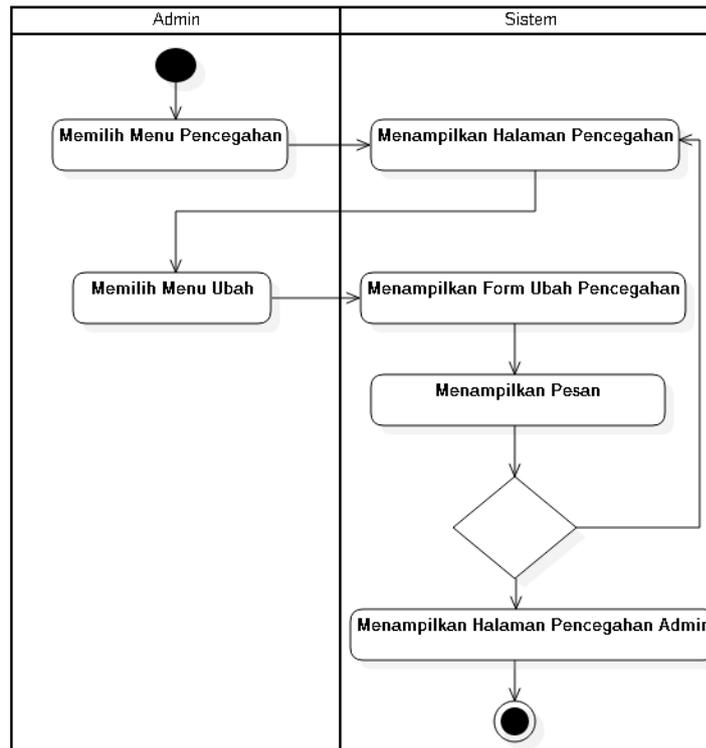
Activity diagram pada gambar 3.15 menggambarkan admin dalam melakukan proses menambahkan pencegahan



**GAMBAR: 3.15 Activity Diagram Menambahkan Pencegahan**

## 12. Activity Diagram Mengubah Pencegahan

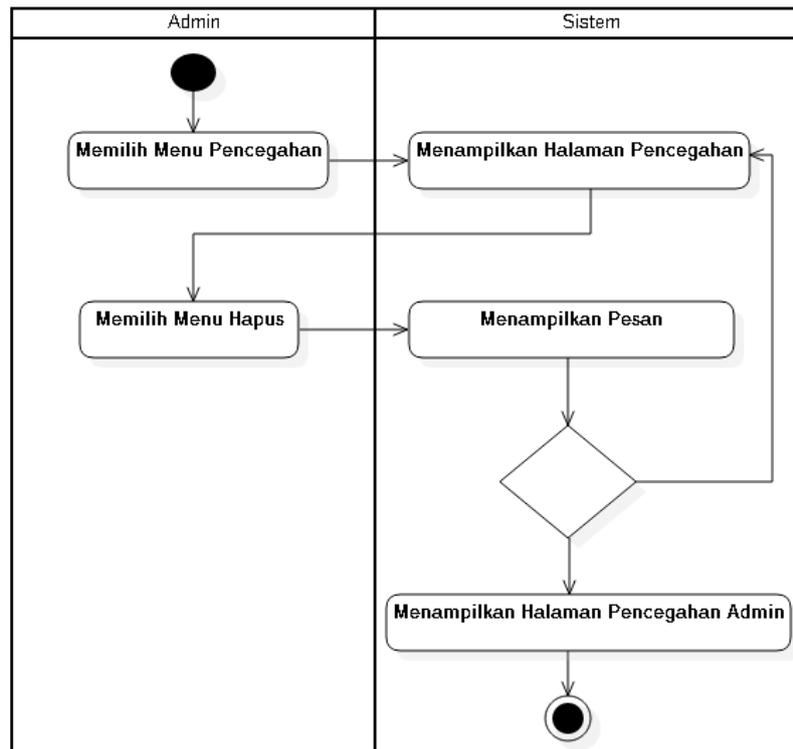
Activity diagram pada gambar 3.16 menggambarkan admin dalam melakukan proses mengubah pencegahan



**GAMBAR: 3.16** Activity Diagram Mengubah Pencegahan

### 13. Activity Diagram Menghapus Pencegahan

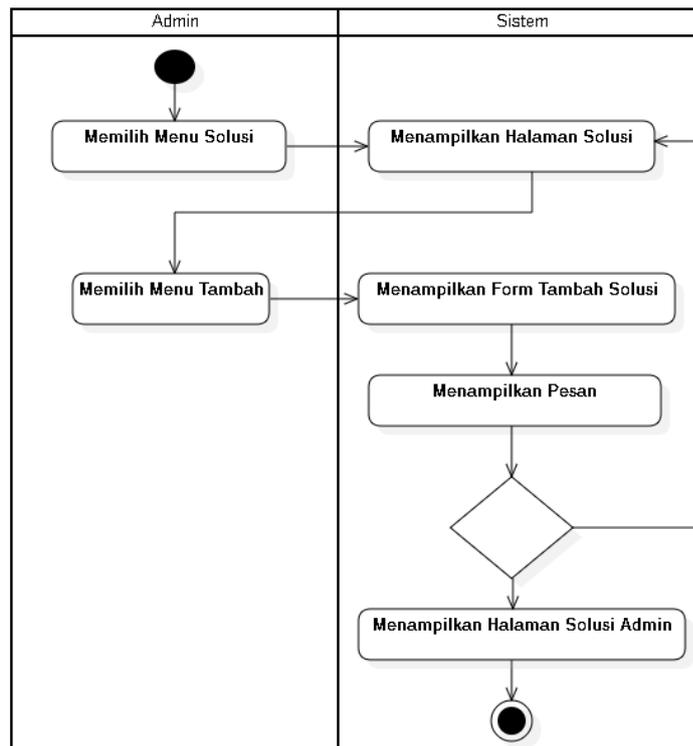
Activity diagram pada gambar 3.17 menggambarkan admin dalam melakukan proses menghapus pencegahan



**GAMBAR: 3.17** Activity Diagram Menghapus Pencegahan

#### 14. Activity Diagram Menambahkan Solusi

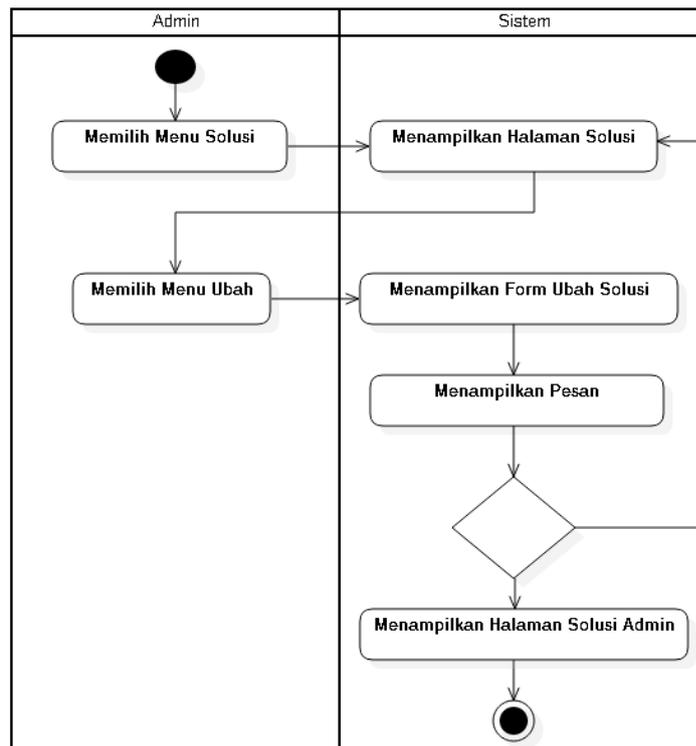
Activity diagram pada gambar 3.18 menggambarkan admin dalam melakukan proses menambahkan solusi



**GAMBAR: 3.18** Activity Diagram Menambahkan Solusi

### 15. Activity Diagram Mengubah Solusi

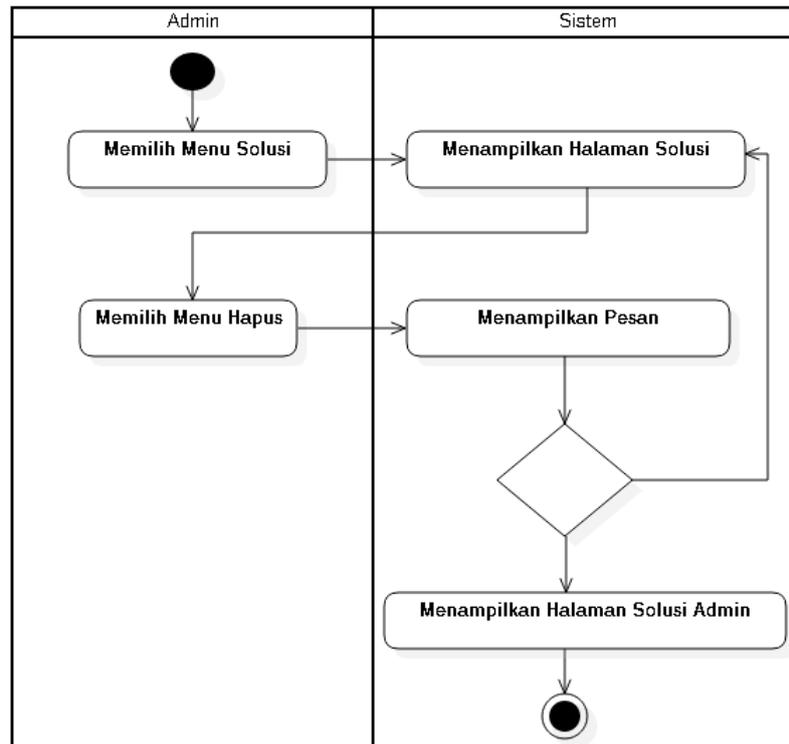
Activity diagram pada gambar 3.19 menggambarkan admin dalam melakukan proses mengubah solusi



**GAMBAR: 3.19** Activity Diagram Mengubah Solusi

## 16. Activity Diagram Menghapus Solusi

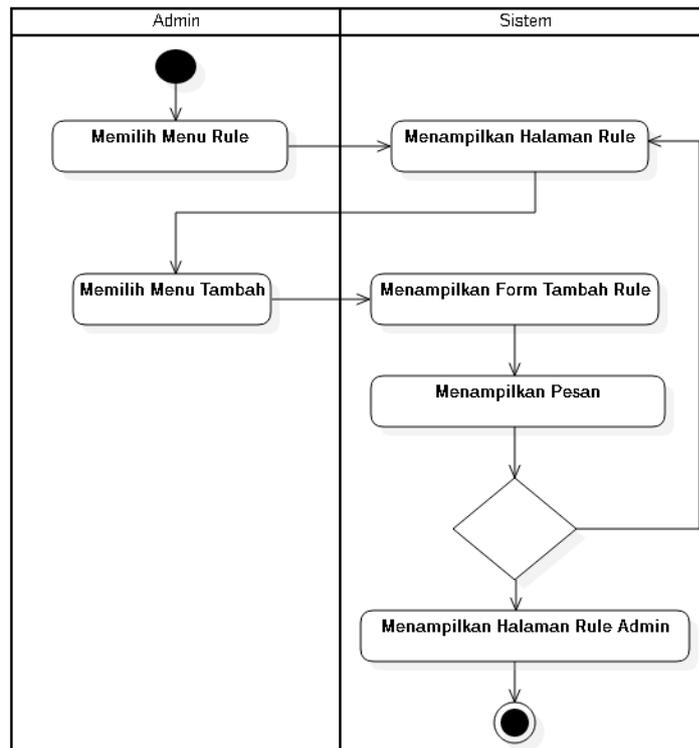
Activity diagram pada gambar 3.20 menggambarkan admin dalam melakukan proses menghapus solusi



**GAMBAR: 3.20** Activity Diagram Menghapus Solusi

### 17. Activity Diagram Menambahkan Rule

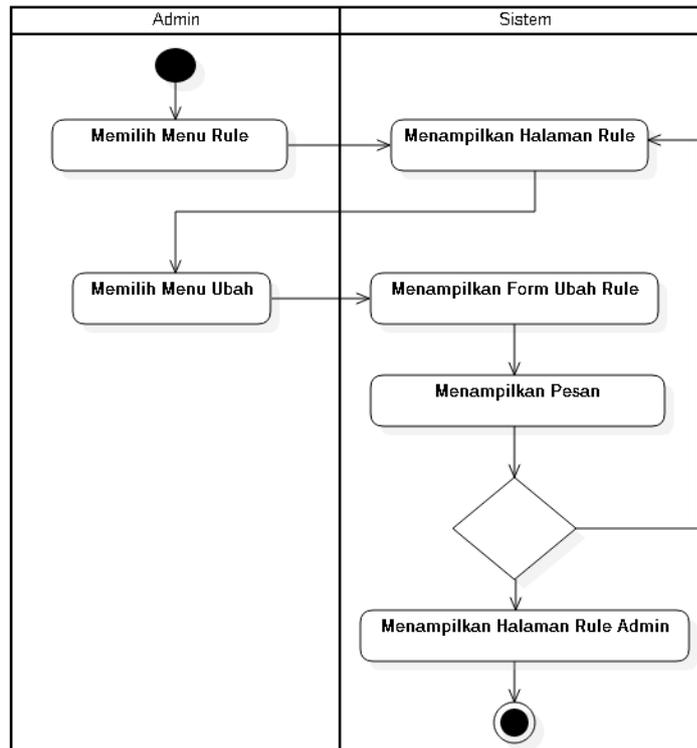
Activity diagram pada gambar 3.21 menggambarkan admin dalam melakukan proses menambahkan rule



**GAMBAR: 3.21 Activity Diagram Menambahkan Rule**

### 18. Activity Diagram Mengubah Rule

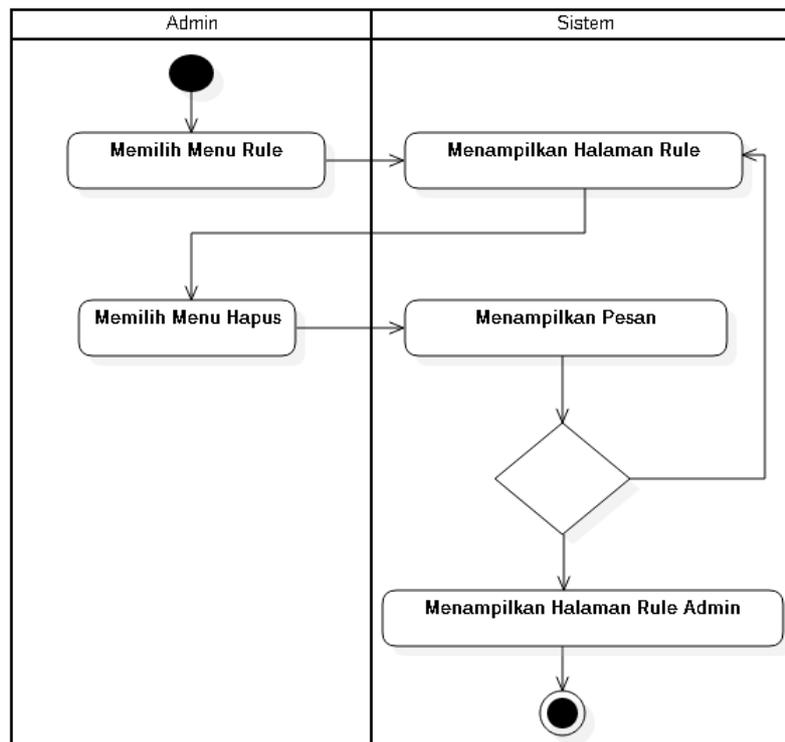
Activity diagram pada gambar 3.22 menggambarkan admin dalam melakukan proses mengubah rule



**GAMBAR: 3.22 Activity Diagram Mengubah Rule**

### 19. Activity Diagram Menghapus Rule

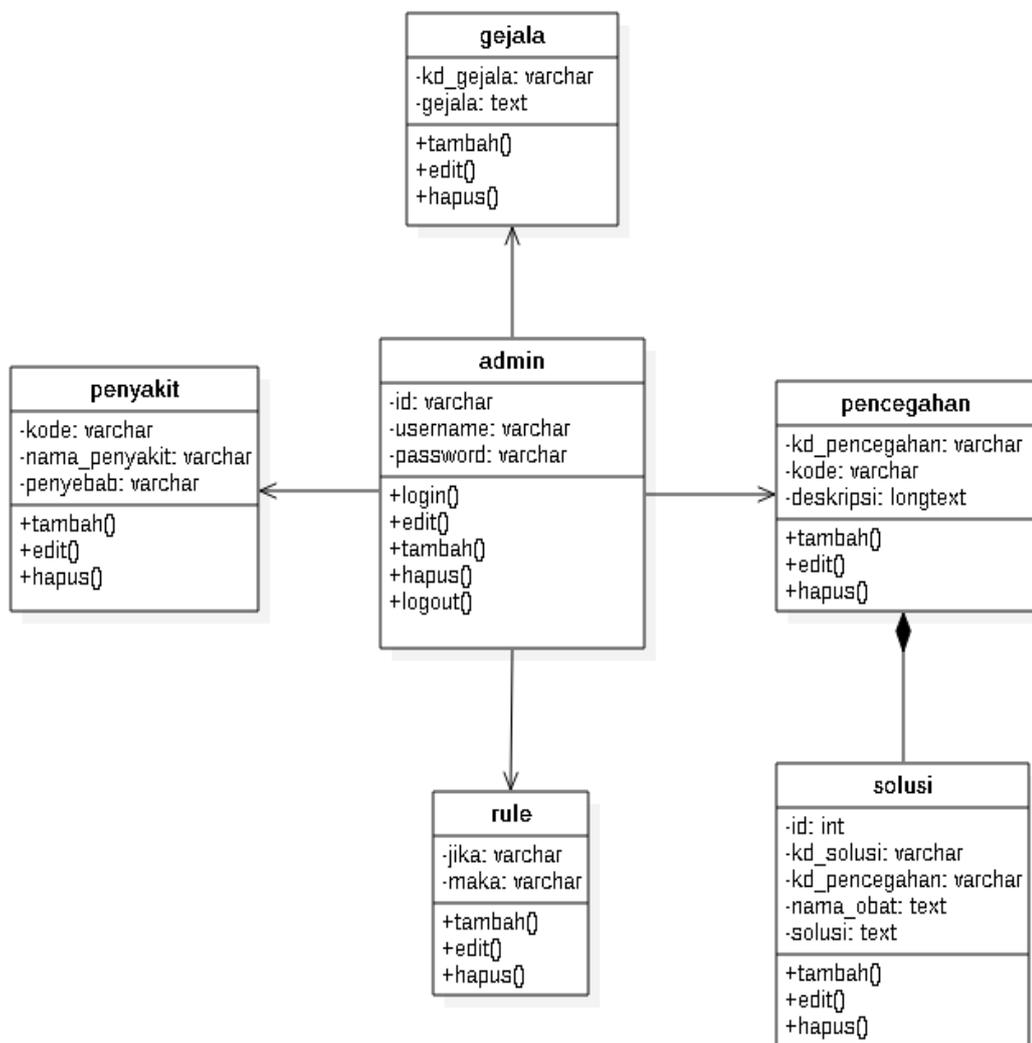
Activity diagram pada gambar 3.23 menggambarkan admin dalam melakukan proses menghapus rule



**GAMBAR: 3.23** Activity Diagram Menghapus Rule

### 3.3.6 Class Diagram

*Class* diagram digunakan untuk menampilkan beberapa kelas yang ada didalam perangkat lunak yang akan dikembangkan. *Class* diagram menggambarkan struktur dan deskripsi beserta hubungan kelas yang satu dengan yang lainnya.



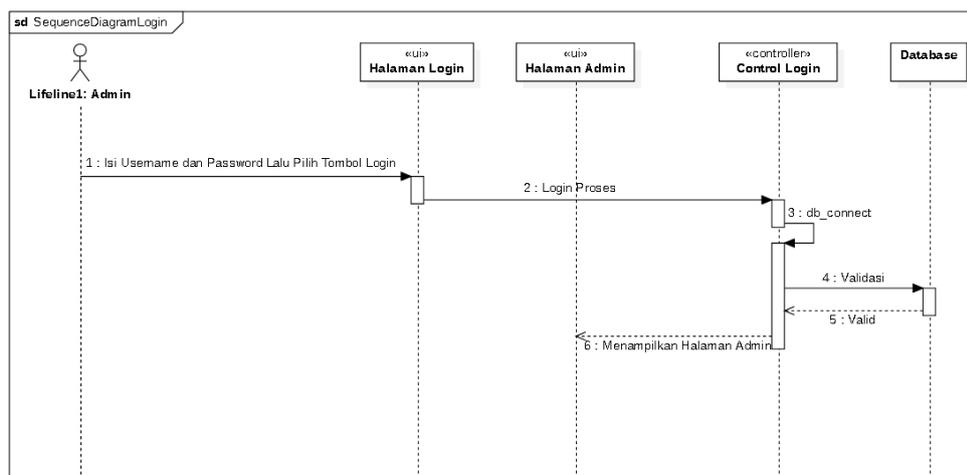
**GAMBAR: 3.24 Class Diagram**

### 3.3.7 Sequence Diagram

Menggambarkan secara grafis bagaimana objek-objek berinteraksi satu sama lain melalui proses pesan selama eksekusi suatu operasi. Mengilustrasikan bagaimana pesan dikirim dan diterima oleh objek dan bagaimana urutannya. Perancangan *sequence* akan dibagi menurut admin dan *user*. Berikut *sequence* diagram yang akan berjalan pada aplikasi.

#### 1. Sequence Diagram Login

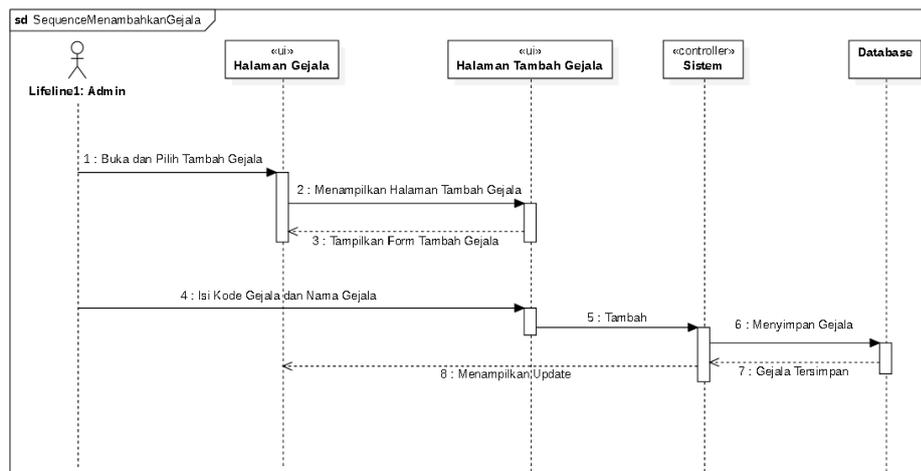
Pada gambar 3.25 menggambarkan *sequence* diagram dari login



**GAMBAR: 3.25 Sequence Diagram Login**

2. Sequence Diagram Menambahkan Gejala

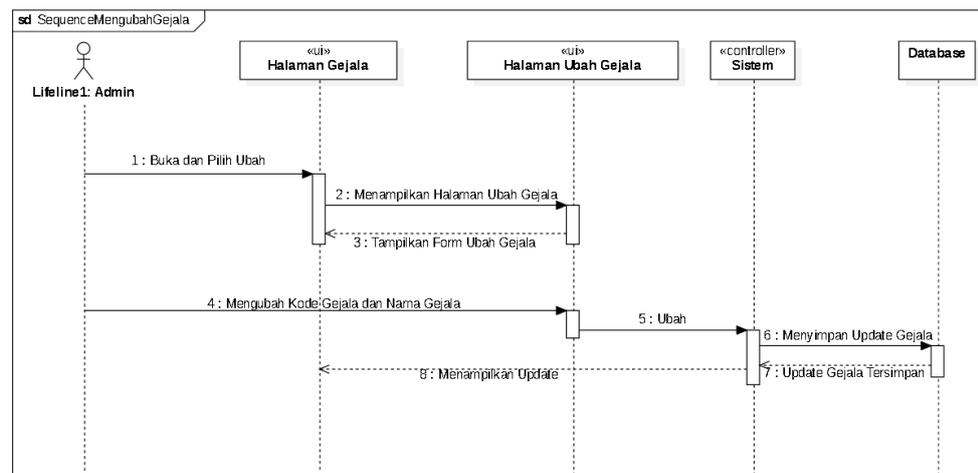
Pada gambar 3.26 menggambarkan *sequence diagram* proses mengelola gejala



**GAMBAR: 3.26** Sequence Diagram Menambahkan Gejala

3. Sequence Diagram Mengubah Gejala

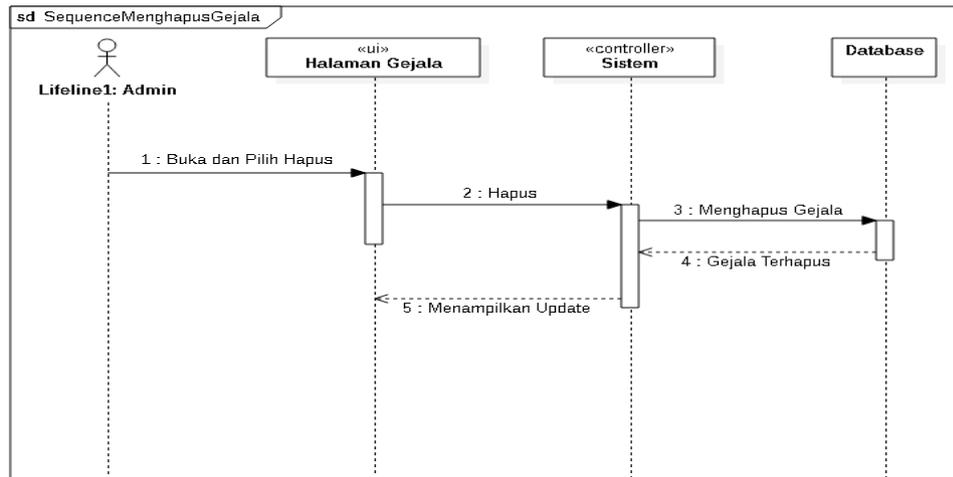
Pada gambar 3.27 menggambarkan *sequence diagram* proses mengubah gejala



**GAMBAR: 3.27** Sequence Diagram Mengubah Gejala

#### 4. Sequence Diagram Menghapus Gejala

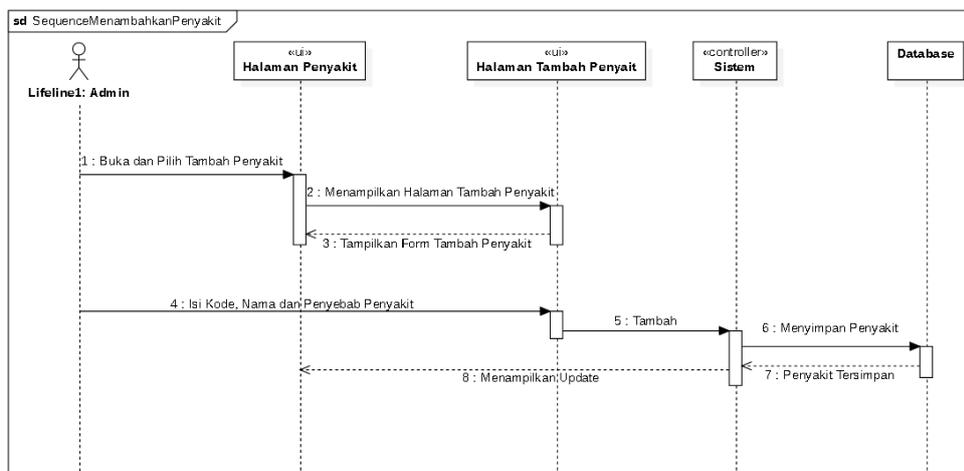
Pada gambar 3.28 menggambarkan *sequence* diagram proses menghapus gejala



**GAMBAR: 3.28** Sequence Diagram Menghapus Gejala

#### 5. Sequence Diagram Menambahkan Penyakit

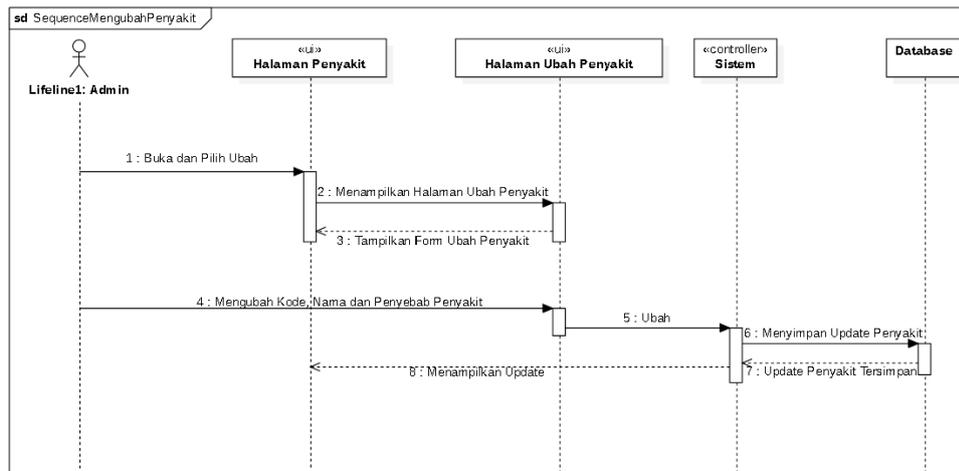
Pada gambar 3.29 menggambarkan *sequence* diagram proses menambahkan penyakit.



**GAMBAR: 3.29** Sequence Diagram Menambahkan Penyakit

## 6. Sequence Diagram Mengubah Penyakit

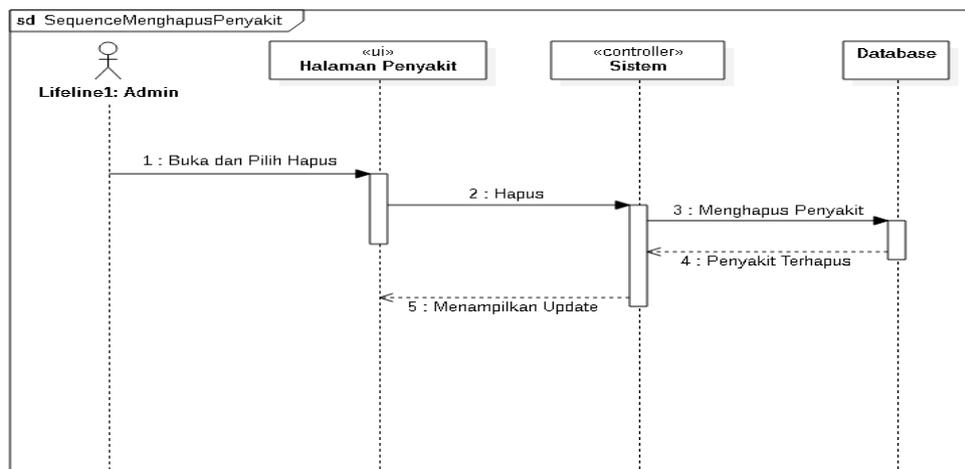
Pada gambar 3.30 menggambarkan *sequence diagram* proses mengubah penyakit



**GAMBAR: 3.30 Sequence Diagram Mengubah Penyakit**

## 7. Sequence Diagram Menghapus Penyakit

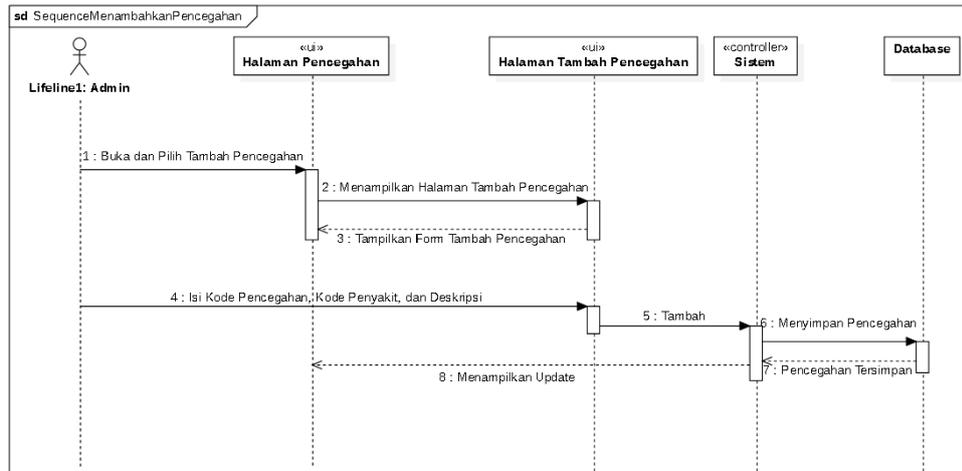
Pada gambar 3.31 menggambarkan *sequence diagram* proses menghapus penyakit



**GAMBAR: 3.31 Sequence Diagram Menghapus Penyakit**

8. Sequence Diagram Menambahkan Pencegahan

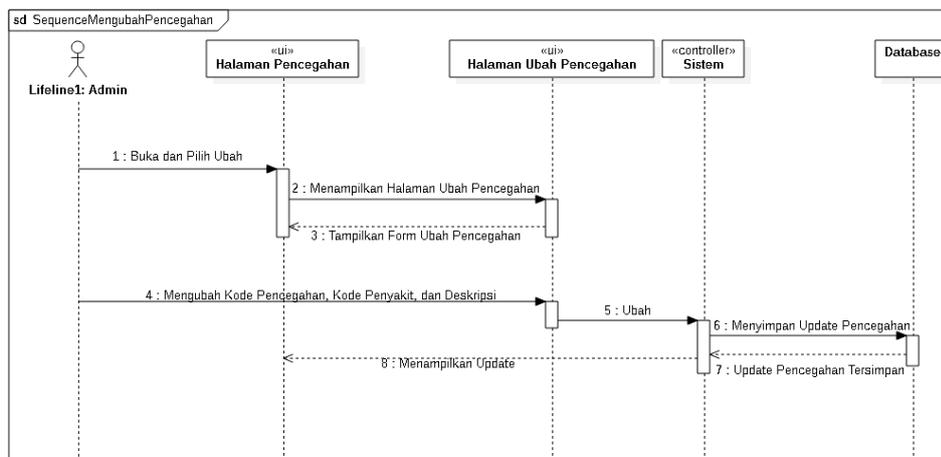
Pada gambar 3.32 menggambarkan *sequence* diagram proses menambahkan pencegahan



**GAMBAR: 3.32** Sequence Diagram Menambahkan Pencegahan

9. Sequence Diagram Mengubah Pencegahan

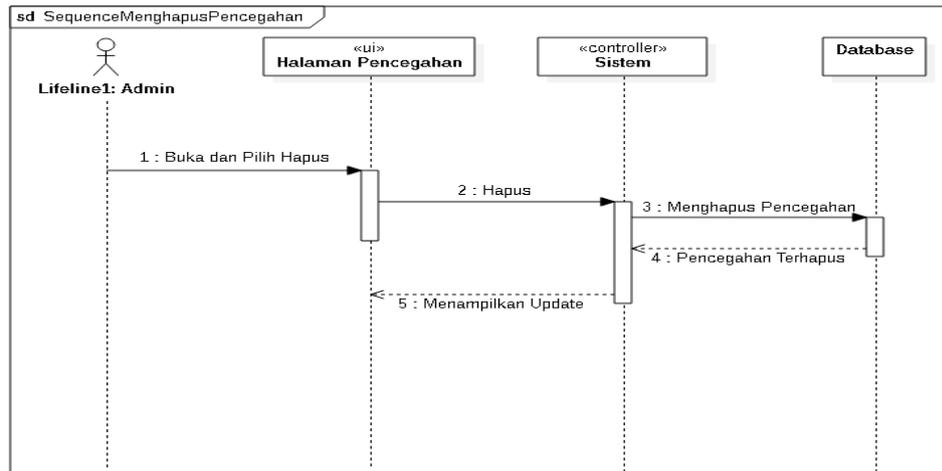
Pada gambar 3.33 menggambarkan *sequence* diagram proses mengubah pencegahan



**GAMBAR: 3.33** Sequence Diagram Mengubah Pencegahan

## 10. Sequence Diagram Menghapus Pencegahan

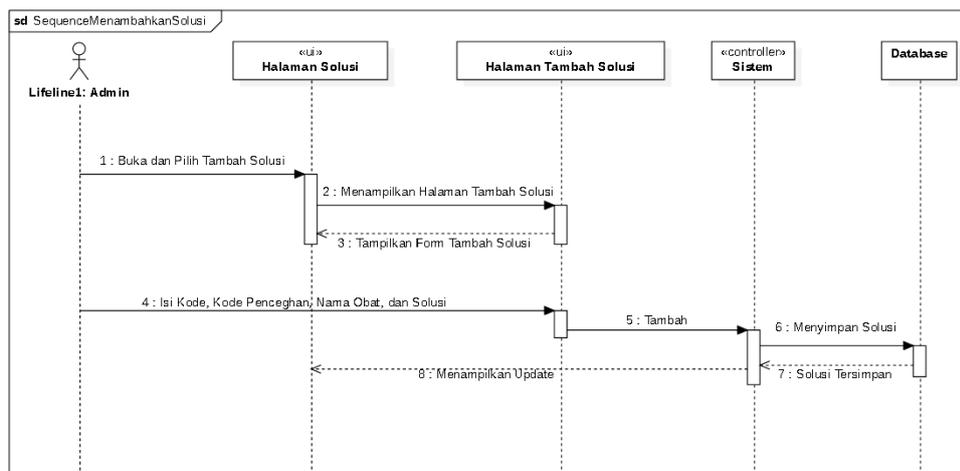
Pada gambar 3.34 menggambarkan *sequence* diagram proses menghapus pencegahan



**GAMBAR: 3.34** Sequence Diagram Menghapus Pencegahan

## 11. Sequence Diagram Menambahkan Solusi

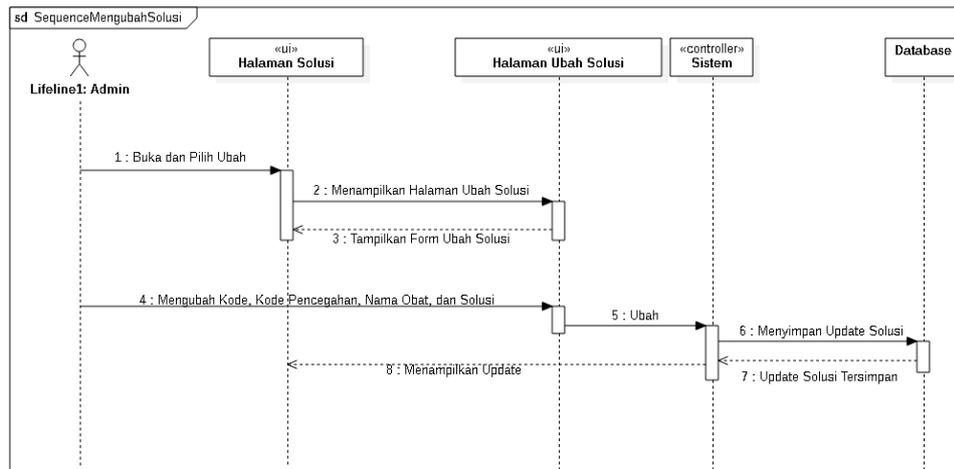
Pada gambar 3.35 menggambarkan *sequence* diagram proses menambahkan solusi



**GAMBAR: 3.35** Sequence Diagram Menambahkan Solusi

## 12. Sequence Diagram Mengubah Solusi

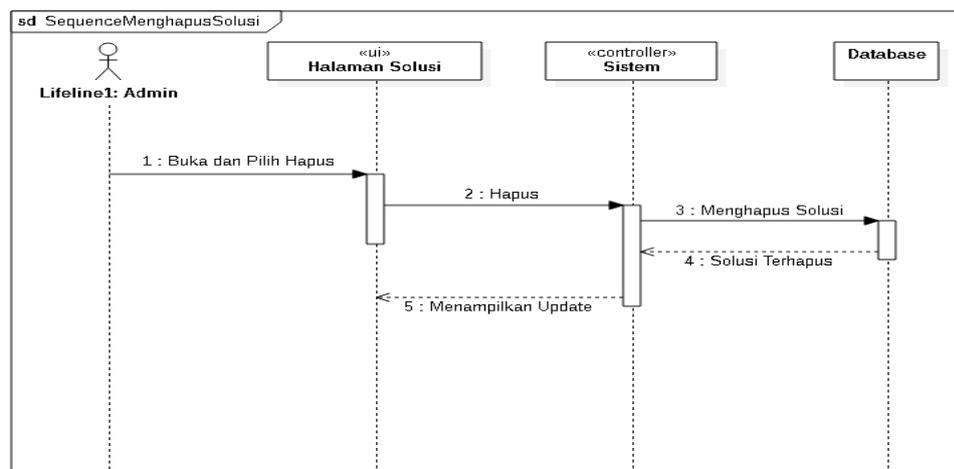
Pada gambar 3.36 menggambarkan *sequence* diagram proses mengubah solusi



**GAMBAR: 3.36** *Sequence* Diagram Mengubah Solusi

## 13. Sequence Diagram Menghapus Solusi

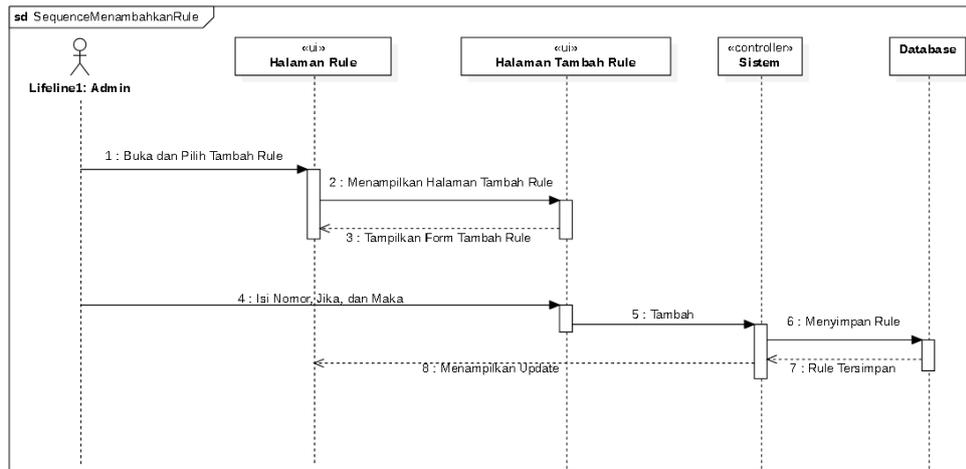
Pada gambar 3.37 menggambarkan *sequence* diagram proses menghapus solusi



**GAMBAR: 3.37** *Sequence* Diagram Menghapus Solusi

14. *Sequence Diagram Menambahkan Rule*

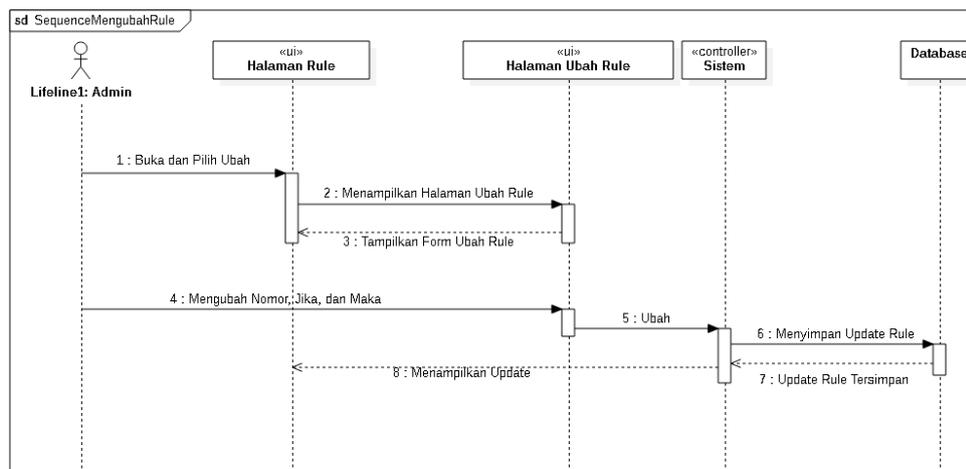
Pada gambar 3.38 menggambarkan *sequence diagram* proses menambahkan rule



**GAMBAR: 3.38** *Sequence Diagram Menambahkan Rule*

15. *Sequence Diagram Mengubah Rule*

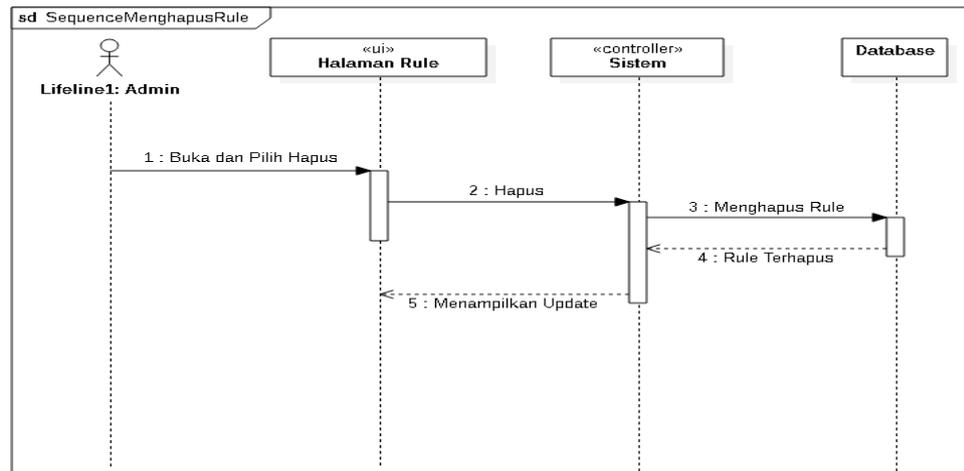
Pada gambar 3.39 menggambarkan *sequence diagram* proses menambahkan rule



**GAMBAR: 3.39** *Sequence Diagram Mengubah Rule*

## 16. Sequence Diagram Menghapus Rule

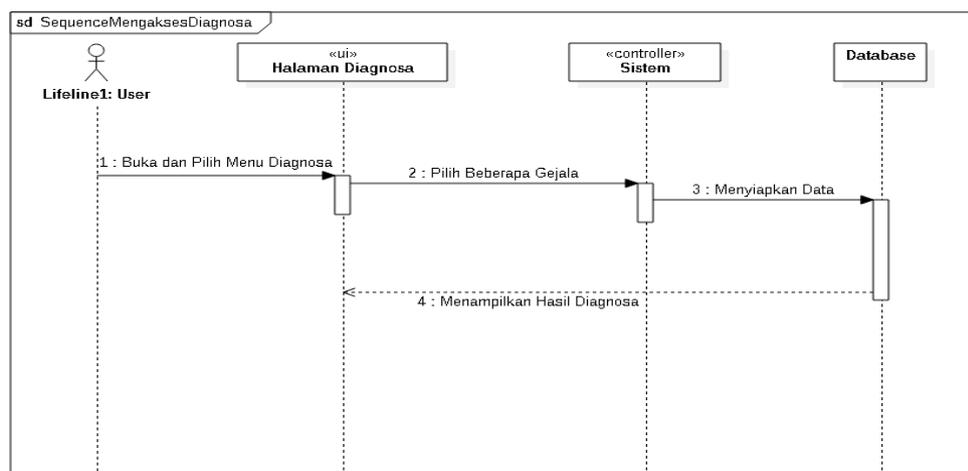
Pada gambar 3.40 menggambarkan *sequence* diagram proses menghapus rule



**GAMBAR: 3.40** Sequence Diagram Menghapus Rule

## 17. Sequence Diagram Mengakses Diagnosa

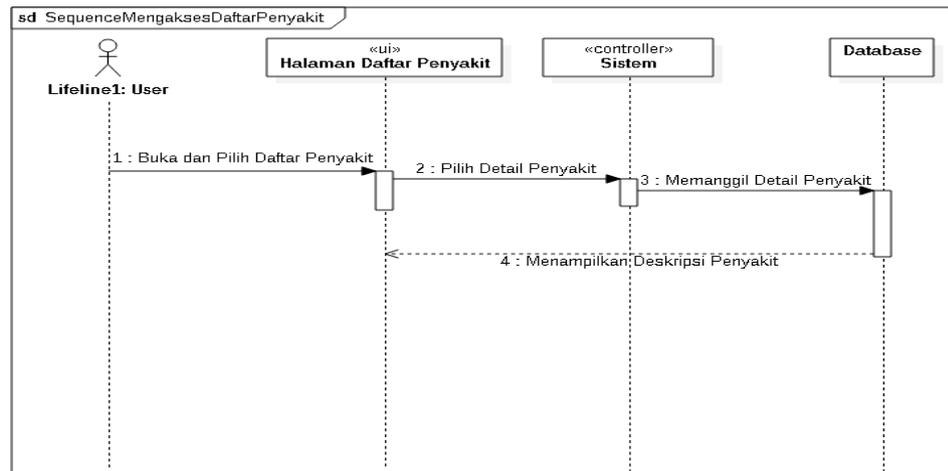
Pada gambar 3.41 menggambarkan *sequence* diagram proses mengakses diagnosa.



**GAMBAR: 3.41** Sequence Diagram Mengakses Diagnosa

### 18. Sequence Diagram Mengakses Daftar Penyakit

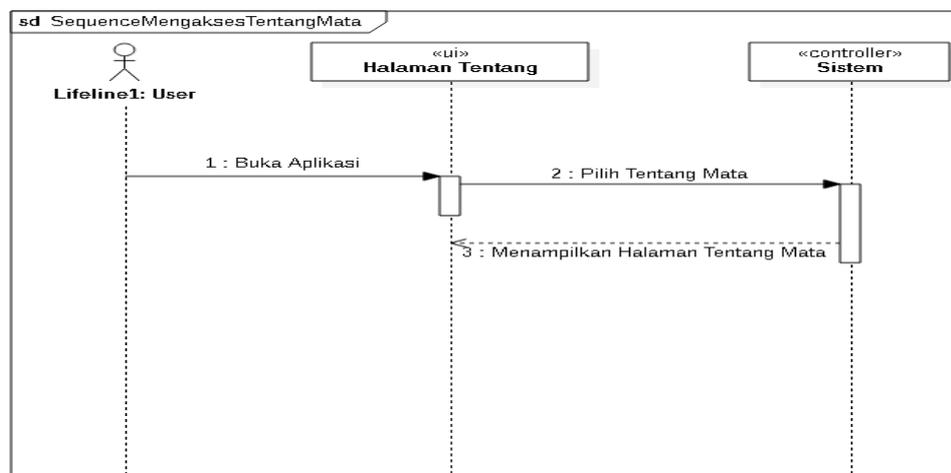
Pada gambar 3.42 menggambarkan *sequence* diagram proses mengakses daftar penyakit



**GAMBAR: 3.42** Sequence Diagram Mengakses Daftar Penyakit

### 19. Sequence Diagram Mengakses Tentang Mata

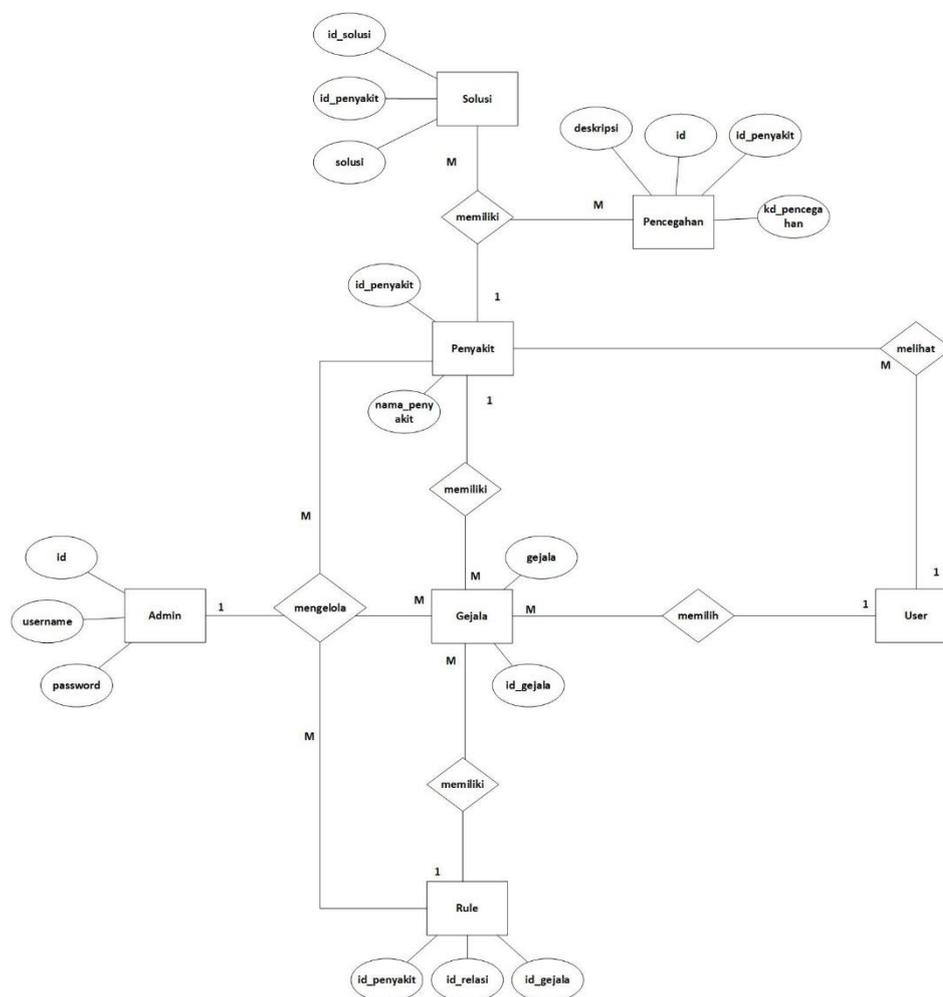
Pada gambar 3.43 menggambarkan *sequence* diagram proses mengakses tentang mata



**GAMBAR 3.43** Sequence Diagram Mengakses Tentang Mata

### 3.3.8 ERD (*Entity Relationship Diagram*)

*Entity Relationship Diagram* (ERD) adalah salah satu model yang digunakan untuk merancang database dengan tujuan menggambarkan data yang berelasi pada sebuah database. Pada gambar 3.44 menggambarkan ERD yang dirancang:



**GAMBAR: 3.44 ERD Sistem Pakar Diagnosa Penyakit Mata**

Terdapat lima tabel dalam aplikasi ini diantaranya tabel gejala, tabel penyakit, tabel pencegahan, tabel solusi, dan tabel rule. Tabel dibawah ini akan menjelaskan isi dari tabel gejala pada sistem yang dirancang:

**TABEL: 3.15 Tabel Gejala**

Nama	Tipe Data	Ukuran	Key
id_gejala	int	25	<i>primary key</i>
Gejala	varchar	100	-

Tabel dibawah ini akan menjelaskan isi dari tabel penyakit pada sistem yang dirancang

**TABEL: 3.16 Tabel Penyakit**

Nama	Tipe Data	Ukuran	Key
id_penyakit	int	25	<i>primary key</i>
Penyakit	varchar	100	-

Tabel dibawah ini akan menjelaskan isi dari tabel pencegahan pada sistem yang dirancang

**TABEL: 3.17 Tabel Pencegahan**

Nama	Tipe Data	Ukuran	Key
id_pencegahan	int	25	<i>primary key</i>
kd_pencegahan	varchar	10	-
Deskripsi	longtext	-	-

Tabel dibawah ini akan menjelaskan isi dari tabel solusi pada sistem yang dirancang

**TABEL: 3.18 Tabel Solusi**

Nama	Tipe Data	Ukuran	Key
id_solusi	int	11	<i>primary key</i>
id_penyakit	int	11	-
Solusi	varchar	100	-

Tabel dibawah ini akan menjelaskan isi dari tabel rule pada sistem yang dirancang

**TABEL: 3.19 Tabel Rule**

Nama	Tipe Data	Ukuran	Key
id_relasi	int	11	<i>primary key</i>
id_gejala	int	11	-
id_penyakit	int	11	-

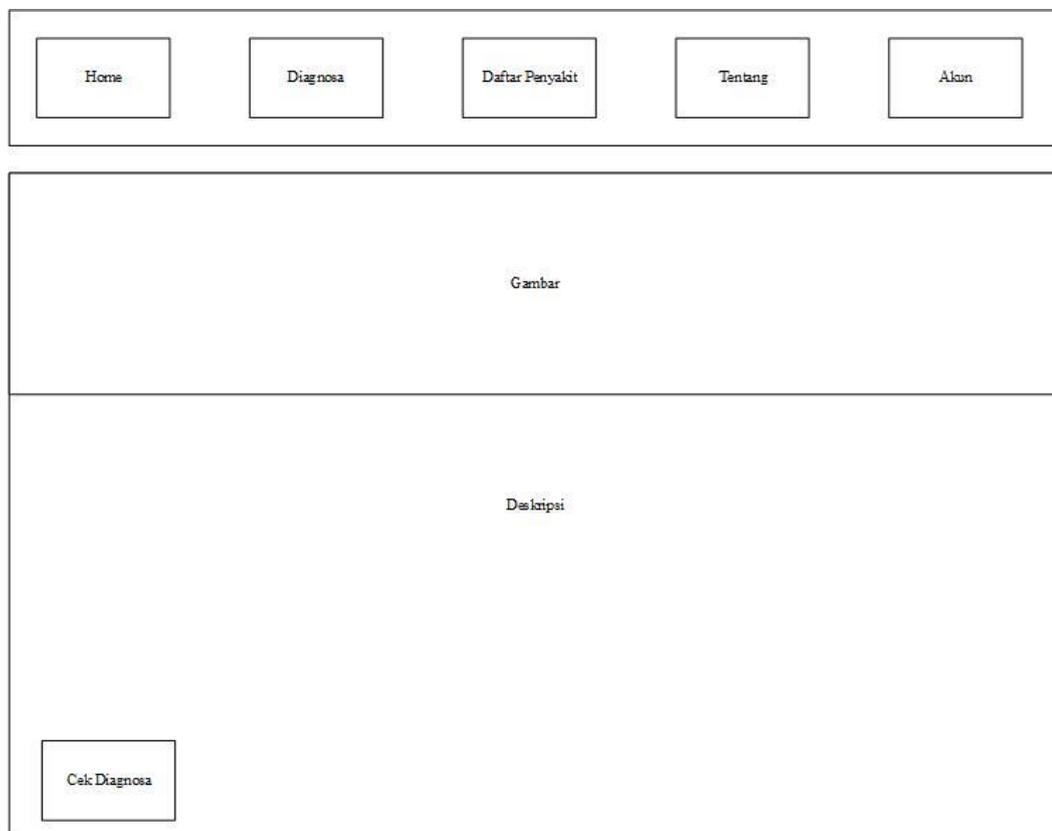
### 3.4 Design Interface

*Design Interace* merupakan rancangan antarmuka yang akan digunakan secara perantara *user* dengan perangkat lunak yang dikembangkan. *Design*

*interface* dari Sistem Pakar Diagnosa Penyakit Mata Berbasis Web ini dapat dilihat pada gambar dibawah ini:

1. Perancangan antarmuka sistem pakar diagnosa penyakit mata

Pada gambar dibawah ini menggambarkan rancangan antarmuka yang akan dibuat pada sistem pakar diagnosa penyakit mata tampilan awal atau *home*



**GAMBAR: 3.45 Rancangan *Interface* Menu Awal**

2. Perancangan antarmuka konsultasi penyakit mata

Pada gambar dibawah ini menggambarkan rancangan antarmuka untuk memulai proses diagnosa penyakit

The image shows a wireframe for a diagnostic menu interface. It consists of a top navigation bar with five buttons: Home, Diagnosa, Daftar Penyakit, Tentang, and About. Below this is a large empty rectangular area labeled 'Gambar'. Underneath the 'Gambar' area is a section with three checkboxes, each followed by the text 'Gejala1', 'Gejala2', and 'Gejala3' respectively. At the bottom left of this section is a button labeled 'Cek Diagnosa'.

Home      Diagnosa      Daftar Penyakit      Tentang      About

Gambar

Gejala1  
 Gejala2  
 Gejala3

Cek Diagnosa

**GAMBAR: 3.46 Rancangan *Interface* Menu Diagnosa**

### 3. Perancangan antarmuka hasil diagnosa

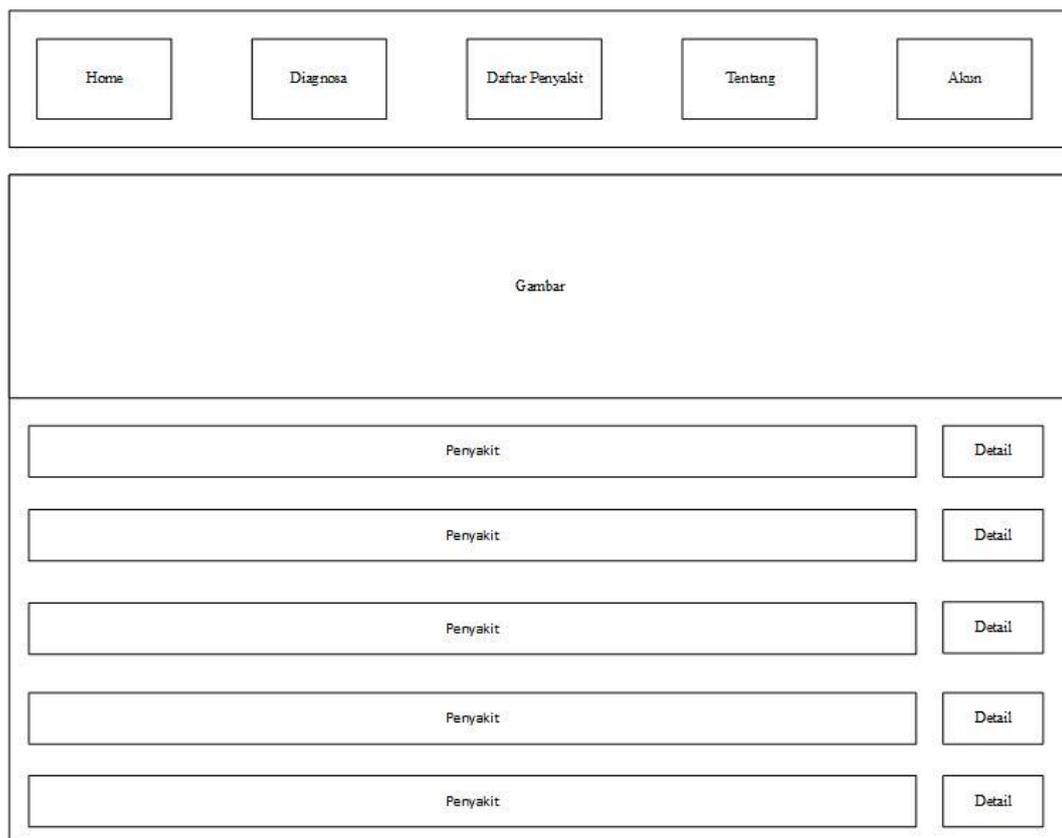
Pada gambar dibawah ini menggambarkan rancangan antarmuka hasil dari diagnosa penyakit



**GAMBAR: 3.47 Rancangan *Interface* Hasil Diagnosa Penyakit**

#### 4. Perancangan antarmuka daftar penyakit

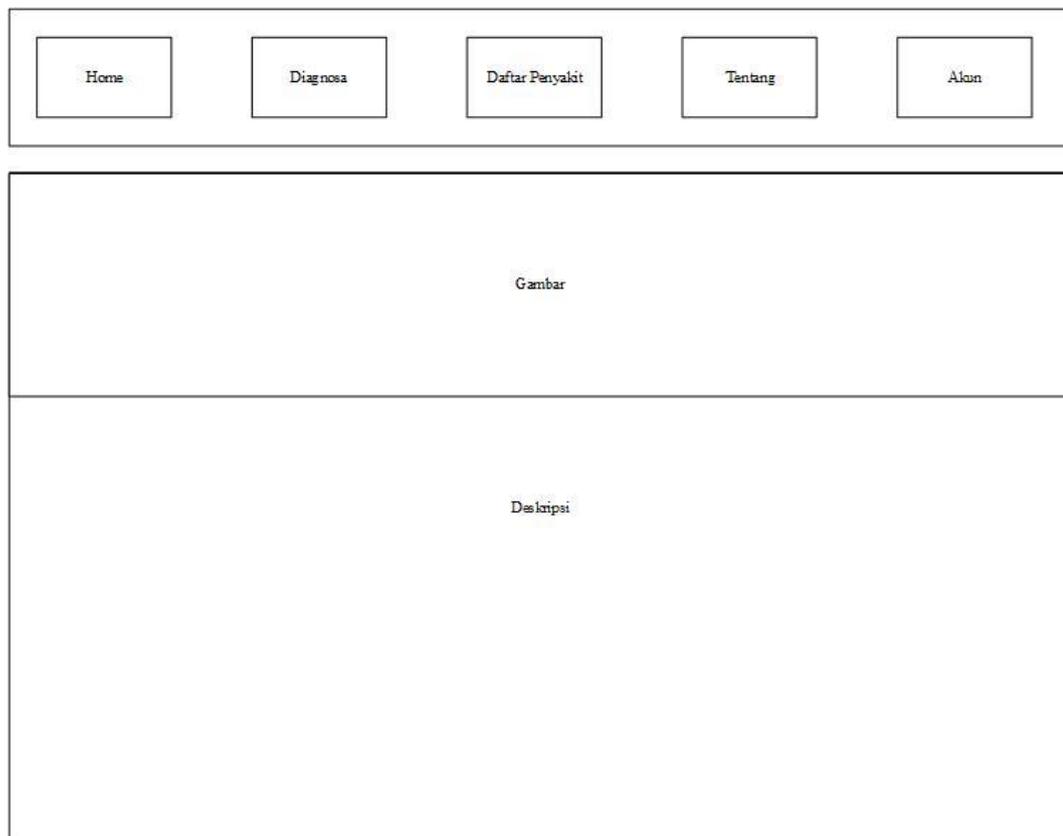
Pada gambar dibawah ini menggambarkan rancangan antarmuka dari kumpulan penyakit yang ada pada sistem



**GAMBAR: 3.48 Rancangan *Interface* Daftar Penyakit**

## 5. Perancangan antarmuka tentang mata

Pada gambar dibawah ini menggambarkan rancangan antarmuka dari menu tentang mata yang ada pada sistem



**GAMBAR: 3.49 Rancangan *Interface* Tentang Mata**

## 6. Perancangan antarmuka halaman *login*

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu login

The image shows a wireframe of a login page interface. It consists of three main sections:

- Navigation Menu:** A horizontal row of five rectangular buttons labeled "Home", "Diagnosa", "Daftar Penyakit", "Tentang", and "Akun".
- Image Placeholder:** A large rectangular area in the center labeled "Gambar".
- Login Form:** A vertical stack of three input fields: "Username", "Password", and "Login".

**GAMBAR: 3.50 Rancangan *Interface* Halaman *Login***

## 7. Perancangan antarmuka halaman admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada halaman awal admin



**GAMBAR: 3.51 Rancangan *Interface* Halaman Awal Admin**

## 8. Perancangan antarmuka halaman gejala admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu gejala admin



**GAMBAR: 3.52 Rancangan *Interface* Halaman Gejala Admin**

### 9. Perancangan antarmuka halaman penyakit admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu penyakit admin



The wireframe illustrates the layout of the Admin Disease Page. It features a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this is a large placeholder area labeled 'Gambar'. The main content area is titled 'Daftar Penyakit' and includes a 'Tambah Penyakit' button. It displays a list of three disease entries, each with a text input field labeled 'Penyakit' and two action buttons: 'Ubah' and 'Delete'.

Home	Gejala	Penyakit	Pencegahan	Solusi	Rule	Logout
------	--------	----------	------------	--------	------	--------

Gambar

Daftar Penyakit

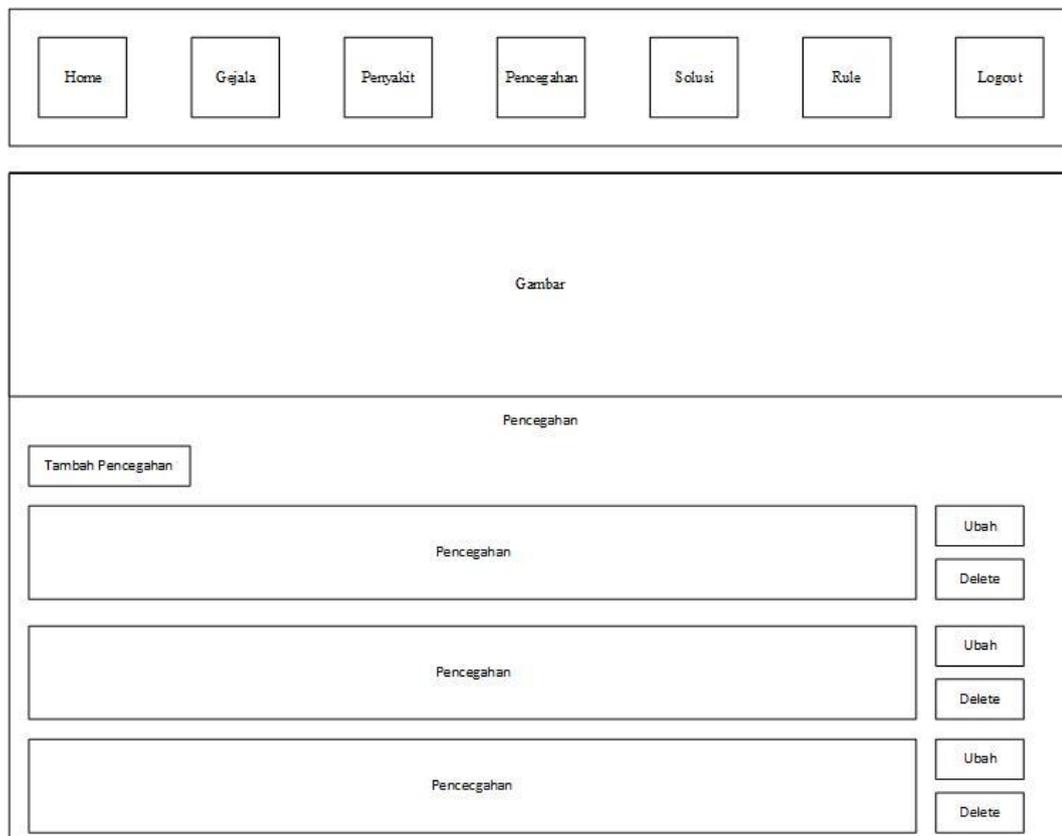
Tambah Penyakit

Penyakit	Ubah	Delete
Penyakit	Ubah	Delete
Penyakit	Ubah	Delete

**GAMBAR: 3.53 Rancangan *Interface* Halaman Penyakit Admin**

## 10. Perancangan antarmuka halaman pencegahan admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu pencegahan admin



**GAMBAR: 3.54 Rancangan *Interface* Halaman Pencegahan Admin**

## 11. Perancangan antarmuka halaman solusi admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu solusi admin



**GAMBAR: 3.55 Rancangan *Interface* Halaman Solusi Admin**

## 12. Perancangan antarmuka halaman rule admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu rule admin



**GAMBAR: 3.56 Rancangan *Interface* Halaman Rule Admin**

### 13. Perancangan antarmuka tambah gejala admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu tambah gejala admin

The image shows a wireframe for the 'Tambah Gejala Admin' (Add Symptom Admin) interface. It consists of a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below the navigation bar is a large empty rectangular area labeled 'Gambar'. Underneath that is a section labeled 'Input Gejala' which contains two text input fields: one labeled 'Kode' and another labeled 'Gejala'. At the bottom of the main content area are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

**GAMBAR: 3.57 Rancangan *Interface* Tambah Gejala Admin**

#### 14. Perancangan antarmuka tambah penyakit admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu tambah penyakit admin

The image shows a wireframe of an administrative interface for adding diseases. It consists of a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this is a large rectangular area containing the main content. The content is titled 'Gambar' and includes a section labeled 'Input Penyakit'. This section contains three horizontal text input fields: the first is labeled 'Kode', the second is labeled 'Penyakit', and the third is labeled 'Penyebab'. At the bottom of the 'Input Penyakit' section, there are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

**GAMBAR: 3.58 Rancangan *Interface* Tambah Penyakit Admin**

### 15. Perancangan antarmuka tambah pencegahan admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu tambah pencegahan admin

The image shows a wireframe of an administrative interface for adding prevention rules. At the top, there is a horizontal navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this, the main content area is enclosed in a large rectangular frame. The top part of this frame is labeled 'Gambar'. The bottom part is labeled 'Input Pencegahan' and contains three stacked text input fields. The first field is labeled 'Kode Pencegahan', the second is 'Kode Penyakit', and the third is 'Pencegahan'. At the bottom left of the input section, there are two buttons: 'Simpan' and 'Kembali'.

**GAMBAR: 3.59 Rancangan *Interface* Tambah Pencegahan Admin**

## 16. Perancangan antarmuka tambah solusi admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu tambah solusi admin



The image shows a wireframe of an admin interface for adding a solution. It features a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below the navigation bar is a large empty rectangular area labeled 'Gambar'. Underneath that is a section titled 'Input Solusi' containing four text input fields: 'Kode Solusi', 'Kode Pencegahan', 'Nama Obat', and 'Solusi'. At the bottom of the input section are two buttons: 'Simpan' and 'Kembali'.

**GAMBAR: 3.60 Rancangan *Interface* Tambah Solusi Admin**

### 17. Perancangan antarmuka tambah rule admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu tambah rule admin

The image shows a wireframe for the 'Tambah Rule Admin' interface. It consists of a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below the navigation bar is a large rectangular area divided into two sections. The top section is labeled 'Gambar' and is currently empty. The bottom section is labeled 'Input Rule' and contains two large text input fields. The first input field is labeled 'Jika' and the second is labeled 'Maka'. At the bottom of the 'Input Rule' section, there are two buttons: 'Simpan' and 'Kembali'.

**GAMBAR: 3.61 Rancangan *Interface* Tambah Rule Admin**

## 18. Perancangan antarmuka ubah gejala admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu ubah gejala admin

The image shows a wireframe of an administrative interface for changing symptoms. It consists of a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this is a large rectangular area containing the following elements:

- A header section labeled "Gambar" (Image).
- A sub-header section labeled "Ubah Gejala" (Change Symptom).
- A text input field labeled "Kode Gejala" (Symptom Code).
- A text input field labeled "Gejala" (Symptom).
- Two buttons at the bottom: "Simpan" (Save) and "Kembali" (Back).

**GAMBAR: 3.62 Rancangan *Interface* Ubah Gejala Admin**

### 19. Perancangan antarmuka ubah penyakit admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu ubah penyakit admin

The image shows a wireframe for an administrative interface to change a disease. It consists of a top navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this is a large rectangular area containing the title 'Ubah Penyakit'. Underneath the title are three horizontal input fields labeled 'Kode Penyakit', 'Nama Penyakit', and 'Penyebab'. At the bottom of this area are two buttons: 'Simpan' and 'Kembali'.

**GAMBAR: 3.63 Rancangan *Interface* Ubah Penyakit Admin**

## 20. Perancangan antarmuka ubah pencegahan admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu ubah pencegahan admin

The image shows a wireframe of an administrative interface for editing prevention rules. At the top, there is a horizontal navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this is a large rectangular area containing the main content. The content is titled 'Ubah Pencegahan' (Edit Prevention). It consists of three stacked input fields. The first field is labeled 'Kode Pencegahan', the second is labeled 'Kode Penyakit', and the third is labeled 'Pencegahan'. At the bottom of this content area, there are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

**GAMBAR: 3.64 Rancangan *Interface* Ubah Penyakit Admin**

## 21. Perancangan antarmuka ubah solusi admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu ubah solusi admin



Home    Gejala    Penyakit    Pencegahan    Solusi    Rule    Logout

Gambar

Ubah Solusi

Kode Solusi

Kode Pencegahan

Nama Obat

Solusi

Simpan    Kembali

**GAMBAR: 3.65 Rancangan *Interface* Ubah Solusi Admin**

## 22. Perancangan antarmuka ubah rule admin

Pada gambar dibawah ini menggambarkan rancangan antarmuka pada menu gejala admin

The image shows a wireframe of an administrative interface for changing rules. At the top, there is a horizontal navigation bar with seven buttons: Home, Gejala, Penyakit, Pencegahan, Solusi, Rule, and Logout. Below this is a large rectangular area containing the main content. The content is organized as follows: a header section labeled 'Gambar', a sub-header 'Ubah Rule', a text input field labeled 'Jika', another text input field labeled 'Maka', and finally two buttons at the bottom: 'Simpan' and 'Kembali'.

**GAMBAR: 3.66 Rancangan *Interface* Ubah Rule Admin**

## **BAB IV**

### **IMPLEMENTASIDAN UJI COBA**

#### **4.1 Construction (Code & Test)**

Dalam tahap ini penelitian berfokus pada pengkodean menggunakan bahasa PHP, setelah itu dilakukan pengujian hasil menggunakan metode *blackbox testing*. *Blackbox testing* ini menguji spesifikasi suatu fungsi atau modul, apakah berjalan dengan semestinya dan sesuai yang diharapkan atau tidak.

##### **4.1.1 Implementasi *Hardware & Software***

Dalam tahap ini peneliti memerlukan spesifikasi *hardware* dan *software* untuk mengoperasikan sistem pakar dan menjalankan aplikasi. Untuk lebih jelasnya akan dijelaskan dibawah ini:

###### **4.1.1.1 Hardware**

*Hardware* yang digunakan diantaranya:

1. *Processor* Intel Core i3 inside
2. *Memory* 2.00 Gyga Byte
3. *Hard disk* Minimal 1 Gyga Byte

###### **4.1.1.2 Software**

1. Sistem Operasi Windows 7
2. *Browser* (*Google Chrome, Opera, Mozilla Firefox, Microsoft Edge*)

3. *PHP*
4. *MySql*
5. *Bootstrap*
6. *Apache*

#### 4.1.2 Implementasi Antarmuka

Pada gambar 4.1 menggambarkan tampilan menu konsultasi yang dimana user akan memilih gejala penyakit mata dengan melakukan checklist pada checkbox yang tersedia.



**GAMBAR: 4.1 Tampilan Memilih Gejala Penyakit Mata**

Pada gambar 4.2 menggambarkan tampilan hasil metode *Forward Chaining* setelah memilih gejala penyakit mata.

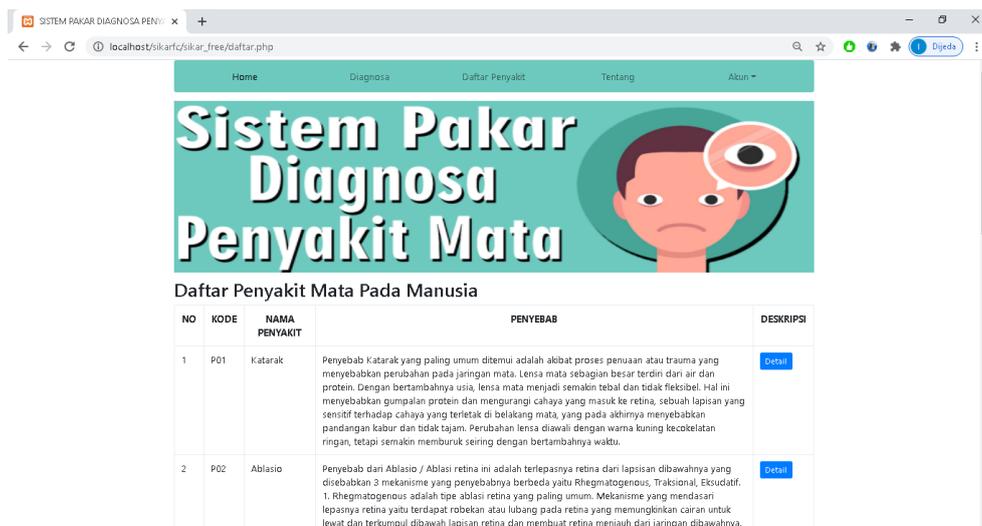


The screenshot shows a web browser window with the URL `localhost/sikarfz/sikar_free/hasil.php`. The page title is "Sistem Pakar Diagnosa Penyakit Mata". Below the header, there is a section titled "Hasil Diagnosa" with the following content:

Gejala	<ul style="list-style-type: none"> <li>Kelopak mata terasa gatal</li> <li>Kelopak mata menjadi lengket</li> <li>Tepi kelopak mata terlihat bengkak</li> </ul>
Penyakit	Blefaritis
Penyebab	Penyebab terjadinya penyakit ini sebenarnya hingga kini belum diketahui pasti. Namun, dalam kebanyakan kasus penyakit mata ini disebabkan oleh bakteri yang membuat tepi kelopak mata yang sensitif sehingga muncul peradangan.
Pencegahan	Penyakit ini dapat dihindari dengan mencuci wajah hingga bersih secara teratur setiap hari. Selain itu, menjaga kebersihan tangan untuk menghindari kontaminasi bakteri ketika menyentuh mata, dan jika terbiasa memakai riasan wajah jangan sampai lupa untuk membersihkannya setiap sebelum tidur malam.
Nama Obat	Antibiotik

**GAMBAR: 4.2 Tampilan Hasil Diagnosa Penyakit Mata**

Pada gambar 4.3 menggambarkan tampilan menu daftar penyakit yang ada pada aplikasi sistem pakar diagnosa penyakit mata

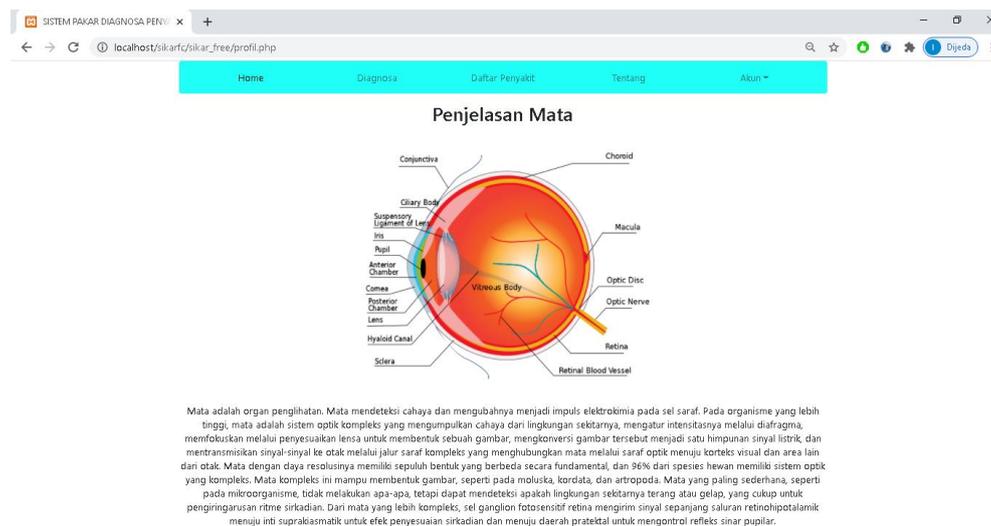


The screenshot shows a web browser window with the URL `localhost/sikarfz/sikar_free/daftar.php`. The page title is "Sistem Pakar Diagnosa Penyakit Mata". Below the header, there is a section titled "Daftar Penyakit Mata Pada Manusia" with the following table:

NO	KODE	NAMA PENYAKIT	PENYEBAB	DESKRIPSI
1	P01	Katarak	Penyebab Katarak yang paling umum ditemui adalah akibat proses penuaan atau trauma yang menyebabkan perubahan pada jaringan mata. Lensa mata sebagian besar terdiri dari air dan protein. Dengan bertambahnya usia, lensa mata menjadi semakin tebal dan tidak fleksibel. Hal ini menyebabkan gumpalan protein dan mengurangi cahaya yang masuk ke retina, sebuah lapisan yang sensitif terhadap cahaya yang terletak di belakang mata, yang pada akhirnya menyebabkan pandangan kabur dan tidak tajam. Perubahan lensa diawali dengan warna kuning kecokelatan ringan, tetapi semakin memburuk seiring dengan bertambahnya waktu.	<a href="#">Detail</a>
2	P02	Ablasio	Penyebab dari Ablasio / Ablasi retina ini adalah terlepasnya retina dari lapisan dibawahnya yang disebabkan 3 mekanisme yang penyebabnya berbeda yaitu Rhegmatogenous, Traksional, Eksudatif. 1. Rhegmatogenous adalah tipe ablasi retina yang paling umum. Mekanisme yang mendasari lepasnya retina yaitu terdapat robekan atau lubang pada retina yang memungkinkan cairan untuk lewat dan terkumpul dibawah lapisan retina dan membuat retina menjauh dari jaringan dibawahnya.	<a href="#">Detail</a>

**GAMBAR: 4.3 Tampilan Menu Dari Daftar Penyakit Mata**

Pada gambar 4.4 menggambarkan tampilan menu tentang mata yang ada pada aplikasi sistem pakar diagnosa penyakit mata



**GAMBAR: 4.4 Tampilan Menu Tentang Mata Dari Aplikasi**

Pada gambar 4.5 menggambarkan tampilan menu dari halaman *login* admin pada aplikasi sistem pakar diagnosa penyakit mata



**GAMBAR: 4.5 Tampilan Menu Login Admin**

Pada gambar 4.6 menggambarkan tampilan menu dari halaman gejala yang telah dimasukkan sebelumnya berdasarkan pengetahuan pakar

NO	KODE	Gejala	Ops
1	G01	Pandangan kabur seperti berkabut	Edit Delete
2	G02	Pandangan ganda	Edit Delete
3	G03	Warna disektar memudar	Edit Delete
4	G04	Melihat kilatan cahaya (Fotopsia)	Edit Delete

**GAMBAR: 4.6 Tampilan Menu Dari Data Gejala Penyakit Mata**

Pada gambar 4.7 menggambarkan tampilan menu dari halaman penyakit yang telah dimasukkan sebelumnya berdasarkan pengetahuan pakar

No.	KODE PENYAKIT	PENYAKIT	PENYEBAB	OPSI
1	P01	Katarak	Penyebab Katarak yang paling umum ditemui adalah akibat proses penuaan atau trauma yang menyebabkan perubahan pada jaringan mata. Lensa mata sebagian besar terdiri dari air dan protein. Dengan bertambahnya usia, lensa mata menjadi semakin tebal dan tidak fleksibel. Hal ini menyebabkan gumpalan protein dan mengurangi cahaya yang masuk ke retina, sebuah lapisan yang sensitif terhadap cahaya yang terletak di belakang mata, yang pada akhirnya menyebabkan pandangan kabur dan tidak tajam. Perubahan lensa diawali dengan warna kuning kecokelatan ringan, tetapi semakin memburuk seiring dengan bertambahnya waktu.	Edit Delete
2	P02	Ablasio	Penyebab dari Ablasio / Ablasi retina ini adalah terlepasnya retina dari lapisan dibawahnya yang disebabkan 3 mekanisme yang penyebabnya berbeda yaitu Rhegmatogenous, Traksional, Eksudatif.	Edit Delete

**GAMBAR: 4.7 Tampilan Menu Dari Data Penyakit Mata**

Pada gambar 4.8 menggambarkan tampilan menu dari halaman pencegahan yang telah dimasukkan sebelumnya berdasarkan pengetahuan pakar.

No.	KODE PENCEGAHAN	KODE PENYAKIT	PENCEGAHAN	OPSI
1	PP01	PD1	Memeriksa mata secara teratur pada dokter spesialis mata, melindungi mata dari benturan dan cahaya matahari yang terlalu lama, dengan menggunakan kacamata yang melindungi dari sinar ultraviolet baik UV-A dan UV-B, mempertahankan kadar gula darah agar tetap normal pada pengidap diabetes, membiasakan kebiasaan menyetr di malam hari, memperbaiki pencahayaan di rumah, dan menggunakan kaca pembesar saat membaca.	Kode Delete
2	PP02	PD2	Ablasi retina tidak selamanya dapat dicegah, misalnya ablasi retina yang disebabkan karena proses peneasan. Namun, beberapa upaya ini dapat mengurangi risiko seseorang agar tidak terjadi ablasi retina, diantaranya: Memeriksa kesehatan mata secara rutin, terlebih jika memiliki faktor resiko seperti diabetes, hipertensi, dan usia tua, menggunakan alat pelindung mata saat berolahraga agar menourangi resiko cedera serius pada mata. Kapan harus ke Dokter Spesialis?	Kode Delete

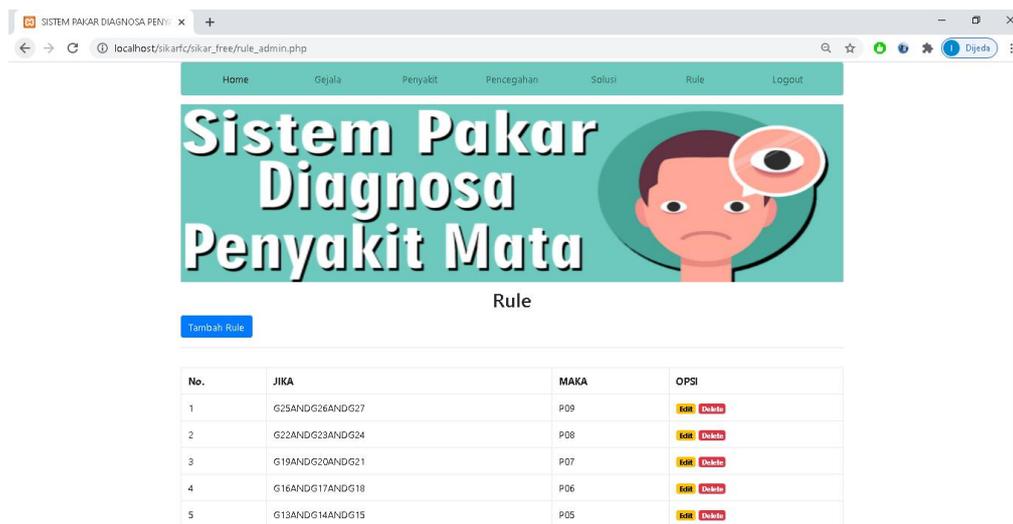
**GAMBAR: 4.8 Tampilan Menu Dari Data Pencegahan Penyakit Mata**

Pada gambar 4.9 menggambarkan tampilan menu dari halaman solusi yang telah dimasukkan sebelumnya berdasarkan pengetahuan pakar

No.	KODE SOLUSI	KODE PENCEGAHAN	NAMA OBAT	SOLUSI	OPSI
1	S01	PP01	Lanosterol, N-acetylcarnosine (NAC), Cydopentolate dan atropin, dan Obat herbal	Jika alabkat katarak semalan memburuk dan sulit untuk menjalani aktivitas sehari-hari, pengobatan katarak hanyalah dengan prosedur operasi. Operasi katarak pada umumnya aman dan tidak membutuhkan rawat inap. Ada dua jenis operasi katarak, yaitu:	Kode Delete
2	S01	PP01	Lanosterol, N-acetylcarnosine	Small incision cataract surgery (phacoemulsification). Operasi ini dilakukan dengan melakukan insisi kecil pada tepi kornea. Selanjutnya, dotter akan menyinarakan	Kode Delete

**GAMBAR: 4.9 Tampilan Menu Dari Data Solusi Penyakit Mata**

Pada gambar 4.10 menggambarkan tampilan menu dari halaman rule yang telah dimasukkan sebelumnya berdasarkan pengetahuan pakar



**GAMBAR: 4.10 Tampilan Menu Dari Data Rule Penyakit Mata**

### 4.1.3 Testing

Testing merupakan bagian penting dalam siklus pengembangan perangkat lunak. *Testing* dilakukan untuk menjamin kualitas dan mengetahui kelemahan dari perangkat lunak. Tujuan dari *testing* perangkat lunak ini adalah untuk menjamin perangkat lunak yang dirancang sedemikian rupa dapat berkualitas dan dapat digunakan dengan baik.

#### 4.1.3.1 Testing Blackbox

Pengujian *black box* digunakan untuk mengetahui fungsi-fungsi khusus dari aplikasi perangkat lunak yang telah dibuat

**TABEL: 4.1 Tabel Rencana Pengujian**

<b>No.</b>	<b>Kelas Uji</b>	<b>Butir Uji</b>	<b>Jenis Pengujian</b>
<b>1</b>	Login Pakar	Verifikasi Username	<i>Blackbox</i>
		Verifikasi Password	
<b>2</b>	Pengolahan Data Gejala	Tambah Data Gejala	<i>Blackbox</i>
		Ubah Data Gejala	
		Hapus Data Gejala	
<b>3</b>	Pengolahan Data Penyakit	Tambah Data Penyakit	<i>Blackbox</i>
		Ubah Data Penyakit	
		Hapus Data Penyakit	
<b>4</b>	Pengolahan Data Pencegahan	Tambah Data Pencegahan	<i>Blackbox</i>
		Ubah Data Pencegahan	
		Hapus Data Pencegahan	
<b>5</b>	Pengolahan Data Solusi	Tambah Data Solusi	<i>Blackbox</i>
		Ubah Data Solusi	
		Hapus Data Solusi	
<b>6</b>	Pengolahan Data Rule	Tambah Data Rule	<i>Blackbox</i>
		Ubah Data Rule	
		Hapus Data Rule	
<b>7</b>	Diagnosa Gejala	Diagnosa	<i>Blackbox</i>

Setelah dilakukan pengujian dengan menggunakan metode *blackbox* maka didapatkan hasil pengujian pada tabel berikut ini:

**TABEL: 4.2 Tabel Hasil Pengujian Login Admin**

No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
1.	Login Admin	<i>Username dan Password</i>	Data <i>username</i> dan <i>password</i> dicari dari <i>database</i> masuk ke menu admin		Diterima

TABEL: 4.3 Tabel Hasil Pengujian Tambah Data Gejala

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2.	Tambah Data Gejala	Kode gejala, dan nama gejala	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2.	Tambah Data Gejala	Kode gejala, dan nama gejala tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

TABEL: 4.4 Tabel Hasil Pengujian Ubah Data Gejala

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2.	Ubah Data Gejala	Kode gejala, dan nama gejala	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2.	Tambah Data Gejala	Kode gejala, dan nama gejala tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

**TABEL: 4.5 Tabel Hasil Pengujian Hapus Data Gejala**

No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
2.	Hapus Data Gejala	Gejala yang akan dihapus	Menampilkan pesan “yakin ingin menghapus data ini?” apabila memilih “Oke” maka data gejala akan dihapus tetapi jika memilih “Batal” data gejala batal untuk dihapus		Diterima

TABEL: 4.6 Tabel Hasil Pengujian Tambah Data Penyakit

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3.	Tambah Data Penyakit	Kode penyakit, nama penyakit, dan penyebab	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3.	Tambah Data Penyakit	Kode penyakit, nama penyakit, dan penyebab tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

TABEL: 4.7 Tabel Hasil Pengujian Ubah Data Penyakit

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3.	Ubah Data Penyakit	Kode penyakit, nama penyakit, dan penyebab	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3.	Ubah Data Penyakit	Kode penyakit, nama penyakit, dan penyebab tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

**TABEL: 4.8 Tabel Hasil Pengujian Hapus Data Penyakit**

No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
3.	Hapus Data Penyakit	Penyakit yang akan dihapus	Menampilkan pesan “yakin ingin menghapus data ini?” apabila memilih “Oke” maka data gejala akan dihapus tetapi jika memilih “Batal” data gejala batal untuk dihapus		Diterima

TABEL: 4.9 Tabel Hasil Pengujian Tambah Data Pencegahan

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4.	Tambah Data Pencegahan	Kode pencegahan, kode penyakit, pencegahan	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4.	Tambah Data Pencegahan	Kode pencegahan, kode penyakit, pencegahan tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

TABEL: 4.10 Tabel Hasil Pengujian Ubah Data Pencegahan

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4.	Ubah Data Pencegahan	Kode pencegahan, kode penyakit, dan pencegahan	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4.	Ubah Data Pencegahan	Kode pencegahan, kode penyakit, pencegahan tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan “harap isi bidang ini”		Diterima

**TABEL: 4.11 Tabel Hasil Pengujian Hapus Data Pencegahan**

No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
4.	Hapus Data Pencegahan	Pencegahan yang akan dihapus	Menampilkan pesan “yakin ingin menghapus data ini?” apabila memilih “Oke” maka data gejala akan dihapus tetapi jika memilih “Batal” data gejala batal untuk dihapus		Diterima

TABEL: 4.12 Tabel Hasil Pengujian Tambah Data Solusi

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5.	Tambah Data Solusi	Kode solusi, kode pencegahan, nama obat, solusi	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5.	Tambah Data Solusi	Kode solusi, kode pencegahan, nama obat, solusi tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

TABEL: 4.13 Tabel Hasil Pengujian Ubah Data Solusi

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5.	Ubah Data Solusi	Kode solusi, kode pencegahan, nama obat, solusi	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5.	Ubah Data Solusi	Kode solusi, kode pencegahan, nama obat, solusi tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan “harap isi bidang ini”		Diterima

TABEL: 4.14 Tabel Hasil Pengujian Hapus Data Solusi

No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
5.	Hapus Data Solusi	Solusi yang akan dihapus	Menampilkan pesan “yakin ingin menghapus data ini?” apabila memilih “Oke” maka data gejala akan dihapus tetapi jika memilih “Batal” data gejala batal untuk dihapus		Diterima

TABEL: 4.15 Tabel Hasil Pengujian Tambah Data Rule

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
6.	Tambah Data Rule	Jika dan Maka	Data tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
6.	Tambah Data Rule	Jika dan maka tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

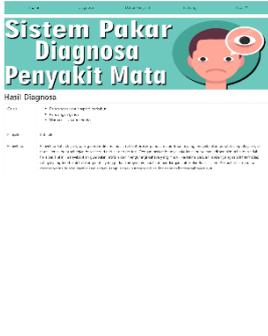
TABEL: 4.16 Tabel Hasil Pengujian Ubah Data Rule

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
6.	Ubah Data Rule	Jika dan maka	Data yang diubah tersimpan kedalam database		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
6.	Ubah Data Rule	Jika dan maka tetapi salah satu form dikosongkan	Tidak terjadi proses simpan, menampilkan pesan "harap isi bidang ini"		Diterima

TABEL: 4.17 Tabel Hasil Pengujian Hapus Data Rule

No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
6.	Hapus Data Rule	Rule yang akan dihapus	Menampilkan pesan “yakin ingin menghapus data ini?” apabila memilih “Oke” maka data gejala akan dihapus tetapi jika memilih “Batal” data gejala batal untuk dihapus		Diterima

TABEL: 4.18 Tabel Hasil Pengujian Diagnosa Gejala

Kasus dan Hasil Uji (Data Normal)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
7.	Diagnosa Gejala	Pilih gejala yang akan di diagnose	Data gejala akan dicocokkan berdasarakan rule yang telah dibuat dan menampilkan hasil diagnosa		Diterima
Kasus dan Hasil Uji (Data Salah)					
No	Kelas Uji	Data Inputan	Hasil Yang Diharapkan	Hasil Yang Ditampilkan Sistem	Kesimpulan
7.	Diagnosa Gejala	Gejala dipilih tidak sesuai rule yang telah dibuat	Tidak terjadi proses hasil diagnosa dan akan menampilkan pesan “Ups! Tidak ditemukan penyakit dengan gejala tersebut silahkan cek daftar penyakit”		Diterima

#### 4.1.3.2 *Testing* Klasifikasi Penyakit Mata

Pengujian klasifikasi penyakit mata digunakan untuk menguji seberapa besar kemampuan sistem dalam mengklasifikasi penyakit mata, dimana pada penelitian ini penyakit mata yang telah diklasifikasikan akan muncul hasil diagnosa dan besarnya akurasi terhadap kesamaan penyakit mata yang diuji dengan penyakit mata yang ada pada *database*.

Pada penelitian ini penulis melakukan *testing* klasifikasi penyakit mata dengan menggunakan kelas uji yang berisi penyakit mata yang akan diklasifikasikan.

**TABEL: 4.19** Tabel Kelas Uji

No.	Kelas Uji
1.	Katarak
2.	Ablasio
3.	Astigmatisme
4.	Glaukoma
5.	Buphthalmus
6.	Konjungtivitis
7.	Pterygium
8.	Blefaritis
9.	Dakriosistitis

#### **4.1.4 Kesimpulan *Testing***

Berdasarkan hasil *testing* yang dilakukan didapat kesimpulan bahwa aplikasi sudah berjalan sesuai dengan apa yang diharapkan, tetapi tidak menutup kemungkinan terdapat kesalahan suatu saat aplikasi sedang digunakan. Sehingga memutuskan proses perbaikan untuk mengetahui kekurangan dari aplikasi ini.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan, maka dapat diambil kesimpulan diantaranya:

1. Dengan adanya aplikasi sistem pakar diagnosa penyakit mata menggunakan *Forward Chaining* ini diharapkan dapat membantu pengguna untuk mengetahui penyakit mata dari gejala yang dirasakan dan mengajak pengguna untuk tetap menjaga kesehatan mata.
2. *Forward Chaining* bisa dimanfaatkan sebagai media pembelajaran untuk membuat manusia tetap menjaga kesehatan organ-organ tubuhnya.

#### **5.2 Saran**

Dengan adanya kesimpulan maka ada beberapa saran yang dapat dikemukakan sebagai bahan pertimbangan selanjutnya, diantaranya:

1. Pada pengembangan selanjutnya diharapkan ditambahkan menu-menu lainnya yang membuat aplikasi ini jauh lebih menarik dan nyaman digunakan
2. Diharapkan diperbanyak aplikasi-aplikasi kesehatan yang dibangun dengan sistem pakar ini.

## DAFTAR PUSTAKA

- Agung Baitul Hikmah, dkk. 2015. *Cara Cepat Membangun Website dari Nol Studi Kasus: Web Dealer Motor*. CV. Andi Offset. Yogyakarta
- Akil, 2017 “*Analisa Efektifitas Metode Forward Chaining Dan Backward Chaining Pada Sistem Pakar*” Jurnal Pilar Nusa Mandiri Volume 13 No.1 Maret
- Astamal, Rio, 2005. *Mastering Kode HTML*. Surabaya.
- Betha, Sidik, 2014. *Pemrograman Web PHP*. Informatika Bandung. Bandung.
- Budhi, G. S., Dewi, L. P., dan Suwarso, G. A., 2015. *Sistem Pakar Untuk Penyakit Anak Menggunakan Metode Forward Chaining*. Jurnal *INFRA*
- Evelyn C, Pearce, 1999. *Anatomi dan Fisiologi untuk Paramedis*, PT. Gramedia Pustaka Utama. Jakarta
- Fowler, 2004. *UML. Distilled\_A Brief Guide to Standart Object*.
- Hananto, P. E., dkk, 2012. *Sistem Pakar Diagnosis Penyakit Tanaman Cengkih Dengan Metode Inferensi*. *Journal of Informatics and Technology*.
- Jogiyanto, H. M., 1999. *Analisis dan Desain Sistem Informasi*. Andi. Yogyakarta.
- Kamus Besar Bahasa Indonesia, 1998. Pustaka Armani. Jakarta
- Komputer, Wahana, 2010. *Panduan Belajar MySQL Database Server*. Mediakita. Jakarta.
- Kustiyahningsih, 2010. *Pemrograman Basis Data Berbasis Web Menggunakan PHP & MySQL*. Graha Ilmu. Yogyakarta.

- Kusumadewi, 2003. *Artificial Intelligence*. Graha Ilmu. Yogyakarta.
- Lestari, 2012. *Definisi Sistem Pakar*. Arsip Teknik Informatika UMMI.
- Murtopo I, dan Sarimurni, 2005, *Pengaruh Radiasi Layar Komputer Terhadap Daya Akomodasi Mata Mahasiswa Pengguna Komputer di Universitas Muhammadiyah Surakarta*, Surakarta.
- Nugroho, Adi, 2010. *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*, Andi. Yogyakarta.
- Nugroho, Bunafit, 2013. *Dasar Pemrograman Web PHP-MySQL dengan Dreamweaver*. Gava Media. Yogyakarta.
- Oktavian, Diar Puji, 2013. *Membuat Website Powerfull Menggunakan PHP*. Mediakom. Yogyakarta.
- Pressman, dan Bruce, 2014. *Software Engineering A Practitioner's Approach Eight Edition*
- Pressman, R. S., 2015. *Rekayasa Perangkat Lunak: Pendekatan Praktisi buku 1*. Andi. Yogyakarta.
- Purba, 2017 “*Perancangan Sistem Pakar Diagnosa Penyakit Mata Katarak Menggunakan Konsep Metode Runut Mundur*” JUSIKOM PRIMA Vol 1 No.1 Juni
- Putra, dkk, 2018 “*Sistem Pakar Diagnosa Penyakit Mata dengan Fuzzy Logic dan Naïve Bayes*” Jurnal Merpati Volume 6 No.1 April
- Raharjo, Budi. 2012. *Pemrograman Web (HTML, PHP, & MySQL)*. Modula. Bandung

- Rahmatullah, dkk, 2018 “*Sistem Pakar Untuk Mendiagnosa Penyakit Mata Dengan Metode Forward Chaining*” *Journal Speed Sentra Penelitian Engineering dan Edukasi* Volume 10 No.2
- Ramli, 1990. *Kamus Kedokteran*. Djambatan. Jakarta.
- Sanjaya, Wina, 2013. *Strategi Pembelajaran Berorientasi Standar Proses Pendidikan*. Kencana. Jakarta
- Saputra, dkk, 2015. *Rancang Bangun Sistem Pendukung Keputusan Penentuan Keluarga Miskin. Kumpulan Jurnal. Ilmu Komputer (KLIK)*.
- Soeherman, Bonnie dan Marion Pinontoan, 2008. *Designing Information System*. PT. Elex Media Komputindo. Jakarta.
- Subtari, Tata, 2012. *Konsep Sistem Informasi*. Andi. Yogyakarta.
- Sukanto, Ariani, Rosa, dan Shalahuddin M, 2015. *Rekayasa Perangkat Lunak dan Berorientasi Objek*. Bandung. Informatika Bandung.
- Yanto, dkk, 2017 “*Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Bawah Lima Tahun Menggunakan Metode Forward Chaining*” *Journal of Information System Engineering and Business Intelligence* Vol.3 No.1 April

## LAMPIRAN

test.php

```
<?php
```

```
include 'function.php';
```

```
if (isset($_SESSION['role'])) {
```

```
    if ($_SESSION['role'] == 0) {
```

```
        header("location: indexAdmin.php");
```

```
    } else if ($_SESSION['role'] == 2) {
```

```
        header("location: indexPakar.php");
```

```
    }
```

```
}
```

```
if(!isset($_SESSION['persentase'])) {
```

```
    $_SESSION['persentase'] = [];
```

```
}
```

```
$gejala = mysqli_query($koneksi, "SELECT * FROM gejala");
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<link
```

```
rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
```

```
integrity="sha384-
```

```
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q
```

```
9Ijfh"
```

```
crossorigin="anonymous"/>
```

```
<link
```

```
href="https://fonts.googleapis.com/css?family=Poppins:300,400,700&display=sw
```

```
ap"
```

```
rel="stylesheet"/>
```

```
<link rel="stylesheet" href="custom.css" />
```

```
<title>Cek Mata Yuk!</title>
```

```
</head>
```

```
<body>
```

```
<nav class="navbar py-2 navbar-expand-lg navbar-light">
```

```
<div class="container">
```

```
<a class="navbar-brand" href="#"
```

```
></a>
```

```
<button
```

```
class="navbar-toggler"
```

```
type="button"
```

```
data-toggle="collapse"
```

```
data-target="#navbarSupportedContent"
```

```
aria-controls="navbarSupportedContent"
```

```
aria-expanded="false"
```

```
aria-label="Toggle navigation"
```

```
>
```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```
<ul class="navbar-nav ml-auto">
```

```
<li>
```

```
<a class="btn px-4 btn-primary ml-2" href="logout.php"
role="button"
```

```
>Log Out</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
<section class="test mt-5">

  <div class="container">

    <div class="row">

      <div class="col align-self-center">

        <h2 class="mb-4">Pertanyaan : </h2>

        <form action="" method="post" enctype="multipart/form-data"
role="form">

          <?php

            $id_penyakit=1;

            // if(!isset($_SESSION['id_penyakit'])){

            //   $_SESSION['id_penyakit'] = $id_penyakit;

            // }else{

            //   $id_penyakit = $_SESSION['id_penyakit'];

            // }

            $id = gejala($id_penyakit);

            $id_gejala = intval($id);

            if(!isset($_SESSION['id_gejala'])){

              $_SESSION['id_gejala'] = $id_gejala;
```

```

    }else{

        $id_gejala = $_SESSION['id_gejala'];

    }

    $data = mysqli_query($koneksi, "SELECT gejala FROM gejala
WHERE id_gejala = '$id_gejala'");

    // var_dump($data);

    $row = mysqli_fetch_assoc($data);

?>

<p class="mb-4">

    Apakah anda mengalami <?= $row['gejala']; ?> ?

</p>

<?php

    echo'<input type="submit" class="btn btn-primary mr-2 px-4 py-2"
name="ya" value="Ya">';

    echo'<input type="submit" class="btn btn-danger px-3 py-2"
name="tidak" value="Tidak">';

    $persentase = $_SESSION['persentase'];

    $temp = 0;

```

```
$_SESSION['id_gejala'] = $id_gejala;

$next_gejala = $_SESSION['id_gejala'];

// $next_penyakit = $_SESSION['id_penyakit'];

if(isset($_POST['ya'])){

    if(isset($id_gejala)){

        $temp = $id_gejala;

        array_push($persentase, $temp);

    }

    $_SESSION['persentase'] = $persentase;

    $next_gejala = $id_gejala + 1;

    $_SESSION['id_gejala'] = $next_gejala;

}

else if(isset($_POST['tidak'])){

    $next_gejala = $id_gejala + 1;

    $_SESSION['id_gejala'] = $next_gejala;

    // $next_penyakit = $id_penyakit += 1;

    // $_SESSION['id_penyakit'] = $next_penyakit;
```

```
}

if($_SESSION['id_gejala'] > 28) {

$Katarak = array(1,2,3,4,5,6,7);

$Buftalmus = array(8,9,10,11,12,13);

$Astigmatisme = array(14,15,16,17,18);

$Dakriosistitis = array(19,20);

$Blefaritis = array(21,22,23);

$Glaukoma = array(24,25,26,27,28);

$nilai = 0;

foreach ($persentase as $value) {

    if (in_array($value, $Katarak)) {

        $nilai += 1;

    }else{

        $nilai += 0;

    }

}

}
```

```
$GinjalAkut = $nilai/count($Katarak);

$Akut = number_format($Katarak,3);

$hasilKatarak = $Katarak *100;

// echo $hasilKatarak;

// echo '<br>';

$_SESSION['Katarak'] = $hasilKatarak;

$nilai = 0;

foreach ($persentase as $value) {

    if (in_array($value, $Blefaritis)) {

        $nilai += 1;

    }else{

        $nilai += 0;

    }

}

$Blefaritis = $nilai/count($Blefaritis);

$Blefaritis = number_format($Blefaritis,3);

$hasilBlefaritis = $Blefaritis *100;
```

```
// echo $hasilBlefaritis;

// echo '<br>';

$_SESSION['Blefaritis'] = $hasilBlefaritis;

$nilai = 0;

foreach ($persentase as $value) {

    if (in_array($value, $buftalmus)) {

        $nilai += 1;

    }else{

        $nilai += 0;

    }

}

$Buftalmus = $nilai/count($buftalmus);

$Buftalmus = number_format($Buftalmus,3);

$hasilBuftalmus = $Buftalmus *100;

// echo $hasilBuftalmus;

// echo '<br>';

$_SESSION['buftalmus'] = $hasilBuftalmus;
```

```

$nilai = 0;

foreach ($persentase as $value) {

    if (in_array($value, $Astigmatisme)) {

        $nilai += 1;

    }else{

        $nilai += 0;

    }

}

$ Astigmatisme = $nilai/count($Astigmatisme);

$ Astigmatisme = number_format($Astigmatisme,3);

$hasil Astigmatisme = $ Astigmatisme *100;

// echo $hasil Astigmatisme;

// echo '<br>';

$_SESSION[' Astigmatisme '] = $hasil Astigmatisme;

$nilai = 0;

foreach ($persentase as $value) {

    if (in_array($value, $ Astigmatisme)) {

```

```
        $nilai += 1;

    }else{

        $nilai += 0;

    }

}

$Glaukoma = $nilai/count($Glaukoma);

$ Glaukoma = number_format($Glaukoma,3);

$hasil Glaukoma = $ Glaukoma *100;

// echo $hasil Glaukoma;

// echo '<br>';

$_SESSION[' Glaukoma '] = $hasil Glaukoma;

$nilai = 0;

foreach ($persentase as $value) {

    if (in_array($value, $ Glaukoma)) {

        $nilai += 1;

    }else{

        $nilai += 0;

    }

}
```

```
    }  
  }  
  
  $ Dakriosistitis = $nilai/count($Dakriosistitis);  
  
  $ Dakriosistitis = number_format($Dakriosistitis,3);  
  
  $hasil Dakriosistitis = $ Dakriosistitis *100;  
  
  // echo $hasil Dakriosistitis;  
  
  // echo '<br>';  
  
  $_SESSION[' Dakriosistitis '] = $hasil Dakriosistitis;  
  
  header('Location:hasil.php');  
  
  }  
  
?>  
  
<br>  
  
</div>  
  
</form>  
  
<div class="col d-none d-sm-block">  
  
  
```

</div>

</div>

</div>

</section>

</body>

<script

src="https://code.jquery.com/jquery-3.4.1.js"

integrity="sha256-

WpOohJOqMqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU="

crossorigin="anonymous"

></script>

<script

src="https://code.jquery.com/jquery-3.4.1.slim.min.js"

integrity="sha384-

J6qa4849bIE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJo

Z+n"

crossorigin="anonymous"

></script>

<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-

Q6E9RHvbIyZFJoft+2mJbHaEWldlV9IOYy5n3zV9zzTtmI3UksdQRVvoxMfoo

Ao"

crossorigin="anonymous"

></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"

integrity="sha384-

wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifw

B6"

crossorigin="anonymous"

></script>

</html>

hasil.php

<?php

```
include 'function.php';

if (isset($_SESSION['role'])) {

    if ($_SESSION['role'] == 0) {

        header("location: indexAdmin.php");

    } else if ($_SESSION['role'] == 2) {

        header("location: indexPakar.php");

    }

} else {

    header("location:index.php");

}

$gejala = mysqli_query($koneksi, "SELECT * FROM gejala");

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<link
```

```
rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
```

```
integrity="sha384-
```

```
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q  
9Ijh"
```

```
crossorigin="anonymous"/>
```

```
<link
```

```
href="https://fonts.googleapis.com/css?family=Poppins:300,400,700&display=sw  
ap"
```

```
rel="stylesheet"/>
```

```
<link rel="stylesheet" href="custom.css" />
```

```
<title>Cek Ginjal Mata!</title>
```

```
</head>
```

```
<body>
```

```
<nav class="navbar py-2 navbar-expand-lg navbar-light">
```

```
<div class="container">

  <a class="navbar-brand" href="#"

  ></a>

  <button

  class="navbar-toggler"

  type="button"

  data-toggle="collapse"

  data-target="#navbarSupportedContent"

  aria-controls="navbarSupportedContent"

  aria-expanded="false"

  aria-label="Toggle navigation"

  >

  <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```
<ul class="navbar-nav ml-auto">
  <li>
    <a class="btn px-2 py-2 btn-success ml-2"
href="function.php?act=ulang" role="button">Cek Ulang</a>
  </li>
  <li>
    <a class="btn px-2 py-2 btn-primary ml-2" href="logout.php"
role="button"
    >Log Out</a>
  </li>
</ul>
</div>
</div>
</nav>
```

```
<section class="hasil mt-4">
```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col align-self-center">
```

```
<h3 class="mb-4">Penyakit yang anda alami :</h3>
```

```
<?php
```

```
    if(isset($_SESSION)) {
```

```
    ?>
```

```
<h5 class="mb-4">
```

```
<div class="py-1">
```

```
<strong>
```

```
    Glaukoma = <?= $_SESSION['glaukoma']; ?>%
```

```
</strong>
```

```
</div>
```

```
<div class="py-1">
```

```
<strong>
```

```
    Glaukoma = <?= $_SESSION[' Glaukoma  ']; ?>%
```

```
</strong>
```

</div>

<div class="py-1">

<strong>

Blefaritis = <?=\$\_SESSION['Blefaritis']; ?>%

</strong>

</div>

<div class="py-1">

<strong>

buftalmus = <?=\$\_SESSION['buftalmus']; ?>%

</strong>

</div>

<div class="py-1">

<strong>

Dakriosistitis = <?=\$\_SESSION['dakriosistitis']; ?>%

</strong>

</div>

<div class="py-1">

```
<strong>
```

```
buftalmus = <?=$_SESSION['buftalmus']; ?>%
```

```
</strong>
```

```
</div>
```

```
</h5>
```

```
<?php } ?>
```

```
<h3 class="mb-4">Solusi untuk penyakit anda adalah : </h3>
```

```
<form action="" method="post" enctype="multipart/form-data"
role="form">
```

```
<?php
```

```
function maximum($a, $b, $c, $d, $e, $f)
```

```
{
```

```
    $max = $a;
```

```
$kode = 1;

if ($b > $max) {

    $max = $b;

    $kode = 2;

}

if ($c > $max) {

    $max = $c;

    $kode = 3;

}

if ($d > $max) {

    $max = $d;

    $kode = 4;

}

if ($e > $max) {

    $max = $e;

    $kode = 5;

}
```

```

        if ($f > $max) {

            $max = $f;

            $kode = 6;

        }

        return $kode;

    }

    $id_penyakit      =      maximum($_SESSION['Katarak'],
$_SESSION['blefaritis'],      $_SESSION['dakriosistitis'],
$_SESSION['buftalmus'],      $_SESSION['Glaukoma'],
$_SESSION['Ablasio']);

    $query = "SELECT * FROM solusi WHERE id_penyakit =
'Sid_penyakit'";

    $data = mysqli_query($koneksi, $query);

    while ($row = mysqli_fetch_array($data)) {

        echo '<p>' . $row['solusi'] . '</p>';

    }

```

?>

</div>

</form>

<div class="col d-none d-sm-block">



</div>

</div>

</div>

</section>

</body>

<script

src="https://code.jquery.com/jquery-3.4.1.js"

integrity="sha256-

WpOohJOqMqyyKL9FccASB9O0KwACQJpFTUBLTYOVvVU="

crossorigin="anonymous"

></script>

<script

src="https://code.jquery.com/jquery-3.4.1.slim.min.js"

integrity="sha384-

J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJo

Z+n"

crossorigin="anonymous"

></script>

<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-

Q6E9RHvbIyZFJoft+2mJbHaEWldlIvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfoo

Ao"

crossorigin="anonymous"

></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"

integrity="sha384-  
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifw  
B6"

crossorigin="anonymous"

></script>

</html>