

**PERANCANGAN SISTEM PAKAR DIAGNOSA  
INFEKSI SALURAN PERNAFASAN AKUT  
MENGUNAKAN METODE FORWARD  
CHAINING BERBASIS WEB**

**SKRIPSI**



**Oleh:  
Harianto Zheng  
140210054**

**PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS PUTERA BATAM  
2018**

**PERANCANGAN SISTEM PAKAR DIAGNOSA  
INFEKSI SALURAN PERNAFASAN AKUT  
MENGUNAKAN METODE FORWARD  
CHAINING BERBASIS WEB**

**SKRIPSI**  
**Untuk memenuhi salah satu syarat**  
**Guna memperoleh gelar Sarjana**



**Oleh**  
**Harianto Zheng**  
**140210054**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**UNIVERSITAS PUTERA BATAM**  
**2018**

## PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 04 Agustus 2018  
Yang membuat pernyataan,

Materai Rp 6.000

Harianto Zheng  
140210054

**PERANCANGAN SISTEM PAKAR DIAGNOSA  
INFEKSI SALURAN PERNAFASAN AKUT  
MENGUNAKAN METODE FORWARD  
CHAINING BERBASIS WEB**

**Oleh  
Harianto Zheng  
140210054**

**SKRIPSI  
Untuk memenuhi salah satu syarat guna  
memperoleh gelar Sarjana**

**Telah disetujui oleh Pembimbing pada tanggal  
seperti tertera di bawah ini**

**Batam, 04 Agustus 2018**

**Nia Ekawati, S.Kom., M.SI.  
Pembimbing**

## ABSTRAK

Infeksi saluran pernafasan akut biasanya disingkat dengan ISPA, penyakit ini umumnya disebabkan oleh virus dan bakteri, penyakit infeksi saluran pernafasan akut dapat menyerang berbagai jenis usia dari balita hingga dewasa dan juga termasuk penyakit yang membutuhkan penanganan cepat, masalah seperti masyarakat tidak dapat menanganin penyakit infeksi saluran pernafasan akut dengan cepat dapat memungkinkan menyebabkan kematian. Kurangnya penyebaran pengetahuan masyarakat mengenai infeksi saluran pernafasan akut menjadi salah satu faktor penting yang menyebabkan lambatnya penanganan terhadap penyakit infeksi saluran pernafasan akut, maka diperlukannya sistem pakar diagnosa infeksi saluran pernafasan akut untuk memberi pengetahuan tentang penyakit infeksi saluran pernafasan akut kepada masyarakat umum. Penelitian ini merancang sistem pakar berbasis *web* menggunakan metode *forward chaining* yang dimaksudkan untuk membantu masyarakat dalam mendiagnosa penyakit infeksi saluran pernafasan akut dan juga memberikan saran-saran pengobatannya. Sistem pakar diagnosa penyakit infeksi saluran pernafasan akut berbasis *web* yang telah dikembangkan mempunyai keunggulan seperti kemudahan pemakaian dan kemudahan akses. Dengan fitur yang berbasis web yang dimiliki, sistem pakar untuk diagnosa penyakit infeksi saluran pernafasan akut ini dapat diakses oleh masyarakat dimanapun, sehingga sistem ini mampu mengatasi persoalan keterbatasan jumlah dokter dalam mendiagnosa penyakit infeksi saluran pernafasan akut. Aplikasi sistem pakar diagnosa infeksi saluran pernafasan akut ini mampu menganalisis jenis penyakit infeksi saluran pernafasan akut berdasarkan gejala-gejala yang dipilih oleh *user*.

Kata kunci: Sistem Pakar, Diagnosa Penyakit, Infeksi Saluran Pernafasan Akut, *Forward Chaining*, *Web*

## **ABSTRACT**

*Acute Respiratory Infection, this disease is commonly caused by virus or bacteria. Acute respiratory infection can affect various type of age from toddler to adult and also a considered type of disease which require immediate treatment. Problem such as people's inability to cure acute respiratory infection immediately might possibly cause fatality for the patient. Lack of awareness among society regarding acute respiratory infection is one of the reasons why acute respiratory infection can not be treated immediately. Therefore, expert system of acute respiratory infection diagnosis is needed to educate common people about this disease. This study designs web-based expert system using forward chaining method with the purpose to help people in diagnosing acute respiratory infection and to recommend the treatment. Currently developed web-based expert system of acute respiratory infection diagnosis has some advantages such as easy to use and easy to access. With web-based's features, this expert system of acute respiratory infection diagnosis can be accessed by people anywhere, thus making this system able to overcome problem of under-staffed doctors to diagnose the disease. This application of expert system of acute respiratory infection able to analyze type of acute respiratory infection based on symptoms chosen by the user.*

*Keywords: Expert System, Diagnosing, Acute Respiratory Infection, Forward Chaining, Web*

## KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Dekan Fakultas Teknik dan Komputer Universitas Putera Batam.
3. Ketua Program Studi Teknik Informatika Universitas Putera Batam.
4. Ibu Nia Ekawati, S.Kom., M.SI, selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
5. Dosen dan Staff Universitas Putera Batam.
6. Bapak dr.H.Iwan Mulyana kepala Puskesmas Kijang yang telah memberikan dukungannya.
7. Keluarga yang selalu memberikan doa dan motivasi yang baik.

8. Rekan-rekan mahasiswa/i Universitas Putera Batam yang turut memberikan doa dan dukungannya.
9. Mitra kerja yang selalu memberikan masukan yang berguna untuk penelitian ini.
10. Serta pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu mencurahkan hidayah serta taufikNya, Amin.

Batam, Agustus 2018

Harianto Zheng



## DAFTAR ISI

	Halaman
<b>HALAMAN SAMBUNG DEPAN</b>	
<b>HALAMAN JUDUL</b> .....	ii
<b>HALAMAN PERNYATAAN</b> .....	iii
<b>HALAMAN PENGESAHAN</b> .....	iv
<b>ABSTRAK</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>KATA PENGANTAR</b> .....	vii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR TABEL</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR LAMPIRAN</b> .....	xiv
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang Penelitian .....	1
1.2 Identifikasi Masalah .....	4
1.3 Pembatasan Masalah .....	4
1.4 Perumusan Masalah.....	5
1.5 Tujuan Penelitian.....	6
1.6 Manfaat Penelitian.....	6
1.6.1 Manfaat Teoritis .....	6
1.6.2 Manfaat Praktis.....	7
<b>BAB II KAJIAN PUSTAKA</b> .....	8
2.1 Teori Dasar .....	8
2.1.1 Kecerdasan Buatan atau <i>Artificial Intelligence</i> (AI) .....	8
2.1.2 Web .....	24
2.1.3 <i>Database</i> (basis data) .....	25
2.1.4 Validasi Sistem.....	25
2.1.5 Infeksi Saluran Pernafasan Akut .....	26
2.2 Variabel Penelitian .....	27
2.2.1 Faringitis.....	28
2.2.2 Laringitis .....	28
2.2.3 Tonsilitis.....	29
2.3 <i>Software</i> Pendukung.....	29
2.3.1 XAMPP .....	29
2.3.2 phpMyAdmin .....	30
2.3.3 <i>Hypertext Preprocessor</i> (PHP) .....	30
2.3.4 HTML ( <i>Hyper Text Markup Language</i> ) .....	32
2.3.5 CSS ( <i>Cascading Style Sheet</i> ).....	32
2.3.6 JavaScript .....	33
2.3.7 jQuery .....	33
2.3.8 MySQL dan SQL.....	34
2.3.9 Sublime Text .....	35
2.3.10 Laravel.....	35

2.3.11	Twitter Bootstrap.....	36
2.3.12	StarUML.....	36
2.4	Penelitian Terdahulu.....	43
2.5	Kerangka Pemikiran .....	48
<b>BAB III METODE PENELITIAN .....</b>		<b>50</b>
3.1	Desain Penelitian .....	50
3.2	Pengumpulan Data .....	55
3.3	Operasional Variabel.....	58
3.4	Metode Perancangan Sistem.....	59
3.4.1	Desain Basis Pengetahuan.....	60
3.4.2	Struktur Kontrol (Mesin Inferensi) .....	69
3.4.3	Desain UML ( <i>Unified Modeling Language</i> ).....	70
3.4.4	Desain <i>Database</i> .....	87
3.4.5	Prototype .....	88
3.5	Lokasi dan Jadwal Penelitian .....	97
3.5.1	Lokasi.....	97
3.5.2	Jadwal Penelitian.....	98
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>99</b>
4.1	Hasil Penelitian.....	99
4.2	Pembahasan .....	109
4.2.1	Pengujian Validasi Sistem.....	110
4.2.2	Pengujian dengan Pakar .....	118
<b>BAB V SIMPULAN DAN SARAN.....</b>		<b>121</b>
5.1	Simpulan.....	121
5.2	Saran .....	122
<b>DAFTAR PUSTAKA</b>		
<b>RIWAYAT HIDUP</b>		
<b>SURAT KETERANGAN PENELITIAN</b>		
<b>SURAT BALASAN KETERANGAN PENELITIAN</b>		
<b>LAMPIRAN</b>		

## DAFTAR TABEL

	Halaman
<b>Tabel 2.1</b> Tabel Keputusan.....	19
<b>Tabel 2.2</b> Alternatif Tabel Keputusan .....	20
<b>Tabel 2.3</b> Simbol <i>Use Case Diagram</i> .....	38
<b>Tabel 2.4</b> Simbol <i>Activity Diagram</i> .....	39
<b>Tabel 2.5</b> Simbol <i>Sequence Diagram</i> .....	41
<b>Tabel 2.6</b> Simbol <i>Class Diagram</i> .....	42
<b>Tabel 3.1</b> Variabel dan Indikator.....	59
<b>Tabel 3.2</b> Data Indikator.....	61
<b>Tabel 3.3</b> Indikator Penyakit Infeksi Saluran Pernafasan Akut .....	61
<b>Tabel 3.4</b> Gejala Penyakit Infeksi Saluran Pernafasan Akut.....	63
<b>Tabel 3.5</b> Gejala <i>Faringitis</i> .....	64
<b>Tabel 3.6</b> Gejala <i>Laringitis</i> .....	64
<b>Tabel 3.7</b> Gejala <i>Tonsilitis</i> .....	65
<b>Tabel 3.8</b> Indikator dan Gejala Penyakit Infeksi Saluran Pernafasan Akut .....	65
<b>Tabel 3.9</b> Tabel Keputusan.....	66
<b>Tabel 3.10</b> Jadwal Penelitian.....	98
<b>Tabel 4.1</b> Pengujian Menu <i>Home</i> .....	110
<b>Tabel 4.2</b> Pengujian Menu Tentang Kami.....	111
<b>Tabel 4.3</b> Pengujian Menu Tentang ISPA.....	111
<b>Tabel 4.4</b> Pengujian Menu Diagnosa .....	112
<b>Tabel 4.5</b> Pengujian Menu <i>Register</i> .....	112
<b>Tabel 4.6</b> Pengujian Menu <i>Login</i> .....	112
<b>Tabel 4.7</b> Pengujian Menu Daftar Akun .....	113
<b>Tabel 4.8</b> Pengujian Menu Daftar Gejala .....	113
<b>Tabel 4.9</b> Pengujian Menu Daftar Penyakit .....	114
<b>Tabel 4.10</b> Pengujian Menu Daftar Relasi .....	115
<b>Tabel 4.11</b> Pengujian Menu Daftar Riwayat Diagnosa .....	115
<b>Tabel 4.12</b> Pengujian Menu <i>Admin Logout</i> .....	116
<b>Tabel 4.13</b> Pengujian Menu Diagnosa.....	116
<b>Tabel 4.14</b> Pengujian Menu Riwayat Diagnosa .....	117
<b>Tabel 4.15</b> Pengujian Menu Hasil Diagnosa .....	117
<b>Tabel 4.16</b> Pengujian Menu <i>User Logout</i> .....	118
<b>Tabel 4.17</b> Tabel Hasil Diagnosa Pakar dan Diagnosa Sistem .....	119

## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b> Struktur Sistem Pakar .....	14
<b>Gambar 2.2</b> Pohon Keputusan .....	19
<b>Gambar 2.3</b> Alternatif Pohon Keputusan .....	21
<b>Gambar 2.4</b> Infeksi Saluran Pernafasan Akut .....	27
<b>Gambar 2.5</b> Kerangka Pemikiran .....	49
<b>Gambar 3.1</b> Desain Penelitian .....	51
<b>Gambar 3.2</b> Pohon Keputusan .....	67
<b>Gambar 3.3</b> <i>Flowchart</i> .....	70
<b>Gambar 3.4</b> <i>Use Case Diagram</i> .....	71
<b>Gambar 3.5</b> <i>Activity Diagram Admin Login</i> .....	72
<b>Gambar 3.6</b> <i>Activity Diagram Admin Mengelola Daftar Akun</i> .....	73
<b>Gambar 3.7</b> <i>Activity Diagram Admin Mengelola Daftar Gejala</i> .....	74
<b>Gambar 3.8</b> <i>Activity Diagram Admin Mengelola Daftar Penyakit</i> .....	74
<b>Gambar 3.9</b> <i>Activity Diagram Admin Mengelola Daftar Relasi</i> .....	75
<b>Gambar 3.10</b> <i>Activity Diagram Admin Mengelola Daftar Riwayat Diagnosa</i> ....	76
<b>Gambar 3.11</b> <i>Activity Diagram Admin Log out</i> .....	76
<b>Gambar 3.12</b> <i>Activity Diagram User Registrasi</i> .....	77
<b>Gambar 3.13</b> <i>Activity Diagram User Login</i> .....	78
<b>Gambar 3.14</b> <i>Activity Diagram User Diagnosa</i> .....	78
<b>Gambar 3.15</b> <i>Activity Diagram User Riwayat Diagnosa</i> .....	79
<b>Gambar 3.16</b> <i>Activity Diagram User Log out</i> .....	79
<b>Gambar 3.17</b> <i>Sequence Diagram Amin Login</i> .....	80
<b>Gambar 3.18</b> <i>Sequence Diagram Admin Mengelola Daftar Akun</i> .....	81
<b>Gambar 3.19</b> <i>Sequence Diagram Admin Mengelola Daftar Gejala</i> .....	81
<b>Gambar 3.20</b> <i>Sequence Diagram Admin Mengelola Daftar Penyakit</i> .....	82
<b>Gambar 3.21</b> <i>Sequence Diagram Admin Mengelola Daftar Relasi</i> .....	83
<b>Gambar 3.22</b> <i>Sequence Diagram Admin Mengelola Daftar Riwayat Diagnosa</i> ..	83
<b>Gambar 3.23</b> <i>Sequence Diagram Admin Log out</i> .....	84
<b>Gambar 3.24</b> <i>Sequence Diagram User Registrasi</i> .....	84
<b>Gambar 3.25</b> <i>Sequence Diagram User Login</i> .....	85
<b>Gambar 3.26</b> <i>Sequence Diagram User Diagnosa</i> .....	85
<b>Gambar 3.27</b> <i>Sequence Diagram User Riwayat Diagnosa</i> .....	86
<b>Gambar 3.28</b> <i>Sequence Diagram User Log out</i> .....	86
<b>Gambar 3.29</b> <i>Class Diagram</i> .....	87
<b>Gambar 3.30</b> <i>Entity Relationship Diagram</i> .....	88
<b>Gambar 3.31</b> Tampilan Menu Utama .....	89
<b>Gambar 3.32</b> Tampilan <i>Login Admin</i> .....	90
<b>Gambar 3.33</b> Tampilan Utama <i>Admin</i> .....	90
<b>Gambar 3.34</b> Tampilan Menu Daftar Akun.....	91
<b>Gambar 3.35</b> Tampilan Menu Daftar Gejala .....	91
<b>Gambar 3.36</b> Tampilan Menu Daftar Penyakit .....	92

<b>Gambar 3.37</b> Tampilan Menu Daftar Relasi .....	92
<b>Gambar 3.38</b> Tampilan Menu Daftar Riwayat Diagnosa.....	93
<b>Gambar 3.39</b> Tampilan Registrasi <i>User</i> .....	94
<b>Gambar 3.40</b> Tampilan <i>Login User</i> .....	94
<b>Gambar 3.41</b> Tampilan Utama <i>User</i> .....	95
<b>Gambar 3.42</b> Tampilan Diagnosa.....	96
<b>Gambar 3.43</b> Tampilan Riwayat Diagnosa.....	96
<b>Gambar 3.44</b> Tampilan Hasil Diagnosa .....	97
<b>Gambar 4.1</b> Menu Utama .....	100
<b>Gambar 4.2</b> <i>Login Admin</i> .....	101
<b>Gambar 4.3</b> Menu Utama <i>Admin</i> .....	102
<b>Gambar 4.4</b> Daftar Akun .....	102
<b>Gambar 4.5</b> Daftar Gejala.....	103
<b>Gambar 4.6</b> Daftar Penyakit.....	104
<b>Gambar 4.7</b> Daftar Relasi .....	105
<b>Gambar 4.8</b> Daftar Riwayat Diagnosa.....	105
<b>Gambar 4.9</b> <i>User</i> Registrasi.....	106
<b>Gambar 4.10</b> <i>User Login</i> .....	106
<b>Gambar 4.11</b> Menu Utama <i>User</i> .....	107
<b>Gambar 4.12</b> Diagnosa .....	108
<b>Gambar 4.13</b> Riwayat Diagnosa.....	108
<b>Gambar 4.14</b> Hasil Diagnosa.....	109

## **DAFTAR LAMPIRAN**

LAMPIRAN I FORM WAWANCARA  
LAMPIRAN II FOTO WAWANCARA  
LAMPIRAN III DATA PENYAKIT

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Penelitian**

Seorang manusia memiliki bermacam-macam organ dan sistem jaringan didalam tubuhnya, dimana masing-masing sistem tersebut memiliki peranan dan dan fungsi serta manfaat tertentu bagi manusia, salah satunya sistem pernafasan.

Berdasarkan penelitian (Mair & Supriadi, 2017:20) diperoleh fakta bernafas merupakan salah satu kebutuhan paling penting untuk menunjang kehidupan manusia dalam melakukan aktivitasnya. Bernafas dibutuhkan oleh manusia dalam setiap waktu untuk menyalurkan oksigen di dalam tubuh. Tanpa oksigen manusia tidak akan bisa bertahan hidup. Sistem pernafasan merupakan organisasi organ yang berfungsi untuk bernafas, hubungan kerja sistem ini mencakup hidung, tenggorokan, cabang batang tenggorok, dan paru-paru.

Infeksi saluran pernafasan akut biasanya disingkat dengan ISPA, penyakit ini umumnya disebabkan oleh virus dan bakteri, penyakit ISPA dapat menyerang berbagai jenis usia dari balita hingga dewasa dan juga termasuk penyakit yang membutuhkan penanganan cepat, masalah seperti masyarakat tidak dapat menanganin penyakit ISPA dengan cepat dapat memungkinkan menyebabkan kematian. Kurangnya penyebaran pengetahuan masyarakat kota Kijang mengenai ISPA menjadi salah satu faktor penting yang menyebabkan lambatnya penanganan terhadap penyakit ISPA, maka diperlukannya sistem pakar diagnosa ISPA untuk

memberi pengetahuan tentang penyakit ISPA kepada masyarakat umum. Dengan merancang suatu sistem pakar diagnosa ISPA dapat mempermudah penderitanya untuk melakukan diagnosa cepat serta dapat juga mempercepat proses diagnosa oleh tenaga medis. Berdasarkan data (Meara et al., 2015:2) ISPA masuk dalam 5 penyakit penyebab kematian tertinggi pada balita tepatnya berada di posisi kedua dengan persentase 16% pada tahun 2013 di Indonesia.

Infeksi saluran pernafasan akut di Indonesia selalu menempati urutan pertama penyebab kematian pada kelompok bayi dan sering menempati urutan pertama angka kesakitan balita. Selain itu infeksi saluran pernafasan akut juga sering berada pada daftar 10 penyakit terbanyak di rumah sakit. Episode penyakit batuk, pilek pada balita di Indonesia diperkirakan 3-6 kali pertahun, artinya seorang balita rata-rata mendapat serangan batuk, pilek sebanyak 3-6 kali setahun (Agrina, Suyanto, 2014:116).

Sistem pakar sebagai salah satu aplikasi yang paling umum dari kecerdasan buatan, sistem pakar mensimulasikan keputusan dan tindakan seseorang yang memiliki fakta dan pengalaman khusus dibidang tertentu. Biasanya, sistem pakar mengandung basis pengetahuan yang berisi akumulasi pengalaman dan seperangkat aturan untuk menerapkan basis pengetahuan untuk setiap situasi tertentu.

Berdasarkan penelitian (Nurhayati, 2013:110) Sistem pakar adalah sistem yang mampu menirukan penalaran seorang pakar agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Pengetahuan yang disimpan didalam sistem pakar umumnya diambil dari seorang manusia yang pakar dalam masalah tersebut. Peran penting seorang pakar dapat digantikan oleh



program komputer yang pada prinsip kerjanya untuk memberikan solusi yang pasti seperti yang biasa dilakukan oleh pakar. Sistem pakar biasanya digunakan untuk konsultasi, analisis, diagnosis dan membantu mengambil keputusan.

Metode pelacakan ke depan atau yang biasanya disebut dengan metode *forward chaining* menggunakan teknik penalaran yang dimulai dari sekumpulan fakta-fakta dengan mencari kaidah yang cocok dengan dugaan atau hipotesa yang untuk mengambil suatu kesimpulan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju.

Pelacakan ke depan atau *forward chaining* adalah pendekatan yang dimotori data (*data-driven*) Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan, mencari fakta yang sesuai dengan bagian *if* dari aturan *if-then* (Bahtiar & Wibawa, 2013:3).

*Website* atau yang biasanya disingkat web dapat diartikan sebagai sekumpulan halaman informasi yang didistribusikan melalui jaringan internet sehingga dapat diakses diseluruh dunia selama terkoneksi dengan jaringan internet tanpa terbatas ruang dan waktu, selain itu *website* juga tidak memerlukan *resource* komputer yang tinggi untuk menyampaikan informasi mengingat kompatibilitasnya yang tinggi.

Berdasarkan penelitian (Hastanti, Purnama, & Wardati, 2015:1) *Website* merupakan kumpulan halaman-halaman yang berisi informasi yang disimpan

diinternet yang bisa diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa mengakses internet itu sendiri seperti komputer. Definisi kata web sebenarnya penyederhanaan dari sebuah istilah dalam dunia komputer yaitu *World Wide Web* yang merupakan bagian dari teknologi internet.

Berdasarkan uraian yang telah dijelaskan diatas, maka penelitian mengambil judul **“PERANCANGAN SISTEM PAKAR DIAGNOSA INFEKSI SALURAN PERNAFASAN AKUT MENGGUNAKAN METODE FORWARD CHAINING BERBASIS WEB”**

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang penelitian diatas, ada beberapa masalah yang teridentifikasi, yaitu sebagai berikut:

1. Kurangnya penyebaran pengetahuan masyarakat mengenai infeksi saluran pernafasan akut.
2. Masyarakat tidak dapat menangani penyakit infeksi saluran pernafasan akut dengan cepat sehingga dapat menyebabkan kematian.

## **1.3 Pembatasan Masalah**

Untuk mencegah meluasnya ruang lingkup pembahasan, maka penulis memberikan pembatasan masalah dengan pernyataan-pernyataan sebagai berikut:

1. Penelitian ini hanya membahas penyakit infeksi saluran pernafasan akut yaitu *Faringitis, Laringitis* dan *Tonsilitis*.
2. Penelitian ini dilakukan di Puskesmas Kijang Kota yang beralamat di Jl.Barek Motor, Kijang Kota, Bintan Timur, Kabupaten Bintan, Kepulauan Riau.
3. Pakar dalam penelitian ialah dr.H.Iwan Mulyana selaku kepala puskesmas dan dokter umum.
4. Penelitian ini menggunakan sistem pakar dengan metode *forward chaining*.
5. Penelitian ini berbasis web dengan menggunakan bahasa pemrograman PHP.
6. Penelitian ini menggunakan *framework* bahasa pemrograman PHP yaitu Laravel.
7. Penelitian ini menggunakan MySQL sebagai media pengolahan *database*.
8. Penelitian menggunakan *text editor* Sublime Text 3 sebagai media penulisan program.
9. Penelitian ini menggunakan Domainsia sebagai media *hosting*.
10. Penelusuran penyakit berdasarkan pengetahuan dokter umum di Puskesmas Kijang.

#### **1.4 Perumusan Masalah**

Berdasarkan identifikasi masalah dijabarkan diatas, maka dapat dirumuskan permasalahan pada penelitian ini yaitu:

1. Bagaimana perancangan sistem pakar mendiagnosa infeksi saluran pernafasan akut menggunakan metode *forward chaining* berbasis web?

2. Bagaimana implementasi sistem pakar mendiagnosa infeksi saluran pernafasan akut menggunakan metode *forward chaining* berbasis *web*?

### **1.5 Tujuan Penelitian**

Berdasarkan uraian latar belakang masalah, maka tujuan penyusunan penelitian ini adalah:

1. Untuk merancang suatu sistem pakar mendiagnosa infeksi saluran pernafasan akut menggunakan metode *forward chaining* berbasis *web*.
2. Untuk mengimplementasi suatu sistem pakar mendiagnosa infeksi saluran pernafasan akut menggunakan metode *forward chaining* berbasis *web*.

### **1.6 Manfaat Penelitian**

#### **1.6.1 Manfaat Teoritis**

Manfaat teoritis yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Meningkatkan pengetahuan dan pemahaman mengenai kecerdasan buatan.
2. Meningkatkan pengetahuan dan pemahaman mengenai implementasi sistem pakar.
3. Meningkatkan pengetahuan dan pemahaman mengenai penyakit infeksi saluran pernafasan akut.

### **1.6.2 Manfaat Praktis**

Manfaat praktis yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Diharapkan dapat menambah kesadaran untuk masyarakat agar lebih aktif dalam menjaga kesehatannya.
2. Diharapkan dapat mempermudah masyarakat dalam mendiagnosa penyakit infeksi saluran pernafasan akut.

## **BAB II KAJIAN PUSTAKA**

### **2.1 Teori Dasar**

Deskripsi teori paling tidak berisi tentang penjelasan terhadap variabel-variabel yang diteliti melalui pendefinisian, dan uraian yang lengkap dan mendalam dari berbagai referensi, sehingga ruang lingkup, kedudukan dan prediksi terhadap hubungan antara variabel yang akan diteliti menjadi lebih jelas dan terarah (Sugiyono, 2014:58).

Di bab ini akan dijelaskan secara singkat beberapa teori dasar mengenai kecerdasan buatan atau *Artificial Intelligence (AI)* dan juga bagian-bagiannya seperti logika *fuzzy (fuzzy logic)*, jaringan syaraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*) dan juga metode-metodenya.

#### **2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)***

Menurut (Budihartono & Suhartono, 2014:2-3) kecerdasan buatan atau *artificial intelligence (AI)* merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa akan datang. Mulai dari yang paling umum hingga yang khusus. Dari *learning* atau *perception* hingga pada permainan catur, pembuktian teori matematika, menulis puisi, mengemudikan mobil, dan melakukan diagnosis penyakit. Pada masa sekarang, perhatian difokuskan pada kemampuan

komputer untuk mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia.

Biasanya konsep suatu kecerdasan buatan dilakukan dengan mengikuti/mencontoh karakteristik dan analogi berpikir dari kecerdasan manusia dan menerapkannya kedalam bentuk algoritma yang dapat dikenali oleh komputer. Untuk membuat suatu sistem menjadi cerdas yang berarti memiliki pengetahuan, pengalaman dan penalaran untuk membuat suatu keputusan dan kemudian mengambil tindakan maka sistem tersebut harus diberi kemampuan untuk menalar dan bekal pengetahuan. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan.

#### **2.1.1.1 Logika *Fuzzy* (*Fuzzy Logic*)**

Logika *Fuzzy* (*Fuzzy Logic*) pertama kali diperkenalkan oleh Jan Lukasiewicz pada tahun 1920-an sebagai teori kemungkinan. Logika kemungkinan ini memperluas jangkauan dari nilai kebenaran untuk semua bilangan riil pada interval antara 0 dan 1. *Fuzzy logic* banyak digunakan karena mirip dengan cara berpikir manusia. Sistem *fuzzy logic* dapat merepresentasikan pengetahuan manusia dalam bentuk matematis dengan menyerupai cara berpikir manusia (Budihartono & Suhartono, 2014:151).

Berdasarkan penelitian (Buana, 2014:138-139) Logika *fuzzy* merupakan suatu cara untuk memetakan suatu ruang masukan ke dalam suatu ruang keluaran. Dalam teori logika *fuzzy* dikenal himpunan *fuzzy* (*fuzzy set*). Merupakan pengelompokan sesuatu berdasarkan variabel bahasa yang dinyatakan dalam fungsi keanggotaan (*membership function*).

Menurut (Budihartono & Suhartono, 2014:152) Beberapa keuntungan yang dapat diambil dalam menggunakan logika *fuzzy* untuk memecahkan suatu masalah yaitu :

1. Konsep *fuzzy logic* mudah dimengerti
2. *Fuzzy logic* sangat fleksibel
3. *Fuzzy logic* memiliki toleransi terhadap data-data yang tidak tepat.
4. *Fuzzy logic* mampu memodelkan fungsi-fungsi non-linear yang sangat kompleks
5. *Fuzzy logic* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui pelatihan.
6. *Fuzzy logic* dapat bekerjasama dengan teknik-teknik kendali secara konvensional
7. *Fuzzy logic* didasarkan pada bahasa alami.

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah (Buana, 2014:139):

1. Metode Tsukamoto

Setiap konsekuen pada aturan yang berbentuk *if-then* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton.



## 2. Metode Mamdani

Sering dikenal dengan nama Metode *Max-Min*.

## 3. Metode Sugeno

Penalaran dengan metode sugeno hampir sama dengan penalaran mamdani, hanya saja output (konsekuen) sistem tidak berupa himpunan *fuzzy*.

## 4. Model Tahani

Analisis data dilakukan untuk mengolah data yang telah di dapat dan mengelompokan data sesuai dengan kebutuhan perancangan.

### **2.1.1.2 Jaringan Saraf Tiruan (*Artificial Neural Network*)**

Jaringan saraf tiruan merupakan merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut *neuron*, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektifitas yang sangat tinggi (Suyanto, 2014:169-170).

Jaringan syaraf tiruan bisa digunakan untuk menyelesaikan berbagai masalah mulai dari klasifikasi, optimasi, kompresi, peramalan (*forecasting*), sistem kontrol, sistem pendeteksian kecurangan (*intrusion detection systems*), dan sebagainya. Pada umumnya jaringan syaraf tiruan sangat sesuai untuk permasalahan yang bernilai kontinyu, seperti pengenalan tulisan tangan, pengenalan wajah, pengenalan

suara, dan sebagainya. Tetapi kita bisa juga menggunakan jaringan syaraf tiruan untuk masalah yang bernilai diskrit (Suyanto, 2014:189).

Berdasarkan penelitian (Sudarsono, 2016:62) jaringan syaraf tiruan terdiri dari beberapa *neuron*, dan terdapat hubungan antara *neuron-neuron* tersebut. Pada jaringan syaraf tiruan, *neuron-neuron* akan dikumpulkan dalam lapisan–lapisan yang disebut dengan lapisan *neuron*. Biasanya *neuron* pada satu lapisan akan dihubungkan dengan lapisan sebelum atau sesudahnya terkecuali lapisan masukan dan lapisan keluaran. Informasi yang diberikan pada jaringan syaraf akan dirambatkan dari lapisan ke lapisan, melalui dari lapisan masukan sampai lapisan keluaran melalui lapisan tersembunyi.

### **2.1.1.3 Sistem Pakar (*Expert System*)**

Secara umum, sistem pakar ialah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer dan dirancang untuk memodelkan kemampuan menyelesaikan masalah layaknya seorang pakar pada bidang tertentu. Dengan menggunakan sistem pakar, orang awan pun dapat menyelesaikan atau mendapatkan informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para pakar dibidangnya.

Pengertian sistem pakar adalah program komputer yang menyimulasi penilaian dan perilaku manusia atau organisasi yang memiliki pengetahuan dan pengalaman ahli dalam bidang tertentu. Biasanya sistem seperti ini berisi basis pengetahuan yang berisi akumulasi pengalaman dan satu set aturan untuk

menerapkan pengetahuan dasar untuk setiap situasi tertentu. Sistem pakar yang canggih dapat ditingkatkan dengan cara menambahkan basis pengetahuan atau set aturan. Diantara banyak pakar yang ada, yang terkenal adalah aplikasi bermain catur dan sistem diagnosis medis (Budihartono & Suhartono, 2014:132).

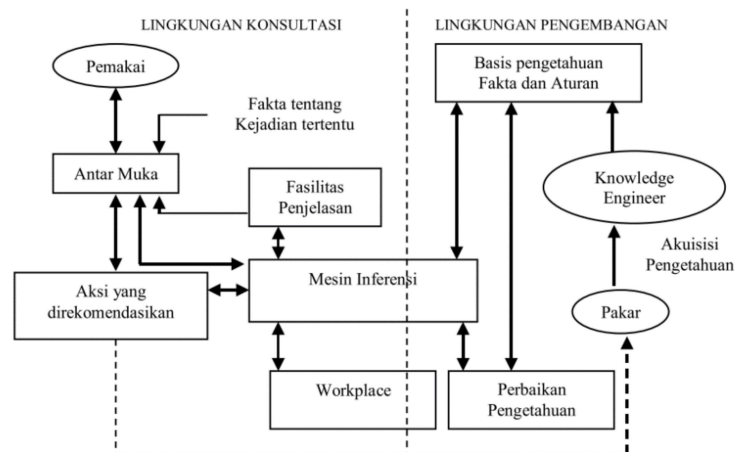
Menurut penelitian (Nurhayati, 2013:110) sistem pakar adalah sistem yang mampu menirukan penalaran seorang pakar agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Pengetahuan yang disimpan didalam sistem pakar umumnya diambil dari seorang manusia yang pakar dalam masalah tersebut. Peran penting seorang pakar dapat digantikan oleh program komputer yang pada prinsip kerjanya untuk memberikan solusi yang pasti seperti yang biasa dilakukan oleh pakar. Sistem pakar biasanya digunakan untuk konsultasi, analisis, diagnosis dan membantu mengambil keputusan.

Menurut (Budihartono & Suhartono, 2014:134) sistem pakar banyak digunakan pada aplikasi terkini dan kompleks karena:

1. Sistem pakar dapat bertindak sebagai konsultan, instruktur, atau pasangan/rekan.
2. Meningkatkan *availability* atau kepakaran tersedia pada semua perangkat komputer.
3. Mengurangi bahaya
4. Permanen
5. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluas sesuai kebutuhan. Program konvensional harus “lengkap” sebelum mereka dapat digunakan.

6. *Database* yang cerdas, sistem pakar dapat digunakan untuk mengakses *database* secara cerdas, misalnya data mining.

Secara umum ada dua bagian penting atau utama yang dibutuhkan dalam membuat aplikasi kecerdasan buatan yaitu basis pengetahuan (*knowledge base*) dan mesin inferensi (*Inference Engine*). Istilah sistem pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant*. Tujuan utama sistem pakar bukan untuk menggantikan kedudukan seorang ahli maupun pakar, tetapi untuk memasyarakatkan pengetahuan dan pengalaman pakar-pakar yang ahli di bidangnya. Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*) (Mukhtar & Samsudin, 2014:22).



**Gambar 2.1** Struktur Sistem Pakar

Menurut (Hartati & Iswanti, 2008:3-6) sistem pakar sebagai suatu program yang berfungsi meniru seorang pakar harus mampu melakukan hal-hal yang dapat dilakukan oleh seorang pakar. Menurut (Giarratano dan Riley, 2005) untuk membangun suatu sistem seperti itu maka dibutuhkan komponen-komponen yang harus dimiliki seperti:

1. Antar Muka Pengguna (*User Interface*)

Sistem pakar harus menyediakan pendukung yang diperlukan oleh pemakainya yang tidak mengerti masalah teknis agar sistem pakar dapat dijadikan sebagai pengganti seorang pakar dalam suatu kondisi tertentu. Antar muka ialah media komunikasi antara sistem dan pemakai sistem pakar, antar muka yang efektif dan ramah pengguna (*user-friendly*) merupakan hal yang penting terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

2. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan adalah komponen yang berisi sekumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Basis pengetahuan bersifat dinamis yaitu selalu berkembang dari waktu ke waktu dikarenakan pengetahuan dapat bertambah dan *terupdate*. Pemisahan basis pengetahuan dan mesin inferensi dalam sistem pakar bertujuan agar perkembangan pada sistem pakar dapat secara leluasa disesuaikan dengan perkembangan pengetahuan pada suatu domain, dimana penambahan dan pengurangan pada basis pengetahuan dapat dilakukan tanpa mengganggu mesin inferensi.

### 3. Mesin Inferensi (*Inference Machine*)

Mesin inferensi ialah otak dari sistem pakar yang berupa perangkat lunak yang melakukan tugas inferensi pada penalaran sistem pakar atau biasa juga disebut sebagai mesin pemikir (*Thinking Machine*). Pada prinsipnya mesin inferensi ialah mesin yang mencari solusi dari suatu permasalahan, adapun konsep-konsep yang biasa digunakan dalam mesin inferensi seperti runut-balik (*top-down*) yaitu proses penalaran yang berawal dari tujuan yang ingin dicapai lalu menelusuri fakta-fakta yang mendukung untuk mencapai tujuan dan runut muju (*bottom-up*) yaitu proses penalaran yang berawal dari kondisi yang diketahui menuju tujuan yang diinginkan.

### 4. Memori Kerja (*Working Memory*)

Memori kerja merupakan bagian dari sistem pakar yang bertugas menyimpan fakta-fakta yang diperoleh saat dilakukannya proses konsultasi. Fakta-fakta ini kemudian akan diolah oleh mesin inferensi berdasarkan pengetahuan yang disimpan dalam basis pengetahuan untuk menentukan suatu keputusan dalam pemecahan masalah. Konklusinya dapat berupa hasil diagnosa, tindakan ataupun akibat.

Menurut (Hartati & Iswanti, 2008:22) representasi pengetahuan dalam sistem pakar bertujuan untuk mengorganisasikan pengetahuan ke dalam bentuk dan format tertentu agar dapat dimengerti oleh komputer.

Pemilihan representasi pengetahuan yang tepat dapat membuat sistem pakar yang dibuat lebih efektif dikarenakan pemilihan representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan dalam

keperluan pembuatan keputusan. Sistem pakar dalam penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi (*Production Rule*).

Menurut (Hartati & Iswanti, 2008:22-26) adapun beberapa model representasi pengetahuan yang penting:

1. Jaringan Semantik (*Semantic Nest*)

Jaringan semantik ialah teknik representasi pengetahuan yang digunakan untuk informasi yang proposional yang dimaksud proporsional disini ialah pernyataan yang mempunyai nilai benar atau salah. Contohnya: sebuah bujur sangkar mempunyai empat sisi. Informasi yang proporsional merupakan bahasa yang deklaratif karena menyatakan fakta.

2. Bingkai (*Frame*)

Teknik representasi bingkai berupa kumpulan slot-slot yang berisi atribut untuk mendeskripsikan pengetahuan, pengetahuan yang dimuat dalam slot dapat berupa kejadian, lokasi, situasi ataupun elemen-elemen lainnya. Representasi pengetahuan bingkai cocok untuk jenis pengetahuan yang memiliki subyek sempit, lebih bersifat pasti dan jarang berubah-ubah isinya kecuali terdapat kondisi khusus.

3. Kaidah Produksi (*Production Rule*)

Kaidah menyediakan cara formal dalam merepresentasikan rekomendasi, arahan atau strategi. Kaidah produksi ditulis dalam bentuk jika-maka (*if-then*). Kaidah jika-maka menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Menurut (Adediji, 1992) adapun berbagai struktur kaidah *if-then* yang menghubungkan obyek atau atribut sebagai berikut:

- a. *IF* premis *THEN* konklusi
- b. *IF* masukan *THEN* keluaran
- c. *IF* kondisi *THEN* tindakan
- d. *IF* antesenden *THEN* konsekuen
- e. *IF* data *THEN* hasil
- f. *IF* aksi *THEN* reaksi
- g. *IF* sebab *THEN* akibat
- h. *IF* gejala *THEN* diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan diagnosa.

Sebelum sampai pada bentuk kaidah produksi, ada beberapa langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut ialah menyajikan pengetahuan yang berhasil didapatkan kedalam bentuk tabel keputusan (*decision table*) kemudian dari tabel keputusan dibuat pohon keputusan (*decision tree*).

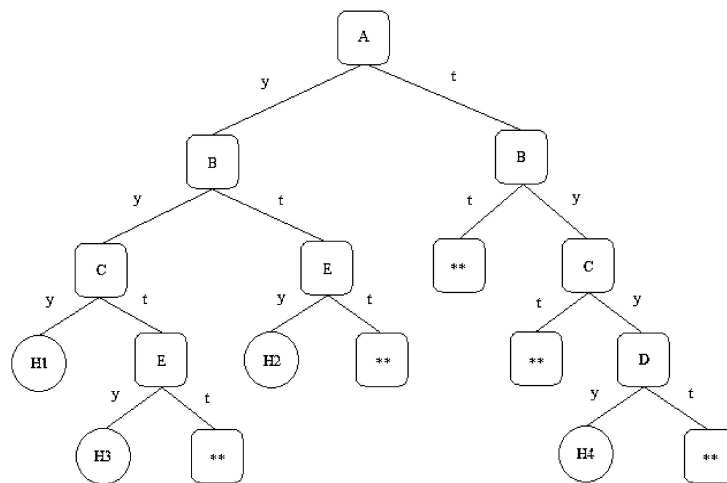


**Tabel 2.1** Tabel Keputusan

<b>Hipotesa</b> <i>Evidence</i>	<b>Hipotesa 1</b>	<b>Hipotesa 2</b>	<b>Hipotesa 3</b>	<b>Hipotesa 4</b>
<i>Evidence A</i>	ya	ya	ya	tidak
<i>Evidence B</i>	ya	tidak	ya	ya
<i>Evidence C</i>	ya	tidak	tidak	ya
<i>Evidence D</i>	tidak	tidak	tidak	ya
<i>Evidence E</i>	tidak	ya	ya	tidak

(Sumber: Hartati & Iswanti, 2008:32)

Mengacu pada tabel keputusan pada tabel 2.1 dapat dihasilkan pohon keputusan sebagai berikut:

**Gambar 2.2** Pohon Keputusan

Keterangan:

A = *evidence A*,      H1 = hipotesa 1,      y = ya  
 B = *evidence B*,      H2 = hipotesa 2,      t = tidak  
 C = *evidence C*,      H3 = hipotesa 3,      \*\* = tidak menghasilkan hipotesa tertentu  
 D = *evidence D*,      H4 = hipotesa 4

Dari gambar 2.2 dapat dilihat bahwa hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak memenuhi.

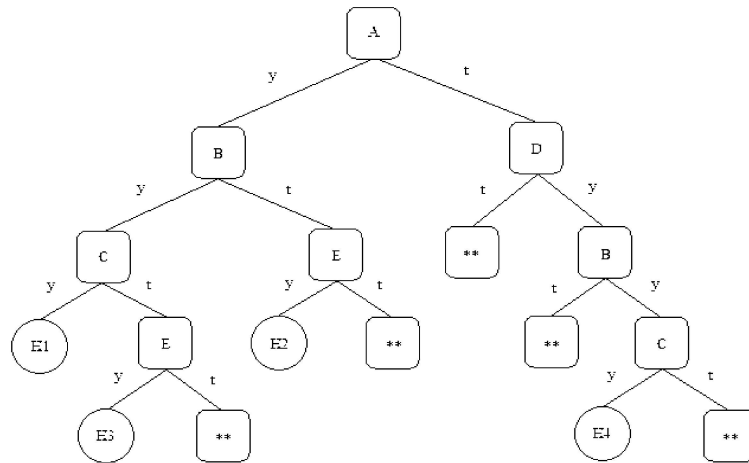
Dalam implementasi sitem pakar terutama dalam sesi konsultasi pada sistem pakar, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

**Tabel 2.2** Alternatif Tabel Keputusan

<b>Hipotesa</b> <i>Evidence</i>	<b>Hipotesa 1</b>	<b>Hipotesa 2</b>	<b>Hipotesa 3</b>	<b>Hipotesa 4</b>
<i>Evidence A</i>	ya	ya	ya	tidak
<i>Evidence D</i>	tidak	tidak	tidak	ya
<i>Evidence B</i>	ya	tidak	ya	ya
<i>Evidence C</i>	ya	tidak	tidak	ya
<i>Evidence E</i>	Tidak	Ya	ya	tidak

(Sumber: Hartati & Iswanti, 2008:34)

Mengacu pada tabel keputusan pada tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



**Gambar 2.3** Alternatif Pohon Keputusan

Keterangan:

A = <i>evidence</i> A,	H1 = hipotesa 1,	y = ya
B = <i>evidence</i> B,	H2 = hipotesa 2,	t = tidak
C = <i>evidence</i> C,	H3 = hipotesa 3,	** = tidak menghasilkan hipotesa tertentu
D = <i>evidence</i> D,	H4 = hipotesa 4	

Dilihat dari gambar 2.3, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Kaidah yang dapat dihasilkan berdasarkan pohon keputusan pada gambar 2.3 adalah sebagai berikut:

- a. Kaidah 1: *IF A AND B AND C THEN H1*
- b. Kaidah 2: *IF A AND B AND E THEN H3*

- c. Kaidah 3: *IF A AND E THEN H2*
- d. Kaidah 4: *IF D AND B AND C THEN H4*

Model representasi pengetahuan kaidah produksi banyak digunakan pada aplikasi sistem pakar karena model representasi ini mudah dipahami dan bersifat deklaratif sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah, dan mudah diinterpretasikan.

#### 4. Logika Predikat (*Predicate Logic*)

Logika predikat berdasarkan pada kebenaran dan kaidah inferensi untuk merepresentasikan simbol-simbol dan hubungan antara satu dengan yang lain. Selain digunakan untuk menentukan kebenaran (*truthfulness*) atau kesalahan (*falsity*) sebuah pernyataan, logika predikat juga dapat digunakan untuk merepresentasikan pernyataan tentang obyek tertentu (Hartati & Iswanti, 2008:40).

Dalam melakukan proses inferensi, sistem pakar memerlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari suatu kondisi yang sesuai dengan kondisi awal atau untuk memastikan kondisi yang sedang berjalan sudah dimasukkan ke dalam basis data (*database*). Proses pengujian itu disebut dengan perunutan, yaitu proses pencocokan fakta atau kondisi tertentu yang tersimpan dalam basis pengetahuan maupun pada memori kerja dengan kondisi yang dinyatakan dalam premis atau bagian kondisi pada suatu kaidah atau aturan (Hartati & Iswanti, 2008:45).

Adapun beberapa konsep perunutan yang dapat digunakan oleh mesin inferensi yaitu:

a. Runut maju (*forward chaining*)

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses perunutan (penalaran) dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information* (*THEN*). Konsep ini dapat dimodelkan sebagai berikut:

*IF* (informasi masukan)

*THEN* (konklusi)

Informasi masukan dapat berupa suatu pengamatan, data, bukti atau temuan. Sedangkan konklusi dapat berupa diagnosis, tujuan, hipotesa atau penjelasan sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosis. Pada metode ini, sistem tidak melakukan praduga apapun terhadap konklusi, namun sistem akan menerima semua gejala yang diberikan pengguna lalu sistem akan memeriksa gejala-gejala tersebut dan selanjutnya mencocokkan dengan konklusi yang sesuai (Hartati & Iswanti, 2008:45-46).

b. Runut balik (*backward chaining*)

Secara umum konsep ini diaplikasikan ketika tujuan ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau perunutan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

Tujuan,

*IF* (kondisi

Proses penalaran pada *backward chaining* dimulai dari tujuan kemudian merunut balik ke jalur yang mengarah ke tujuan tersebut, untuk membuktikan bahwa bagian kondisi pada kaidah atau aturan benar-benar terpenuhi. Proses *internal* selalu memeriksa konklusi (tujuan) terlebih dahulu sebagai praduga awal, kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *output*. Jika sistem menemukan ada bagian kondisi yang tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya (Hartati & Iswanti, 2008:46-47).

### **2.1.2 Web**

Menurut (Trimarsiah & Arafat, 2017:2) web atau *website* merupakan sebuah media informasi yang ada di internet. *Website* tidak hanya dapat digunakan untuk penyebaran informasi saja melainkan bisa digunakan untuk membuat toko online. *Website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (WWW) di Internet. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server *website* untuk ditampilkan kepada para pemakai melalui web browser. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

### **2.1.3 Database (basis data)**

Menurut (A.S & Shalahuddin, 2015:43-44) sistem *database* adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data atau informasi yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. *Database* adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem yaitu *Database Management System (DBMS)*. *DBMS* adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data.

### **2.1.4 Validasi Sistem**

Menurut (A.S & Shalahuddin, 2015:273-276) validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa sistem atau perangkat lunak yang dibangun telah sesuai dengan yang diharapkan. Beberapa pendekatan dalam melakukan pengujian untuk validasi sistem antara lain:

1. *Black-Box Testing* (pengujian kotak hitam)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Tujuannya untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak sudah sesuai dengan spesifikasi yang dibutuhkan.

Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

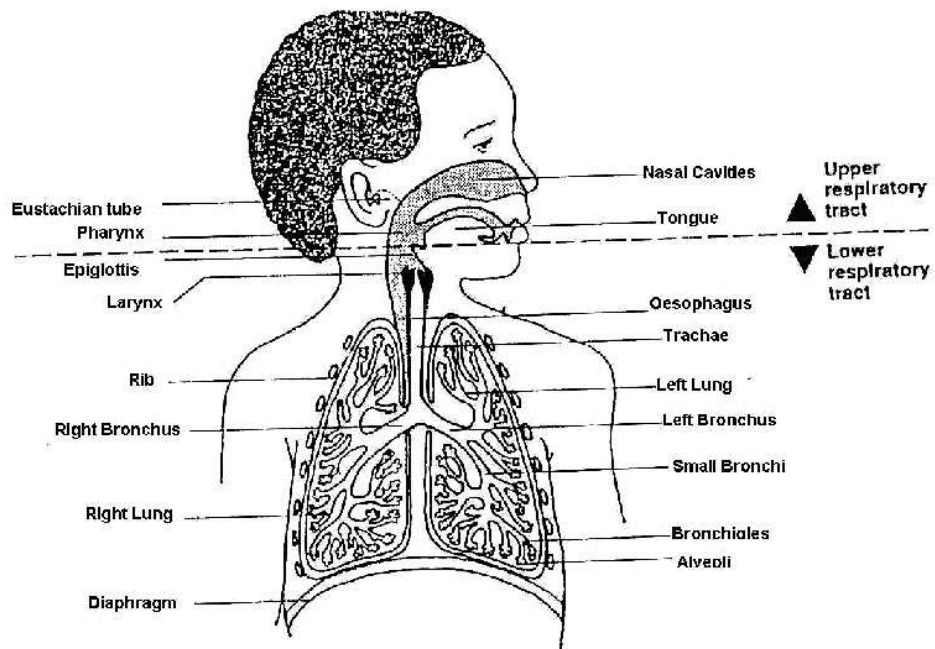
## 2. *White-Box Testing* (pengujian kotak putih)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. *White-box testing* dilakukan dengan memeriksa logika dari kode program. Pembuatan kasus uji dapat mengikuti standar pengujian dari standar pemrograman yang ada.

### **2.1.5 Infeksi Saluran Pernafasan Akut**

Infeksi saluran pernafasan akut (ISPA) secara anatomis mencakup saluran pernafasan bagian atas, saluran pernafasan bagian bawah (termasuk jaringan paru-paru) dan organ adneksa saluran pernafasan. Dengan batasan ini, jaringan paru termasuk dalam saluran pernafasan (*respiratory tract*). Penyakit ISPA didiagnosa masyarakat awam berdasarkan ciri-ciri yang diketahuinya, sehingga masyarakat atau penderita sulit membedakan penyakit ISPA dengan jenis penyakit lainnya akibatnya penyakit tersebut ditangani dengan cara yang sama (Nurhayati, 2013:110).





**Gambar 2.4** Infeksi Saluran Pernafasan Akut

## 2.2 Variabel Penelitian

Variabel penelitian merupakan segala sesuatu yang berbentuk apa saja yang dimiliki orang, obyek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya (Sugiyono, 2014:38). Pada penelitian ini yang menjadi variabel penelitian adalah penyakit infeksi saluran pernafasan akut (ISPA) khususnya Faringitis, Laringitis dan Tonsilitis.

### **2.2.1 Faringitis**

Faringitis merupakan peradangan dinding faring yang dapat disebabkan oleh virus (40- 60%), bakteri (5-40%), alergi, trauma, toksin dan lain-lain. Virus dan bakteri melakukan invasi ke faring dan menimbulkan reaksi inflamasi lokal. Infeksi bakteri group A streptokokus  $\beta$  hemolitikus dapat menyebabkan kerusakan jaringan yang hebat, karena bakteri ini melepaskan toksin ekstraseluler yang dapat menimbulkan demam reumatik, kerusakan katup jantung, glomerulonefritis akut karena fungsi glomerulus terganggu akibat terbentuknya kompleks antigen antibodi (Nadeak, 2017:57).

### **2.2.2 Laringitis**

Laringitis adalah peradangan pada laring yang dapat disebabkan oleh virus, bakteri, atau jamur. Laringitis juga merupakan akibat dari penggunaan suara yang berlebihan, pajanan terhadap polutan eksogen, atau infeksi pada pita suara. Refluks gastroesofageal, bronkitis, dan pneumonia juga dapat menyebabkan laringitis. Laringitis pada anak sering diderita oleh anak usia 3 bulan hingga 3 tahun, dan biasanya disertai inflamasi pada trakea dan bronkus dan disebut sebagai penyakit croup. Penyakit ini seringkali disebabkan oleh virus, yaitu virus parainfluenza, adenovirus, virus influenza A dan B, RSV, dan virus campak (Zainuddin et al., 2016:351).

### **2.2.3 Tonsilitis**

Tonsilitis adalah peradangan tonsil palatina yang merupakan bagian dari cincin Waldeyer. Cincin Waldeyer terdiri atas susunan jaringan limfoid yang terdapat di dalam rongga mulut yaitu: tonsil faringeal (adenoid), tonsil palatina (tonsil faucial), tonsil lingual (tonsil pangkal lidah), tonsil tuba *Eustachius* (*lateral band* dinding faring/ *Gerlach's tonsil*). Penyakit ini banyak diderita oleh anak-anak berusia 3 sampai 10 tahun (Zainuddin et al., 2016:355).

## **2.3 Software Pendukung**

### **2.3.1 XAMPP**

XAMPP adalah sebuah software *web server apache* yang didalamnya sudah tersedia *database server MySQL* dan dapat mendukung pemrograman *PHP*. *XAMPP* merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server, MySQL Database Server, PHP Support* (*PHP 4 dan PHP 5*) dan beberapa module lainnya (Februariyanti & Zuliarso, 2012:129).

### 2.3.2 phpMyAdmin

phpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *MySQL* melalui Jejaring Jagat Jembar (*World Wide Web*). phpMyAdmin mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel- tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain. Pada dasarnya, mengelola basis data dengan *MySQL* harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data.

Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu. Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam *MySQL*, salah satunya adalah *phpMyAdmin*. Dengan *phpMyAdmin*, seseorang dapat membuat *database*, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya (Barri, Lumenta, & Wowor, 2015:25).

### 2.3.3 Hypertext Preprocessor (PHP)

*PHP (Hypertext Preprocessor)* merupakan bahasa pemrograman pada sisi *server* yang memperbolehkan *programmer* menyisipkan perintah-perintah perangkat lunak *web server (Apache, IIS, atau apapun)* akan dieksekusi sebelum

perintah itu dikirim oleh halaman ke *browser* yang *me-request*-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman web setiap kali tampilan tanggal dibutuhkan. Sesuai dengan fungsinya yang berjalan di sisi server maka *PHP* adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application*. *PHP* telah menjadi bahasa *scripting* untuk keperluan umum yang pada awalnya hanya digunakan untuk pembangunan web yang menghasilkan halaman web dinamis. Untuk tujuan ini, kode *PHP* tertanam ke dalam dokumen sumber *HTML* dan diinterpretasikan oleh *server web* dengan modul *PHP* prosesor, yang menghasilkan dokumen halaman web. Sebagai bahasa pemrograman untuk tujuan umum, kode *PHP* diproses oleh aplikasi penerjemah dalam modus baris-baris perintah modus dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program di *channel output* standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. *PHP* tersedia sebagai prosesor untuk *server web* yang paling modern dan sebagai penerjemah mandiri pada sebagian besar sistem operasi dan komputer *platform* (Februariyanti & Zuliarso, 2012:128).

*PHP* merupakan salah satu bahasa pemrograman web *server-side* yang populer dan banyak digunakan sampai saat ini, *PHP* dirancang oleh Rasmus Ledrof pada tahun 1994. Awalnya *PHP* digunakan untuk mendeteksi *user* yang berkunjung pada situs (Utomo, 2014:1-2).

### **2.3.4 HTML (*Hyper Text Markup Language*)**

*HyperText Markup Language (HTML)* adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan *formatting hypertext* sederhana yang ditulis kedalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format *ASCII normal* sehingga menjadi *homepage* dengan perintah- perintah *HTML*. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML (Standard Generalized Markup Language)*, *HTML* adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. *HTML* saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium(W3C)*. *HTML* dibuat oleh kolaborasi Robert Cailliau dan Tim Berners-Lee ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa) (Harison & Syarif, 2016:43).

### **2.3.5 CSS (*Cascading Style Sheet*)**

*CSS (Cascading Style Sheet)* adalah *stylesheet language* yang digunakan untuk mendeskripsikan penyajian dari dokumen yang dibuat dalam *mark up language*. *CSS* merupakan sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman web, inti dari dokumen ini adalah memformat

halaman web standar menjadi bentuk web yang memiliki kualitas yang lebih indah dan menarik (Binarso, Sarwoko, & Bahtiar, 2012:76).

### 2.3.6 JavaScript

*JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode *HTML* dan di proses di sisi *client*. *JavaScript* digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap *HTML* melalui eksekusi perintah di sisi *browser*. *JavaScript* dapat merespon perintah *user* dengan cepat dan menjadikan halaman web menjadi responsif. *JavaScript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen *HTML* atau berdiri sebagai satu kesatuan aplikasi (Yatini, 2014:2).

### 2.3.7 jQuery

*jQuery* adalah *JavaScript library* yang dirancang untuk meringkas kode-kode *JavaScript*, sehingga dapat menyederhanakan penulisan skrip program, sesuai dengan slogan “*write less, do more*”. *jQuery* pertama kali dirilis oleh John Resig pada tahun 2006, pada perkembangannya *jQuery* tidak hanya sebagai *framework JavaScript*, namun memiliki kelebihan antara lain:

1. Kemudahan mengakses dan memanipulasi elemen-elemen *HTML*.
2. Memanipulasi *CSS*.
3. Penanganan *event HTML*.

4. Efek-efek *JavaScript* dan animasi.
5. Memodifikasi elemen *HTML DOM*.

Sintak dasar *jQuery*  $\$(selector).action()$ , tanda \$ untuk mendefinisikan *jQuery*, *jQuery selector* digunakan untuk mendapatkan elemen *HTML*, *action* adalah tindakan yang dilakukan *jQuery* pada elemen () (Yatini, 2014:2).

### 2.3.8 MySQL dan SQL

Menurut (Saputra, 2012:77-78) *MySQL* ialah salah satu *database* kelas dunia yang sangat cocok jika dipadukan dengan bahasa pemrograman *PHP*. *MySQL* bekerja menggunakan bahasa *SQL (Structure Query Language)* yang merupakan bahasa standar yang berfungsi untuk manipulasi *database*. Ada beberapa alasan yang menjadikan *MySQL* sangat diminati oleh para *programmer*, diantaranya:

1. Bersifat *open-source*
2. Menggunakan bahasa *SQL (Structure Query Language)* yang merupakan standar bahasa dalam pengolahan data.
3. *Performance* dan *reliable*, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*ease of use*).
5. Memiliki dukungan (*group*) pengguna *MySQL*.
6. Lintas *platform*, dapat digunakan pada berbagai sistem operasi berbeda.
7. *Multiuser*, dimana *MySQL* dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.
8. Dan masih banyak lagi.



### 2.3.9 Sublime Text

*Sublime Text Editor* adalah *editor* teks yang dapat digunakan untuk berbagai bahasa pemrograman termasuk pemrograman *PHP*. *Sublime Text Editor* merupakan *editor text* lintas-*platform* dengan *Python application programming interface (API)*. *Sublime Text Editor* juga mendukung banyak bahasa pemrograman dan bahasa *markup*, dan fungsinya dapat ditambah dengan *plugin*, dan *Sublime Text Editor* tanpa lisensi perangkat lunak, *Text Editor* merupakan suatu kebutuhan wajib yang harus dimiliki dalam menulis suatu pemrograman terutama pemrograman web (Abdullah, 2017:4).

### 2.3.10 Laravel

*Laravel* merupakan salah satu *framework PHP* yang dapat digunakan secara gratis. *Laravel* dikembangkan oleh *programmer* asal Amerika yaitu Taylor Otwell pada tahun 2011. Secara sederhana *framework* dapat diartikan sebagai kumpulan kode-kode program yang akan selalu digunakan pada setiap pembuatan aplikasi. Dikarenakan sering atau selalu digunakan, maka kode-kode tersebut dikumpulkan dan disusun secara rapi pada folder-folder agar dapat mudah digunakan dan jadilah sebuah *framework*, *Laravel* telah menjadi salah satu *framework* terbaik dan paling banyak digunakan oleh *programmer* dunia (Abdullah, 2017:1-2).

### 2.3.11 Twitter Bootstrap

*Twitter Bootstrap* adalah *framework* yang kuat menyediakan *set* kelas *CSS* dan fungsi *JavaScript* untuk memudahkan proses pembangunan antarmuka halaman web. Mengaktifkan fitur desain responsif dukungan untuk menampilkan *desktop* maupun *mobile*. Situs dikembangkan dapat bekerja dengan baik pada *desktop* maupun *mobile*. *Developer* tidak harus bekerja dengan *CSS* untuk membuat *website* terlihat menarik atau mendukung prinsip desain *responsive*, kecuali diperlukan.

*Twitter Bootstrap* dapat diunduh secara gratis di *website* resminya, setelah itu tinggal memanggil *file CSS Twitter Bootstrap* pada *file project website* yang akan menggunakan *Twitter Bootstrap*. Begitu selesai memanggil *Bootstrap*, maka secara otomatis akan mengubah tampilan *website* tanpa harus melakukan pengetikan sintak-sintak *CSS* seperti biasa dilakukan (Rosid & Jakaria, 2016:130).

### 2.3.12 StarUML

*StarUML* merupakan suatu *CASE (Computer-Aided Software Engineering) tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang membantu dalam pembuatan suatu sistem perangkat lunak. *StarUML* termasuk salah satu *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak seperti membuat diagram ER (*Entity Relationship*),

DFD (*Data Flow Diagram*), dan diagram perencanaan atau desain perangkat lunak lainnya (A.S & Shalahuddin, 2015:122-123).

UML (*Unified Modeling Language*) adalah suatu standar bahasa yang banyak digunakan untuk mendefinisikan kebutuhan atau *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (OOP) dalam industri perangkat lunak (A.S & Shalahuddin, 2015:133).

Menurut (A.S & Shalahuddin, 2015:140-141) dalam *UML 2.3* terdapat 13 macam diagram yang dibagi menjadi 3 kelompok yaitu:

1. *Structure Diagrams*

Dalam *Structure Diagrams* terdiri dari kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan, Diagram yang termasuk dalam *Structure Diagrams* antara lain *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.

2. *Behavior Diagrams*

Dalam *Behavior Diagrams* terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem, yang termasuk dalam *Behavior Diagrams* antara lain *use case diagram*, *activity diagram* dan *state machine diagram*.

3. *Interaction Diagrams*

Dalam *Interaction Diagrams* terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem, yang termasuk dalam *Interaction Diagrams*

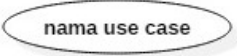

antara lain *sequence diagram*, *communication diagram*, *timing diagram* dan *interaction overview diagram*.

Menurut (A.S & Shalahuddin, 2015:18) *use case*, *class diagram* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:


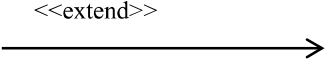

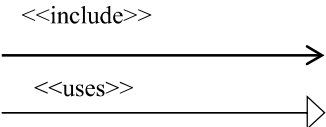
#### 1. *Use case diagram*

*Use case diagram* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 hal utama yang terdapat pada *use case* yaitu aktor dan *use case*. Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S & Shalahuddin, 2015:155-156).

**Tabel 2.3** Simbol *Use Case Diagram*

Simbol	Deskripsi
<p data-bbox="363 1430 456 1455"><i>Use case</i></p> 	<p data-bbox="807 1430 1320 1541">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p data-bbox="363 1629 483 1654">Aktor/<i>actor</i></p> 	<p data-bbox="807 1629 1320 1797">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i></p>

**Tabel 2.3 Lanjutan**


Asosiasi/association 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor
Ekstensi/extend 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu. Arah panah mengarah pada use case yang ditambahkan.
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada use case yang menjadi generalisasinya (umum)
Menggunakan/include/uses 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya use case ini. Arah panah mengarah pada use case yang ditambahkan

(Sumber: A.S & Shalahuddin, 2015:156-158)





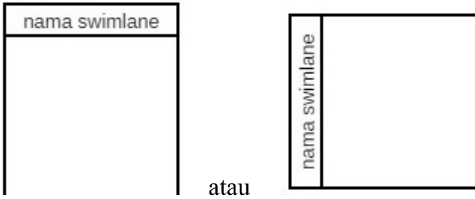
## 2. Activity Diagram

Menurut (A.S & Shalahuddin, 2015:161-163) *Activity Diagram* ialah diagram aktivitas yang menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Dibawah ini akan menjelaskan simbol-simbol *Activity Diagram*:

**Tabel 2.4 Simbol Activity Diagram**

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal

Tabel 2.4 Lanjutan



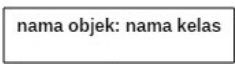

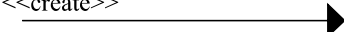
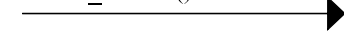

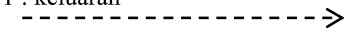

Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status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

(Sumber: A.S & Shalahuddin, 2015:162-163)

### 3. Sequence Diagram

Menurut (A.S & Shalahuddin, 2015:165:167) diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang akan dikirimkan dan diterima antar objek. Oleh karena itu objek-objek yang terlibat dalam *use case* beserta metode-metode yang dimiliki setiap kelas yang diinstansiasi menjadi objek harus diketahui untuk menggambar diagram sekuen. Dibawah ini akan menjelaskan simbol-simbol *Sequence Diagram* yaitu:

Tabel 2.5 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor/actor</p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
<p>Garis hidup/lifeline</p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
<p>Pesan tipe create</p> <p>&lt;&lt;create&gt;&gt;</p> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat
<p>pesan tipe call</p> <p>1 : nama_metode()</p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
<p>Pesan tipe send</p> <p>1 : masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju
<p>pesan tipe return</p> <p>1 : keluaran</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima
<p>Pesan tipe destroy</p> <p>&lt;&lt;destroy&gt;&gt;</p> <p>2 :</p> 	Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri

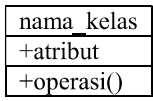


(Sumber: A.S & Shalahuddin, 2015:165-167)

#### 4. *Class Diagram*

Menurut (A.S & Shalahuddin, 2015:141-148) *Class diagram* digunakan untuk melakukan visualisasi struktur kelas-kelas yang akan dibuat untuk memabangun suatu sistem *Class diagram* dibuat bertujuan agar pembuat program membuat kelas-kelas sesuai dengan rancangan dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat pada perangkat lunak, sehingga tidaklah berguna lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai atau sama. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:



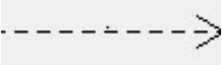
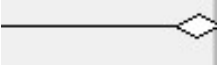
- a. Kelas *main*
- b. Kelas yang menangani tampilan sistem (*view*)
- c. Kelas yang diambil dari pendefinisian *use-case* (*controller*)
- d. Kelas yang diambil dari pendefinisian data (*model*)

**Tabel 2.6** Simbol *Class Diagram*

<b>Simbol</b>	<b>Deskripsi</b>
Kelas 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>



Tabel 2.6 Lanjutan

Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

(Sumber: A.S & Shalahuddin, 2015:146-147)

## 2.4 Penelitian Terdahulu

Berikut peneliti mencatumkan beberapa penelitian terdahulu dibidang sistem pakar untuk mendukung teori yang berkaitan dengan penelitian.

Penelitian (Nurlaela, 2013:76) dengan judul “Sistem Pakar Untuk Mendiagnosa Penyakit Gigi Pada Manusia”. Sistem pakar merupakan suatu sistem yang dirancang untuk membantu dalam mendeteksi penyakit dengan basis pengetahuan yang dinamis. Pengetahuan ini didapat dari pakar yaitu dokter gigi. Dalam sistem pakar ini menggunakan metode inferensi *forward chaining*. Dalam penelitian ini menggunakan metode wawancara dengan dokter gigi. Adapun untuk tujuan penelitian adalah menghasilkan suatu sistem pakar untuk membantu dokter gigi dalam mendokumentasikan ilmunya. Dan untuk manfaatnya dapat menerapkan ilmu yang telah didapatkan di perkuliahan untuk membantu

memberikan pelayanan kepada masyarakat. Hasil yang diperoleh dari penelitian ini adalah sistem pakar untuk mendeteksi penyakit gigi pada manusia dengan memanfaatkan komputer sebagai alat bantu untuk mengakses data.

Penelitian (Lisnawita, Lhaura Van & Lindra, 2016:95) dengan judul “Sistem Pakar Diagnosa Penyakit THT”. Dalam mendiagnosa penyakit THT (Telinga, Hidung, tenggorokan), seorang dokter memerlukan data berupa gejala-gejala yang sedang dialami oleh si penderita. Gejala-gejala ini dapat di peroleh dari pemeriksaan fisik atau laboratorium. Sebagai seorang manusia yang memiliki keterbatasan, begitu juga dengan dokter THT tentunya memiliki kelemahan. Apabila terdapat penyakit THT baru atau lupa tentang jenis penyakit dan pengobatannya, maka seorang ahli THT mencari kembali buku-buku atau dokumen-dokumen yang membahas tentang penyakit THT tersebut. Cara ini tentu saja akan memakan waktu yang lama. Penelitian ini bertujuan untuk merancang sistem pakar untuk mendiagnosa penyakit THT. Perancangan ini di mulai dari pembuatan basis pengetahuan dilanjutkan dengan identifikasi *rule* dan perancangan *input-output* dengan metode *forward chaining*. Hasil rancangan pada penelitian ini diharapkan mampu membantu pekerjaan dokter menjadi lebih mudah sehingga memberikan hasil diagnosa penyakit THT menggunakan sistem pakar.

Penelitian (Syamsuddin & Ahyuna, 2014:64) dengan judul “Sistem Pakar untuk Diagnosa Penyakit yang Disebabkan oleh Nyamuk Berbasis Web”. Sistem pakar ini bertujuan untuk mendiagnosa penyakit yang disebabkan oleh nyamuk. Sistem pakar ini dirancang dengan alat pemodelan sistem *Unified Modeling Language(UML)*, *MySQL* sebagai sistem manajemen basis data *SQL* dan *PHP*

sebagai bahasa pemrograman. Setelah sistem dapat diimplementasikan maka dilakukanlah pengujian sistem dengan metode *Black Box*. Hasil dari sistem yang dibangun adalah sebuah sistem pakar yang mampu melakukan diagnosa penyakit yang disebabkan oleh nyamuk dengan tingkat akurasi yang baik dan hampir tidak ditemukan kesalahan yang ada pada tiap form komponen yang diuji.

Penelitian (Pasalli, Poekoel, & Najoran, 2016:1) dengan judul “Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode *Forward Chaining* Berbasis *Mobile*”. Tujuan Penelitian ini adalah membuat aplikasi sistem pakar berbasis *mobile* dalam mendiagnosis penyakit anak. Dengan menggunakan teknik penalaran *forward chaining* diagnosa dilakukan dengan memulai dari sekumpulan gejala-gejala, nantinya dapat melihat kesimpulan jenis penyakit pada anak. Metode yang digunakan sebagai tahapan penelitian ini adalah metode *Extreme Programming (XP)* yang merupakan metode rancang bangun perangkat lunak yang menekankan pada 4 tahap dalam pengembangan perangkat lunak.

Penelitian (Sinaga, 2014:101) dengan judul “Sistem Pakar Mendeteksi Penyakit Tanaman Terong Belanda dengan Menggunakan Metode *Forward Chaining*”. Sistem pakar ini menggunakan mesin inferensi dengan metode *forward chaining*. Penalaran dilakukan berdasarkan dari gejala-gejala yang tampak secara fisik terhadap tanaman Terong Belanda. Berdasarkan gejala-gejala tersebut dibuatlah *rule-rule* yang kemudian menjadi knowledge base yang akan diterapkan ke dalam mesin inferensi untuk mengetahui penyakit apa yang dialami oleh tanaman Terong Belanda tersebut. Hasil program ini menunjukkan bahwa sistem pakar dapat digunakan sebagai suatu media yang dapat memberikan

informasi tentang tanaman Terong Belanda. Sistem pakar ini dapat digunakan untuk mempercepat pencarian dan pengaksesan terhadap pengetahuan oleh orang-orang yang membutuhkan informasi.

Penelitian (Ashari & Muniar, 2016:1) dengan judul “Penerapan Sistem Pakar Untuk Mendiagnosa Penyakit Pencernaan Dengan Pengobatan Alami”. Penerapan sistem pakar untuk mendiagnosa penyakit pencernaan dengan pengobatan dari bahan alami menggunakan metode *forward chaining*. Metode *forward chaining* merupakan metode inferensi untuk penalaran dari suatu masalah dengan memberikan solusinya. Penelitian ini sebagai produk teknologi terapan yang diharapkan memberi manfaat sebagai media konsultasi atau instruktur bagi masyarakat pada umumnya, dan terkhusus bagi dokter dan paramedis pada klinik, puskesmas dan rumah sakit dalam memberikan alternatif pencegahan dan pengobatan secara alami. Perancangan sistem telah dilakukan melalui aktivitas pengumpulan data, perancangan rules, perancangan proses dan pengujian sistem. Tahun pertama menghasilkan produk aplikasi sistem pakar yang telah diimplementasikan pada sejumlah puskesmas dan pada tahun kedua dilakukan pengujian model integrasi sistem pakar terhadap berbagai gejala penyakit pencernaan yang terjadi dengan memberikan alternatif solusi pencegahan penyakit berdasarkan hasil inference yang ditemukan dengan pengobatan cara alami. Hasil pengujian sistem dinyatakan baik dengan tingkat akurasi 91,56%.

Penelitian (Supartini & Hindarto, 2016:147) dengan judul “Sistem Pakar Berbasis Web Dengan Metode *Forward Chaining* Dalam Mendiagnosis Dini Penyakit Tuberkulosis di Jawa Timur”. Tuberkulosis adalah suatu penyakit menular

berbahaya yang disebabkan oleh kelompok *Mycobacterium*, yaitu *Microbacterium* Tuberkulosis. Setiap pasien Tuberkulosis dapat menularkan penyakitnya pada orang lain yang berada di sekelilingnya dan atau yang berhubungan erat dengannya. Karena masih banyak orang yang tidak mengetahui gejala-gejala penyakit suatu sistem pakar mendiagnosis secara dini penyakit tuberkulosis menggunakan metode forward chaining berbasis web, dapat dikenali dengan melihat gejala-gejala dengan mendeteksi penyakit sejak dini, dilakukan pencegahan terhadap penyakit tuberkulosis. Diagnosis sistem pakar, memiliki nilai keakuratan 93,333 % dan nilai eror 6,667 % untuk uji coba pada 15 pasien. Sehingga dapat disimpulkan bahwa sistem pakar cukup layak untuk digunakan oleh pasien dalam mendiagnosis dini pada penyakit tuberkulosis.

Penelitian (Ali & Saudi, 2014:1) dengan judul "*An expert system for the diagnosis and management of oral ulcers*". Kondisi-kondisi oral ulcer seperti *aphthous ulcers, chancre, traumatic ulcers, histoplasmosis ulcers, acute herpetic ulcers, burns, herpangina, tuberculosis, syphilis, gonorrhoea, acute necrotizing ulcerative gingivitis, herpes labialis, herpes zoster, erythema multiform, Steven Johnson's syndrome*. Kondisinya juga termasuk alergi, sindrom *behcet*, neutropenia, sialometaplasia necrotizing, pemfigus, ulseratif lichen planus, defisiensi vitamin, anemia, lupus eritematosus, *epidermolisis bulosa, pemphigoid, karsinoma sel squamous, gingivitis deskuamatif* dan lain-lain. Hasil menunjukkan bahwa rata-rata kualitas penilaian keseluruhan program adalah  $3,0 \pm 0,7$  dengan tingkat keberhasilan 75%. Dapat disimpulkan bahwa sistem pakar yang

diperkenalkan sangat membantu bagi mahasiswa pascasarjana dalam diagnosis dan pengobatan ulkus oral yang paling umum.

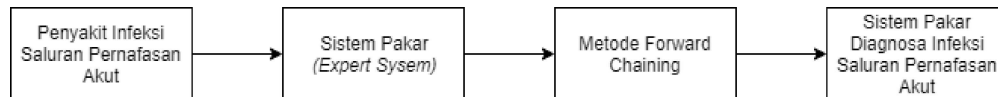
Penelitian (Singla, Grover, & Bhandari Asst, 2014:36) dengan judul “*Medical Expert Systems for Diagnosis of Various Diseases*”. Penyakit seharusnya dirawat dengan baik dan tepat waktu. Jika mereka tidak diobati tepat waktu, mereka dapat menyebabkan banyak masalah kesehatan dan masalah ini dapat menjadi penyebab kematian. Masalah-masalah ini menjadi lebih buruk karena kelangkaan spesialis, praktisi, dan fasilitas kesehatan. Dalam upaya untuk mengatasi masalah tersebut, penelitian ini dibuat sebagai upaya untuk merancang dan mengembangkan sistem pakar yang dapat memberikan saran bagi dokter dan pasien untuk memfasilitasi diagnosis dan merekomendasikan pengobatan pasien. Ulasan makalah ini menyajikan studi komprehensif sistem pakar medis untuk diagnosis berbagai penyakit. Ini memberikan gambaran singkat tentang sistem pakar diagnostik medis dan menyajikan analisis studi yang sudah ada.

## **2.5 Kerangka Pemikiran**

Munurut (Sugiyono, 2014:60) kerangka pemikiran ini merupakan penjelasan sementara terhadap gejala-gejala yang menjadi obyek permasalahan dan juga merupakan sintesa tentang hubungan antar variabel yang disusun dari berbagai teori yang telah dideskripsikan.

Permasalahan seperti kurangnya penyebaran pengetahuan masyarakat mengenai infeksi saluran pernafasan akut dan masyarakat tidak dapat menanganin

penyakit infeksi saluran pernafasan akut dengan cepat yang berujung menyebabkan kematian merupakan permasalahan yang teridentifikasi dalam penelitian ini. Untuk menjawab permasalahan tersebut berikut kerangka pemikiran pada penelitian ini:



**Gambar 2.5** Kerangka Pemikiran

Data-data yang dibutuhkan berkaitan dengan penyakit infeksi saluran pernafasan akut dianalisis terlebih dahulu agar lebih sederhana atau mudah dilakukan proses pengolahan datanya. Data-datanya tersebut kemudian diolah menggunakan sistem pakar menggunakan metode *forward chaining*. Sistem pakar yang menggunakan metode *forward chaining* ini menggunakan bahasa pemrograman *PHP* dan *database MySQL* yang dapat digunakan untuk mendiagnosa penyakit infeksi saluran pernafasan akut dan menghasilkan *output* (hasil diagnosa).

## **BAB III METODE PENELITIAN**

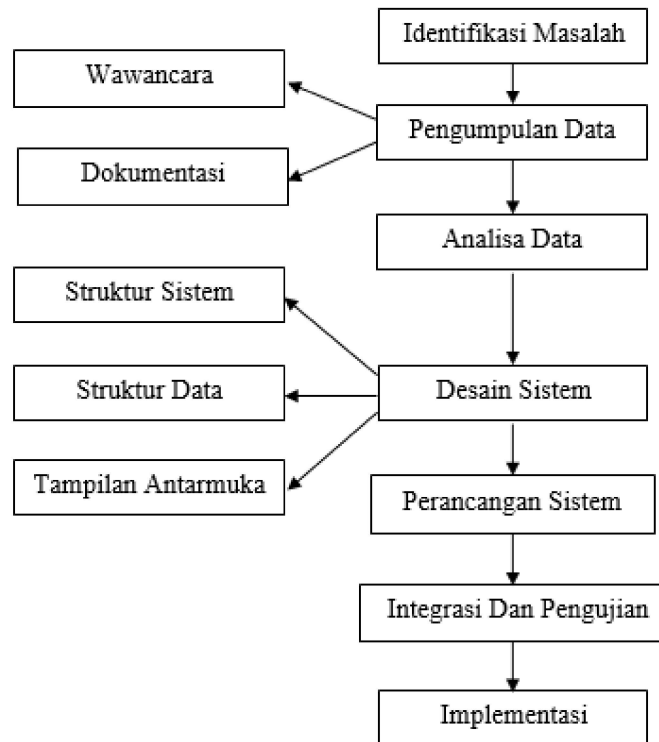
Metode penelitian pada dasarnya merupakan cara ilmiah untuk menemukan data dengan tujuan dan kegunaan tertentu. Data yang telah diperoleh dari penelitian dapat digunakan untuk memahami, memecahkan dan mengantisipasi masalah (Sugiyono, 2014:2-3).

### **3.1 Desain Penelitian**

Menurut (Sudaryono, 2015:3) desain penelitian berisi rumusan tentang langkah-langkah penelitian, dengan pendekatan, metode penelitian, teknik pengumpulan data, dan sumber data tertentu serta alasan-alasan mengapa menggunakan metode tersebut.

Penelitian ini menggunakan desain penelitian dengan beberapa tahap proses penelitian seperti yang terlihat pada gambar di bawah ini.





**Gambar 3.1** Desain Penelitian

Berikut ini adalah penjelasan dari desain penelitian yang ada pada gambar di atas:

#### 1. Identifikasi masalah

Menurut (Sudaryono, 2015:3) kegiatan penelitian dimulai dengan pengidentifikasian isu-isu dan masalah-masalah yang penting (esensial), hangat (aktual), dan mendesak (krusial) yang dihadapi saat ini, dan yang paling banyak arti atau kegunaanya bila diteliti.

Masalah yang ada pada penelitian ini: Kurangnya penyebaran pengetahuan masyarakat mengenai infeksi saluran pernafasan akut dan. Masyarakat tidak dapat menanganin penyakit infeksi saluran pernafasan akut dengan cepat sehingga dapat menyebabkan kematian.

## 2. Pengumpulan Data

Menurut (Sudaryono, 2015:3) kegiatan pengumpulan data didahului dengan penentuan teknik serta penyusunan dan pengujian instrumen pengumpul data yang akan digunakan. Dalam pelaksanaan pengumpulan data, selain objektivitas dan keakuratan data yang akan diperoleh, segi-segi legal dan etis juga perlu diperhatikan.

Pengumpulan data dengan metode Wawancara dan Dokumentasi.

- a. Wawancara dengan seorang pakar mengenai variabel yang akan diteliti. Seperti gejala penyakit, solusi dan konselingnya. Wawancara pada penelitian ini dilakukan dengan seorang dokter umum yaitu dr.H.Iwan Mulyana.
- b. Data yang digunakan dalam dokumentasi didapatkan dengan cara mengumpulkan buku dan jurnal yang berkaitan dengan topik penelitian yaitu penyakit infeksi saluran pernafasan akut. Buku yang dipakai sebagai referensi adalah buku Panduan Praktik Klinis Bagi Dokter karya (Zainuddin et al., 2016), jurnal *Upper Respiratory Infections* karya (Grief, 2013) dan jurnal *Global Surgery 2030: Evidence and solutions for achieving health, welfare, and economic development* karya (Meara et al., 2015)

## 3. Analisis Data

Menurut (Sugiyono, 2014:244) analisis data adalah proses mencari dan menyusun secara sistematis data yang diperoleh dari hasil wawancara, catatan lapangan, dan bahan-bahan lain, sehingga dapat mudah dipahami, dan temuannya dapat diinformasikan kepada orang lain. Analisis data dilakukan dengan cara mengorganisasikan data ke dalam kategori, menjabarkan ke dalam unit-unit,

melakukan sintesa, menyusun ke dalam pola, memilih mana yang penting dan yang akan dipelajari, dan membuat kesimpulan sehingga mudah dipahami oleh diri sendiri maupun orang lain.

Analisis data yang diterapkan pada mesin inferensi pada penelitian sistem pakar diagnosa infeksi saluran pernafasan akut ini menggunakan metode perunutan maju (*forward chaining*). Dimana teknik runut maju melakukan proses perunutan (penalaran) dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information (THEN)*.

#### 4. Desain Sistem

Menurut (A.S & Shalahuddin, 2015:23) desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. Kualitas perangkat lunak biasanya dinilai dari segi kepuasan pengguna perangkat lunak terhadap perangkat lunak yang digunakan.

Sistem pakar diagnosa infeksi saluran pernafasan akut pada penelitian ini di desain menggunakan aplikasi *Adobe Experience Design* untuk tampilan *prototype* yaitu berupa *wireframe* tampilan antarmuka yang akan dijelaskan lebih dalam pada sub-bab *prototype*.

## 5. Perancangan Sistem

Menurut (A.S & Shalahuddin, 2015:27) perancangan atau pengembangan sistem mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan; membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program; peninjauan pengujian.

Sistem pakar diagnosa infeksi saluran pernafasan akut ini dirancang menggunakan bahasa pemrograman *PHP* dengan menggunakan *framework Laravel* dan *Javascript* dengan menggunakan *library jQuery*. Untuk tampilan antarmuka dibangun dengan menggunakan bahasa *markup* seperti *HTML* dan *CSS* dengan menggunakan *framework Twitter Bootstrap*. Pengolahan basis data menggunakan *MySQL* dan *SQL*

## 6. Integrasi dan Pegujian

Menurut (A.S & Shalahuddin, 2015:27) integrasi dan pengujian mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan user.

Untuk pengujian pada sistem pakar diagnosa infeksi saluran pernafasan akut ini menggunakan pendekatan *Black-Box Testing* (pengujian kotak hitam) yaitu sebuah pendekatan yang melakukan pengujian sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program.

## 7. Implementasi

Menurut (A.S & Shalahuddin, 2015:27) implementasi perangkat lunak pada lingkungan produksi (lingkungan pada user) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian. Implementasi perangkat lunak dapat di lihat di Bab 4.

### **3.2 Pengumpulan Data**

Metode pengumpulan data ialah cara atau teknik yang digunakan peneliti dalam mendapatkan data-data yang berkaitan dengan pokok bahasan dalam rangka untuk mendukung penelitian yang sedang dilakukan. Adapun metode pengumpulan data diantaranya seperti angket, pengamatan, wawancara, tes, analisis dokumen, dan sebagainya (Sudaryono, 2015:83). Metode pengumpulan data yang digunakan dalam penelitian ini adalah:

#### 1. Wawancara

Menurut (Sudaryono, 2015:88) wawancara atau interviu merupakan salah satu cara pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Wawancara digunakan apabila kita ingin mengetahui hal-hal dari responden, yang jumlahnya sedikit, secara lebih mendalam.

Penelitian ini mendapatkan data-data penyakit dengan cara wawancara langsung pada dr.H.Iwan Mulyana di Puskesmas Kijang. Pedoman wawancara yang dilakukan berupa garis-garis besar permasalahan yang akan ditanyakan yaitu hal-

hal yang berkaitan dengan penyakit infeksi saluran pernafasan akut. Berkas hasil wawancara akan dilampirkan pada halaman lampiran.

## 2. Dokumentasi

Menurut (Sudaryono, 2015:92) dokumentasi ditujukan untuk memperoleh data langsung dari tempat penelitian, meliputi buku, peraturan, laporan kegiatan, foto, film dokumenter, dan data yang relevan dengan penelitian. Dokumen merupakan catatan peristiwa yang sudah berlalu. Dokumen bisa berbentuk tulisan, gambar, atau karya monumental seseorang. Dokumen yang berbentuk tulisan, misalnya, adalah catatan harian, sejarah kehidupan, cerita, biografi, peraturan, dan kebijakan; dokumen yang berbentuk gambar: foto, gambar hidup, sketsa, dan sebagainya; dokumen yang berbentuk karya: gambar, patung, film, dan sebagainya.

Buku yang dipakai sebagai referensi adalah buku Panduan Praktik Klinis Bagi Dokter (Zainuddin et al., 2016), buku Webtips PHP, HTML 5 dan CSS3 (Saputra, 2012), buku Kolaborasi PHP 5 & MySQL 5 untuk Pengembangan (Utomo, 2014), buku Metodologi Riset di Bidang TI (Sudaryono, 2015), buku Sistem Pakar & Pengembangannya (Hartati & Iswanti, 2008), buku Membuat Aplikasi POS dengan Laravel dan AJAX (Abdullah, 2017), buku Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek (A.S & Shalahuddin, 2015), buku *Artificial Intelligence Searching – Reasoning – Planning – Learning* (Suyanto, 2014), buku *Artificial Intelligence: Konsep dan Penerapannya* (Budihartono & Suhartono, 2014), buku Metode Penelitian Kuantitatif, Kualitatif dan R&D (Sugiyono, 2014), jurnal *Upper Respiratory Infections* (Grief, 2013), jurnal *Global Surgery 2030: Evidence and solutions for achieving health, welfare, and economic development* (Meara et al.,

2015), jurnal Implementasi *Framework Twitter Bootstrap* Dalam Perancangan Aplikasi Penerimaan Mahasiswa Baru Berbasis Web (Rosid & Jakaria, 2016), jurnal Perancangan Replikasi Basis Data *MySQL* Dengan Mekanisme Pengamanan Menggunakan *SSL Encryption* (Herman, 2014), jurnal Aplikasi Pengolahan Citra Berbasis Web Menggunakan *Javascript* dan *Jquery* (Yatini, 2014), jurnal Sistem Informasi Geografis Pemetaan Sarana Pada Kabupaten Pasaman Barat (Harison & Syarif, 2016), jurnal Perancangan Aplikasi *SMS Gateway* Untuk Pembuatan Kartu Perpustakaan di Fakultas Teknik Unsrat (Barri et al., 2015), jurnal Pembangunan Sistem Informasi Alumni Berbasis Web pada Program Studi Teknik Informatika Universitas Diponegoro (Binarso et al., 2012), jurnal Rancang Bangun Sistem Perpustakaan untuk Jurnal Elektronik (Februariyanti & Zuliarso, 2012), jurnal Karakteristik Penderita Faringitis Akut di Tentara Binjai Tahun 2017 (Nadeak, 2017), jurnal Aplikasi Pendiagnosis Hama dan Penyakit Tanaman Palawija menggunakan Metode Inferensi *Forward Chaining* (Bahtiar & Wibawa, 2013), Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode *Backpropagation* (Sudarsono, 2016), jurnal Sistem Pakar Diognosa Penyakit ISPA (Infeksi Saluran Pernapasan Akut) Dengan Menggunakan Metode *Certainty Factor* (Nurhayati, 2013), jurnal Sistem Penjualan Berbasis Web (*E-Commerce*) Pada Tata Distro Kabupaten Pacitan (Hastanti et al., 2015), jurnal Analisis Dan Perancangan Website Sebagai Sarana Informasi Pada Lembaga Bahasa Kewirausahaan dan Komputer AKMI Baturaja (Trimarsiah & Arafat, 2017), Analisa Aspek Balita terhadap Kejadian Infeksi Saluran Pernafasan Akut (ISPA) di rumah (Agrina, Suyanto, 2014), jurnal Media Pembelajaran Sistem

Pernapasan Pada Manusia Berbasis Multimedia (Mair & Supriadi, 2017), jurnal Penerapan Fuzzy Mamdani Untuk Sistem Pendukung Keputusan Pemilihan Telepon Seluler (Buana, 2014), jurnal Sistem Pakar Diagnosa Dampak Penggunaan *Soflens* Menggunakan Metode *Backward Chaining* (Mukhtar & Samsudin, 2014), jurnal *Medical Expert Systems for Diagnosis of Various Diseases* (Singla et al., 2014), jurnal *An expert system for the diagnosis and management of oral ulcers* (Ali & Saudi, 2014), jurnal Sistem Pakar Berbasis Web Dengan Metode *Forward Chaining* Dalam Mendiagnosis Dini Penyakit Tuberkulosis di Jawa Timur (Supartini & Hindarto, 2016), jurnal Sistem Pakar Diagnosa Penyakit THT (Lisnawita, Lhaura Van & Lindra, 2016), jurnal Sistem Pakar untuk Diagnosa Penyakit yang Disebabkan oleh Nyamuk Berbasis Web (Syamsuddin & Ahyuna, 2014), jurnal Penerapan Sistem Pakar Untuk Mendiagnosa Penyakit Pecernaan Dengan Pengobatan Bahan Alami (Ashari & Muniar, 2016), jurnal Sistem Pakar Mendeteksi Penyakit Tanaman Terong Belanda dengan Menggunakan Metode *Forward Chaining* (Sinaga, 2014), Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode *Forward Chaining* Berbasis *Mobile* (Pasalli et al., 2016) dan jurnal Sistem Pakar Untuk Mendeteksi Penyakit Gigi Pada Manusia (Nurlaela, 2013).

### **3.3 Operasional Variabel**

Menurut (Sudaryono, 2015:17-18) variabel adalah segala sesuatu yang memiliki variasi nilai. Contohnya, nilai ujian bervariasi dari 0-100; tingkat motivasi



bisa bervariasi dari sangat rendah hingga sangat tinggi; tingkat kepuasan konsumen bervariasi dari sangat rendah hingga sangat tinggi. Variabel mempunyai tiga ciri, yaitu memiliki variasi nilai, membedakan satu objek dengan objek lain dalam satu populasi, dan dapat diukur. Karena membedakan objek-objek dalam satu populasi, variabel harus mempunyai nilai yang bervariasi. Sebagai contoh, dari populasi penduduk yang mendiami suatu wilayah, jenis pekerjaan atau profesi bukan merupakan variabel apabila seluruh penduduk memiliki pekerjaan atau profesi yang sama.

Variabel yang digunakan dalam penelitian ini adalah penyakit infeksi saluran pernafasan akut yaitu khususnya penyakit *Faringitis*, *Laringitis* dan *Tonsilitis*.

**Tabel 3.1** Variabel dan Indikator

Variabel	Indikator
Penyakit Infeksi Saluran Pernafasan Akut	<i>Faringitis</i> <i>Laringitis</i> <i>Tonsilitis</i>

(Sumber: Pengolahan Data Penelitian, 2018)

### 3.4 Metode Perancangan Sistem

Perancangan sistem dapat kita gambarkan seperti suatu bagan alir yang menjelaskan keseluruhan proses yang kita lakukan secara detail sebelum kita membuat program, yang dapat meliputi perancangan sistem, perancangan masukan

dan keluaran, perancangan proses, perancangan antarmuka dan perancangan struktur data (Sudaryono, 2015:230).

Pada penelitian ini perancangan sistem dimulai dari melakukan studi pendahuluan seperti melakukan wawancara dengan pakar dan dokumentasi untuk memperoleh data-data yang berkaitan dengan pokok bahasan penelitian yang digunakan untuk membangun sistem, kemudian perancangan proses, antarmuka dan basis data dilakukan secara bersamaan.

### **3.4.1 Desain Basis Pengetahuan**

Sebelum desain basis pengetahuan dilakukan, peneliti telah melakukan proses akuisi pengetahuan dengan mengumpulkan fakta dan pengetahuan dari sumber-sumber yang tersedia. Sumber pengetahuan dan fakta diperoleh melalui wawancara dengan dokter umum dan dokumentasi tentang materi yang berkaitan dengan penyakit infeksi saluran pernafasan akut. Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan penyakit infeksi saluran pernafasan akut, gejala penyakit dan juga solusi mengatasinya. Pengetahuan dan fakta tersebut ditampilkan dengan tabel-tabel berikut:

Tabel data indikator berikut menjelaskan kode yang digunakan pada penelitian ini kepada masing-masing indikator.

**Tabel 3.2** Data Indikator

<b>Kode</b>	<b>Nama indikator</b>
IND001	Faringitis
IND002	Laringitis
IND003	Tonsilitis

(Sumber: Pengolahan Data Penelitian, 2018)

Tabel indikator penyakit infeksi saluran pernafasan akut berikut menjelaskan indikator pada penelitian ini yaitu *faringitis*, *laringitis* dan *tonsilitis* berserta gejala, solusi dan konselingnya.

**Tabel 3.3** Indikator Penyakit Infeksi Saluran Pernafasan Akut

<b>Indikator</b>	<b>Gejala Penyakit</b>	<b>Solusi</b>	<b>Konseling dan Edukasi</b>
<i>Faringitis</i>	<ol style="list-style-type: none"> <li>1. Nyeri tenggorakan saat menelan</li> <li>2. Demam</li> <li>3. Sekret dari hidung</li> <li>4. Nyeri Kepala</li> <li>5. Mual</li> <li>6. Muntah</li> <li>7. Rasa lemah pada tubuh</li> <li>8. Nafsu makan berkurang</li> </ol>	<ol style="list-style-type: none"> <li>1. Istirahat cukup</li> <li>2. Minum air putih yang cukup</li> <li>3. Berkumur dengan air yang hangat dan berkumur dengan obat kumur antiseptik untuk menjaga kebersihan mulut.</li> <li>4. Konsumsi anti virus Isoprinosine dengan dosis 60-100mg/kgBB dibagi dalam 4-6 x/hari pada orang dewasa dan pada anak &lt;5 tahun diberikan 50mg/kgBB dibagi dalam 4-6 x/hari</li> </ol>	<ol style="list-style-type: none"> <li>1. Menjaga daya tahan tubuh dengan mengkonsumsi makanan bergizi dan olahraga teratur</li> <li>2. Berhenti merokok bagi anggota keluarga yang merokok</li> <li>3. Menghindari makan makanan yang dapat mengiritasi tenggorakan</li> <li>4. Selalu menjaga kebersihan mulut dan tangan</li> </ol>

Tabel 3.3 Lanjutan

<i>Laringitis</i>	<ol style="list-style-type: none"> <li>1. Suara serak</li> <li>2. Sesak nafas</li> <li>3. Nyeri tenggorakan saat menelan</li> <li>4. Demam</li> <li>5. Batuk kering disertai dengan dahak kental</li> </ol>	<ol style="list-style-type: none"> <li>1. Istirahat suara</li> <li>2. Rehabilitasi suara</li> <li>3. Tingkatkan asupan cairan</li> <li>4. Konsumsi Parasetamol atau Ibuprofen sebagai antipiretik dan analgetik</li> </ol>	<ol style="list-style-type: none"> <li>1. Menjaga daya tahan tubuh dengan mengkonsumsi makanan bergizi dan olahraga teratur</li> <li>2. Berhenti merokok bagi anggota keluarga yang merokok</li> <li>3. Istirahat berbicara dan bersuara atau tidak bersuara berlebihan</li> <li>4. Menghindari makanan yang mengiritasi atau meningkatkan asam lambung</li> </ol>
<i>Tonsilitis</i>	<ol style="list-style-type: none"> <li>1. Rasa kering di tenggorakan</li> <li>2. Nyeri pada tenggorakan saat menelan</li> <li>3. Nyeri menyebar sampai di telinga</li> <li>4. Demam yang sangat tinggi</li> <li>5. Nyeri Kepala</li> <li>6. Rasa lemah pada tubuh</li> <li>7. Nafsu makan berkurang</li> <li>8. Mulut berbau</li> <li>9. Ludah menumpuk</li> </ol>	<ol style="list-style-type: none"> <li>1. Istirahat yang cukup</li> <li>2. Makan makanan lunak dan menghindari makan makan mengiritasi</li> <li>3. Menjaga kebersihan mulut</li> <li>4. Konsumsi Obat topikal dapat berupa obat kumur antiseptik</li> <li>5. Konsumsi obat soral sistemik</li> </ol>	<ol style="list-style-type: none"> <li>1. Menghindari makanan dan minuman yang mengiritasi</li> <li>2. Melakukan pengobatan yang adekuat karena risiko kekambuhan cukup tinggi</li> <li>3. Menjaga daya tahan tubuh dengan mengkosumsi makanan bergizi dan olahraga teratur</li> <li>4. Berhenti merokok</li> <li>5. Selalu menjaga kebersihan mulut.</li> <li>6. Mencuci tangan secara teratur.</li> </ol>

(Sumber: Pengolahan Data Penelitian, 2018)

Tabel gejala penyakit infeksi saluran pernafasan akut berikut menjelaskan kode yang digunakan pada penelitian ini kepada masing-masing gejala.

**Tabel 3.4** Gejala Penyakit Infeksi Saluran Pernafasan Akut

<b>Kode</b>	<b>Nama gejala</b>
GJL001	Nyeri tenggorokan saat menelan
GJL002	Demam
GJL003	Sekret dari hidung
GJL004	Mual
GJL005	Muntah
GJL006	Nyeri kepala
GJL007	Rasa lemah pada tubuh
GJL008	Nafsu makan berkurang
GJL009	Rasa kering ditenggorokan
GJL010	Nyeri menyebar sampai di telinga
GJL011	Demam yang sangat tinggi
GJL012	Mulut berbau
GJL013	Ludah menumpuk
GJL014	Suara serak
GJL015	Sesak nafas
GJL016	Batuk kering disertai dengan dahak kental

(Sumber: Pengolahan Data Penelitian, 2018)

Tabel gejala *faringitis* berikut menjelaskan kode gejala yang merupakan gejala dari penyakit *faringitis*.

**Tabel 3.5** Gejala *Faringitis*

<b>Kode</b>	<b>Nama Gejala</b>
GJL001	Nyeri tenggorokan saat menelan
GJL002	Demam
GJL003	Sekret dari hidung
GJL004	Mual
GJL005	Muntah
GJL006	Nyeri kepala
GJL007	Rasa lemah pada tubuh
GJL008	Nafsu makan berkurang

(Sumber: Pengolahan Data Penelitian, 2018)

Tabel gejala *laringitis* berikut menjelaskan kode gejala yang merupakan gejala dari penyakit *laringitis*.

**Tabel 3.6** Gejala *Laringitis*

<b>Kode</b>	<b>Nama Gejala</b>
GJL001	Nyeri tenggorokan saat menelan
GJL002	Demam
GJL014	Suara serak
GJL015	Sesak nafas
GJL016	Batuk kering disertai dengan dahak kental

(Sumber: Pengolahan Data Penelitian, 2018)

Tabel gejala *tonsilitis* berikut menjelaskan kode gejala yang merupakan gejala dari penyakit *tonsilitis*.

**Tabel 3.7** Gejala *Tonsilitis*

Kode	Nama Gejala
GJL001	Nyeri tenggorokan saat menelan
GJL006	Nyeri kepala
GJL007	Rasa lemah pada tubuh
GJL008	Nafsu makan berkurang
GJL009	Rasa kering ditenggorokan
GJL010	Nyeri menyebar sampai di telinga
GJL011	Demam yang sangat tinggi
GJL012	Mulut berbau
GJL013	Ludah menumpuk

(Sumber: Pengolahan Data Penelitian, 2018)

Tabel indikator dan gejala penyakit infeksi saluran pernafasan akut berikut menjelaskan hubungan antara penyakit dan gejala dalam bentuk kode penyakit dan kode gejala.

**Tabel 3.8** Indikator dan Gejala Penyakit Infeksi Saluran Pernafasan Akut

Kode Penyakit	Kode Gejala
IND001	GJL001, GJL002, GJL003, GJL005, GJL005, GJL006, GJL007, GJL008
IND002	GJL001, GJL002, GJL014, GJL015, GJL016
IND003	GJL001, GJL006, GJL007, GJL008, GJL009, GJL010, GJL011, GJL012, GJL013

(Sumber: Pengolahan Data Penelitian, 2018)

Berdasarkan data aturan yang telah disusun, maka kaidah (*rule*) yang akan digunakan dalam sistem pakar adalah sebagai berikut:

1. Kaidah 1: IF GJL001 AND GJL002 AND GJL003 AND GJL004 AND GJL005 AND GJL006 AND GJL007 AND GJL008 THEN IND001
2. Kaidah 2: IF GJL001 AND GJL002 AND GJL014 AND GJL015 AND GJL016 THEN IND002
3. Kaidah 3: IF GJL001 AND GJL006 AND GJL007 AND GJL008 AND GJL009 AND GJL010 AND GJL011 AND GJL012 AND GJL013 THEN IND003

Berdasarkan kaidah yang telah disusun, maka berikut tabel keputusan yang digunakan dalam penelitian ini:

**Tabel 3.9** Tabel Keputusan

<b>Penyakit Gejala</b>	<b>IND 001</b>	<b>IND 002</b>	<b>IND 003</b>
GJL001	√	√	√
GJL002	√	√	
GJL003	√		
GJL004	√		
GJL005	√		
GJL006	√		√
GJL007	√		√
GJL008	√		√
GJL009			√
GJL010			√
GJL011			√
GJL012			√
GJL013			√
GJL014		√	

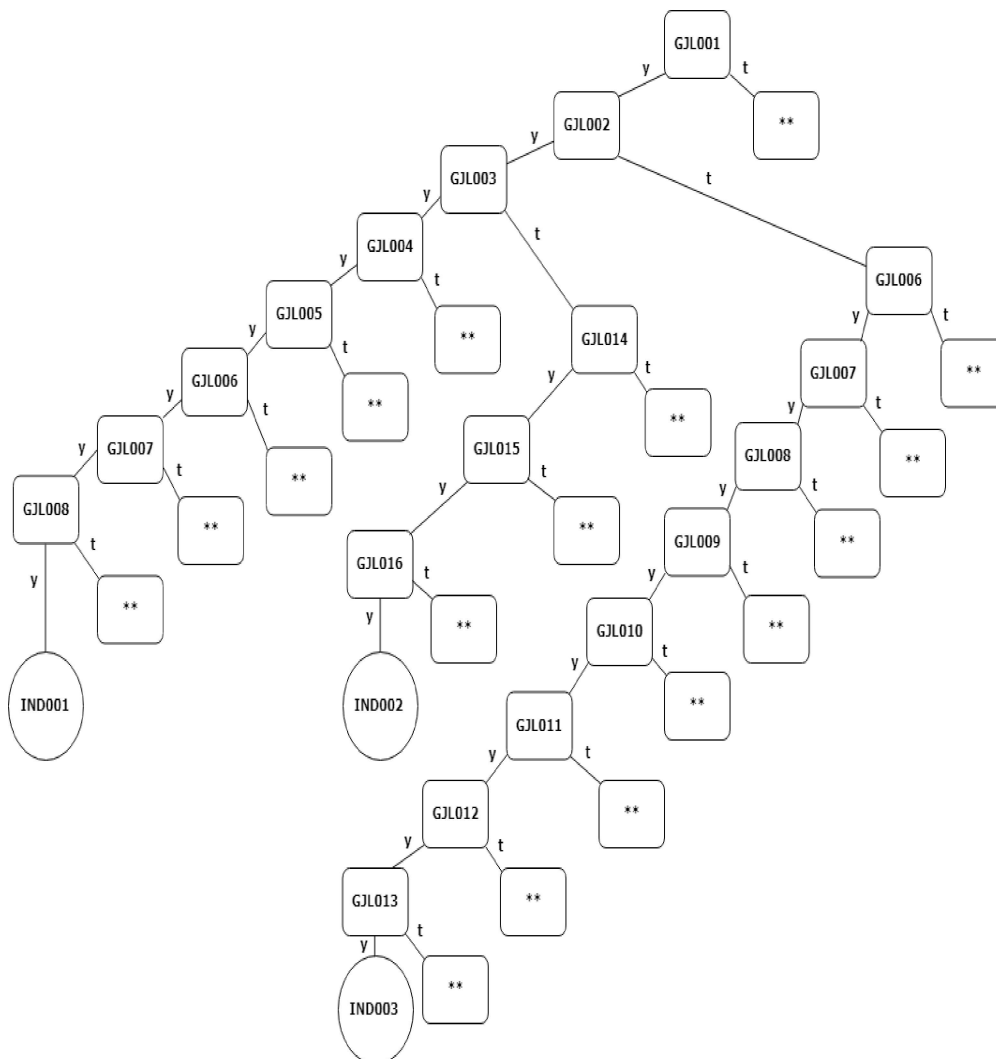


**Tabel 3.9 Lanjutan**

GJL015		√	
GJL016		√	

(Sumber: Pengolahan Data Penelitian, 2018)

Berdasarkan tabel keputusan tersebut maka pohon keputusannya adalah sebagai berikut:

**Gambar 3.2** Pohon Keputusan

## Keterangan:

GJL001 = Gejala 001	GJL009 = Gejala 009	IND001 = Indikator 001
GJL002 = Gejala 002	GJL010 = Gejala 010	IND002 = Indikator 002
GJL003 = Gejala 003	GJL011 = Gejala 011	IND003 = Indikator 003
GJL004 = Gejala 004	GJL012 = Gejala 012	y = ya
GJL005 = Gejala 005	GJL013 = Gejala 013	t = tidak
GJL006 = Gejala 006	GJL014 = Gejala 014	** = tidak menghasilkan simpul
GJL007 = Gejala 007	GJL015 = Gejala 015	
GJL008 = Gejala 008	GJL016 = Gejala 016	

Data gejala ditentukan sebagai keadaan awal dalam sistem saat melakukan penelusuran sebelum diperoleh sebuah kesimpulan. Pohon keputusan pada gambar 3.2 digunakan untuk memperlihatkan hubungan terkait antar gejala yang ada. Arah penelusuran pada pohon keputusan tersebut dimulai dari simpul akar (yang paling atas) ke bawah. Alur penelusuran sistem pakar ini dimulai dari GJL001, yaitu nyeri tenggorokan saat menelan. Gejala ini dipilih sebagai keadaan awal dalam penelusuran karena gejala ini adalah gejala yang dialami oleh semua penyakit pada penelitian ini.

Proses penelusuran selanjutnya tergantung bagaimana jawaban yang diberikan pengguna. Jika pengguna memberikan jawaban “ya”, maka penelusuran menuju simpul kiri pada level berikutnya GJL002 dan jika pengguna memberikan jawaban “tidak”, maka penelusuran menuju simpul kanan. Begitu seterusnya sampai penelusuran menemukan simpul IND atau simpul \*\*. Simpul IND tersebut merupakan bagian dari Indikator. Misalnya GJL001 yaitu indikator berada di IND 001, yaitu *Faringitis*. Simpul \*\* berarti tidak menghasilkan kesimpulan tertentu.

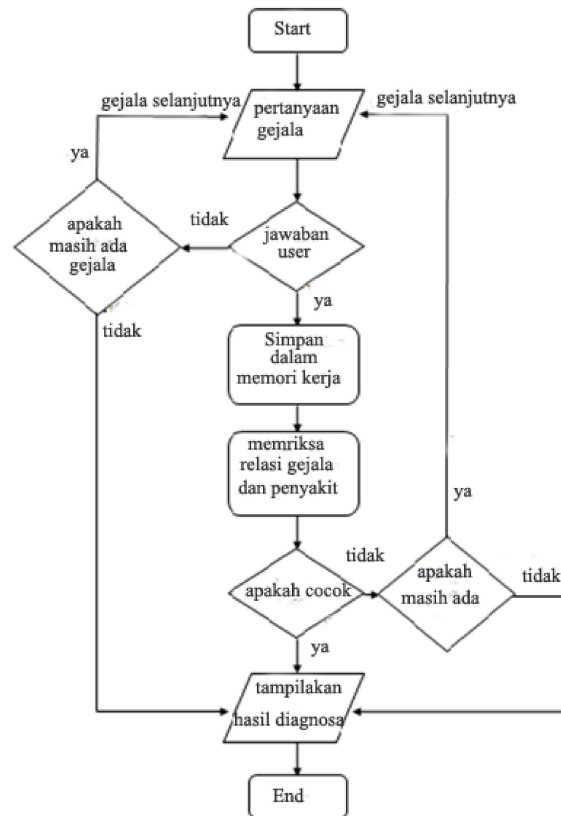
Pada sistem pakar ini, jika penelusuran menemukan simpul \*\* maka sistem akan menampilkan hasil dugaan tidak terjangkau penyakit infeksi saluran pernafasan akut.

### 3.4.2 Struktur Kontrol (Mesin Inferensi)

Mesin inferensi dalam sistem pakar ini menggunakan metode penelusuran *forward chaining*. Langkah-langkah yang digunakan dalam proses penelusurannya adalah sebagai berikut:

1. Mengajukan pertanyaan tentang gejala penyakit kepada pengguna.
2. Jika jawaban pengguna “Ya” maka sistem akan melakukan langkah 3. Jika jawaban pengguna “Tidak” maka sistem akan melakukan langkah 4.
3. Menyimpan gejala dalam memori kerja lalu memeriksa relasi gejala dengan penyakit yang telah dibuat. Jika ada relasi yang cocok maka sistem akan melakukan langkah 5. Jika tidak ada aturan yang cocok maka sistem akan melakukan langkah 4.
4. Memeriksa apakah masih ada gejala lain yang belum ditanyakan. Jika masih ada, maka sistem akan mengajukan pertanyaan tentang gejala penyakit selanjutnya kepada pengguna dan ulangi langkah 2 sampai dengan 4. Jika tidak ada, maka sistem akan melakukan langkah 5.
5. Menampilkan hasil diagnosa.

Berikut ini adalah gambar *flowchart* mesin inferensi yang digunakan dalam sistem pakar ini.



**Gambar 3.3** Flowchart Mesin Inferensi

### 3.4.3 Desain UML (*Unified Modeling Language*)

Desain sistem pada penelitian ini menggunakan bahasan permodelan *Unified Modeling Language (UML)* yang digambarkan dengan menggunakan aplikasi *StarUML* versi 3.0.1. Diagram UML yang digunakan dalam penelitian ini antara lain:

### 1. Use Case Diagram

Aktor yang digunakan dalam sistem pakar ini terdiri dari 2 orang yaitu admin dan *user*. Dalam sistem pakar ini yang berperan sebagai admin adalah peneliti sendiri sedangkan *user* adalah masyarakat umum yang ingin menangani permasalahan yang berkaitan dengan penyakit infeksi saluran pernafasan akut. *Use case* yang terdapat dalam sistem antara lain *login*, registrasi, mengelola daftar akun, mengelola daftar gejala, mengelola daftar penyakit, mengelola daftar relasi, mengelola daftar riwayat diagnosa, diagnosa, riwayat diagnosa dan *log out*.

*Use case diagram* yang dirancang untuk sistem pakar dalam penelitian ini dapat dilihat pada gambar dibawah ini.



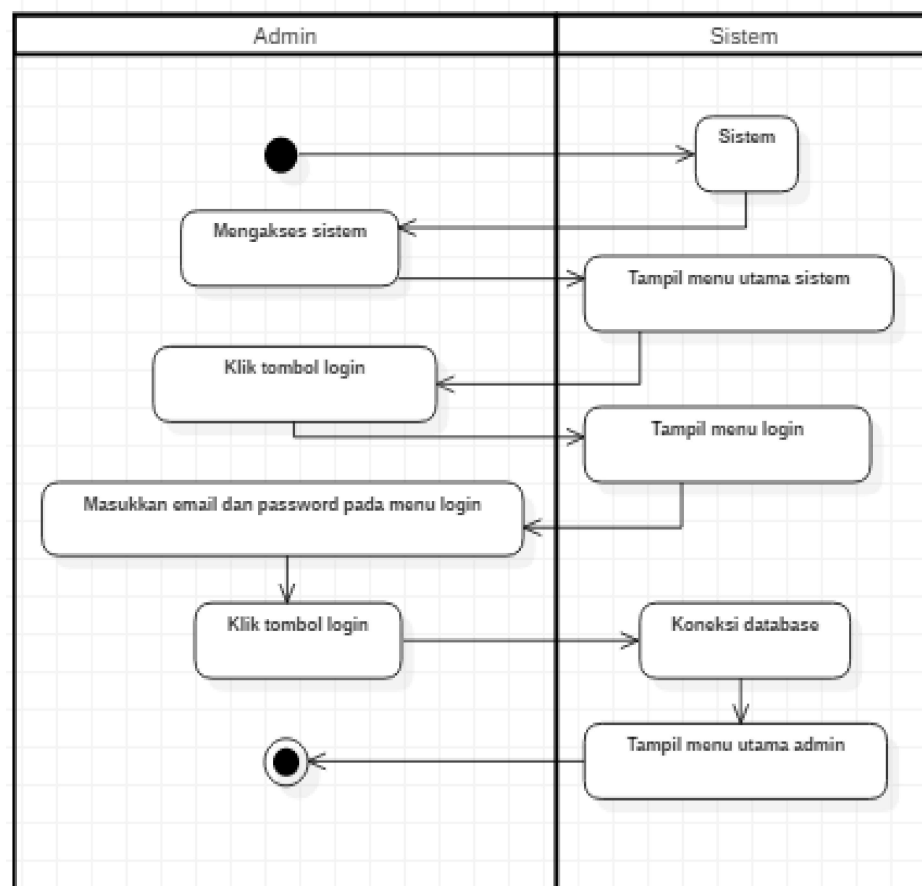
**Gambar 3.4** Use Case Diagram

## 2. Activity Diagram

*Activity diagram* menggambarkan aktifitas yang dapat dilakukan oleh sistem atau menu yang ada pada perangkat lunak, bukan apa yang dilakukan oleh aktor (A.S & Shalahuddin, 2015:161). *Activity diagram* yang dirancang untuk sistem pakar dalam penelitian ini akan ditunjukkan melalui gambar-gambar dibawah ini.

### a. Activity Diagram Admin Login

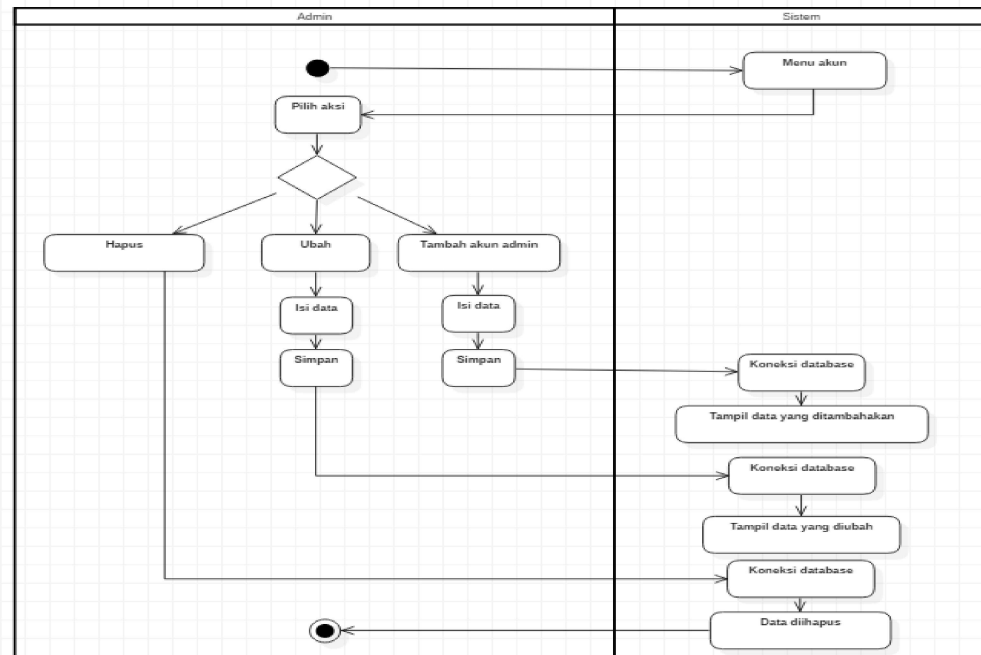
Gambar *Activity Diagram Admin Login* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *Admin* melakukan *Login*.



**Gambar 3.5** Activity Diagram Admin Login

b. *Activity Diagram Admin Mengelola Daftar Akun*

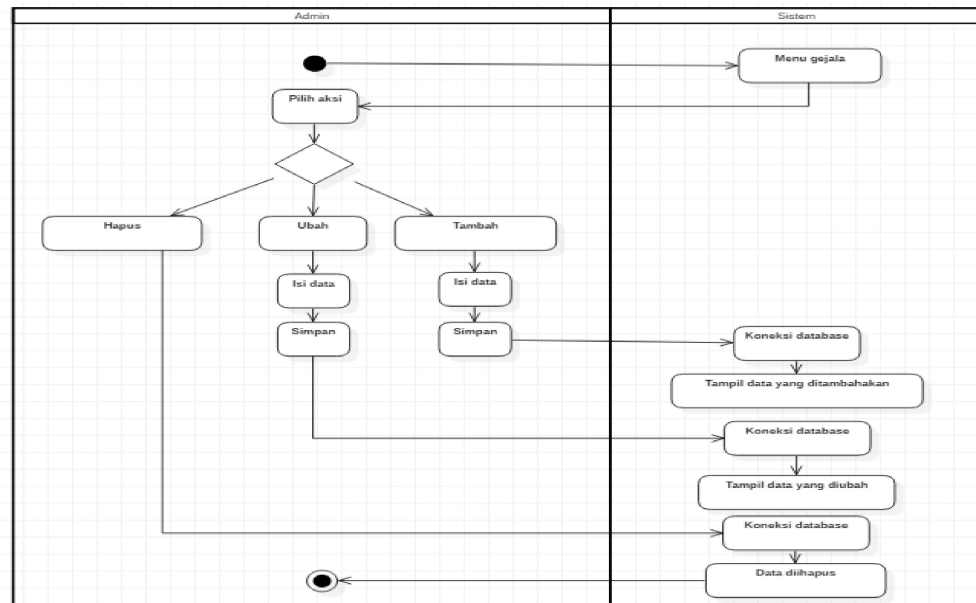
Gambar *Activity Diagram Admin Mengelola Daftar Akun* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *Admin* melakukan pengolahan daftar akun.



**Gambar 3.6** *Activity Diagram Admin Mengelola Daftar Akun*

c. *Activity Diagram Admin Mengelola Daftar Gejala*

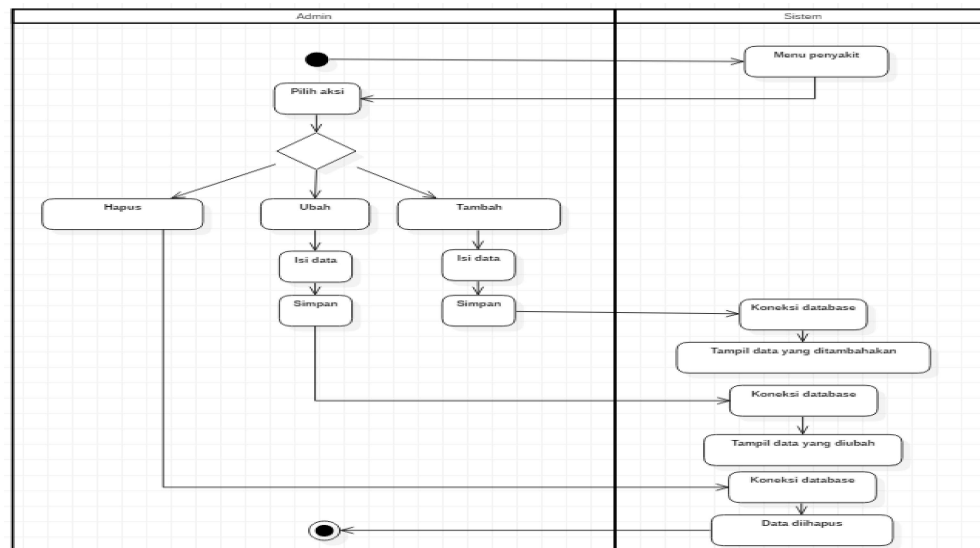
Gambar *Activity Diagram Admin Mengelola Daftar Gejala* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *Admin* melakukan pengolahan daftar gejala.



**Gambar 3.7** Activity Diagram Admin Mengelola Daftar Gejala

d. Activity Diagram Admin Mengelola Daftar Penyakit

Gambar Activity Diagram Admin Mengelola Daftar Penyakit berikut menjelaskan aktifitas yang dilakukan oleh sistem saat Admin melakukan pengolahan daftar penyakit.

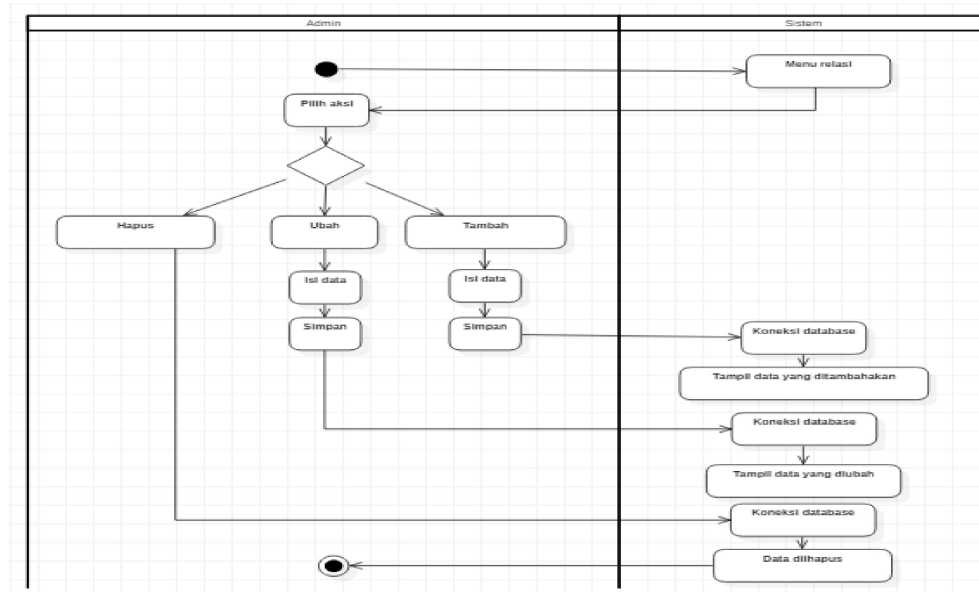


**Gambar 3.8** Activity Diagram Admin Mengelola Daftar Penyakit



e. *Activity Diagram Admin Mengelola Daftar Relasi*

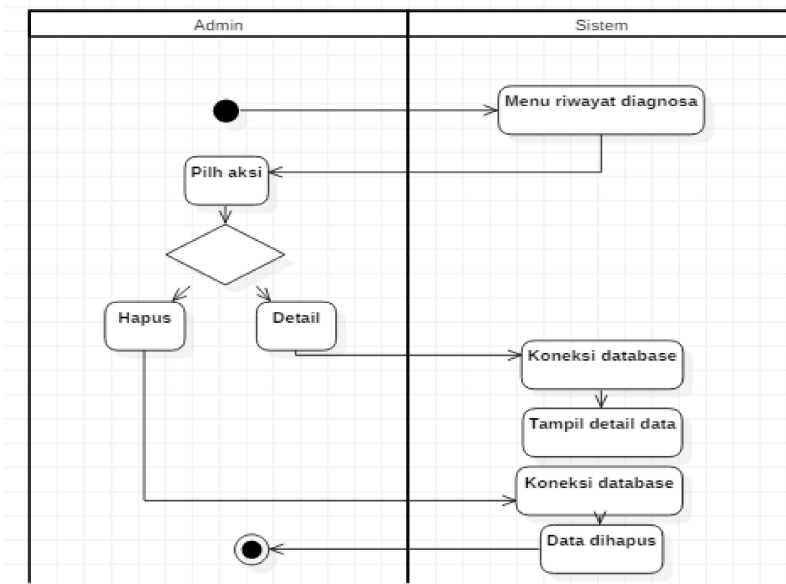
Gambar *Activity Diagram Admin Mengelola Daftar Relasi* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *Admin* melakukan pengolahan daftar relasi.



**Gambar 3.9** *Activity Diagram Admin Mengelola Daftar Relasi*

f. *Activity Diagram Admin Mengelola Daftar Riwayat Diagnosa*

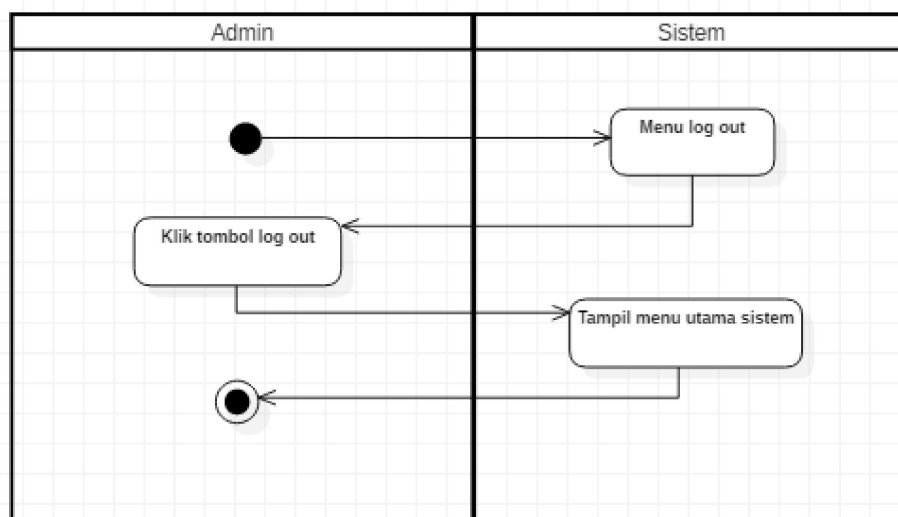
Gambar *Activity Diagram Admin Mengelola Daftar Riwayat Diagnosa* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *Admin* melakukan pengolahan daftar riwayat diagnosa.



**Gambar 3.10** Activity Diagram Admin Mengelola Daftar Riwayat Diagnosa

g. Activity Diagram Admin Log out

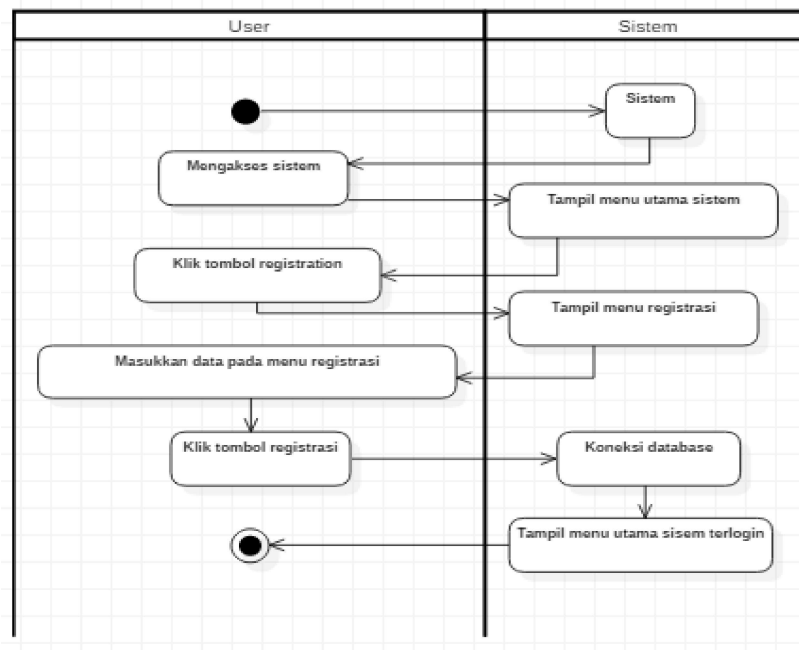
Gambar Activity Diagram Admin Log out berikut menjelaskan aktifitas yang dilakukan oleh sistem saat Admin melakukan Log out.



**Gambar 3.11** Activity Diagram Admin Log out

#### h. *Activity Diagram User Registrasi*

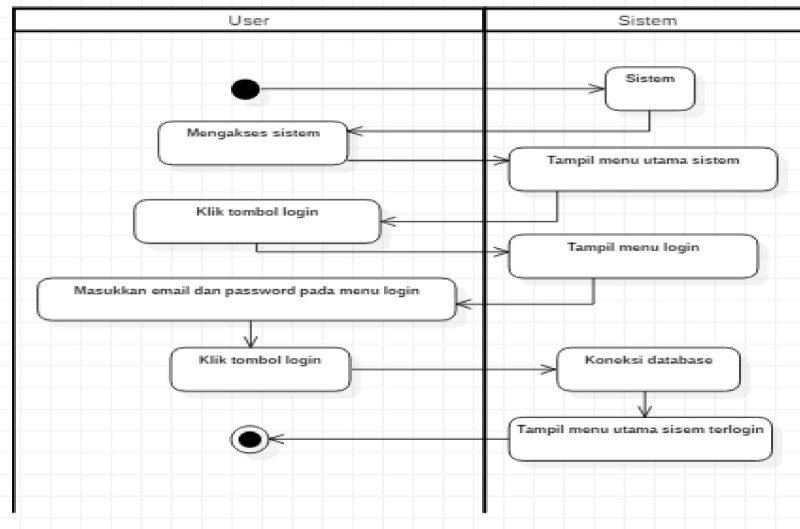
Gambar *Activity Diagram User Registrasi* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *User* melakukan registrasi.



**Gambar 3.12** *Activity Diagram User Registrasi*

#### i. *Activity Diagram User Login*

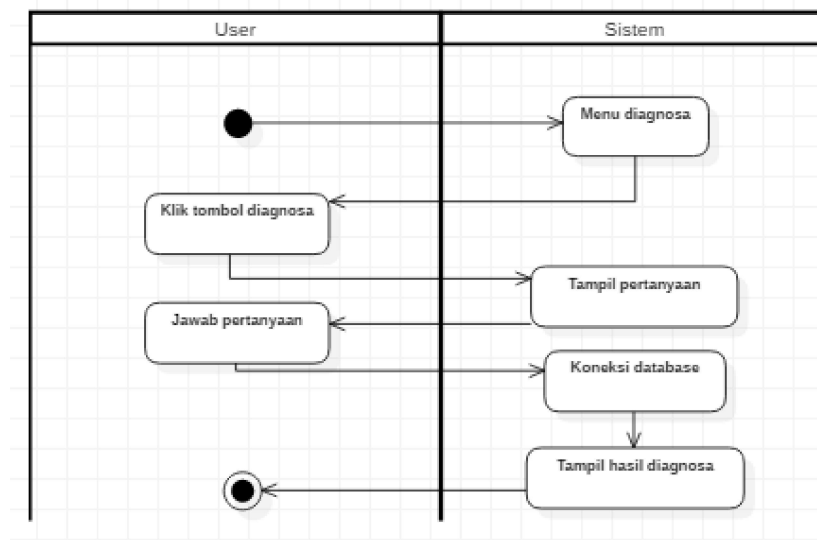
Gambar *Activity Diagram User Login* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *User* melakukan *Login*.



**Gambar 3.13** Activity Diagram User Login

j. Activity Diagram User Diagnosa

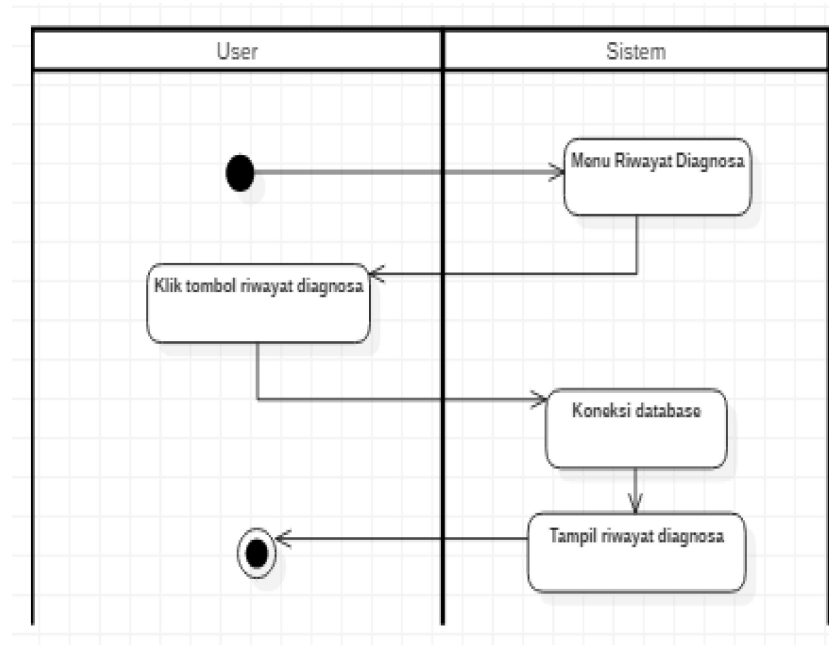
Gambar Activity Diagram User Diagnosa berikut menjelaskan aktifitas yang dilakukan oleh sistem saat User melakukan diagnosa.



**Gambar 3.14** Activity Diagram User Diagnosa

k. *Activity Diagram User Riwayat Diagnosa*

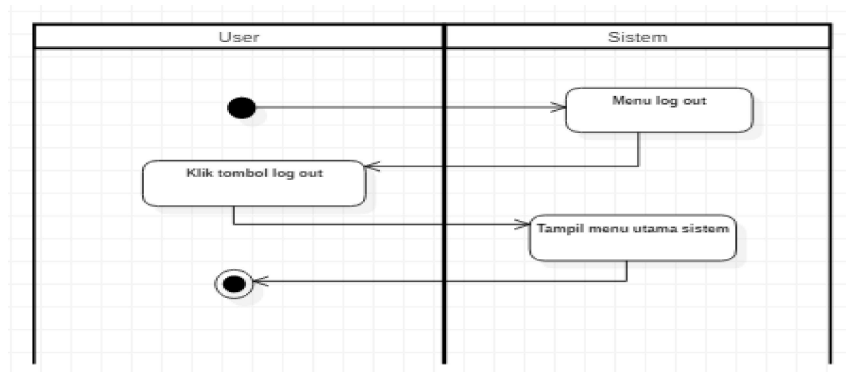
Gambar *Activity Diagram User Riwayat Diagnosa* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *User* mengakses menu riwayat diagnosa.



**Gambar 3.15** *Activity Diagram User Riwayat Diagnosa*

l. *Activity Diagram User Log out*

Gambar *Activity Diagram User Log out* berikut menjelaskan aktifitas yang dilakukan oleh sistem saat *User* melakukan *Log out*.



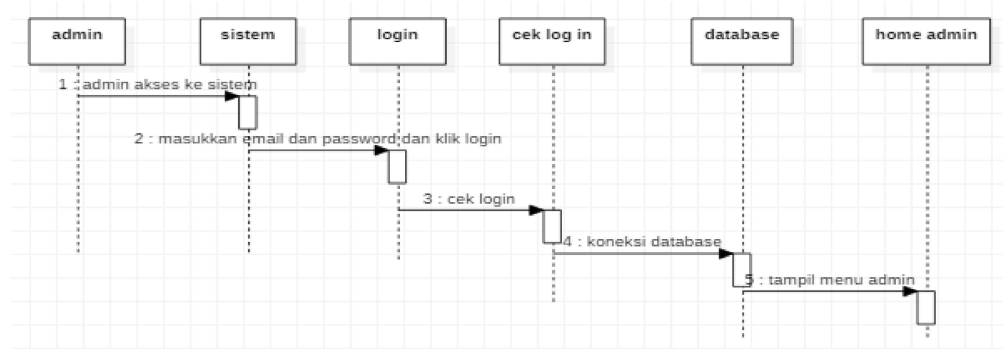
**Gambar 3.16** *Activity Diagram User Log out*

### 3. *Sequence Diagram*

*Sequence diagram* merupakan diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (A.S & Shalahuddin, 2015:165). Berikut ini adalah gambar-gambar *sequence diagram* yang digunakan dalam sistem pakar pada penelitian ini.

#### a. *Sequence Diagram Admin Login*

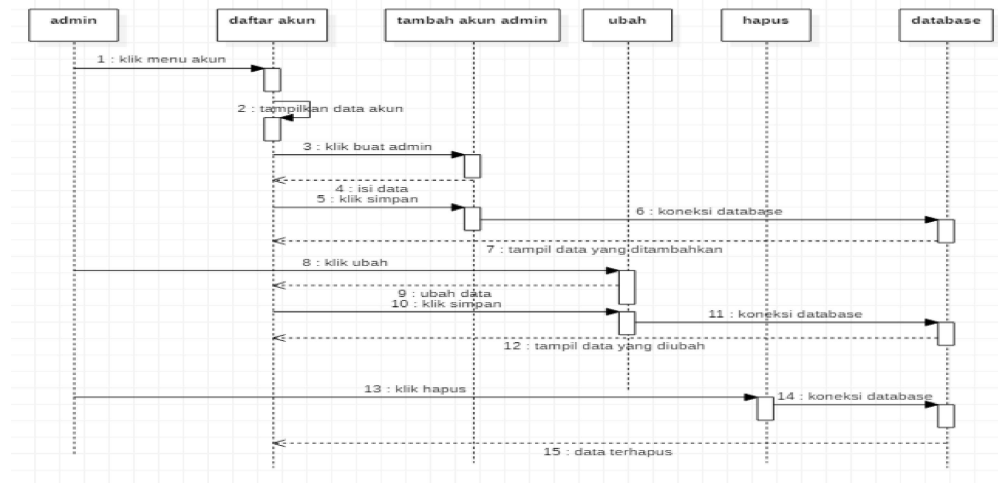
Gambar *Sequence Diagram Admin Login* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan *login*.



**Gambar 3.17** *Sequence Diagram Amin Login*

#### b. *Sequence Diagram Admin Mengelola Daftar Akun*

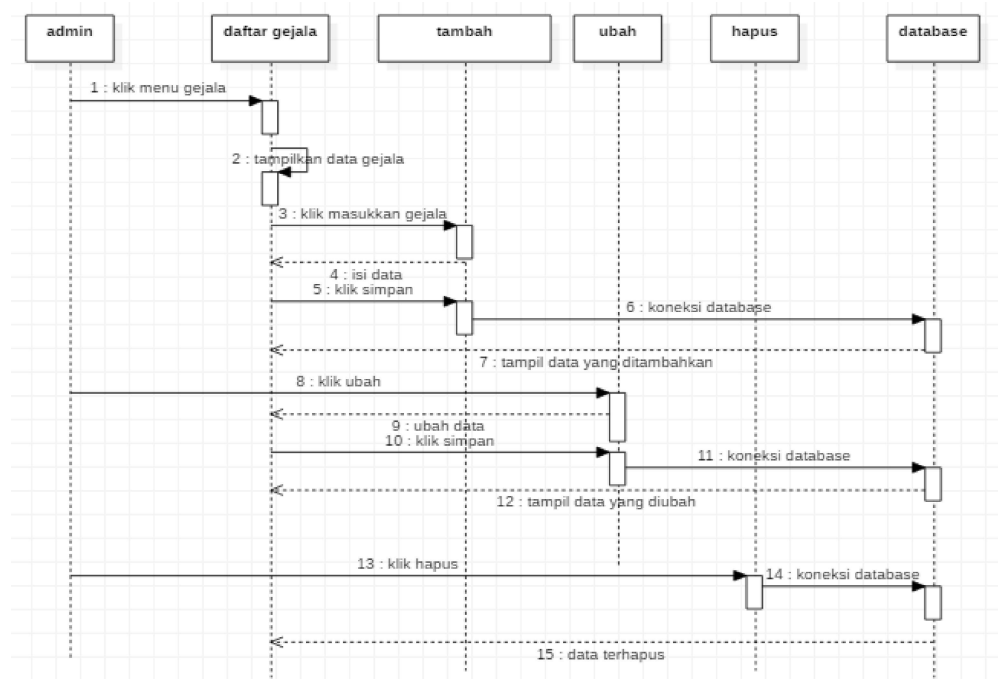
Gambar *Sequence Diagram Admin Mengelola Daftar Akun* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan pengolahan daftar akun.



**Gambar 3.18** *Sequence Diagram Admin Mengelola Daftar Akun*

c. *Sequence Diagram Admin Mengelola Daftar Gejala*

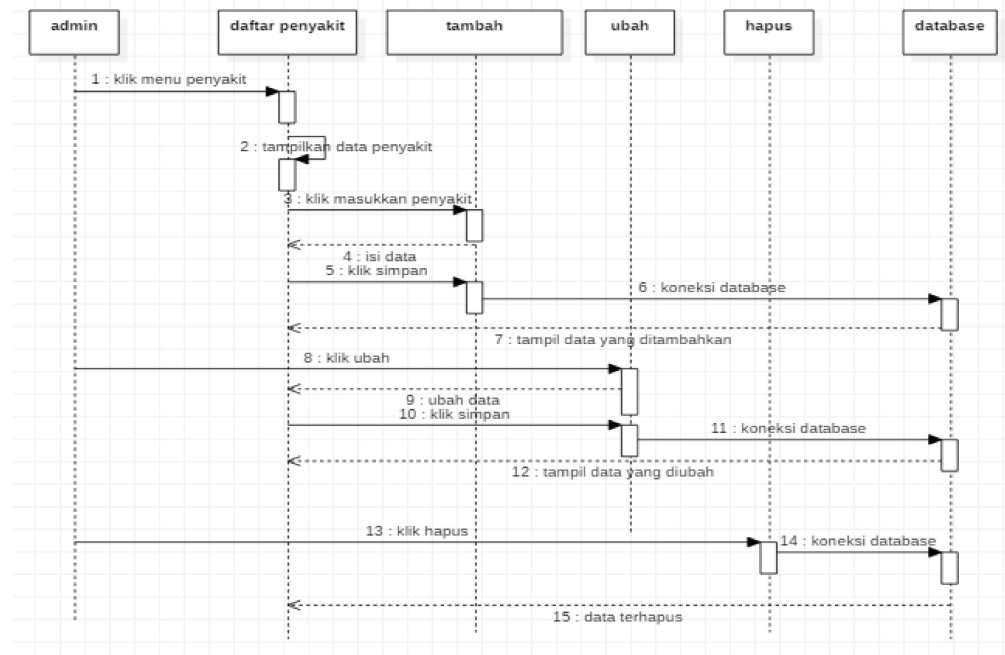
Gambar *Sequence Diagram Admin Mengelola Daftar Gejala* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan pengolahan daftar gejala.



**Gambar 3.19** *Sequence Diagram Admin Mengelola Daftar Gejala*

d. *Sequence Diagram Admin Mengelola Daftar Penyakit*

Gambar *Sequence Diagram Admin Mengelola Daftar Penyakit* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan pengolahan daftar penyakit.

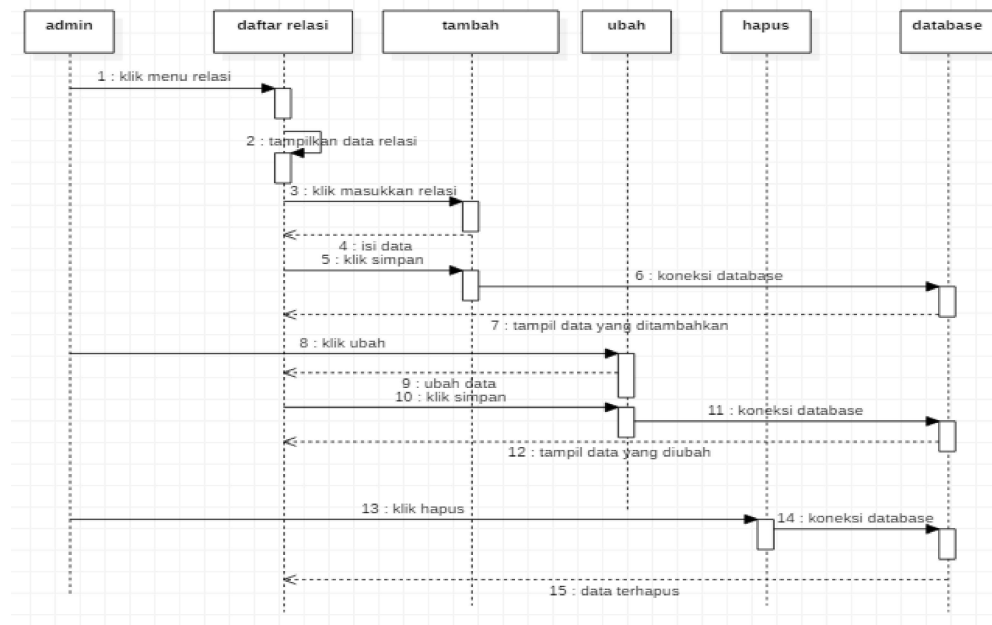


**Gambar 3.20** *Sequence Diagram Admin Mengelola Daftar Penyakit*

e. *Sequence Diagram Admin Mengelola Daftar Relasi*

Gambar *Sequence Diagram Admin Mengelola Daftar Relasi* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan pengolahan daftar relasi.

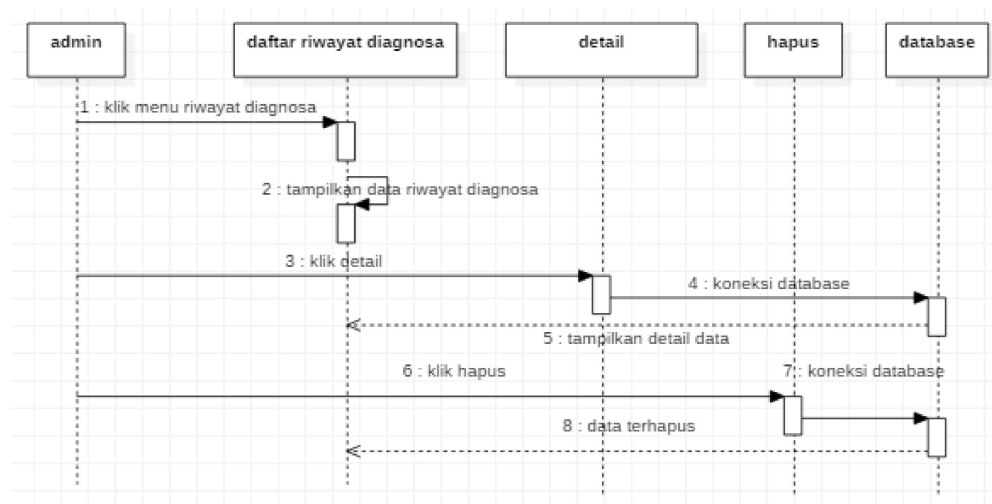




**Gambar 3.21** *Sequence Diagram Admin Mengelola Daftar Relasi*

f. *Sequence Diagram Admin Mengelola Daftar Riwayat Diagnosa*

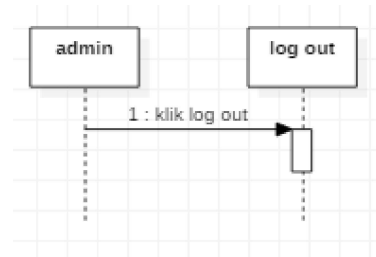
Gambar *Sequence Diagram Admin Mengelola Daftar Riwayat Diagnosa* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan pengolahan daftar riwayat diagnosa.



**Gambar 3.22** *Sequence Diagram Admin Mengelola Daftar Riwayat Diagnosa*

g. *Sequence Diagram Admin Log out*

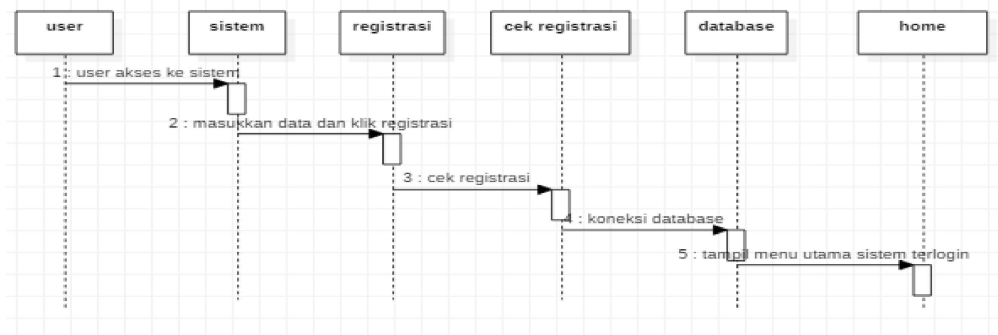
Gambar *Sequence Diagram Admin Log out* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *admin* melakukan *log out*.



**Gambar 3.23** *Sequence Diagram Admin Log out*

h. *Sequence Diagram User Registrasi*

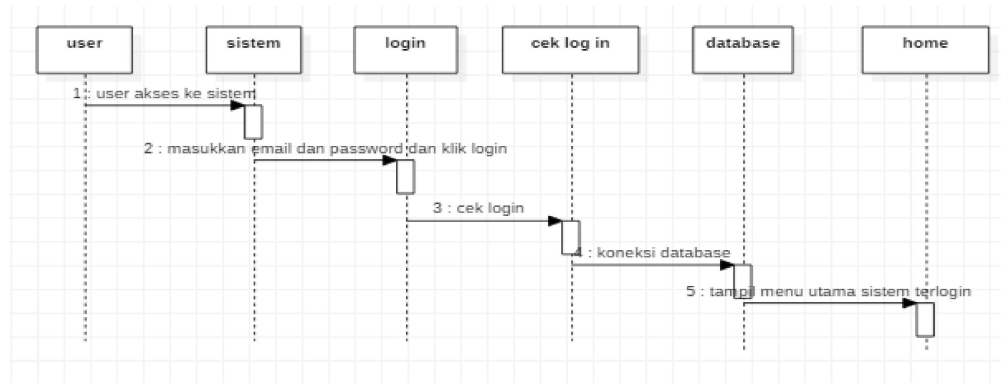
Gambar *Sequence Diagram User Registrasi* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *user* melakukan registrasi.



**Gambar 3.24** *Sequence Diagram User Registrasi*

i. *Sequence Diagram User Login*

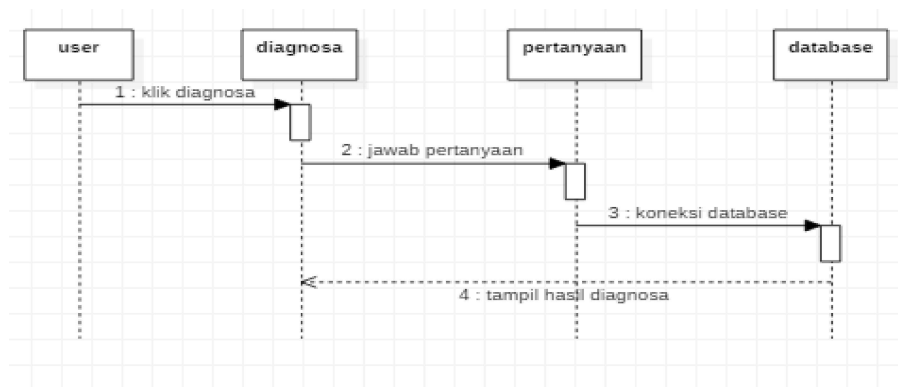
Gambar *Sequence Diagram User Login* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *user* melakukan *Login*.



**Gambar 3.25** *Sequence Diagram User Login*

j. *Sequence Diagram User Diagnosa*

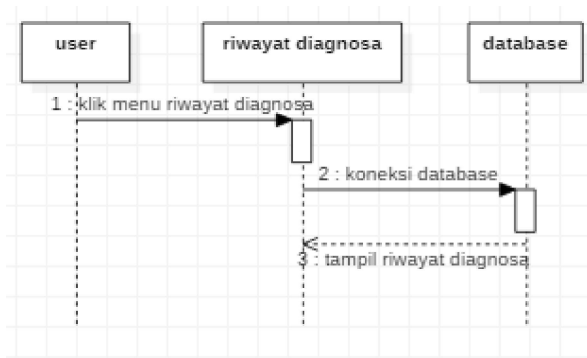
Gambar *Sequence Diagram User Diagnosa* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *user* melakukan diagnosa.



**Gambar 3.26** *Sequence Diagram User Diagnosa*

k. *Sequence Diagram User Riwayat Diagnosa*

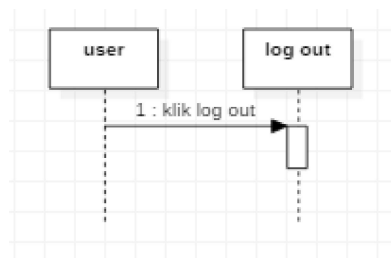
Gambar *Sequence Diagram User Riwayat Diagnosa* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *user* mengakses menu riwayat diagnosa.



**Gambar 3.27** *Sequence Diagram User Riwayat Diagnosa*

1. *Sequence Diagram User Log out*

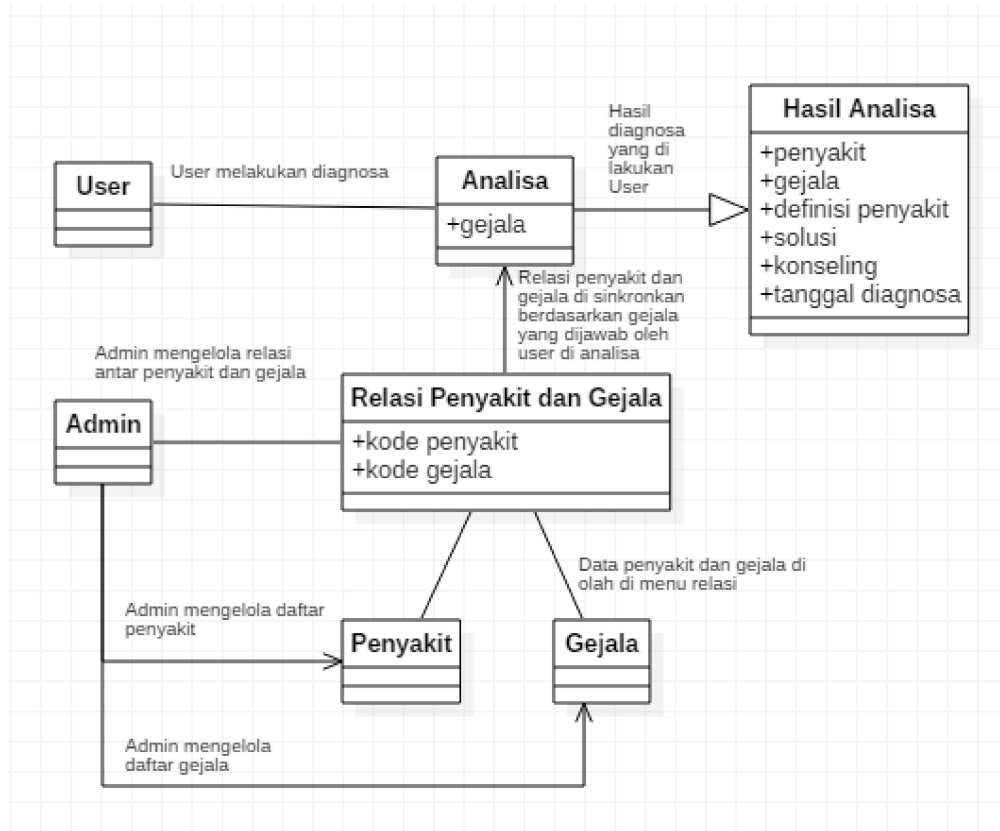
Gambar *Sequence Diagram User Log out* berikut menggambarkan kelakuan objek terhadap *use case* sistem saat *user* melakukan *log out*.



**Gambar 3.28** *Sequence Diagram User Log out*

4. *Class Diagram*

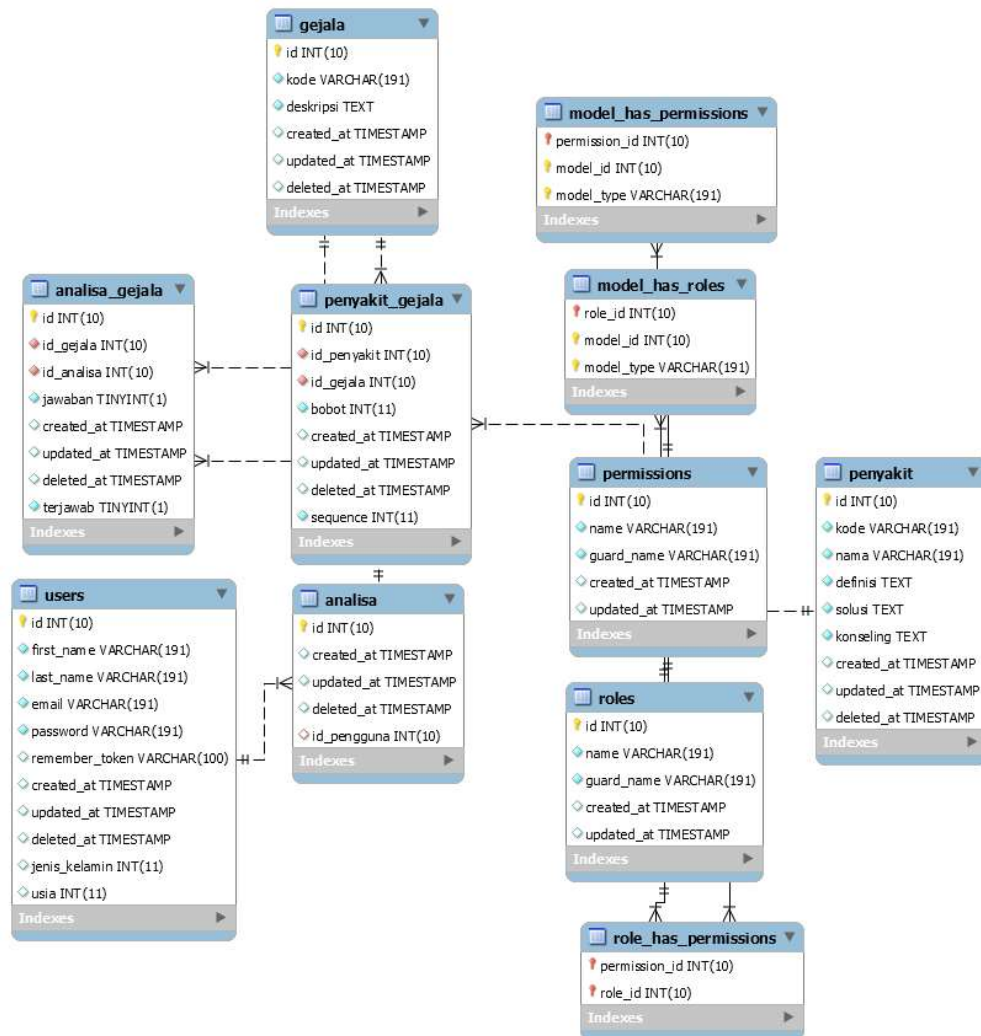
Dalam penelitian ini, peneliti membuat desain *database* menggunakan teknik pemodelan *Class diagram*. Berikut ini adalah gambar model relasional yang digunakan dalam sistem pakar ini:



**Gambar 3.29** *Class Diagram*

#### 3.4.4 Desain Database

Dalam penelitian ini, peneliti membuat desain dataase dengan menggunakan teknik permodelan *Entity Relationship Diagram (ERD)*. Berikut ini adalah gambar model relasional yang digunakan pada sistem pakar ini:



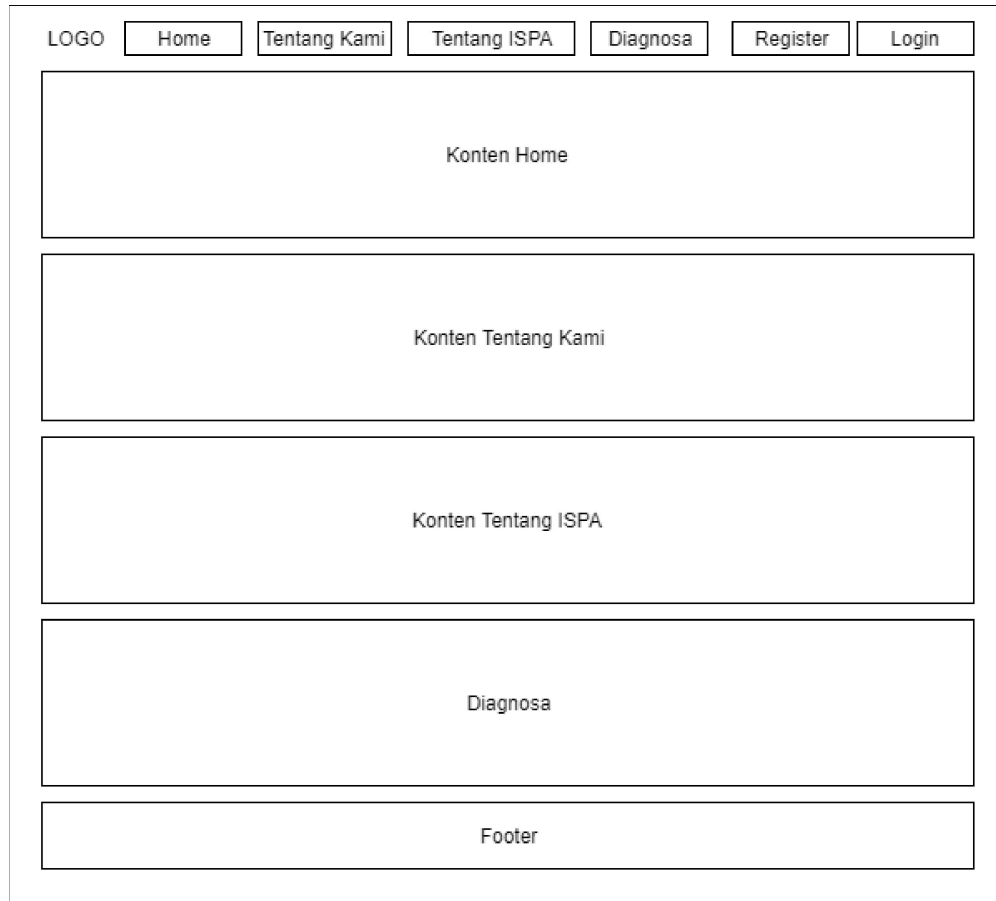
**Gambar 3.30** Entity Relationship Diagram

### 3.4.5 Prototype

Berikut adalah desain tampilan sistem pakar untuk mendiagnosa penyakit infeksi saluran pernafasan akut:

#### 1. Menu Utama

Tampilan awal atau *landing page* sistem pakar diagnosa infeksi saluran pernafasan akut.



**Gambar 3.31** Tampilan Menu Utama

## 2. Menu *Admin*

Tampilan dalam sistem pakar diagnosa infeksi saluran pernafasan akut yang diakses oleh *Admin*.

### a. Tampilan *Login Admin*

Berikut ialah desain tampilan *login admin* pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk *login* ke dalam tampilan utama *admin* sistem.

Sign In	
email	
password	
<input type="checkbox"/>	Remember Me
LOGIN	

**Gambar 3.32** Tampilan *Login Admin*

*b.* Tampilan Utama *Admin*

Berikut ialah desain tampilan utama *admin* pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk mengelolah akun, gejala, penyakit, relasi dan riwayat diagnosa sistem.

<b>Admin Panel</b>		
Nama Admin		
Log out	Dashboard	
<b>Main menu</b>	Jumlah user      Jumlah diagnosa terdaftar          dilakukan	
Akun		
Gejala		
Penyakit		
Relasi		
Riwayat diagnosa		
Footer		

**Gambar 3.33** Tampilan Utama *Admin*



c. Tampilan Menu Daftar Akun

Berikut ialah desain tampilan menu daftar akun pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk mengelolah daftar akun.

<b>Admin Panel</b>					
Nama Admin					
Log out	Daftar Akun				Buat Admin
<b>Main menu</b>	first name	last name	email	admin	action
Akun	text	text	text	yes/no	ubah hapus
Gejala					
Penyakit					
Relasi					
Riwayat diagnosa					
Footer					

**Gambar 3.34** Tampilan Menu Daftar Akun

d. Tampilan Menu Daftar Gejala

Berikut ialah desain tampilan menu daftar gejala pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk mengelolah daftar gejala.

<b>Admin Panel</b>					
Nama Admin					
Log out	Daftar Gejala				Masukkan Gejala
<b>Main menu</b>	kode gejala	deskripsi	action		
Akun	text	text	ubah hapus		
Gejala					
Penyakit					
Relasi					
Riwayat diagnosa					
Footer					

**Gambar 3.35** Tampilan Menu Daftar Gejala

e. Tampilan Menu Daftar Penyakit

Berikut ialah desain tampilan menu daftar penyakit pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk mengelolah daftar penyakit.

<b>Admin Panel</b>							
Nama Admin							
Log out	Daftar Penyakit						Masukkan Penyakit
<b>Main menu</b> Akun Gejala Penyakit Relasi Riwayat diagnosa	kode penyakit	nama	definisi	solusi	konseling	action	
	text	text	text	text	text	ubah	hapus
Footer							

**Gambar 3.36** Tampilan Menu Daftar Penyakit

f. Tampilan Menu Daftar Relasi

Berikut ialah desain tampilan menu daftar relasi pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk mengelolah daftar relasi.

<b>Admin Panel</b>							
Nama Admin							
Log out	Daftar Relasi						Masukkan Relasi
<b>Main menu</b> Akun Gejala Penyakit Relasi Riwayat diagnosa	nama penyakit	gejala	bobot	urutan	action		
	text	text	text	text	ubah	hapus	
Footer							

**Gambar 3.37** Tampilan Menu Daftar Relasi

g. Tampilan Menu Daftar Riwayat Diagnosa

Berikut ialah desain tampilan menu daftar riwayat diagnosa pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *admin* untuk mengelolah daftar riwayat diagnosa.

<b>Admin Panel</b>					
Nama Admin					
Log out	Daftar Riwayat Diagnosa				
<b>Main menu</b>	nama pengguna	usia	jenis kelamin	tanggal	action
Akun	text	text	text	text	detail hapus
Gejala					
Penyakit					
Relasi					
Riwayat diagnosa					
Footer					


**Gambar 3.38** Tampilan Menu Daftar Riwayat Diagnosa

3. Menu *User*

Tampilan dalam sistem pakar diagnosa infeksi saluran pernafasan akut yang diakses oleh *User*.

a. Tampilan Registrasi *User*

Berikut ialah desain tampilan registrasi *user* pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *user* untuk melakukan proses registrasi akun.

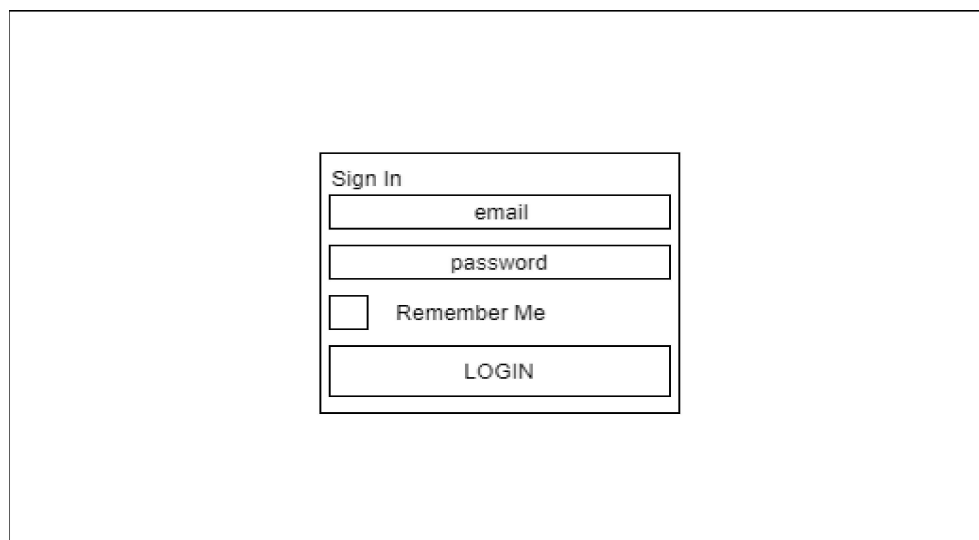


The image shows a registration form titled "Register". It contains the following fields from top to bottom: "nama depan", "nama belakang", "email", "password", "confirm password", "usia", and "jenis kelamin" (which is a dropdown menu). Below these fields is a "Register" button.

**Gambar 3.39** Tampilan Registrasi *User*

b. Tampilan *Login User*

Berikut ialah desain tampilan *login user* pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *user* untuk *login* ke dalam tampilan utama *user*.

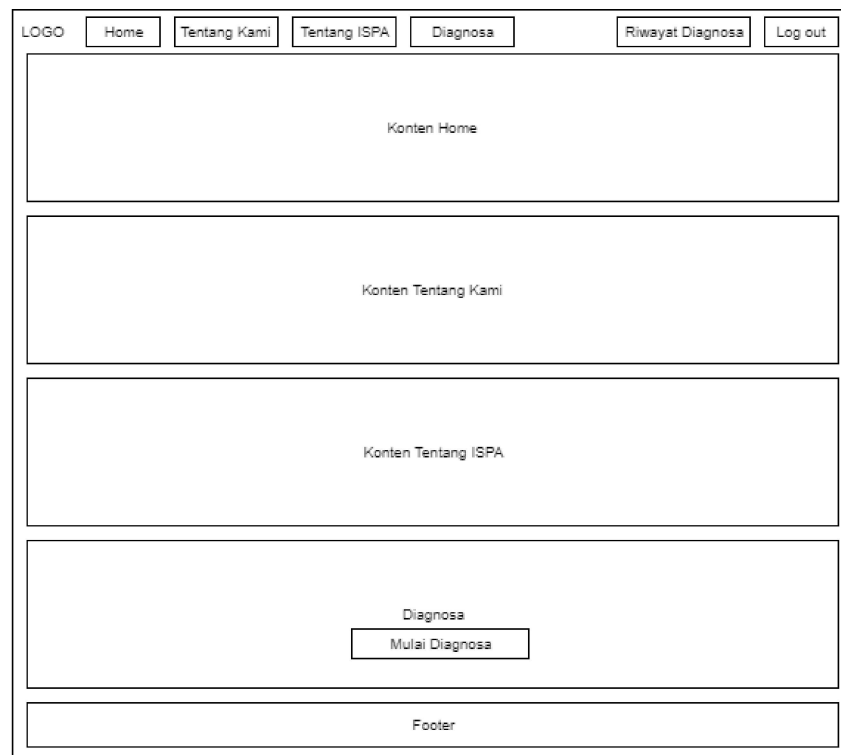


The image shows a login form titled "Sign In". It contains the following fields from top to bottom: "email", "password", a "Remember Me" checkbox, and a "LOGIN" button.

**Gambar 3.40** Tampilan *Login User*

c. Tampilan Utama *User*

Berikut ialah desain tampilan utama *user* pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *user* untuk mengakses menu *Home*, *Tentang Kami*, *Tentang ISPA*, *Diagnosa*, *Riwayat Diagnosa* dan melakukan *Log out*.



**Gambar 3.41** Tampilan Utama *User*

d. Tampilan Diagnosa

Berikut ialah desain tampilan diagnosa pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *user* untuk melakukan proses diagnosa.

The screenshot shows a web interface for a diagnostic system. At the top left is the text "LOGO" and at the top right is a "Log out" button. Below these is a large rectangular box containing the text "Informasi pengguna". Underneath this box is another section titled "Pertanyaan". This section contains two radio buttons labeled "Ya" and "Tidak", and a "Next" button positioned below them.

**Gambar 3.42** Tampilan Diagnosa

e. Tampilan Riwayat Diagnosa

Berikut ialah desain tampilan riwayat diagnosa pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *user* untuk melihat riwayat diagnosa.

The screenshot shows a web interface for a diagnostic history system. At the top left is the text "LOGO" and at the top right is a "Log out" button. Below these is a large rectangular box containing the text "Informasi pengguna". Underneath this box is another section titled "Riwayat Diagnosa". This section contains the text "Tanggal Diagnosa" and a "Detail" button positioned to its right.

**Gambar 3.43** Tampilan Riwayat Diagnosa

#### f. Tampilan Hasil Diagnosa

Berikut ialah desain tampilan hasil diagnosa pada sistem pakar diagnosa infeksi saluran pernafasan akut yang digunakan oleh *user* untuk melihat hasil diagnosa.



**Gambar 3.44** Tampilan Hasil Diagnosa

### 3.5 Lokasi dan Jadwal Penelitian

#### 3.5.1 Lokasi

Penelitian ini dilakukan di Puskesmas Kijang yang beralamat di Jln. Berek Motor, Kijang Kota, Kecamatan Bintan Timur, Kabupaten Bintan. Alasan peneliti memilih instansi ini sebagai lokasi penelitian adalah:

1. Ketersediaan data.
2. Penyakit infeksi saluran pernafasan akut merupakan penyakit dengan jumlah penderita kedua terbanyak di Puskesmas Kijang.

### 3.5.2 Jadwal Penelitian

Setiap rancangan penelitian perlu dilengkapi dengan jadwal penelitian yang dilaksanakan yang berisi jadwal kegiatan apa saja yang akan dilakukan selama penelitian (Sugiyono, 2014:286). Berikut ini adalah tabel jadwal kegiatan yang dilakukan selama penelitian berlangsung.

**Tabel 3.4** Jadwal Penelitian

No	Kegiatan	Tahun 2018																			
		Maret				April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengajuan Judul	■	■																		
2	Penyusunan Bab I		■	■	■	■	■														
3	Penyusunan Bab II				■	■	■	■													
4	Penyusunan Bab III							■	■	■	■	■	■	■							
5	Penyusunan Bab IV														■	■	■				
6	Penyusunan Bab V, Daftar Pustaka, Lampiran																■	■	■	■	■

(Sumber: Pengolahan Data Penelitian, 2018)