

**SISTEM PAKAR MENDIAGNOSIS PENYAKIT  
LAMBUNG MENGGUNAKAN METODE *FORWARD*  
*CHAINING* BERBASIS WEB**

**SKRIPSI**



Oleh  
**Yohana Sitorus**  
**140210107**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
2020**

**SISTEM PAKAR MENDIAGNOSIS PENYAKIT  
LAMBUNG MENGGUNAKAN METODE *FORWARD*  
*CHAINING* BERBASIS WEB**

**SKRIPSI**



Oleh  
**Yohana Sitorus**  
**140210107**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
2020**

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini saya:

Nama : Yohana Sitorus  
NPM : 140210107  
Fakultas : Teknik dan Komputer  
Program Studi : Teknik Informatika

Menyatakan bahwa “**skripsi**” yang saya buat dengan judul:

**Sistem Pakar Mendiagnosis Penyakit Lambung Menggunakan Metode  
*Forward Chaining* Berbasis Web**

Adalah hasil karya sendiri dan bukan “duplikasi” dari karya orang lain. Sepengetahuan saya, didalam naskahs skripsi ini tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia naskah Skripsi ini digugurkan dan gelar akademik saya saya peroleh dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa ada paksaan dari siapa pun

Batam, 18 Februari 2020

**Yohana Sitorus**  
**14021010**

**SISTEM PAKAR MENDIAGNOSIS PENYAKIT  
LAMBUNG MENGGUNAKAN METODE FORWARD  
CHAINING BERBASIS WEB**


**SKRIPSI**

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

**Oleh  
Yohana Sitorus  
14021017**

**Telah disetujui oleh pembimbing pada tanggal  
seperti tertera dibawah ini**

**Batam, 18 Februari 2020**

  
**Anggia Dasa Putri, S.kom., M.kom  
Pembimbing**

## ABSTRAK

Lambung berbentuk kantong menyerupai huruf J yang terletak dibagian atas perut. Organ pencernaan yang satu ini diisi dengan otot dan pembuluh darah, ukuran perut bisa berubah bentuk sesuai dengan postur manusia dan berapa banyak makanan yang ada di perut saat makan. Ada empat penyakit yang menyerang lambung diantaranya Gastritis, Dispepsia, Kanker Lambung, dan Gerd. Terbatasnya pengetahuan seseorang dari suatu gejala awal dari suatu penyakit pada tubuh merupakan faktor penyebab menjadi parah ketika penderita ditangani medis, dengan bantuan sistem pakar seseorang yang bukan ahli/pakar dapat menjawab atau menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. Teknologi Kecerdasan Buatan merupakan mesin yang mampu berfikir, menimbang tindakan yang akan di ambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Dan kecerdasan buatan terbagi menjadi beberapa bidang ilmu, salah satunya yaitu sistem pakar terkait yang merupakan bidang ilmu pengetahuan dan pengalaman yang dimasukan oleh satu atau banyak pakar ke dalam sebuah mesin atau perangkat lunak sehingga mesin tersebut mampu memecahkan masalah yang membutuhkan keahlian manusia. Penelitian ini mendiagnosis penyakit pada lambung dengan membangun sebuah sistem pakar yang menganalisa gejala-gejala penyakit menjadi sebuah keputusan nama penyakit dengan menggunakan metode *forward chaining* yang berbasis web. Penelitian ini diharapkan dapat membantu masyarakat untuk mendiagnosis penyakit lambung sejak dini, serta memudahkan dan meringankan pihak medis untuk menjelaskan pada pasien hanya dengan menunjukkan monitor maka pasien akan lebih mudah memahami dalam mendiagnosis penyakit lambung.

Kata Kunci: Penyakit Lambung; Sistem Pakar; *Forward Chaining*.

## **ABSTRACT**

*Bag-shaped stomach resembles the letter J which is located at the top of the stomach. This digestive organ is filled with muscles and blood vessels, the size of the stomach can change shape according to human posture and how much food is in the stomach when eating. There are four diseases that attack the stomach including Gastritis, Dyspepsia, Gastric Cancer, and Gerd. The limited knowledge of a person from an initial symptom of a disease in the body is a contributing factor to be severe when the patient is treated medically, with the help of an expert system someone who is not an expert / expert can answer or solve problems and make decisions that are usually made by an expert. Artificial Intelligence Technology is a machine that is able to think, weigh the actions to be taken, and be able to make decisions like those carried out by humans. And artificial intelligence is divided into several fields of science, one of which is a related expert system which is a field of science and experience that is entered by one or many experts into a machine or software so that the machine is able to solve problems that require human expertise. This research diagnoses diseases in the stomach by building an expert system that analyzes the symptoms of a disease into a disease name decision by using a web-based forward chaining method. This research is expected to help the community to diagnose gastric disease early on, as well as facilitate and ease the medical authorities to explain to patients only by showing a monitor so that patients will more easily understand in diagnosing gastric disease.*

*Keywords: Stomach Disease; Expert System; Forward Chaining.*

## 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. Ketua Program Studi Teknik Informatika Universitas Putera Batam
3. Anggia Dasa Putri, S.Kom.,M.Kom, selaku pembimbing Skripsi pada program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. dr. Supardi di Klinik Nira Medika Batam yang telah memberikan dukungannya.
6. Linda Sirait (Ibu) yang selalu memberikan doa dan motivasi yang terbaik.
7. Abang dan Adek saya yang selalu membarikan suport dan motivasi terbaik.
8. Rekan-rekan mahasiswa/i Universitas Putera Batam yang turut memberikan doa dan dukungan nya.
9. Rekan kerja yang selalu memeberikan masukan yang berguna untuk penelitian ini.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu memberikan kesehatan, Amin.

Batam, 18 Januari 2020

Yohana Sitorus

## DAFTAR ISI

<b>HALAMAN SAMPUL</b> .....	i
<b>HALAMAN JUDUL</b> .....	ii
<b>SURAT PERNYATAAN</b> .....	iii
<b>HALAMAN PENGESAHAN</b> .....	<b>Error! Bookmark not defined.</b>
<b>ABSTRAK</b> .....	iv
<b>ABSTRACT</b> .....	vi
<b>KATA PENGANTAR</b> .....	vii
<b>DAFTAR ISI</b> .....	viii
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>BAB I PENDAHULUAN</b>	
1.1 LatarxBelakangxPenelitian .....	1
1.2 Identifikasi Masalah.....	3
1.3 Pembatasan Masalah.....	4
1.4 Perumusan Masalah .....	4
1.5 Tujuan Penelitian.....	5
1.6 Manfaat Penelitian.....	5
1.6.1 Aspek Teoritis.....	6
1.6.2 Aspek praktis .....	6
<b>BAB II KAJIAN PUSTAKA</b>	
2.1 Teori Dasar .....	7
2.1.1 Kecerdasan Buatan (Artificial Intelligence) .....	7
2.1.2 Sistem Pakar .....	12
2.1.3 Forward Chaining.....	28
2.1.4 World Wide Web .....	30
2.2 Penyakit Lambung .....	33
2.3 Software Pendukung.....	41
2.4 Penelitian Terdahulu.....	69
2.5 Kerangka Pemikiran .....	71
<b>BAB III METODE PENELITIAN</b>	
3.1 Desain Penelitian .....	73
3.2 Pengumpulan Data.....	76
3.3 Operasional Variabel .....	78
3.4 Metode Perancangan Sistem .....	80
3.4.1 Desain Basis Pengetahuan.....	80
3.4.2 Aturan (Rule).....	82
3.5 Perancangan Sistem.....	85
3.5.1 <i>Use Case Diagram</i> .....	85
3.5.2 Desain UML (Unified Modeling Language) .....	86
3.4.2. Desain Antarmuka.....	107
3.5. Lokasi dan Jadwal Penelitian	



3.5.1. Lokasi Penelitian.....	114
3.5.2. Jadwal Penelitian.....	114

#### **BAB IV HASIL DAN PEMBAHASAN**

4.1 Hasil Penelitian .....	116
4.2 Pembahasan .....	125
4.2.1 Pengujian Validasi .....	125
4.2.2 Pengujian Keakuratan dengan Pakar.....	128

#### **BAB V SIMPULAN DAN SARAN**

5.1 Simpulan.....	130
5.2 Saran.....	130

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN**

Lampiran 1 Pendukung Penelitian

Lampiran 2 Daftar Riwayat Hidup

Lampiran 3 Surat Keterangan Penelitian

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Kecerdasan Buatan .....	8
<b>Gambar 2.2</b> <i>logika fuzzy</i> .....	9
<b>Gambar 2.3</b> Jaringan Saraf Tiruan.....	10
<b>Gambar 2.4</b> Sistem Pakar.....	12
<b>Gambar 2.5</b> Tim Pengembang sistem pakar.....	22
<b>Gambar 2.6</b> Pohon Keputusan.....	26
<b>Gambar 2.7</b> pohon keputusan.....	29
<b>Gambar 2.8</b> <i>Diagram Alur Forward Chaining</i> .....	29
<b>Gambar 2.9</b> World Wide Web.....	31
<b>Gambar 2.10</b> Penyakit Lambung.....	33
<b>Gambar 2.11</b> Gastritis.....	34
<b>Gambar 2.12</b> Dispepsia.....	36
<b>Gambar 2.13</b> Kanker Lambung.....	37
<b>Gambar 2.14</b> Gerd .....	39
<b>Gambar 2.15</b> Logo HTML.....	42
<b>Gambar 2.16</b> Logo PHP.....	44
<b>Gambar 2.17</b> Logo phpMyAdmi .....	50
<b>Gambar 2.18</b> Tampilan awal phpMyadmin .....	51
<b>Gambar 2.19</b> Logo CSS .....	51
<b>Gambar 2.20</b> Logo <i>Notepad++</i> .....	54
<b>Gambar 2.21</b> Tampilan awal notepad++.....	54
<b>Gambar 2.22</b> Logo XAMPP.....	55
<b>Gambar 2.23</b> Tampilan awal xampp.....	56
<b>Gambar 2.24</b> Logo MySQL .....	57
<b>Gambar 2.25</b> Logo StarUML .....	58
<b>Gambar 2.26</b> Tampilan awal Star UML .....	58
<b>Gambar 2.27</b> <i>Usecase Diagram</i> .....	62
<b>Gambar 2.28</b> Class Diagram .....	64
<b>Gambar 2.29</b> <i>Usecase Diagram</i> .....	66
<b>Gambar 2.30</b> Sequennce Diagram.....	69
<b>Gambar 2.31</b> Kerangka Pemikiran .....	72
<b>Gambar 3.1</b> Desain Penelitian 2019 .....	73
<b>Gambar 3.2</b> Pohon Keputusan.....	84
<b>Gambar 3.3</b> Use case Diagram.....	86
<b>Gambar 3.4</b> Activity <i>Diagram</i> Login .....	87
<b>Gambar 3.5</b> Activity <i>Diagram</i> .....	89
<b>Gambar 3.6</b> Activity <i>Diagram</i> Mengelola Gejala.....	90
<b>Gambar 3.7</b> Activity <i>Diagram</i> Mengelola Menu Relasi .....	92
<b>Gambar 3.8</b> Activity <i>Diagram</i> Mengelola Menu Lihat Relasi .....	93

<b>Gambar 3.9</b> Activity <i>Diagram</i> Logout .....	94
<b>Gambar 3.10</b> Activity <i>Diagram</i> Melihat Menu Beranda .....	95
<b>Gambar 3.11</b> Activity <i>Diagram</i> Menu Diagnosis .....	96
<b>Gambar 3.12</b> Sequence <i>Diagram</i> Login Admin .....	97
<b>Gambar 3.13</b> Sequence <i>Diagram</i> Mengelola Menu Penyakit .....	98
<b>Gambar 3.14</b> Sequence <i>Diagram</i> Mengelola Gejala.....	99
<b>Gambar 3.15</b> Sequence <i>Diagram</i> Mengelola Menu Relasi .....	101
<b>Gambar 3.16</b> Sequence <i>diagram</i> Mengelola Menu Lihat Relasi.....	102
<b>Gambar 3.17</b> Sequence <i>diagram</i> logout admin .....	103
<b>Gambar 3.18</b> Sequence <i>diagram</i> Melihat Menu Beranda.....	103
<b>Gambar 3.19</b> Sequence <i>Diagram</i> Menu Diagnosis .....	104
<b>Gambar 3.20</b> Class <i>Diagram</i> Admin.....	105
<b>Gambar 3.21</b> Class <i>Diagram</i> User .....	106
<b>Gambar 3.22</b> Gambar Desain Database .....	107
<b>Gambar 3.23</b> Halaman Beranda .....	108
<b>Gambar 3.24</b> Halaman Diagnosis.....	108
<b>Gambar 3.25</b> Halaman Diagnosis Pertanyaan.....	109
<b>Gambar 3.26</b> Halaman Diagnosis Solusi .....	110
<b>Gambar 3.27</b> Halaman Admin.....	110
<b>Gambar 3.28</b> Halaman <i>Home admin</i> .....	111
<b>Gambar 3.29</b> Halaman Menu Penyakit.....	111
<b>Gambar 3.30</b> Halaman Tambah Penyakit .....	112
<b>Gambar 3.31</b> Halaman Menu Gejala .....	112
<b>Gambar 3.32</b> Halaman Tambah Gejala.....	113
<b>Gambar 3.33</b> Halaman Menu Relasi.....	113
<b>Gambar 3.34</b> Rancangan Halaman Menu Konsultasi .....	114
<b>Gambar 4.1</b> Tampilan Menu Beranda.....	116
<b>Gambar 4.2</b> Tampilan <i>Form</i> Pendaftaran .....	117
<b>Gambar 4.3</b> Tampilan Menu Diagnosa Dengan Pesan Kesalahan .....	118
<b>Gambar 4.4</b> Tampilan Hasil Diagnosa.....	119
<b>Gambar 4.5</b> Tampilan Menu Petunjuk.....	119
<b>Gambar 4.6</b> Tampilan Menu Buku Tamu .....	120
<b>Gambar 4.7</b> Tampilan <i>Form Login Administrator</i> .....	121
<b>Gambar 4.8</b> Tampilan Daftar Pengguna .....	122
<b>Gambar 4.9</b> Tampilan Daftar Pengguna ( <i>Administrator</i> ).....	122
<b>Gambar 4.10</b> Tampilan Menu Penyebab.....	123
<b>Gambar 4.11</b> Tampilan Menu <i>Edit Data</i> Penyebab .....	124
<b>Gambar 4.12</b> Tampilan Menu Dari Tombol Lihat Gejala.....	125

## DAFTAR TABEL

<b>Tabel 2.1</b> Perbandingan sistem tradisional dan pakar .....	20
<b>Tabel 2.2</b> Keputusan.....	26
<b>Tabel 2.3</b> Alternatif Tabel Keputusan .....	27
<b>Tabel 2.4</b> Indikator Dan Gejala .....	40
<b>Tabel 2.5</b> Simbol Use Case Diagram.....	61
<b>Tabel 2.6</b> Simbol Class Diagram.....	63
<b>Tabel 2.7</b> Simbol Activity Diagram .....	65
<b>Tabel 2.8</b> Simbol Sequence Diagram .....	66
<b>Tabel 3.1</b> Operasional Variabel penyakit pada lambung.....	78
<b>Tabel 3.2</b> Indikator, Gejala Dan Solusi .....	79
<b>Tabel 3.3</b> Jenis Penyakit, penyebab dan Solusi.....	80
<b>Tabel 3.4</b> Tabel Gejala Penyakit .....	81
<b>Tabel 3.5</b> Aturan (Rule).....	82
<b>Tabel 3.6</b> Tabel Keputusan .....	83
<b>Tabel 3.7</b> Tabel Jadwal Penelitian.....	115
<b>Tabel 4.1</b> Tabel pengujian validitas sistem.....	126
<b>Tabel 4.2</b> Tabel Hasil Diagnosa Pakar dan Diagnosa Sistem .....	128

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Penelitian**

Lambung bagian organ berbentuk kantong menyerupai huruf J yang terletak di bagian atas perut. Organ pencernaan satu ini dipenuhi oleh otot berongga berukuran besar dan terdiri dari tiga bagian, yaitu kardia, fundus, dan antrum pembuluh darah, ukuran lambung dapat berubah bentuk mengikuti postur manusia dan seberapa banyak makanan yang ada di dalam lambung pada saat makan, asam lambung memegang peranan penting dalam fungsi pencernaan yang bisa mengurai makanan agar mudah diserap oleh tubuh. Hanya saja ketika sampai asam lambung dapat membuat efek yang tidak baik (Ida, 2015).

Kasus yang sering terjadi adalah asam lambung keluar dari lambung dan naik kekrongkongan. Dalam dunia kedokteran penyakit ini muncul akibat naiknya asam lambung dikenal dengan istilah *gastro esophageal reflux disease* (GERD). Gejala yang muncul pada penyakit lambung ini seperti kembung, mual, muntah, nyeri pada bagian ulu hati, nyeri perut yang muncul setelah mengonsumsi makanan tertentu (Ida, 2015).

Menurut pakar (dr. Supardi) di Klinik Nira Medika Batam pasien yang menderita penyakit lambung umumnya pada pekerja dengan *range* usia 17 tahun sampai dengan 40 tahun. Penyakit yang menyerang lambung, diantaranya adalah *Gastritis*, *Dispepsia*, kanker lambung, dan *Gastroesophageal Reflux Disease* (GERD). Penyakit lambung merupakan penyakit yang tidak bisa disepelekan,

jika dibiarkan terus menerus dapat mengakibatkan penyakit yang lain muncul dan bisa menyebabkan kematian, jika tidak segera di tangani penyakit pada lambung dapat disebabkan oleh infeksi.

Terbatasnya pengetahuan seseorang dari suatu gejala awal dari suatu penyakit pada tubuh yang merupakan faktor penyebab penyakit menjadi parah ketika penderita ditangani medis. Dengan bantuan sistem pakar seseorang yang bukan ahli / pakar dapat menjawab atau menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar.

Teknologi Kecerdasan Buatan merupakan mesin yang mampu berfikir, menimbang tindakan yang akan di ambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Dan kecerdasan buatan terbagi menjadi beberapa bidang ilmu, salah satunya yaitu sistem pakar terkait yang merupakan bidang ilmu pengetahuan dan pengalaman yang dimasukan oleh satu atau banyak pakar ke dalam sebuah mesin atau perangkat lunak sehingga mesin tersebut mampu memecahkan masalah yang membutuhkan keahlian manusia (Kusumadewi, 2003)

Menurut (Zulfian, 2017), *forward chaining* adalah strategi untuk mengantisipasi masalah atau menemukan solusi yang dimulai dengan serangkaian fakta yang diketahui dan kemudian mendapatkan fakta berbasis aturan baru yang sesuai dengan fakta yang diketahui. Proses akan berlanjut sampai tujuan tercapai atau tidak ada aturan lain yang mencakup fakta yang diketahui olehnya. Fakta atau pernyataan acak mulai dari kiri (pertama). Dengan kata lain, pemikiran dimulai dengan fakta untuk membuktikan kebenaran hipotesis.

Menurut (Abbdullah, 2018) penggunaan website kian hari kian meluas, rasanya sudah menjadi suatu keharusan kepada perusahaan atau instansi untuk membuat suatu website, mulai dengan yang sederhana semisal profil perusahaan, penjualan, keuangan, bahkan samapai pembuatan sistem terintegrasi.

Sistem yang dibuat diharapkan mampu memberikan informasi yang tepat kepada masyarakat mengenai penyakit lambung berdasarkan masukan gejala-gejala serta solusi penanganan dan rekomendasi obat yang tepat. Penelitian dilakukan untuk alasan ini, dengan judul **"SISTEM PAKAR UNTUK MENDIAGNOSIS PENYAKIT LAMBUNG MENGGUNAKAN METODE FORWARD CHAINING BERBASIS WEB"**

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang penelitian di atas, masalah tersebut dapat diidentifikasi sebagai berikut:

1. Penyakit lambung rentan menyerang kalangan dari usia 17 tahun sampai 40 tahun
2. Minimnya informasi yang tepat tentang penyakit lambung dan masyarakat sering beranggapan sabagai masuk angin biasa
3. Banyak masyarakat yang sudah dalam kondisi kronis baru memeriksakan diri ke dokter

### 1.3 Pembatasan Masalah

Agar terhindar dari penyimpangan judul serta tujuan aktual dan untuk mengingat batas waktu, ruang lingkup dan batas-batas masalah yang dibuat dalam penelitian ini telah dibuat:

1. Sampel dalam penelitian ini adalah orang dewasa dari 17 hingga 40 tahun.
2. Indikasi dalam penelitian ini yang mendiagnosis penyakit lambung adalah gastritis, dispepsia, kanker lambung, dan gerd
3. Pakar dalam penelitian ini adalah dokter umum di Klinik Kedokteran Nira Batam, yaitu: Supardi Sp.Ok.
4. Untuk mendiagnosis penyakit lambung menggunakan Sistem pakar berbasis web dengan metode *forward chaining*.
5. Tools yang digunakan adalah *web server* mozilla firefox, *database* menggunakan MySQL dan bahasa pemrogramannya menggunakan HTML, PHP dan CSS.
6. Output penelitian ini untuk mempermudah masyarakat mendeteksi penyakit lambung sejak dini, dan akan dihosting.

### 1.4 Perumusan Masalah

Dari masalah yang telah dijelaskan keterbatasan dan masalah dalam penelitian ini dapat dibentuk:

1. Bagaimana cara mengetahui mengenali serta mendiagnosis penyakit lambung?



2. Bagaimana cara menerapkan sistem pakar untuk diagnosis penyakit lambung dengan *forward chaining*?
3. Bagaimana merancang aplikasi berbasis web untuk membantu mendiagnosis penyakit lambung ?

### **1.5 Tujuan Penelitian**

Penelitian ini memiliki tujuan yaitu untuk:

1. Ketahui cara untuk mengenali serta mendiagnosis penyakit lambung
2. Mengetahui implementasi sistem pakar metode *forward chaining* untuk mendiagnosis penyakit lambung.
3. Merancang aplikasi mendiagnosa penyakit lambung berbasis *web*.

### **1.6 Manfaat Penelitian**

Manfaat penelitian merupakan dampak dari pencapaiannya tujuan. Seandainya dalam penelitian, tujuan dapat tercapai dan rumusan masalah dapat dipecahkan secara tepat dan akurat, maka apa manfaatnya secara praktis maupun secara teoritis. Kegunaan penelitian mempunyai dua hal yaitu mengembangkan ilmu pengetahuan (secara teoritis) dan membantu mengatasi, memecahkan dan mencegah masalah yang ada pada objek yang diteliti.

### **1.6.1 Aspek Teoritis**

Penelitian ini bertujuan untuk memperluas pengetahuan dan persepsi yang terkait dengan metode yang digunakan oleh sistem pakar dan mampu menerapkan hasilnya di lingkungan masyarakat.

### **1.6.2 Aspek praktis**

Manfaat penelitian adalah:

1. Mempermudah masyarakat untuk melakukan diagnosa diri tentang penyakit lambung sebelum menemui seorang pakar atau dokter.
2. Dapat menambah wawasan dan informasi tentang penyakit lambung bagi masyarakat.
3. Sebagai alat bantu bagi pengguna dalam mengdiagnosa penyakit lambung yang diderita masyarakat.
4. Memudahkan dan meringankan pihak medis untuk menjelaskan pada pasien hanya dengan menunjukkan monitor maka pasien akan lebih mudah memahami dalam mendiagnosa penyakit lambung.

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Teori Dasar**

Menurut (Sugiyono, 2011) Teori dasarnya adalah bahwa teori dasar lain yang digunakan dalam penulisan skripsi. Dengan demikian penelitian dapat berjalan dengan lancar, membutuhkan dasar untuk jalannya penelitian dalam bentuk teori yang ada. Dalam penelitian ini, kami akan menjelaskan secara singkat kecerdasan buatan, sistem pakar, metode pengajaran, uml, basis data.

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

(Sutojo et al., 2011), Kecerdasan buatan berasal dari "*Artificial Intelligence*" atau disingkat AI, yaitu, kecerdasan memiliki kata sifat yang berarti cerdas, sedangkan buatan berarti buatan. Kecerdasan buatan yang dimaksud mengacu pada mesin yang mampu berpikir, menimbang tindakan yang diambil dan mampu membuat keputusan manusia. (Simon, 2011: 2) berpendapat bahwa kecerdasan buatan (*artificial intelligence*) adalah bidang penelitian, aplikasi dan instruksi dalam pemrograman komputer untuk melakukan sesuatu yang menurut orang cerdas. Pada awalnya, komputer hanya berfungsi sebagai alat perhitungan.



**Gambar 2.1** Kecerdasan Buatan

**Sumber :** (Sutojo et al., 2011)

Kecerdasan buatan atau “*Artificial intelligence*” itu sendiri di munculkan oleh seorang profesor dari *Massachusetts Institute of Technology* yang bernama John McCarthy pada tahun 1956 pada *Dartmouth Conference* yang dihadiri oleh para peneliti AI. Dia juga menemukan bahasa pemrograman lisp (Sutojo et al., 2011).

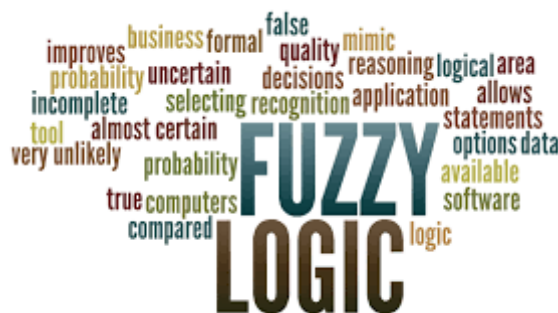
### *1. Fuzzy Logic*

Menurut (Sutojo, Mulyanto, Suhartono, 2011: ) walaupun konsep ini diperkenalkan oleh Zadeh pada tahun 1962, Carl Prof.Lotfi Fuzzy system adalah sistem kontrol metodologi penyelesaian masalah yang diterapkan dalam kaitannya dengan yang lain yang dapat diimplementasikan dalam sistem, mulai dari sistem kecil yang sederhana, embedded system, jaringan PC, multi -channel atau workstation-akuisisi dan sistem kontrol data berbasis. Metode ini dapat diimplementasikan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik telah dinyatakan bahwa itu adalah biner, yaitu hanya dua kemungkinan. 'benar atau salah.' "Baik atau buruk, itu saja." Selama

ini nilai keanggotaan Dapa 0 dan 1, atau nilai "Ya" atau "Tidak" pada saat yang sama, tergantung pada berat keanggotaannya.

Defenisi metode *fuzzy* adalah suatau acara ntuk memetakan ruang input ke ruang output. Ada beberapa metode pada *fuzzy* yaitu:

1. Metode Mamdani : Menghasilkan suatau hasil (*output*) pada himpunan *fuzzy*.
2. Metode Tsukamoto : Perluasan dari penalaran monoton.
3. Metode Sugeno : Untuk atauran yang di presentasikan dalam bentuk IF-THEN, dimana output tidak berupa himpunan *fuzzy* melainkan konstanta.
4. Metode Himpunan Crisp dan Himpunan *Fuzzy* : didefenisikan oleh item-item yang ada pada himpunan.
5. Metode Penalaran Monoton : digunakan sebagai dasar untuk teknik implikasi *fuzzy*.



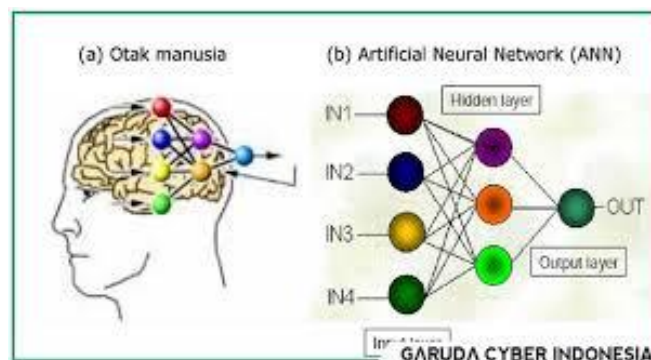
**Gambar 2.2** *logika fuzzy*

**Sumber:** (Sutojo et al., 2011)

## 2. Jaringan Saraf Tiruan

(Sutojo, Mulyanto, Suhartono, 2011) Paradigma jaringan syaraf tiruan dihitung terinspirasi oleh sistem saraf biologis, seperti informasi proses otak manusia. Paradigma ini merupakan faktor kunci, dan - struktur dari sistem komputer yang terdiri dari sejumlah elemen pengolahan saling berhubungan (neuron) yang bekerja sama untuk memecahkan masalah tertentu.. Ada beberapa metode pada jaringan saraf tiruaan:

1. 1. Metode Perhitungan Vektor (LVQ), melakukan pelatihan bersarang yang tidak dipantau dalam kompetisi.
2. Metode *backpropagation*, menyesuaikan bobot nilai dari perbedaan output yang diinginkan.
3. Metode *pereptron*, melakukan pengklasifikasian secara linear.



**Gambar 2.3** Jaringan Saraf Tiruan

**Sumber:** (Sutojo et al., 2011)

## 2. *Machine learning*

*Machine learning* merupakan yang digunakan untuk meniru manusia untuk melakukan atau menyelesaikan masalah. Sama seperti namanya pembelajaran mesin yaitu proses untuk menirukan bagaimana cara manusia bekerja. Ciri khas dari *Machine learning* adanya proses untuk pelatihan. Metode yang populer pada *Machine learning* ini yaitu seperti sistem pengambilan keputusan. Berikut ilmu besar di PK untuk mencapai tujuan:

- a. *Neural Network (NN)*,
- b. *Support Vector Machines*,
- c. *Fuzzy Logic System*.

## 3. Sistem pakar

Menurut (Sutojo, Mulyanto, Suhartono, 2011) Sistem pakar adalah program kecerdasan buatan yang mensimulasikan para ahli dengan mengintegrasikan basis pengetahuan dan sistem keraguan. Sistem pakar adalah sistem yang mencoba membawa pengetahuan manusia ke komputer sehingga komputer dapat memecahkan masalah seperti yang biasanya dilakukan oleh para profesional. Bagian dari Sistem pakar adalah terdiri dari dua komponen utama yaitu *knowledge base* yang berisi *knowledge* dan mesin inferensi yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respon dari sistem pakar atas permintaan pengguna.

### 2.1.2 Sistem Pakar

Menurut (Sutojo, Mulyanto, Suhartono, 2011) Sistem Pakar didasarkan pada sistem pakar berbasis pengetahuan, sistem yang dirancang untuk meniru kemampuan para ahli untuk menjawab pertanyaan dan mereplikasi masalah. Sistem Pakar akan menyelesaikan semua masalah yang diangkat oleh dialog pengguna. Menggunakan sistem pakar, seseorang yang bukan pakar dapat memecahkan dan memecahkan masalah dan membuat keputusan yang biasanya dibuat oleh para ahli. Sistem pakar didefinisikan sebagai sistem komputer yang memiliki kemampuan untuk memberikan pengetahuan dan keterampilan kepada spesialis tertentu. Secara umum, sistem pakar dibangun menjadi dua komponen utama:

Menurut (Sutojo, Mulyanto, Suhartono, 2011) sistem pakar adalah suatu sistem yang dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari dalam menjawab pertanyaan dan memecahkan suatu masalah. Dalam bukunya kecerdasan buatan (2011:160), ada beberapa definisi sistem pakar menurut para ahli:



**Gambar 2.4** Sistem Pakar

**Sumber:** (Sutojo et al., 2011)



### 1. Penguasaan pengetahuan

Subsistem ini digunakan oleh ahli untuk memasukkan pengetahuan melalui pengembangan pengetahuan sehingga dapat diproses oleh komputer dan ditempatkan dalam basis pengetahuan dalam format tertentu (dalam bentuk presentasi pengetahuan). Sumber pengetahuan dapat diperoleh dari para ahli, buku, dokumen, multimedia, database, laporan penelitian khusus, dan informasi yang dipublikasikan secara online.

### 2. Basis pengetahuan

Basis pengetahuan berisi pengetahuan yang dibutuhkan untuk memahami, merumuskan dan memecahkan masalah.

Dasar pengetahuan terdiri dari dua elemen dasar, yaitu:

- a. Fakta seperti situasi, kondisi dan masalah yang ada
- b. Aturannya adalah untuk memecahkan masalah untuk memperbaiki pengetahuan pengguna.

### 3. Mesin Inferensi

Mesin Inferensi adalah program yang bekerja untuk memandu proses penalaran untuk suatu kondisi berdasarkan pada basis pengetahuan yang ada, manipulasi dan arah kebijakan, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai basis pengetahuan solusi dan kesimpulan.

### 4. Area kerja (file)

Untuk mencatat hasil sementara yang akan digunakan sebagai solusi dan menjelaskan apa yang terjadi, sistem pakar membutuhkan papan tulis, yang merupakan basis data yang berfungsi sebagai basis data. Tiga jenis hasil dapat ditulis di papan tulis:

- a. Desain: bagaimana menyelesaikan masalah.
- b. Agenda: tindakan yang berpotensi tertunda.
- c. Solusi: kandidat untuk tindakan perlu dibangkitkan.

#### 5. Antarmuka pengguna (*User Interface*)

Digunakan sebagai alat komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling baik dilakukan dalam bahasa alami dan dilengkapi dengan grafik, menu, dan bentuk elektronik. Bagian ini membahas dialog antara sistem pakar dan pengguna.

#### 6. Subsistem Penjelasan (Pengadilan)

Tujuannya adalah untuk menjelaskan kepada pengguna bagaimana atau untuk menyimpulkan. Kemampuan ini sangat penting bagi pengguna untuk memahami proses mentransfer pengetahuan dan memecahkan masalah.

#### 7. Sistem Pemurnian Pengetahuan (*Knowledge Refining System*)

Fungsi untuk meningkatkan pengetahuan (sistem pemurnian pengetahuan) dari para ahli diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, dan kemudian menambah pengetahuan untuk menggunakannya di masa depan. Analisis efisiensi dari seri tersebut diperlukan oleh program untuk menganalisis penyebab keberhasilan dan kegagalan untuk menarik kesimpulan.

Dengan cara ini, basis pengetahuan yang baik dan penalaran yang lebih efektif akan dikembangkan.

## 8. Pengguna (*User*)

Secara umum, pengguna sistem non-pakar (*nonexpert*) membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai masalah.

### 1. Ciri-ciri Sistem Pakar

Karakteristik sistem pakar (Sutojo et al., 2011) adalah sebagai berikut:

- a. Terbatas pada bidang keahlian tertentu.
- b. Memberikan alasan untuk informasi yang tidak lengkap atau tidak lengkap.
- c. Masalah bisa dijelaskan dengan jelas. Bekerja berdasarkan kaidah/*rule* tertentu.
- d. Putar ringan.
- e. Tuliskan basis pengetahuan dan desain.
- f. Rekomendasi rekomendasi.
- g. Sistem dapat mengaktifkan kebijakan dengan cara yang benar, dipandu oleh komunikasi.

## 2. Keuntungan Sistem Pakar

(Sutojo et al., 2011), sistem pakar menjadi sangat populer karena banyak peluang dan keunggulannya, termasuk: Membuat seorang yang awam bekerja seperti layaknya seorang pakar.

1. dikarenakan penggunaan system pakar yang cepat sehingga dapat meningkatkan produktivitas.
2. peningkatan pada kualitas memberikan saran yang tetap dan tidak memperbanyak kesalahan.
3. Kemampuan untuk mencatat pengetahuan dan kompetensi Anda.
4. Dapat beroperasi di lingkungan berbahaya.
5. Memfasilitasi akses ke pengetahuan pakar.
6. Sistem pakar yang andal tidak akan pernah membosankan, lelah atau sakit.
7. Tingkatkan kemampuan sistem komputer Anda. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efisien dan berisi lebih banyak aplikasi.
8. Kemampuan untuk bekerja dengan informasi yang tidak lengkap atau tidak jelas. Tidak seperti sistem komputer tradisional, sistem pakar dapat menangani informasi yang tidak lengkap. Pengguna dapat menjawab "Saya

tidak tahu" atau "Saya tidak yakin" untuk satu atau lebih pertanyaan selama konsultasi, dan sistem pakar terus menjawab.

9. Dapat digunakan sebagai media tambahan dan pelatihan. Penguin pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena kemampuan interpretif bekerja sebagai guru.

10. Tingkatkan keterampilan Anda dalam memecahkan masalah karena sistem pakar disediakan oleh banyak pakar untuk sumber pengetahuan.

### 3. Area Masalah dalam Sistem

Menurut (Sutojo et al., 2011), program pakar sistem biasanya mengandalkan bidang masalah berikut:

1. *Interpretasi*: Membuat laporan kondisi berdasarkan pada input data.
2. *Prediksi*: Untuk mengevaluasi kemungkinan hasil suatu situasi.
3. *Evaluasi*: merangkum situasi berdasarkan gejala yang divalidasi (gejala)
4. *Panning*: Atur tindakan untuk dilakukan
5. *Desain*: Desain sesuai dengan batasan
6. *Audit*: Perbandingan sistem pemantauan dan kontrol
7. *Resolusi*
8. *Kompensasi*: lanjutkan dengan paket upgrade

9. Tip: Ikuti instruksi untuk mendiagnosis, mengatasi masalah dan memperbarui prosedur.

10. Kontrol: Mengontrol efek interpretasi, diagnostik, debugging, pemantauan dan memperbarui sistem.

#### 4. Aspek kunci dari sistem pakar

##### 1. Keahlian (Spesialisasi)

Spesialisasi adalah pengetahuan yang diperoleh melalui pelatihan, membaca, dan pengalaman. Keahlian memungkinkan para profesional untuk membuat keputusan lebih cepat dan lebih baik daripada yang bukan ahli. Pengetahuan ini juga akan mencakup tentang pengetahuan (Sutojo et al., 2011):

- a. Adanya kebenaran masalah spesifik.
- b. Adanya fakta area masalah tertentu
- c. Hal-hal umum peraturan dan regulasi lapangan.
- d. Adanya pemecahan masalah yang dalam situasi tertentu harus bekerja.
- e. Cara global untuk menuntaskan masalah spesifik.
- f. *Meta-knowledge* atau pengetahuan-pengetahuan (Sutojo et al., 2011).

##### 2. Pakar (*Expert*)

Seorang ahli adalah pengetahuan seseorang, pengalaman, dan metode khusus dan dapat diterapkan untuk menyelesaikan masalah atau memberikan saran. Pakar harus mampu memberi dan memecahkan beberapa hal-hal baru yang

terkait dengan pokok permasalahan masalahnya, jika perlu, mampu menata kembali pengetahuan yang didapat, dan dapat memecah aturan dan menentukan relevansi keahlian. (Sutojo et al., 2011).

### 3. Transfer keahlian (*Transferring Expertise*)

Menurut (Sutojo et al., 2011) tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar kedalam computer, kemudian di transfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan yaitu:

1. Dapatkan pengetahuan (dari para ahli atau sumber lain).
2. Representasi pengetahuan (komputer)
3. Ketidaktahuan.
4. Transfer pengetahuan kepada pengguna.

### 4. kesimpulan (*Inferensi*)

Sikap adalah program yang bisa memberi sebab. Indikasi ditampilkan dalam komponen yang disebut Mesin Inferensi, yang mencakup pemecahan masalah. Semua pengetahuan yang diperoleh oleh para ahli ini disimpan berdasarkan pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah menarik kesimpulan secara ilmiah (Sutojo et al., 2011).

### 5. Aturan (*rule*)

Sebagian besar perangkat lunak perusahaan adalah sistem berbasis aturan, di mana pengetahuan disimpan sebagai aturan penyelesaian masalah (Sutojo et al., 2011).

### 6. Kemampuan Menjelaskan (Kekuatan Deskriptif)

Bidang perencanaan ahli lainnya adalah kemampuan Anda untuk menyoroti saran atau rekomendasi yang Anda tawarkan. Definisi tersebut dibuat dalam suatu sistem yang disebut Teknologi Informasi (deskripsi). sistem memungkinkan untuk menghitung tujuan dan menjelaskan efisiensinya. Fitur dan kemampuan program kebugaran berbeda dari program reguler. Perbedaan ini ditunjukkan pada tabel di bawah ini (Sutojo et al., 2011).

**Tabel 2.1** Perbandingan sistem tradisional dan pakar

<b>Sistem Normal</b>	<b>Sistem Pakar</b>
Informasi dan pemrosesan bias digabungkan.	basis pengetahuan dipisahkan dengan baik dari mekanika informasi
Program tidak membuat kesalahan (yang membuat kesalahan: pemrograman atau pengguna).	Aplikasi mungkin salah
Biasanya tidak dijelaskan mengapa datanya. Input diperlukan atau saat output dihasilkan.	Disistem pakar penjelasan bagian terpenting.
Mengubah program itu sulit	Sangat mudah untuk mengubah aturan
Sistem hanya dapat bekerja setelah selesai	Sistem dapat bekerja dengan beberapa aturan (sebagai prototipe awal)



Pengisian dilakukan langkah demi langkah (algoritmik).	Pengisian dilakukan dengan menggunakan matematika dan logika di bidang pengetahuan.
Diperlukan informasi lengkap per operator produk	Menangani informasi yang tidak lengkap atau tidak pasti.
Penggunaan basis data besar secara efisien.	Pemasaran yang efektif dari basis pengetahuan yang besar
Data Eksperimen	Menggunakan pengetahuan
Tujuan utama: efisiensi.	Tujuan akhir: efisiensi.
Manajemen informasi akan berkurang	Jadikan informasi yang berkualitas mudah dilihat
Pertimbangkan, tambahkan, dan distribusikan akses ke data atau informasi numerik.	Abadikan, tambahkan dan sebarkan akses ke ide dan wawasan.

**Sumber:**(Sutojo et al., 2011)

## 6. Struktur Sistem Pakar

Lingkungan Pengembangan dan ruangan konsultasi merupakan dua bagian penting disistem professional. Didalam vendor sistem professional lingkungan pengembangan menggunakannya untuk meyusun komponen mereka dan membawa pengetahuan ke basis pengetahuan.sedangkan di dalam Lingkungan konsultasi digunakan oleh pengguna untuk mengarahkan, sehingga pengguna dapat mencapai pengetahuan dan saran dari sistem pakar, seperti bertanya dengan pakar. (Sutojo et al., 2011).

## 7. Tim pengembangan sistem pakar

Di bawah ini adalah representasi dari kelompok pakar pengembangan sistem, yang kami kembangkan dalam bentuk gambar dan penjelasan.



**Gambar 2.5** Tim Pengembang sistem pakar  
**Sumber:** (Sutojo et al., 2011)

### 1. Teknik inferensi

Sastra-teknisnya, tekno-narator, yaitu, jika penulis pengetahuan berasal dari kumpulan budaya, maka, seperti dalam TI, meninggalkan catatan. skrip, ketika dikirim ke status saat ini. Datanglah ke mesin inferensi untuk semua aturan yang disimpan di basis Otentikasi yang dibuat oleh sistem. Jika IF (kondisi) membuat file baru di negara tertaut (Sutojo et al., 2011).

### 2. Keuntungan dari sistem pakar

Menurut(Sutojo, Mulyanto, Suhartono, 2011: 160-161) adasebelas manfaat yang bisa dicapai dengan adanya sistem khusus.

- a) Pekerjaan pakar dapat dikerjakan oleh orang awam.
- b) Cepatnya sistem pada saat dijalankan.
- c) Memperkecil kesalahan dengan membuat kualitas yang meningkat
- d) Keterampilan dan pengetahuan harus dipahami.
- e) Itu dapat bekerja bahkan di lingkungan yang sulit.
- f) Memfasilitasi akses ke para ahli.
- g) Jangan pernah bosan dan lelah atau sakit.
- h) Meningkatkan kualitas sistem komputer, memasukkan lebih banyak aplikasi dan membuat sistem lebih efisien.
- i) Itu dapat menjawab bahkan jika pengguna dapat menjawab "Saya tidak tahu" atau "Saya tidak yakin" satu atau lebih pertanyaan selama membaca.
- j) Digunakan sebagai penelitian dan sebagai aditif.
- k) Sistem pakar mengandalkan sumber pengetahuan banyak pakar untuk memecahkan masalah.

### 3. Kelemahan sistem pakar

Ada Tiga macam kelemahan sistem pakar(Sutojo, Mulyanto, Suhartono, 201:161)

- a. Di sini Ada banyak biaya yang harus dikeluarkan.

- b. Sulit bagi para ahli di bidangnya untuk berkembang.
- c. Tidak 100% benar.

#### 4. Representasi Pengetahuan

Pemrosesan yang dilakukan oleh sistem pakar merupakan pemrosesan pengetahuan, bukan pemrosesan data seperti yang dikerjakan oleh sistem informasi. Pengetahuan (*knowledge*) adalah suatu yang hadir dan terwujud dalam jiwa dan pikiran seseorang karena adanya reaksi, sentuhan dan hubungan dengan lingkungan dan alam disekitar. Menurut (Zulfian, 2017). Berikut ini adalah contoh struktur aturan IF-THEN untuk menghubungkan objek:

JIKA premis KEMUDIAN konklusi

- a. JIKA masukan KEMUDIAN keluaran
- b. JIKA kondisi KEMUDIAN tindakan
- c. JIKA anteseden KEMUDIAN konsekuen
- d. JIKA data KEMUDIAN hasil
- e. JIKA tindakan KEMUDIAN tujuan
- f. JIKA aksi KEMUDIAN reaksi
- g. JIKA gejala KEMUDIAN diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Sebelum tindakan dapat diambil kondisi harus tertuju pada keadaan yang harus berlaku. *Anteseden* mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Sebuah hasil dapat diperoleh dari data

yang mengacu pada informasi yang tersedia. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Menurut (Hartati & Iswanti) adanya pemeriksaan disebabkan oleh gejala yang mengacu pada keadaan yang menyebabkan kerusakan tertentu.

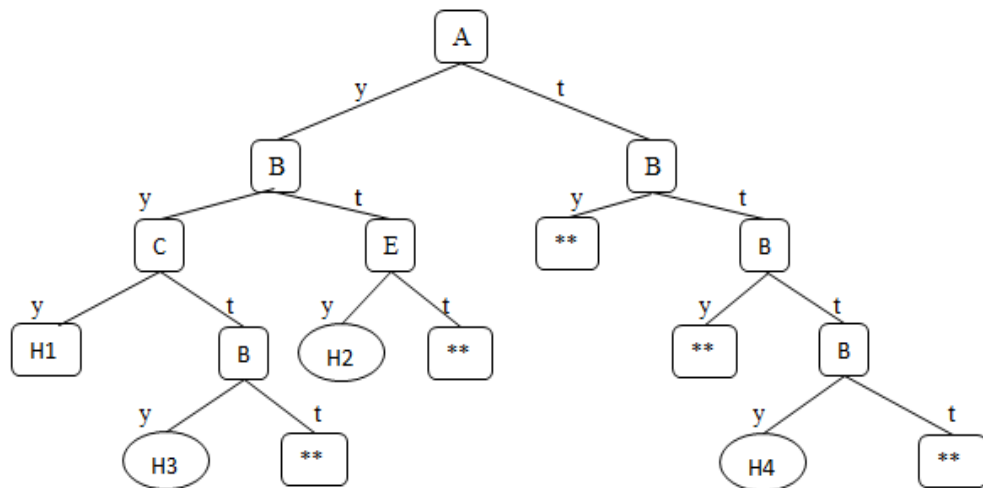
Sebelum sampai pada bentuk kaidah produksi, terdapat langkah-langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut adalah menyampaikan pengetahuan yang sudah berhasil didapatkan dalam bentuk tabel keputusan (*decision table*) kemudian dari tabel keputusan dibuat pohon keputusan (*decision tree*).

Tabel keputusan merupakan cara untuk menyimpan pengetahuan. Tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Meskipun kaidah secara langsung dapat dihasilkan dari tabel keputusan tetapi untuk mendapatkan kaidah yang efisien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan terlebih dahulu. Dari pohon keputusan dapat diketahui atribut (kondisi) yang dapat direduksi sehingga menghasilkan kaidah yang efisien dan optimal. Berikut ini variasi contoh terkait dengan penyajian dalam bentuk tabel keputusan dan pohon keputusan.

**Tabel 2.2** Keputusan

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

**Sumber:** (Zulfian, 2017)

**Gambar 2.6** Pohon Keputusan

**Sumber:** (Zulfian, 2017)

Pada gambar 2.6 berikut keterangannya:

A = bukti A, H1 = hipotesa 1, y = ya

B = bukti B, H2 = hipotesa 2, t = tidak

C = bukti C, H3 = hipotesa 3, \*\* = tidak menghasilkan hipotesa tertentu

D = bukti D, H4 = hipotesa 4

Dari gambar 2.6 dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi bukti A, B, dan C. Hipotesa H2 terpenuhi jika memiliki bukti A dan bukti E. Hipotesa H3 akan terpenuhi jika memiliki bukti A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi bukti B, C, dan D. Notasi “y” mengandung arti memenuhi simpul (bukti) di atasnya, notasi “t” artinya tidak memenuhi.

Dalam implementasi sistem pakar terutama dalam sesi konsultasi pada, simpul-simpul yang mewakili bukti biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.5 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki bukti A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan bukti 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 2.3** 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>
Bukti A	Ya	Ya	Ya	Tidak
Bukti D	Tidak	Tidak	Tidak	Ya
Bukti B	Ya	Tidak	Ya	Ya
Bukti C	Ya	Tidak	Tidak	Ya
Bukti E	Tidak	Ya	Ya	Tidak

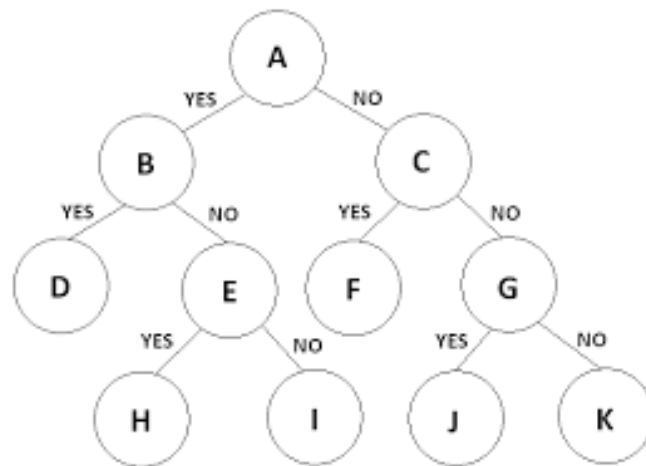
**Sumber:**(Zulfian, 2017)

### 2.1.3 *Forward Chaining*

*Forward chaining* adalah cara dalam mengambil keputusan yang diawali sebuah fakta menuju kesimpulan atau akhir. Dengan menanyakan pertanyaan yang benar merupakan aspek penting dari perolehan fakta. Pertanyaan benar yang diajukan menghasilkan efisiensi dalam menentukan jawaban yang benar. Untuk hal ini adalah sistem pakar hanya akan menanyakan pertanyaan yang berhubungan dengan hipotesis yang dicoba dibuktikan (Sutojo et al., 2011).

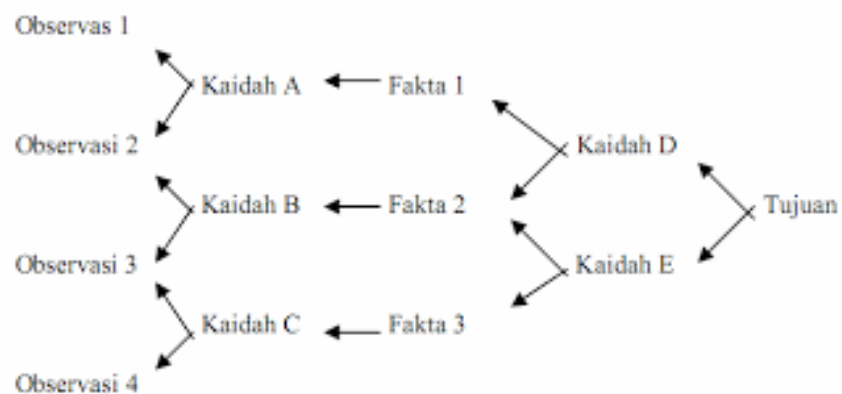
*Forward chaining* adalah teknik mencari yang dimulai dengan fakta apa yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari rules *IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. *first search* (DFS), *breadth-first search* (BFS) atau best first search merupakan metode dalam pencarian (Sutojo et al., 2011).





**Gambar 2.7** pohon keputusan

**Sumber:** (Sutojo et al., 2011)



**Gambar 2.8** Diagram Alur Forward Chaining

**Sumber:** (Sutojo et al., 2011)

Kelebihan:

- Kelebihan utama dari *Forward Chaining* yaitu metode ini akan bekerja dengan ketika masalah bermula dari menyusun dan mencari informasi lalu kemudian mengali kesimpulan yang terdapat pada informasi tersebut.
- Metode ini dapat menyimpan informasi dari jumlah banyak ke jumlah kecil datanya.

Kelemahan:

- a. Kelemahan utama metode ini yaitu tidak tahu bagaimana cara untuk tau fakta penting dari pada fakta yang lain.
- b. Sistem bisa saja memberi pertanyaan yang tidak berhubungan. walaupun jawaban dari pertanyaan tersebut penting. Namun hal ini akan membuat bingung user untuk menjawab karena pada subjek yang tidak berhubungan. 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.(Sutojo, Mulyanto, Edy Suhartono 2011).

#### **2.1.4 World Wide Web**

*World wide web* (www), lebih dikenal dengan *web*, merupakan layanan yang pemakainya terhubung dengan *internet*. *Web* pada awalnya adalah ruang informasi dalam *internet*, dengan menggunakan teknologi *hyperteks*, untuk dapat mencari informasi pemakai dituntun untuk mengikuti *link* yang didalam halaman *web* yang berada dalam *browser web*. Kini *internet* identik dengan *web*, karena kepopuleran *web* sebagai standar *interface* pada layanan-layanan yang ada di *internet*, dari awalnya sebagai penyedia informasi, saat ini juga kita dapat berbincang-bincang tanpa tatap muka dari *email* dan *chattingan*, sampai melakukan sebuah bisnis (*commerce*). Dalam *Web* dapat membantu pengguna komputer untuk berkomunikasi dengan pengguna internet lainnya dan mencari

beberapa informasi di *internet*. Selain itu, Karena beberapa alasan *web* telah diadopsi oleh perusahaan sebagai sebagian dari rencana teknologi informasinya, (Gani, 2017):

1. Setup server lebih gampang,
2. Akses informasi gampang,
3. Dalam mendistribusikan Informasi mudah , dan
4. Bebas platform; pada browser web informasi dapat disajikan pada sistem operasi mana saja karena adanya ketentuan dokumen berbagai tipe data dapat disajikan (Gani, 2017)



**Gambar 2.9** World Wide Web  
**Sumber :** (Sutojo et al., 2011)

Untuk memahami cara kerja pada sebuah web , maka langkah kerjanya adalah

#### 1. Langkah Kerja *Web*

Langkah kerja *web* yang pertama yaitu, dokumen-dokumen yang ada informasi *web* disimpan yang disebut dengan halaman-halaman *web* (*web pages*), *web page* adalah dalam komputer disimpan *file-file* disebut dengan *server-server web* (*web servers*). Selanjutnya *web page* dapat dibaca oleh computer disebut sebagai *web client*, *web client* menampilkan page dengan menggunakan program

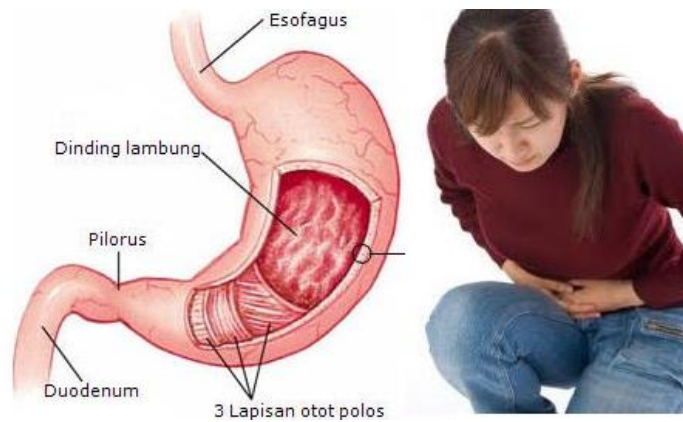
yang disebut dengan browser web (web browser), Internet Explorer dan Netscape Navigator merupakan browser web yang populer. (Gani, 2017)

Browser web adalah dari server web dapat pengguna lihat informasi yang ditampilkan. *User interface grafis* merupakan software yang dikembangkan saat ini oleh *server web*, sehingga ‘*point* dan *click*’ dapat dipakai oleh pemakai untuk pindah antar dokumen. *Internet Explorer* dan *Netscape Navigator* merupakan dua *browser web* GUI yang saat ini populer. Sehingga Kedua *browser* ini bersaing untuk merebut pemakainya, dengan mendekati standar spesifikasi dokumen HTML yang direkomendasikan oleh W3C. Suatu *browser* mengambil sebuah *web* page dari server dengan sebuah request. Sebuah request adalah sebuah permintaan HTTP standar yang berisi sebuah page address. Sebuah page address terlihat seperti berikut: *http// www.kita.web.id/ page htm*. Instruksi-instruksi yang ditampilkan ada didalam . *web page Browser* menampilkan page dengan membaca instruksi-instruksi ini. Tag HTML merupakan Instruksi yang paling umum untuk ditampilkan (Abbdullah, 2018)

*Server web* adalah yang komputer gunakan sebagai penyimpanan dokumen-dokumen, komputer ini akan melayani permintaan dokumen web dari kliennya. *Browser web* seperti *Explorer* atau *Navigator* berkomunikasi melalui jaringan (termasuk jaringan internet) dengan *server web*, menggunakan HTTP. *Browser* akan mengirimkan *request* kepada *server* untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh *server*. *Server* memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protokol HTTP (Abbdullah, 2018).

## 2.2 Penyakit Lambung

Menurut (Ida, 2015) Penyakit dalam merupakan penyakit yang berkaitan dengan banyak organ vital dalam tubuh manusia, salah satunya Lambung. Lambung terletak oblik dari kiri ke kanan menyilang di bdomen atas tepat dibawah diafragma. Dalam keadaan kosong lambung menyerupai tabung berbentuk J, dan bila penuh berbentuk seperti buah pir raksasa. Lambung merupakan organ penting dalam tubuh karena merupakan salah satu organ pencernaan makanan dan minuman yang dikonsumsi manusia. Penyakit lambung kurang diketahui oleh masyarakat karena kurangnya informasi dan pengetahuan tentang penyakit lambung sehingga masyarakat mengabaikan gejala-gejala yang timbul.



**Gambar 2.10** Penyakit Lambung  
**Sumber:** (Ida, 2015)

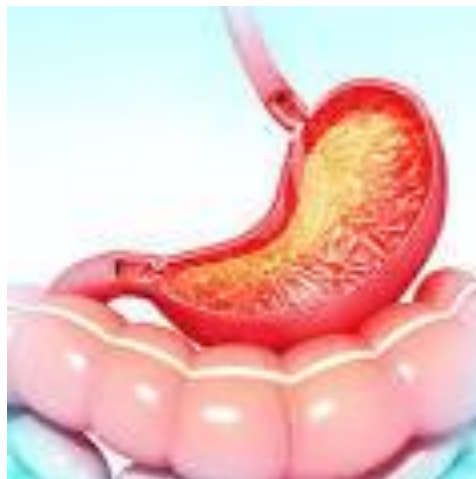
Penyakit lambung merupakan penyakit yang tidak bisa dianggap remeh, karena jika dibiarkan terus menerus dapat menyebabkan kematian jika tidak segera ditangani. Penyakit lambung dapat disebabkan oleh pola makan yang tidak sesuai, beban pikiran dan juga infeksi yang disebabkan bakteri. Beberapa penyakit

yang menyerang lambung, diantaranya adalah *Gastritis*, *Dispepsia*, kanker lambung dan *Gerd*

Jenis jenis penyakit lambung yang akan dibahas pada penelitian ini adalah :

1. *Gastritis*

Menurut (Suratun, 2010:57) *Gastritis* adalah radang mukosa lambung dan manifestasi klinisnya tampak melalui perubahan histopatologis yang dapat menyebabkan metaplasia, displasia, dan karsinoma. Gastritis akut merupakan penyakit yang sering terjadi; pada penderita gastritis akut selintas 80-90% pasien yang dirawat di IC. Menurut Fahrial (2009) berdasarkan hasil penelitian Devisi Gastroentologi Departemen Ilmu Penyakit Dalam FKUI pada tahun 2009 yang menderita penyakit gastritis di Indonesia sebanyak 86,41 %. Terlepas dari hal tersebut, gastritis akut biasanya jinak dan sembuh dengan sendirinya. Berikut ini penyebab gastritis (Syamsudin, M.Biomed.2015:36):



**Gambar 2.11** Gastritis  
**Sumber:** (Ida, 2015)

### 1. Penyebab gastritis

Ada beberapa penyebab yang dapat mengakibatkan seseorang menderita gastritis antara lain mengkonsumsi alkohol, terapi radiasi, reflu empedu, zat korosif (cuka, lada) kemotrapi

### 2. Gejala

Gejala yang ditimbulkan seperti nyeri tekan abdomen (nyeri pada perut), dehidrasi, muntah

### 3. Pengobatan

Mengonsumsi obat sitoprotektif, penetralisir asam lambung (antasida), farmakoterapi untuk *H.pylori*

### 4. Diagnosis

pemeriksaan laboratorium, pencatatan riwayat medis

### 2. Dispepsia

Dispepsia adalah rasa nyeri atau tidak nyaman dibagian ulu hati. Kondisi ini dianggap gangguan di dalam tubuh yang dikibatkan reaksi tubuh terhadap lingkungan sekeliling. Reaksi ini menimbulkan gangguan ketidkseimbangan metabolisme, dan seringkali menyerang individu usia produktif, yakni usia 30-50 tahun (Arif & Sari, 2011)



**Gambar 2.12** Dispepsia  
**Sumber:** (Ida, 2015)

Dispepsia meliputi kumpulan gejala klinis yang terdiri dari rasa tidak nyaman atau sakit menetap atau mengalami kekambuhan pada perut bagian atas (Arif & Sari, 2011) keluhan akan gejala-gejala klinis tersebut kadang-kadang disertai dengan rasa panas di dada dan perut, rasa lekas kenyang, anoreksia, kembung, regurgitasi, dan banyak mengeluarkan gas asam dari mulut.

#### 1. Penyebab dispepsia

Ada beberapa penyebab yang dapat mengakibatkan seseorang menderita dispepsia antara lain faktor asam lambung pasien, kelainan psikis, stres dan faktor lingkungan, dan gangguan motilitas (makanan berlemak, kopi, alkohol, rokok)

#### 2. Gejala

Gejala yang ditimbulkan seperti rasa sakit dan tidak enak di ulu hati, perih, mual, sering bersendawa

#### 3. Pengobatan



Suportif perubahan pada pola kebiasaan terutama jenis makanan, Endoskopi, USG (bila diduga ada kelainan di pankreas)

#### 4. Dianosis

Pemeriksaan laboratorium, Pemeriksaan endoskopi

#### 3. Kanker Lambung

Kanker lambung merupakan suatu keganasan yang terjadi di lambung, sebagian besar adalah dari jenis adenokarsinoma. Jenis kanker lambung lainnya adalah leiomyosarkoma (kanker otot polos) dan limfoma. Kanker lambung lebih sering terjadi pada usia lanjut, kurang dari 25% kanker tertentu terjadi pada orang dibawah usia 50 tahun. Kanker lambung pada pria merupakan keganasan terbanyak ketiga setelah kanker paru dan kanker kolorektal, sedangkan pada wanita merupakan peringkat keempat setelah kanker payudara, kanker serviks, dan kanker kolorektal.



**Gambar 2.13** Kanker Lambung  
**Sumber:** (Ida, 2015)

### 1. Penyebab kanker lambung

Ada beberapa penyebab yang dapat mengakibatkan seseorang menderita kanker lambung antara lain Gastritis kronis, sering makan daging dengan cara dipanggang, bakar atau diasapkan, terlalu sering makan makanan pedas

### 2. Gejala

Gejala yang ditimbulkan seperti penurunan berat badan, perih perut bagian bawah, asites (perut membesar), darah yang nyata atau samar dalam tinja

### 3. Pengobatan

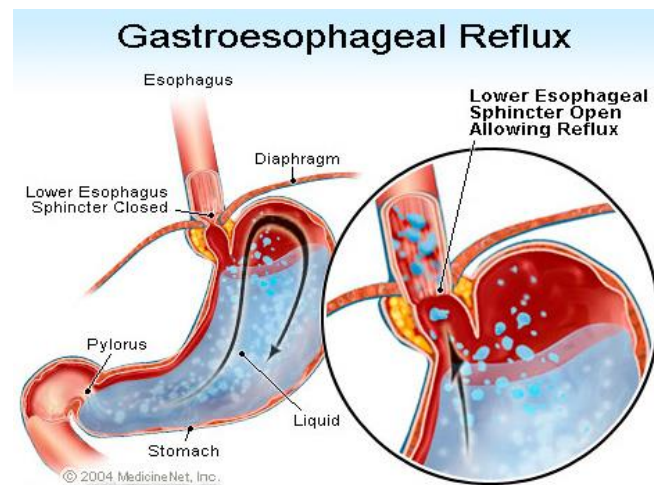
Melaksanakan pola hidup yang sehat, mengkonsumsi makanan yang bergizi dan sehat, mengkonsumsi buah dan sayur

### 4. Diagnosis

Pencatatan riwayat medis, Pemeriksaan fisik, Teknik-teknik diagnosti

### 4. GERD

*Gastroesophageal Reflux Disease* ( GERD ) merupakan suatu keadaan patologis yang merupakan akibat dari *reflux* kandung lambung ke dalam esofagus. Kondisi ini menimbulkan berbagai gejala mengganggu, baik di esofagus maupun ekstra esofagus dan atau komplikasi.



**Gambar 2.14** Gerd  
**Sumber:** (Ida, 2015)

Pada orang normal, *reflux* dapat diatasi dengan posisi tegak setelah makan. Posisi tegak tersebut di bantu oleh kontraksi peristaltik primer, sehingga isi lambung yang mengalir masuk ke esofagus segera dikembalikan ke lambung. *Reflux* sejenak ini tidak merusak mukosa esofagus dan tidak menimbulkan keluhan atau gejala, oleh karena itu dinamakan refluks fisiologis.

Keadaan *refluks* tersebut dikatakan patologis apabila terjadi berulang-ulang dan menyebabkan esofagus distal terkena pengaruh isi lambung dalam waktu yang lama. Istilah *esofagitis refluks* berarti kerusakan esofagus akibat refluks cairan lambung, seperti erosi dan ulserasi epitel skuamosa esofagus.

#### 1. Penyebab gerd

Ada beberapa penyebab yang dapat mengakibatkan seseorang menderita kanker lambung antara lain Menurunnya tonus *lower esophageal sphincter* (LES),

Bersihkan asam dari lumen esofagus menurun, Ketahanan epitel esofagus menurun,  
Bahan refluksat mengenai dinding esofagus

## 2. Gejala

Gejala yang ditimbulkan seperti muntah, nyeri pada dada, batuk

## 3. Pengobatan

Tidak mengkonsumsi makanan yang berlemak (obesitas), Menghindari obat-obatan yang menurunkan kekuatan LES ( spinter esofagus bawah), Penggunaan obat-obatan prokinetik

## 4. Diagnosis

Pemeriksaan fisik dan riwayat medis, Pemantauan PH esafagus, Endoskopi dan biopsi, Radiografi kontras barium

**Tabel 2.4** Indikator Dan Gejala

<b>Indikator</b>	<b>Gejala</b>
1. Gastritis	1. nyeri tekan abdomen 2. dehidrasi 3. muntah
2. Dyspepsia	1. rasa sakit dan tidak enak pada ulu hati 2. perih 3.mual 4. mual 5. sering bersendawa
3. Kanker Lambung	1. penurunan berat badan 2. muntah 3. asites (perut membesar)

	4. darah yang nyata atau samar dalam tinja
4. Gerd	1. muntah 2. nyeri pada dada 3. batuk

**Sumber:** Data Penelitian 2019

### 2.3 Software Pendukung

Software pendukung merupakan perangkat yang mendukung dalam menjalankan system pakar dalam penelitian. Perangkat lunak tersebut antara lain: HTML, PHP, *phpMyAdmin*, CSS, *Notepad++*, XAMPP, dan MySQL.

#### 1. Hyper Text Markup Language (HTML)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut digunakan untuk menunjukkan dan menjalankan tampilan didalam halaman *website* dan bahasa paling dasar dan penting. Menurut sumber yang penulis kutip dari Wikipedia, *hyper text markup language* digunakan untuk menunjukkan berbagai informasi didalam sebuah penjelajah *webInternet* dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII (*American Standard Code for Information Interchange*) agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata, disimpan ke dalam format ASCII normal sehingga menjadi *homepage* dengan perintah-perintah **HTML**. *Hyper text markup language* menggunakan dua macam *ekstensi file* yaitu **.htm** dan **.html**. Format

ekstensi berformat **.htm** awalnya hanyalah untuk mengakomodasi penggunaan html dalam operasi DOS (Jubliee, 2018).



**Gambar 2.15** Logo HTML  
**sumber:** (Jubliee, 2018)

### 1. Struktur Dasar HTML (*Hyper Text Markup Language*)

(Saputra, 2012) dokumen *hyper text markup language* memiliki sebuah struktur yang harus kita ikuti aturan pembuatannya. Beberapa elemen wajib yang ada pada *file hyper text markup language* apabila kita ingin membangun suatu pondasi kerangka *website*. Elemen tersebut diantaranya:

#### 1. Elemen *hyper text markup language* (html)

Elemen html merupakan *tag* dasar apabila kita ingin memulai suatu dokumen html. Secara logika, jika kita menemukan *tag* ini, berarti secara jelas dapat didefinisikan bahwa dokumen ini merupakan suatu dokumen html. *Tag* ini merupakan tag pertama dalam dokumen html yang wajib bagi pemrograman *web*. Contoh *tag*-nya adalah: `<html>` dan diakhiri dengan `</html>`

#### 2. Elemen *Head*

*Head* merupakan *tag* berikutnya setelah elemen `html` (`<html>`), yang berfungsi untuk menjelaskan keterangan tentang dokumen *web* yang akan ditampilkan. Dan diakhiri dengan tanda penutup `</head>`. Jadi jika secara runtun dapat dituliskan format berikut:

```
<html>
<head>
</head>
</html>
```

### 3. .Elemen *Title*

Elemen *title* merupakan elemen yang harus dituliskan didalam *elemen head* yang digunakan untuk membuat judul/informasi pada *caption browser web* tentang topik/tema atau judul dari suatu dokumen *web* yang ditampilkan pada *browser*. Berikut struktur penggunaannya:

```
<html>
<head>
    <title Tuliskan Judul disini </title>
</head>
</html>
```

### 4. Elemen *Body*

Elemen *body* merupakan bagian dokumen *web* yang utama. Untuk menampilkan suatu teks, maka kita harus meletakkan teks pada elemen *body*.. Struktur elemennya sebagai berikut:

```
<html>
<head>
```

```

<title> Tuliskan Judul disini </title>
</body>
</html>

```

## 2. PHP (*Hypertext Preprocessor*)

PHP (*hypertext preprocessor*) adalah bahasa *scrip* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP (*hypertext preprocessor*) banyak dipakai untuk pemrograman situs *web* dinamis. PHP (*hypertext preprocessor*) dapat digunakan untuk membangun sebuah CMS (*content management system*). Untuk menggunakan *hypertext preprocessor* ada beberapa *sftoware* yang harus kita *install*, diantaranya *Apache server*, *Php*, *php Myadmin*, dan *Mysql server*(Jubliee, 2018).



**Gambar 2.16** Logo PHP

**Sumber:** (Jubliee, 2018)

(Jubliee, 2018) untuk menggunakan bahasa pemrograman *hypertext preprocessor*, kita harus memulainya dengan tanda khusus seperti:

```

<?php
// Sintaks PHP
?>

```

Terdapat empat macam format yang bisa kita digunakan untuk memulai pemrograman *hypertext preprocessor* didalam kode Anda, yaitu:



<?php ..... ?>

<? ..... ?>

<script language="PHP"> ..... </script>

<% ..... %>

Dari ke-empat macam format tersebut, sintaks <?php ... ?> dan <? ... ?> merupakan programmer banyak yang gunakan. *Echo* dan *print* perintah untuk menampilkan suatu kalimat atau *string*, terserah anda ingin menggunakan yang mana.

## 1. Variabel dan Tipe Data

Variabel adalah menyimpan sementara didalam memori komputer. Penulisan variabel didalam pemrograman *hypertext preprocessor*, ada aturan tertentu yang harus kita ketahui, diantaranya (Jublee, 2018):

1. Simbol dollar (\$) untuk penulisan variable diawal.
2. Setelah simbol dollar (\$), harus huruf tidak boleh angka.

Contoh penggunaan yang salah: **\$123**

Contoh penggunaan yang benar: **\$hore**

3. Setelah simbol dollar dan huruf, maka karakter selanjutnya boleh menggunakan angka.

Contoh: **\$hore123**

## 2. Parameter (*POST* dan *GET*)

PHP (*hypertext preprocessor*) *POST* DAN *GET* metode yang digunakan untuk kirim data dari client (*browser*) ke *server*. Metode *POST*, digunakan untuk mengirim data dari *client* ke *server* dimana data tersebut akan dikirimkan melalui *http header*. Metode *GET*, yang bekerja dengan perintah fungsi **\$\_GET** akan mengirimkan data atau berkomunikasi melalui parameter yang akan terlihat pada URL (Jubilee, 2018).

## 3. Fungsi Waktu

PHP (*hypertext preprocessor*) memiliki banyak sekali fungsi untuk menampilkan waktu, diantaranya:

1. *Checkdate()*, digunakan untuk memastikan keabsahan tanggal dan waktu. Format dasar: *Checkdate* (bulan, hari, tahun).
2. *Date()*, digunakan untuk melihat waktu saat ini. Format dasar: *Date* ('fungsi tanggal').
3. *Date\_Default\_Timezone\_Set*, digunakan untuk menentukan/menerapkan zona waktu. Zona waktu berdasarkan daerah dimana Anda tinggal. Format: *Date\_default\_timezone\_set* ('zona daerah').
4. *Getdate()*, digunakan untuk mendapatkan informasi seputar tanggal dan waktu saat ini dalam bentuk array. Format: *Getdate* (timestamp).
5. *Idate()*, digunakan untuk mengatur waktu dan tanggal lokal ke dalam bentuk *integer* (bilangan bulat) sehingga tanggal lokal yang

mengandung format *string*, tidak akan ditampilkan. Format: *idate* (format, timestamp).

6. *Strtotime()*, digunakan untuk memperbarui tipe *string* menjadi format penanggalan/waktu. Format: *Strtotime (time, now)*.

Array merupakan variabel tunggal yang dapat menampung banyak nilai.

Tiap nilai ditampung dalam index array yang berbeda pula. Array biasanya digunakan untuk menampung data yang dapat disusun berdasarkan nomor tertentu.

#### 4. Kalimat Pengulang

Sering kali kita mendapatkan kondisi dimana terdapat suatu proses yang mengharuskan kita menggunakan kalimat pengulangan, contohnya: menampilkan nomor bilangan 1 s/d 10. Kalo kita menuliskan secara manual, maka sangat tidak efisien apalagi jika kita ingin menampilkan bilangan 1 s/d 1000. Ada beberapa fungsi yang dapat kita gunakan, diantaranya *For*, *For Loop*, *While*, *Do... While*, dan *Foreach* (Saputra, 2012).

1. Pengulangan *For*, digunakan untuk mengerjakan pengulangan yang terhitung. Sintaks dasar:

*For* (awal; kondisi; penambahan) {Kode untuk di jalankan}

2. Berbeda dengan perintah *for* yang hanya bisa mengerjakan perulangan yang terhitung, perintah *while* bisa melakukan perulangan baik yang terhitung maupun tidak terhitung. Fungsi *while*, yang menunjukkan seluruh data dari table ke halamana *web*..

3. *Do ... While*, merupakan suatu perulangan yang akan terus dilakukan jika nilai yang dihasilkan terpenuhi dan akan terhenti jika nilai tidak terpenuhi. Kode yang didalam *Do While* akan dieksekusi minimal sekali.
4. *Foreach*, merupakan sebuah perulangan pada variabel bertipe array.
5. Pernyataan Kontrol

Pada pemrograman PHP, kita dapat melakukan pengkodean dengan menggunakan perintah bersyarat atau perintah *If Else* yaitu apabila persyaratan terpenuhi maka perintah tersebut yang akan dieksekusi. PHP memiliki empat macam model / kerangka fungsi pernyataan kontrol, diantaranya *if*, *if - else*, *if - elseif - else*, dan *switch*(Jubliee, 2018).

1. Pernyataan *If*, digunakan apabila kita hanya memiliki satu kondisi dan kondisi tersebut akan dijalankan jika pernyataan dianggap benar. Format dasarnya:

```
If (kondisi) {
    // Pernyataan / kondisi
}
```

2. Pernyataan *If-Else*, digunakan apabila kita memiliki dua kondisi perbandingan. Jika kondisi/pernyataan bernilai *TRUE*, maka pernyataan 1 akan dijalankan, namun jika kondisi bernilai sebaliknya, maka pernyataan 2 akan dijalankan. Formatnya:

```
If (kondisi) {
    // Pernyataan / kondisi 1
}
else{
    // Pernyataan / kondisi 2
}
```

3. Pernyataan *If - Elseif - Else*, pernyataan ini digunakan apabila kita memiliki banyak kondisi (minimal tiga kondisi). Formatnya:

```

if (kondisi) {
    // Pernyataan 1
}
elseif (kondisi2) {
    // Pernyataan 2
}
else{
    // Pernyataan 3
}

```

4. Switch, merupakan alternatif pengganti If. Perintah switch menentukan dan mencocokkan kondisi yang diberikan dan konstanta yang ada didalam case. Konstanta pertama sampai terakhir awal Perbandingan akan dimulai. Jika kondisi didapatkan maka program akan membaca kode didalam konstanta tersebut, dan ketika bertemu perintah break.

```

Switch (ekspresi) {
Case "Kondisi 1";
Pernyataan 1;      Pernyataan 2
Break;      Break;
Case "kondisi 2"; }

```

#### 6. Operasi CRUD

CRUD merupakan singkatan dari *Craet*, *Read*, *Update*, dan *Delete*, yang merupakan suatu fungsi yang mengatur suatu data secara dinamis., kita harus terhubung dengan database, Untuk menggunakan operasi tersebut. Berikut penjelasan dari CRUD (Saputra, 2012):

##### 1. *Creat* (Menambahkan Data)

*Creat*, istilah lain dari proses “Menambah Data”, digunakan apabila kita ingin menambah suatu data kedalam tabel pada database.

##### 2. *Read* (Membaca/Menampilkan Data)

Setelah data berhasil disimpan, maka langkah selanjutnya timbul pertanyaan, bagaimana untuk menampilkan data yang telah disimpan tersebut? Proses inilah yang dinamakan dengan proses “*Read*”. Untuk menampilkan data, kita akan menggunakan fungsi *mysql\_fetch\_array*.

### 3. *Update* (Merubah Data)

Data yang telah kita simpan terkadang mengandung kesalahan *input* atau perlu diperbaharui karena itu, kita perlu proses mengupdate data.

### 4. *Delete* (Menghapus Data)

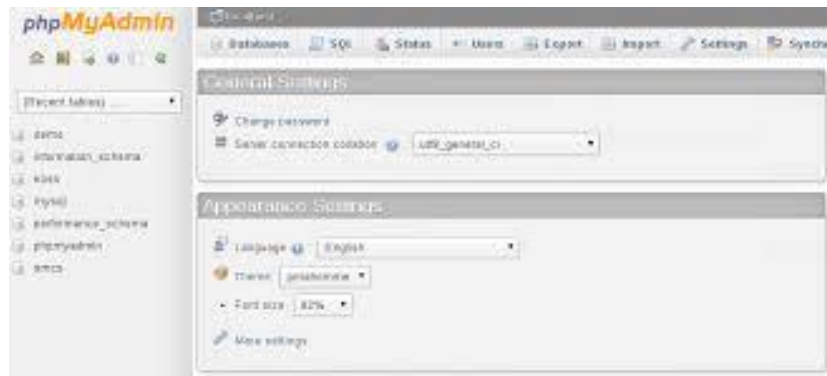
Ketika data sudah tidak dibutuhkan lagi, Anda bisa menghapusnya dari *database* agar hilang dari muka bumi ini (Saputra, 2012).

### 3. *phpMyAdmin*

*PhpMyAdmin* adalah perangkat lunak yang ditulis dalam bahasa pemograman *PHP*, dimaksudkan untuk menangani administrasi *MySQL* melalui *Web*, berikut logo dari *phpMyAdmin*.



**Gambar 2.17** Logo phpMyAdmi  
**Sumber:** (Jubliee, 2018)



**Gambar 2.18** Tampilan awal phpMyadmin  
**Sumber:** (Jubliee, 2018)

*PhpMyAdmin* adalah perangkat lunak yang ditulis dalam bahasa pemrograman *PHP*, dimaksudkan untuk menangani administrasi *MySQL* melalui *Web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL* dan *MariaDB*. Operasi (mengelola database, tabel, hubungan, *indeks*, *users*, *permissions*, dan lain-lain) dapat dilakukan melalui antarmuka pengguna masih memiliki kemampuan langsung mengeksekusi pernyataan *SQL*.

#### 4. *Cascading Style Sheet (CSS)*

*CSS* merupakan singkatan dari *cascading style sheet*, yaitu bahasa pemrograman web yang didesain khusus untuk mengendalikan dan membangun komponen dalam web sehingga tampilan web lebih rapih, terstruktur dan seragam.



**Gambar 2.19** Logo CSS  
**Sumber:** (Jubliee, 2018)

*Cascading style sheet* merupakan salah satu pemrograman wajib disamping *html* yang harus dikuasai oleh setiap pemrogram *web*, terlebih lagi itu adalah *web designer*. Tujuan utama dari *cascading style sheet* adalah memisahkan konten utama dengan tampilan dokumen lainnya (*html* dan sejenisnya). Dengan adanya pemisahan ini, akses konten pada *web* meningkat. *Web* yang menggunakan *cascading style sheet* akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* yang tidak menggunakan *cascading style sheet*. Perbedaan ini akan semakin terasa ketika *web* yang anda buka mempunyai data yang banyak. Tujuan lainnya adalah untuk mempercepat pembuatan halaman *web*. Anda hanya perlu membuat satu properti dan properti tersebut dapat digunakan pada elemen lainnya, artinya anda tidak perlu menulis ulang kode program yang digunakan berulang kali. *Cascading style sheet* saat ini dikembangkan oleh *World Wide Web Consortium* atau yang biasa lebih dikenal dengan istilah W3C. Sehingga *Cascading Style Sheet* menjadi bahasa standard dalam pembuatan *web*. *Cascading Style Sheet* bukan menggantikan kode hyper text markup language, tetapi hanya difungsikan sebagai penopang atau pendukung (pelengkap) dari *file hyper text markup language* yang berperan dalam penataan kerangka dan *layout*

#### 1. Mengenal Berbagai Macam Versi *Cascading Style Sheet*

*Cascading style sheet* (CSS) saat ini sudah mencapai versi 3 dimana pada setiap versi pasti ada peningkatan yang dilakukan. Versi yang pertama adalah CSS-1, masih kuno, CSS hanya dikembangkan dan digunakan untuk *formatting* dokumen *hyper text markup language*, sedangkan versi kedua yaitu CSS-2, disini sudah mulai terasa mantapnya. *Cascading style sheet* sudah mulai



menggunakan *font*, *table-layout*, dan berbagai media untuk *printer*. Maksudnya, *cascading style sheet* dikembangkan untuk memenuhi kebutuhan terhadap format dokumen supaya bisa ditampilkan di *printer*. Versi selanjutnya adalah CSS-3 merupakan *cascading style sheet* pengembangan dari versi sebelumnya. Peningkatan yang paling mencolok pada versi ketiga ini adalah peningkatan fitur yang mengarah pada efek *animasi*, seperti *crop*, *slide images*, dan lain-lain. Namun, satu yang harus diingat, saat ini belum semua *web browser* mendukung *Cascading style sheet 3* (Jubliee, 2018).

## 2. Mengapa *Cascading Style Sheet* (CSS)

(Saputra, 2012) banyak yang selalu bertanya, mengapa CSS? Bukankah memakai tabel kode *html* saja sudah cukup? Memang, tanpa CSS pun *web* dapat dibangun, namun tahukah Anda, salah satu kunci kesuksesan *web* agar banyak pengunjung adalah tampilan yang menarik, interaktif, cepat diakses dan lain-lain. Jika *web* Anda sangat lambat hanya saat *loading*, tentu pengunjung juga akan merasa bosan dan menutup *browser web* mereka sebelum seluruh informasi dalam *web* Anda berhasil ditampilkan. Dengan menggunakan *cascading style sheet*, akan banyak yang dapat kita peroleh, diantaranya:

1. Memisahkan pembuatan dokumen (CSS dan HTML).
2. Mempermudah dan mempersingkat pembuatan dan pemeliharaan dokumen *web*.
3. Akses *web* lebih cepat saat di-*loading* (mempercepat pembacaan HTML).

4. Fleksibel, interaktif, tampilan lebih menarik dan nyaman dipandang.
5. Lebih kecil ukuran *file* sehingga *bandwith* yang digunakan juga otomatis menjadi lebih kecil.
6. Dapat digunakan pada semua *web browser*.

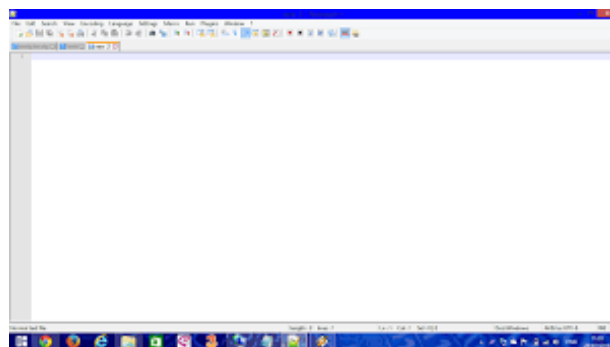
## 5. Notepad++

Menurut (Gilmore, 2010) *Notepad++* merupakan *editor teks open source* yang matang dan diakui sebagai pengganti *Notepad*, berikut logo dari *Notepad++*.



**Gambar 2.20** Logo *Notepad++*

**Sumber:** (Jubilee, 2018)



**Gambar 2.21** Tampilan awal notepad++

**Sumber:** (Jubilee, 2018)

*Notepad++* tersedia untuk platform *Windows* yang dapat digunakan untuk menulis kode dengan beberapa pilihan bahasa (pemrograman). *Notepad++* menawarkan beragam kenyamanan fitur yang diharapkan dari setiap kemampuan IDE (*Integrated Development Environment*), termasuk kemampuan untuk menunjukkan baris tertentu dari suatu dokumen sebagai referensi yang mudah; *sintaks*, tanda kurung, *indentation highlighting*, fasilitas pencarian yang tangguh, *macro recording* untuk tugas-tugas seperti memasukkan template komentar, dan sebagainya. Salah satu kelebihan *Notepad++* adalah dukungan dasar untuk *auto-completion* dari nama fungsi yang ditawarkan sehingga akan mengurangi beberapa proses pengetikan kode.

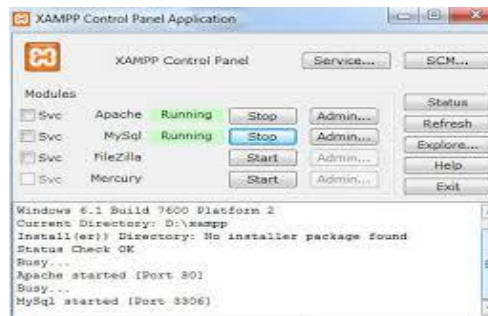
#### 6. XAMPP (X Apache MySQL PHP Perl)

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program, berikut logo dari XAMPP.



**Gambar 2.22** Logo XAMPP

**Sumber:** (Jubliee, 2018)



**Gambar 2.23** Tampilan awal xampp

**Sumber:** (Jubliee, 2018)

Fungsinya adalah sebagai *server* yang berdiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

## 7. MySQL

MySQL merupakan salah satu database kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. MySQL bekerja menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standard yang digunakan untuk manipulasi database. Pada umumnya, perintah yang sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, *structure query language* juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data (Jubliee, 2018).



**Gambar 2.24** Logo MySQL

**Sumber:** (Jubliee, 2018)

### 1. Mengapa MySQL

Ada beberapa alasan yang menjadikan database MySQL sangat diminati oleh para *programmer*, diantaranya:

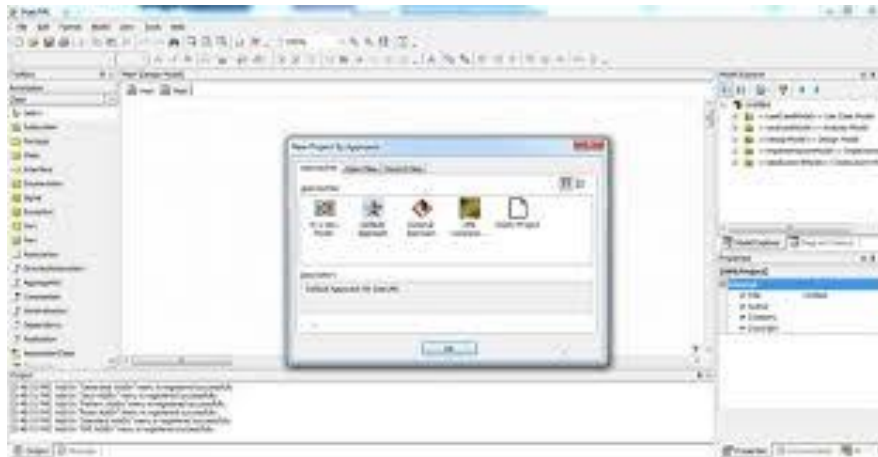
- a. Bersifat open source.
- b. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dalam pengolahan data.
- c. *Performance* dan *reliable*, pemrosesan *database*-nya sangat cepat dan stabil.
- d. Sangat mudah dipelajari (*ease of use*).
- e. Memiliki dukungan (*group*) pengguna MySQL.
- f. Lintas *Platform*, dapat digunakan pada berbagai sistem operasi berbeda.
- g. *Multiuser*, dimana MySQL dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.

### 8. StarUML

Salah satu pemodelan yang saat ini paling banyak digunakan adalah UML (*Unified Modeling Language*), berikut logo UML.



**Gambar 2.25** Logo StarUML  
**Sumber:** (Jubilee, 2018)



**Gambar 2.26** Tampilan awal Star UML  
**Sumber:** (Jubilee, 2018)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (S & Shalahuddin, 2013).

Menurut (A.S & Shalahuddin, n.d.) UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML tidak terbatas pada metodologi pemrograman tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek.

*StarUML* merupakan salah satu *CASE* (*Computer-Aided Software Engineering*) *tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upperCASEtools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak menjadi 3 kategori, yaitu (S & Shalahuddin, 2013):

### 1. *Structure diagrams*

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

### 2. *Behaviour diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Diagram UML yang termasuk dalam kategori ini antara lain *use case diagram*, *activity diagram*, dan *state machine diagram*.

### 3. *Interaction diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram UML yang termasuk dalam kategori ini

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

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

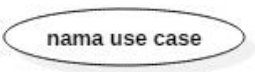


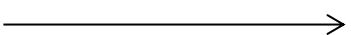
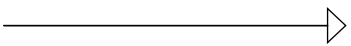
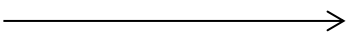
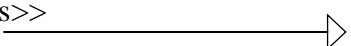
#### 1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem 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 dua hal utama yang terdapat pada *use case* yaitu aktor dan *use case*.

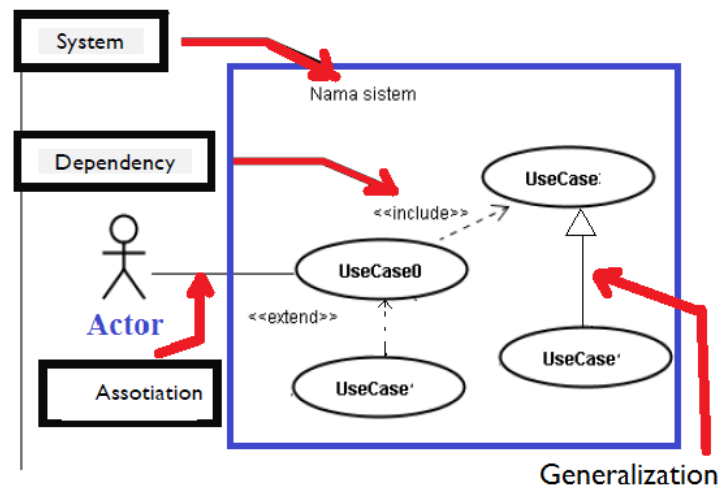
Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S & Shalahuddin, n.d.)



Tabel 2.5 Simbol Use Case Diagram

Simbol	Deskripsi
<p><b>Use case</b></p> 	<p>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><b>Aktor/actor</b></p> 	<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>
<p><b>asosiasi/association</b></p> 	<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>
<p><b>Ekstensi/extend</b></p> <p>&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p><b>generalisasi/generalization</b></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p><b>Menggunakan/include/uses</b></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini. Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

**Sumber:**(A.S & Shalahuddin, n.d.)



**Gambar 2.27** *Usecase Diagram*  
**Sumber:** (A.S & Shalahuddin, n.d.)

## 2. Class Diagram

Menurut (A.S & Shalahudin, 2013: 14) diagram kelas atau *class diagram* yang menggambarkan struktur dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunaan struktur kelas yang pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

### 1. Kelas Main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.


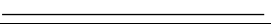




3. Kelas yang diambil dari pendefinisian *usse case*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *usse case*.

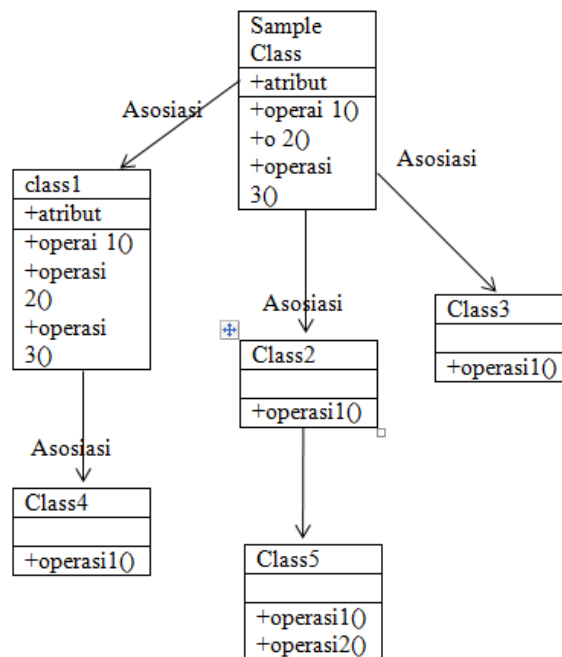
4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

**Tabel 2.6** Simbol Class Diagram

Simbol	Deskripsi
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <div style="border-bottom: 1px solid black; padding-bottom: 2px;">nama_kelas</div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;">+atribut</div> <div style="padding-bottom: 2px;">+operasi</div> </div>	Kelas pada struktur sistem
Antarmuka/interface  nama_interface	Sama dengan konsep interface dalam pemrograman dalam berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>defedency</i> 	Relasi antara kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

**Sumber:** (A.S & Shalahuddin, n.d, 2013)








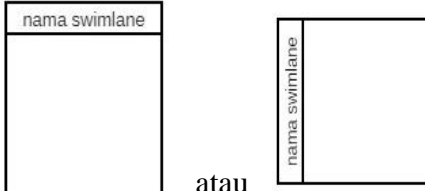
**Gambar 2.28** Class Diagram

**Sumber:** (A.S & Shalahuddin, n.d.)

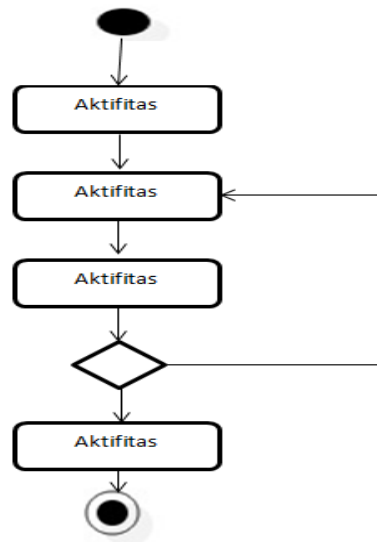
### 3. Activity Diagram

Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Jadi dapat dikatakan bahwa activity diagram menggambarkan aktifitas sistem, bukan apa yang dilakukan oleh aktor. Simbol-simbol yang digunakan dalam activity diagram ditampilkan dalam tabel berikut (A.S & Shalahuddin, n.d.)

**Tabel 2.7** Simbol *Activity Diagram*

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

**Sumber:** (S & Shalahuddin, 2013)


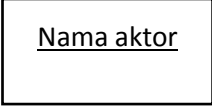




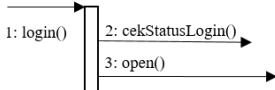

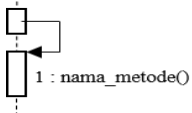
**Gambar 2.29** *Usecase Diagram*  
**Sumber:** (A.S & Shalahuddin, 2018)


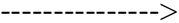
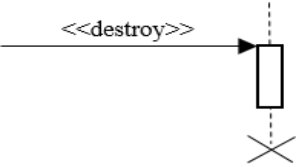
#### 4. Sequence Diagram

Menurut (A.S & Shalahuddin, 2018) Diagram sekuen ini menggambarkan tanggapan atau reaksi objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek.

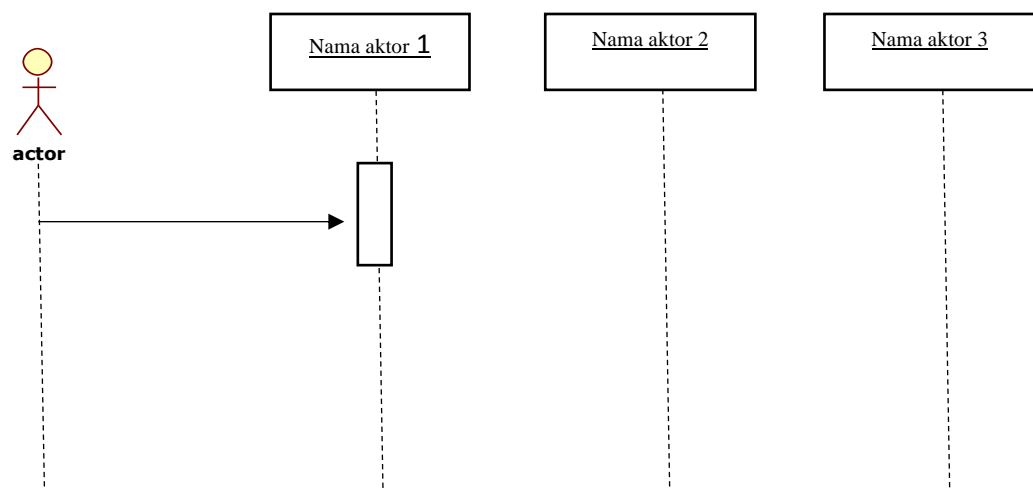
**Tabel 2.8** Simbol *Sequennce Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p>  <p><u>Nama aktor</u></p> <p>Tanpa waktu aktif</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

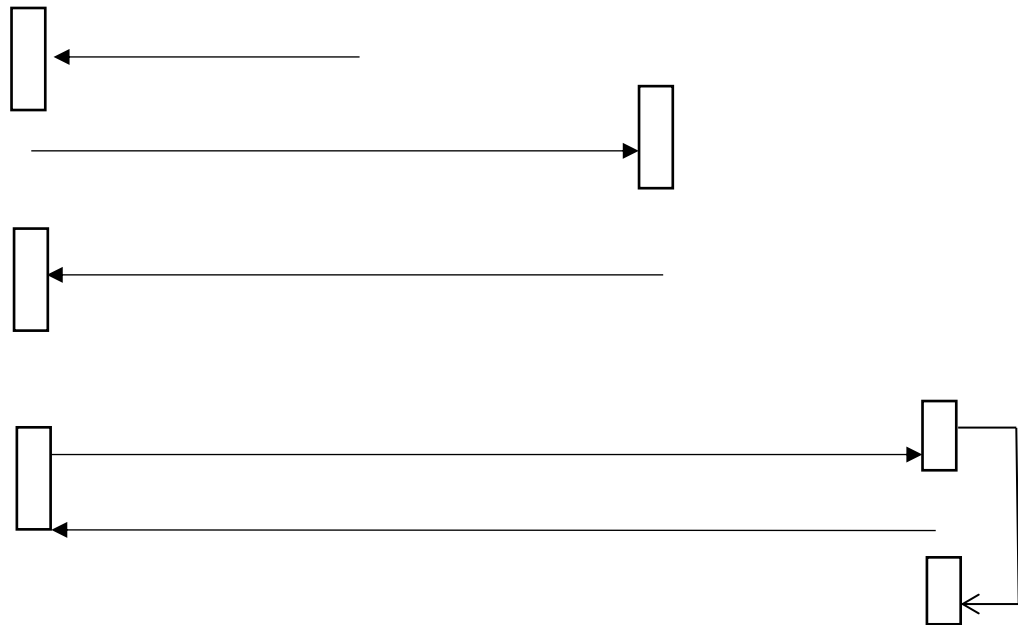
<p>Garis hidup/ <i>lifeline</i></p> 	<p>Konunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada dan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor, menyatakan suatu objek.</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p>Nama objek : nama kelas</p> </div>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan dan berinteraksi,</p>  <p>semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan dalamnya. Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe <i>create</i></p> <p>&lt;&lt;create&gt;&gt;</p>	<p>Menyatakan suatu abjek membuat objek yang lain.</p> <p>Arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i> →</p> <p>1: nama_metode()</p> 	<p>operasi/metode yang ada pada objek lain atau diri sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang</p>

	berinteraksi.
Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

**Sumber:**(A.S & Shalahuddin, n.d.)







**Gambar 2.30** Sequence Diagram  
**Sumber:** (A.S & Shalahuddin, 2018)

## 2.4 Penelitian Terdahulu

Pada sub judul ini dijabarkan jurnal dari beberapa penelitian yang berhubungan dengan judul yang di angkat pada penelitian ini yang telah dilakukan oleh peneliti lain sebelumnya, yang dapat mendukung sebagai dasar pembahasan penelitian serta digunakan untuk memperkuat dan menambah referensi penelitian.

1. (Samsudin, 2018) **Penerapan Sistem Pakar Diagnosa Dini Penyakit Lambung Menggunakan Metode Forward Chaining** diperoleh fakta bahwa, Penyakit dalam merupakan penyakit yang berkaitan dengan banyak organ penting dalam tubuh kerana merupakan salah satu organ pencernaan makanan dan

minuman yang di konsumsi manusia. Penyakit lambung kurang di ketahui oleh masyarakat karena kurang nya informasi dan prengetahuan tentang penyakit lambung sehingga masyarakat mengabaikan gejala-gejala yang timbul.

2. (Murni & Riandari, 2018) **Penerapan Metode Teorema Bayes Pada Sistem Pakar Untuk Mendiagnosa Penyakit** diperoleh fakata bahawa, Penyakit lambung ini dapat terkena gangguan atau terinfeksi penyakit sehingga perlu diwaspadai oleh masyarakat. Oleh karena itu, penyebaran informasi tentang penyakit ini sangat diperlukan untuk mengetahui terlebih dini jenis penyakit yang dideritanya.

3. (Winanda et al., 2016) **Perancangan Sistem Pakar Diagnosa Penyakit Dalam Berbasis Android**di peroleh fakata bahwa, Suatu gejala penyakit dapat merupakan indikasi dari suatu penyakit yang akan diderita. Setiap orang wajib menjaga kesehatannyamasing-masing, tetapi pada kenyataannya banyak sekali orang yang lupa atau bahkan meremehkan gejala penyakit yang dideritanya. Maka dengan adanya kemajuan teknologi saat ini, suatu penyakit akan terdeteksi dengan lebih cepat melalui gejala-gejala tersebut.

4. (Ritonga, 2013) **Siatem Pakar Mendiagnosa Penyakit Lambung Menggunakan Metode Certainty Factordi** peroleh fakta bahawa, Penyakit lambung antara lain adalah sakit maag (*gastritis*), dispepsia dan *gastroesophageal reflux disease* (GERD). Penyakit maag diakibatkan oleh asam lambung yang berlebihan, sehingga dinding lambung lama-lama tidak kuat menahan asam lambung tadi sehingga timbul luka.

5. (Putri & Suhendra, 2016) **Sistem Pakar Untuk Mendeteksi Kerusakan Air Kondisioner Menggunakan Metode Forward Chaining Berbasis Web**

peroleh fakta bahwa, Kemampuannya untuk mendinginkan ruangan telah menarik minat pengguna untuk membantu mereka memperoleh kenyamanan dalam melakukan kegiatan sehari-hari terutama pada saat cuaca sedang panas.

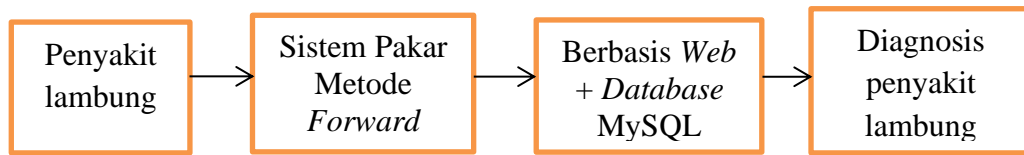
6. (Soepomo, 2014) **Sistem Pakar Mendiagnosa Penyakit Lambung Dengan Implementasi Metode CBR (Case-Based Reasoning) Berbasis Web**

peroleh fakta bahwa, Penyakit lambung merupakan penyakit yang tidak bisa dianggap remeh, karena jika dibiarkan terus menerus dapat mengakibatkan penyakit yang lain muncul dan bisa juga menyebabkan kematian jika tidak segera ditangani. Penyakit lambung dapat disebabkan oleh pola makan yang tidak sesuai, beban pikiran dan juga infeksi yang disebabkan oleh bakteri.

## 2.5 Kerangka Pemikiran

Secara teoritis, kerangka berfikir yang baik akan menjelaskan peraturan antar variabel yang akan diteliti. Kerangka berfikir dalam penelitian perlu dikemukakan apabila penelitian berkaitan dengan dua variabel atau lebih. Kerangka pemikiran adalah penjelasan sementara terhadap gejala-gejala yang menjadi objek permasalahan. Jadi, kerangka berfikir adalah sintesis tentang hubungan antar variabel yang disusun dari berbagai teori yang telah dideskripsikan. Sintesis itulah yang selanjutnya digunakan untuk merumuskan hipotesis (Sugiyono, 2011).

Dengan ini dapat dibuat kerangka pemikiran sebagai berikut:



**Gambar 2.31** Kerangka Pemikiran

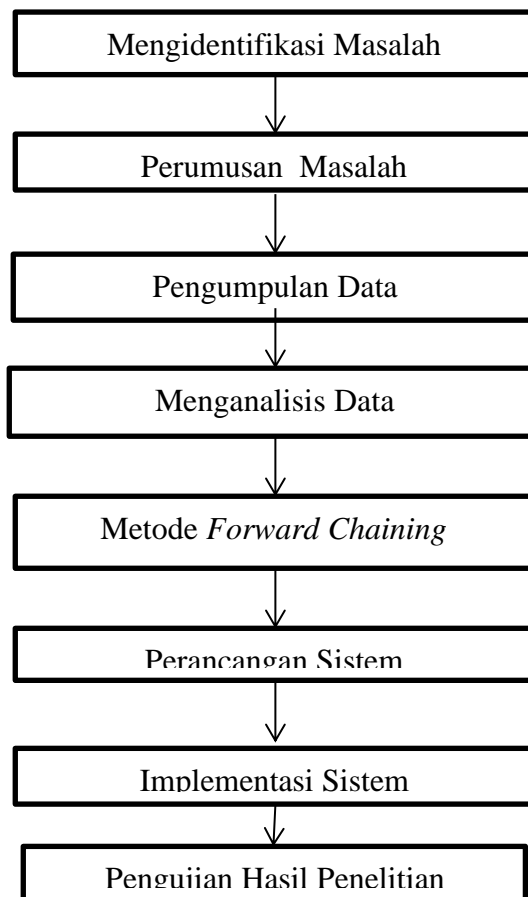
**Sumber:** Data Penelitian 2019

Langkah pertama akan mengidentifikasi apa yang menjadi masalah, dengan melihat apa yang menjadi penyebab sehingga mengakibatkan Lambungbermasalah. Setelah mengidentifikasi masalah tersebut, maka akan melakukan dan menyelesaikannya dengan sistem pakar menggunakan metode *forward chaining*, dari sistem pakar maka akan menemukan apa yang menjadi penyebabnya dan menghasilkan apa saja yang menjadi penyebab penyakit lambung, kemudian akan menyelesaikannya dengan mesin inferensi dengan metode *forward chaining*, jika sesuai dengan aturan *rule* maka akan dilanjutkan dengan mengaplikasikannya dalam sebuah bahasa pemograman berbasis *web* dengan menggunakan *database* MySQL, jika tidak sesuai maka akan kembali mengidentifikasi masalah yang terjadi. Agar lebih mudah di pahami, maka penulis akan memasukkan data dari gejala penyakit ke dalam bahasa pemograman dengan menginputkan gejala-gejala yang dialami oleh lambung (melakukan diagnosa penyakit lambung). Setelah melakukan pengimputan maka akan menghasilkan (*output*) status penyakit lambung, keterangannya, jenis penyakit apa yang terjadi pada lambung tersebut, dan solusi yang dapat mencegah penyakit lambung tersebut tersebut.

## BAB III METODE PENELITIAN

### 3.1 Desain Penelitian

Desain penelitian adalah semua langkah yang diperlukan untuk mengatur dan melakukan penelitian. Pertama-tama, ini adalah deskripsi hubungan antara volatilitas, data, dan analisis data (Noor, 2011). Dibawah ini penelitian menggunakan desain seperti berikut:



**Gambar 3.1** Desain Penelitian 2019  
**Sumber:** Data Penelitian 2019

Desain penelitian yang digunakan untuk mendeteksi penyakit lambung adalah struktur penelitian yang digunakan untuk memperoleh bukti-bukti empiris, adapun uraian yang dilakukan yaitu:

1. Mengidentifikasi masalah

Langkah yang diambil dalam penelitian ini adalah penerapan identifikasi masalah untuk menentukan objek yang harus dipelajari, seperti penyakit lambung.

2. Perumusan Masalah

penelitian ini merumuskan masalah penyakit pada lambung yang akan dibahas setelah mendapatkan informasi yang spesifik maka, berdasarkan identifikasi masalah yang telah dilakukan yaitu: Gastritis, Dispepsia, Kanker Lambung, dan Gerd.

3. Penentuan Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui bagaimana Sistem Pakar untuk penyakit Lambung menggunakan metode *forward chaining* berbasis web. Selanjutnya peneliti melakukan landasan teori. Landasan Teori

Peneliti mencari dan mempelajari sumber-sumber pengetahuan berupa buku-buku teori, jurnal-jurnal penelitian, dan sumber-sumber pustaka otentik lainnya yang berkaitan dengan penelitian, untuk membantu jalannya penelitian, diantaranya yaitu buku mengenai kecerdasan buatan, sistem pakar, penyakit lambung, pemrograman web, dan *UML*. Berikutnya dilakukan pengumpulan data.

#### 4. Mempelajari Kepustakaan (Literatur)

Peneliti mencari dan mempelajari sumber-sumber pengetahuan berupa buku-buku teori, jurnal-jurnal penelitian dan sumber pustaka yang berkaitan dengan penelitian. untuk mendukung jalannya penelitian, maka diantaranya kecerdasan buatan, sistem pakar, PHP, UML, dan *mySQL*

#### 5. Pengumpulan data

Tujuan dalam pengumpulan data adalah untuk mendapatkan data-data berupa gejala-gejala dari penyakit lambung melalui wawancara dengan dr.Supardi dari Klinik Nira Medika Batam dan *study* pustaka mengenai gejala-gejala penyakit lambung. Mengolah data dengan metode *forward chaining*, dan dilakukan analisa

#### 6. Mengolah dan Mengalisa Data dengan metode *forward chaining*

Penelitian ini pertama-tama mengeksplorasi variabel dan kemudian membahas metode yang telah digunakan. Sistem pakar ini menggunakan model representasi pengetahuan berdasarkan aturan produksi. Sistem pakar dapat menghasilkan suatu kesimpulan berdasarkan atau kaidah yang ada. Oleh sebab itu data yang sudah dianalisa kemudian diolah dengan metode *forward chaining* untuk membuat aturan (*rule*) yang digunakan pada saat sistem pakar melakukan penelusuran sebelum menyimpulkan hasil. Setelah itu dilakukan perancangan sistem berbasis *web*.

#### 7. Perancangan Sistem Berbasis *Web*

Pada bagian ini, para peneliti mengembangkan desain berdasarkan pada pengembangan pengetahuan, desain UML, desain basis data dan antarmuka.

Kemudian dilakukan menggunakan HTML, PHP, *JavaScript* dan *jQuery* setelah membuat database menggunakan phpMyAdmin, yang ada di program XAMMP. Langkah selanjutnya adalah mengembangkan sistem pakar yang dikembangkan dan mengevaluasi hasil penelitian ini.

#### 8. Pengujian hasil Penelitian

proses ini bertujuan untuk mengurangi kesalahan dan memastikan keluran yang dihasilkan sesuai dengan yang diinginkan. Selanjutnya tahap terakhir yaitu menarik kesimpulan. Setelah pembuatan aplikasi sistem pakar yang telah dirancang kemudian dilakukanlah pengujian hasil penelitian.

#### 9. Menarik Kesimpulan

Hasil dapat diperoleh ketika informasi diperoleh, diidentifikasi, diproses, dan dilakukan berdasarkan jenis penelitian dan evaluasi penelitian maka dapat ditarik kesimpulan.

### 3.2 Pengumpulan Data

Metode pengumpulan data adalah metode dan metode yang dapat digunakan untuk mengumpulkan informasi. Pengumpulan data untuk penelitian dimaksudkan untuk menyediakan konten, informasi, fakta, dan informasi yang dapat diandalkan. Untuk keperluan pengumpulan data, berbagai metode dapat digunakan dalam penelitian ini, seperti kuesioner, observasi, wawancara, tes, analisis dokumen. Survei bisa menggunakan salah satunya, tergantung masalahnya (Sugiyono, 2011).



Dalam penelitian ini, para peneliti menggunakan teknik pengumpulan data primer menggunakan metode wawancara dan data sekunder menggunakan metode dokumentasi.

#### 1. Wawancara

Wawancara atau *interview* adalah metode pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumber dalam wawancara yang dilakukan tatap muka (Sugiyono, 2011). Teknik pengumpulan data ini digunakan pertanyaan secara lisan kepada subjek peneliti **pakar** (ahli) yang ada di Klinik Nira Medika Batam. Dengan wawancara ini diharapkan mendapatkan informasi yang sebenarnya untuk mendiagnosa penyakit lambung guna memperoleh data secara tepat dan akurat dalam pengolahan data lebih lanjut.

#### 2. Studi Literatur

Menurut (Noor, 2011) Studi literatur bertujuan untuk menemukan variabel yang akan diteliti, membedakan hal-hal yang sudah dilakukan dan menentukan hal yang perlu dilakukan, melakukan sintesa dan memperoleh perspektif baru, dan menentukan makna dan hubungan antar variabel. Studi literatur diperoleh melalui mengumpulkan, membaca dan memahami referensi teoritis yang berasal dari buku-buku teori, buku elektronik (*e-book*), jurnal-jurnal penelitian, dan sumber pustaka lainnya. Sifat utama dari studi literatur adalah tak terbatas pada ruang dan waktu sehingga memberi peluang kepada peneliti untuk mengetahui hal-hal yang pernah terjadi di waktu yang lalu.

### 3.3 Operasional Variabel

Operasi dilakukan untuk memfasilitasi pengumpulan data dan untuk mencegah interpretasi dan untuk membatasi ruang lingkup variabel. Dengan definisi operasional, maka dapat ditentukan cara yang di pakai untuk mengukur variabel, tidak terdapat arti dan istilah-istilah ganda yang apabila tidak dibatasi akan menimbulkan tafsiran yang berdeda. Variabel yang dimasukkan dalam definisi operasional adalah variabel kunci/penting yang dapat diukur secara operasional dan dapat dipertanggung jawabkan (Sugiyono, 2011).

Berikut beberapa variabel untuk menentukan jenis penyakit apa yang diderita seseorang yang berkaitan dengan penyakit lambung yaitu:

**Tabel 3.1** Operasional Variabel penyakit pada lambung

Variabel	Indikator
PENYAKIT LAMBUNG	Gastritis
	Dispepsia
	Kanker Lambung
	Gerd

**Sumber:** Pengolahan Data Penelitian (2019)

Pada **tabel 3.1** data menjelaskan variabelnya adalah penyakit lambung hubungan antara variabel dan indikator, sedangkan indikatornya ada empat jenis yaitu: Gastritis, Dispepsia, Kanker Lambung, dan Gerd

**Tabel 3.2** Indikator, Gejala Dan Solusi

<b>Indikator</b>	<b>Gejala</b>	<b>Penyebab</b>	<b>Solusi</b>
Gastritis	1. Nyeri tekan abdomen 2. Dehidrasi 3. Muntah	1. Konsumsi alkohol 2. Terapi radiasi 3. Kemoterapi 4. Zat-zat korosif (cuka, lada)	Awasi tingkat stres, batasi konsumsi alkohol, dan berhenti merokok
Dispepsia	1. Rasa sakit dan tidak enak di ulu hati 2. Perih 3. Mual 4. Sering sendawa	1. Faktor asam lambung pasien 2. Kelainan psikis 3. Gangguan motilitas	Makan dengan porsi kecil, hindari makanan pedas, minuman yang mengandung gas, menjaga berat badan dan olahraga secara teratur
Kanker Lambung	1. Penurunan berat badan 2. Muntah 3. Asites (perut membesar) 4. Darah yang samar atau nyata dalam tinja	1. Gastritis kronis 2. Sering makan daging dengan cara dipanggang, dibakar atau diasapkan 3. Sering makan makanan pedas	Makan makanan yang sehat, tidak merokok, dan aturan penggunaan obat
Gerd	1. Muntah 2. Nyeri pada dada 3. Batuk	1. Menurunnya tonus 2. Ketahanan epitel esofagus menurun 3. Bersihan asam dari lumen esofagus menurun 4. Bahan refluks mengenai dinding esofagus	Turunkan berat badan, hindari makan jumlah besar, hindari berbaring setelah makan, hindari rokok, dan hindari makanan yg dapat mengiritasi mukosa

**Sumber:** Pengolahan Data Penelitian 2019

### 3.4 Metode Perancangan Sistem

Dalam melakukan perancangan sistem dan membangun sistem pakar mendiagnosa penyakit lambung dengan menggunakan metode *forward chaining*, pengkodean yang harus dibentuk dalam perancangan ini yaitu (nama penyakit, gejala), memberikan aturan (*rule*), membuat model inferensi dan pohon keputusan. Perancangan sistem disini menggunakan metode perancangan UML (*Unified Modelling Language*). UML adalah "bahasa model terpadu" yang merupakan metode pemodelan visual untuk alat desain sistem berorientasi objek, UML dapat digambarkan sebagai bahasa yang terstandarisasi dalam visualisasi, desain, dan dokumentasi *software* sistem perangkat lunak. Langkah-langkah yang diambil adalah sebagai berikut:

#### 3.4.1 Desain Basis Pengetahuan

Peneliti dapat merancang basis pengetahuan setelah melakukan data dan proses pengumpulan fakta melalui wawancara dengan para ahli dan studi literatur tentang topik yang terkait dengan data. Sumber fakta berupa data penyakit lambung. Pengetahuan dan fakta ditampilkan dalam tabel berikut:

**Tabel 3.3** Jenis Penyakit, penyebab dan Solusi

Kode	Jenis Penyakit	Gejala	Solusi
JP01	Gastritis	1. Nyeri tekan abdomen 2. Dehidrasi 3. Muntah	Awasi tingkat stres, batasi konsumsi alkohol, dan berhenti merokok
JP02	Dispepsia	1. Sakit dan tidak enak di ulu hati 2. Perih 3. Mual 4. Sering sendawa	Makan dengan porsi kecil, hindari makanan pedas, minuman yang mengandung gas, menjaga berat badan dan olahraga secara teratur

JP03	Kanker Lambung	1. Berat badan menurun 2. Muntah 3. Asites (perut membesar)	Makan makanan yang sehat, tidak merokok, dan aturan penggunaan obat
JP04	Gerd	1. Darah yang nyata atau samar dalam tinja 2. Nyeri pada dada 3. Batuk	Turunkan berat badan, hindari makan jumlah besar, hindari berbaring setelah makan, hindari rokok, dan hindari makanan yg dapat mengiritasi mukosa

Sumber: Data Penelitian (2019)

**Tabel 3.4** Tabel Gejala Penyakit

Kode	Gejala
G001	Nyeri tekan abdomen
G002	Dehidrasi
G003	Muntah
G004	Sakit dan tidak enak di ulu hati
G005	Perih
G006	Mual
G007	sering sendawa
G008	Berat badan menurun
G009	muntah
G010	asites (perut membesar)
G011	darah yang nyata atau samar dalam tinja
G012	nyeri pada dada
G013	Batuk

**Sumber:** (Data Penelitian (2019))

Dalam tabel 3.2 solusi yang harus menjelaskan di lakukan sesuai jenis penyakit Lambung yang diderita. Kemudian pemberian kode untuk setiap jenis penyakit Lambung tersebut. Sistem pakar yang menggunakan metode *forward chaining* dalam penelitian ini digunakan untuk mendiagnosis penyakit lambung pada manusia sehingga data tidak dienkripsi.

Pada tabel 3.3 menjelaskan gejala dari masing-masing jenis penyakit lambung dan kemudian diberi kode.

Data aturan ini disusun untuk memudahkan peneliti dalam menyusun kaidah yang akan digunakan sebagai basis pengetahuan dalam sistem pakar pada penelitian ini. Data aturan merupakan data yang berisi relasi antara data jenis penyakit dan data gejala penyakit yang telah diberi kode sebelumnya. Relasi antara data tersebut disusun berdasarkan sumber pengetahuan dan fakta yang telah didapatkan.. Susunan data aturan yang digunakan dapat dilihat pada tabel dibawah ini.

### 3.4.2 Aturan (Rule)

**Tabel 3.5** Aturan (Rule)

Kode	Jenis Penyakit Lambung
JP01	G001, G002, G003
JP02	G004, G005, G006, G007
JP03	G008, G009, G010,
JP04	G011, G012, G013

**Sumber:** Data Penelitian (2019)

Dari **Tabel 3.5** diatas, Dalam penelitian ini, pengetahuan diwakili oleh aturan berikut: JIKA-LALU. Berikut adalah aturan yang harus diterapkan dalam sistem pakar sebagai berikut:

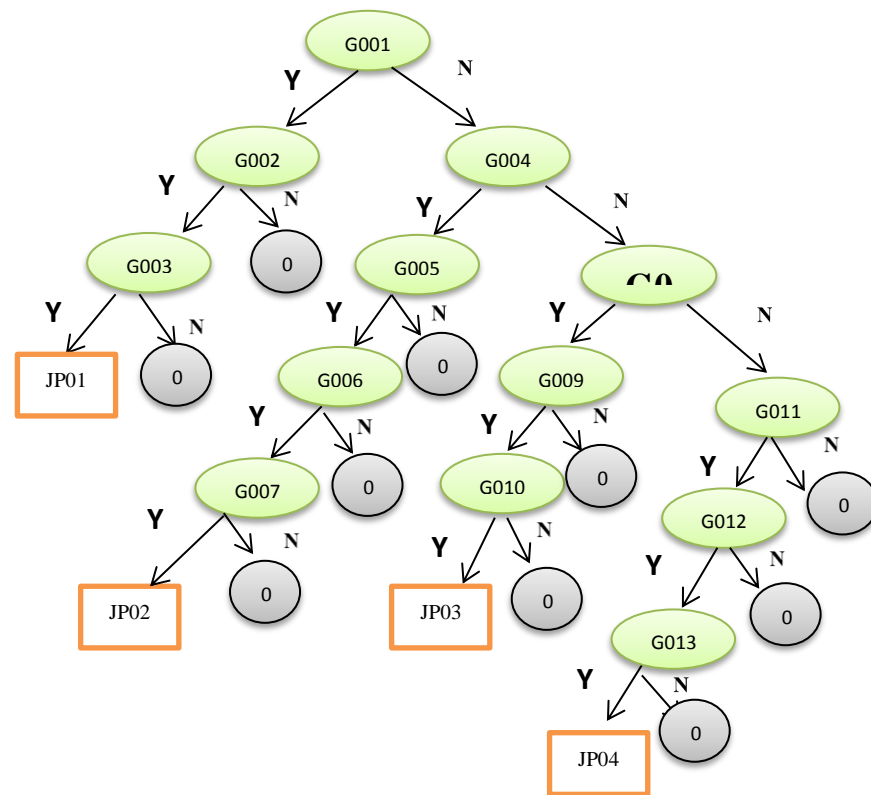
1. kaidah 1: *IF G001 AND G002 AND G003 THEN JP01*
2. kaidah 2: *IF G004 AND G005 AND G006 AND G007 THEN JP02*
3. kaidah 3: *IF G008 AND G009 AND G010 THEN JP03*
4. kaidah 4: *IF G011AND G012AND G013THEN JP04*

Sebelum membuat pohon keputusan, disini harus terlebih dahulu membuat tabel keputusan dengan mengikuti tabel keputusan: berikut tabel keputusan:

**Tabel 3.6** Tabel Keputusan

Gejala	P001	P002	P003	P004
G001	√			
G002	√			
G003	√			
G004		√		
G005		√		
G006		√		
G007		√		
G008			√	
G009			√	
G010			√	
G011				√
G012				√
G013				√

**Sumber:** Data penelitian 2019



**Gambar 3.2** Pohon Keputusan  
**Sumber:** Pengolahan Data Penelitian (2019)

Sebagai keadaan awal dalam data gejala ditentukan sistem pada saat melakukan penelusuran ssebelum diperoleh sebuah kesimpulan. Pohon keputusan pada gambar 3.6 memperlihatkan hubungan terkait antar gejala yang ada. Digunakan untuk arah penelusuran pada pohon keputusan tersebut dimulai dari simpul akar (yang paling atas) kebawah. Alur penelusuran sistem pakar dimulai dari G001, yaitu konsumsi alkohol. Gejala ini dipilih sebagai keadaan awal dalam penelusuran karena gejala ini adalah gejala yang paling mudah di priksa dan mudah di ketahui.

Proses pencarian tergantung pada umpan balik yang diberikan pengguna. Jika pengguna mengatakan ya, maka Pencarian akan pergi ke tanda paling bawah.



Pencarian akan "G001-G007. Demikian pula, jika pengguna menjawab" Tidak ", maka pencarian akan dikirim kepenelusuran selanjutnya sampai selesai.

### **3.5 Perancangan Sistem**

Pembuatan sebuah sistem informasi juga memerlukan beberapa buah tahapan sebagaimana proses pembuatan sebuah perangkat lunak,. Tahapan-tahapan tersebut dimulai dari desain, perancangan, hingga implementasi dan pengujian. Berdasarkan sudut pandang keilmuan informatika, langkah-langkah tersebut termasuk dalam kajian penelitian. Pembuatan sebuah sistem informasi diawali dengan kebutuhan pengguna berdasarkan permasalahan yang terjadi. UML (*Unified Modelling Language*) adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek (Pratama, 2014)

#### **3.5.1 Use Case Diagram**

*Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sistem dan merepresentasikan interaksi antara aktor dengan sistem. *Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

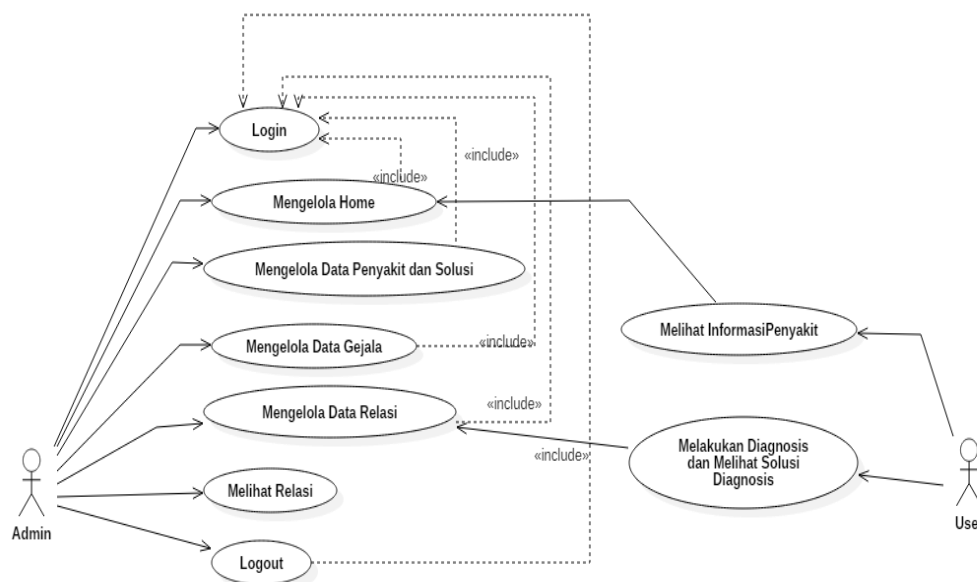
### 3.5.2 Desain UML (Unified Modeling Language)

*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang digunakan di sebagian besar industri di seluruh dunia untuk mengidentifikasi persyaratan, melakukan analisis dan desain, dan menggambarkan arsitektur pemrograman berorientasi objek (A.S. & Shalahuddin)

Berikut ini adalah diagram UML yang digunakan dalam perancangan program. Desain UML dibuat untuk memudahkan dalam pembuatan program. Pemodelan UML menggunakan alat bantu software StarUML versi 2.8.0.

#### 1. Use case Diagram

*Use case diagram* yang akan digunakan pada sistem diagnosis penyakit lambung seperti pada **Gambar 3.3:**



**Gambar 3.3** Use case Diagram

**Sumber:** Data Penelitian (2019)

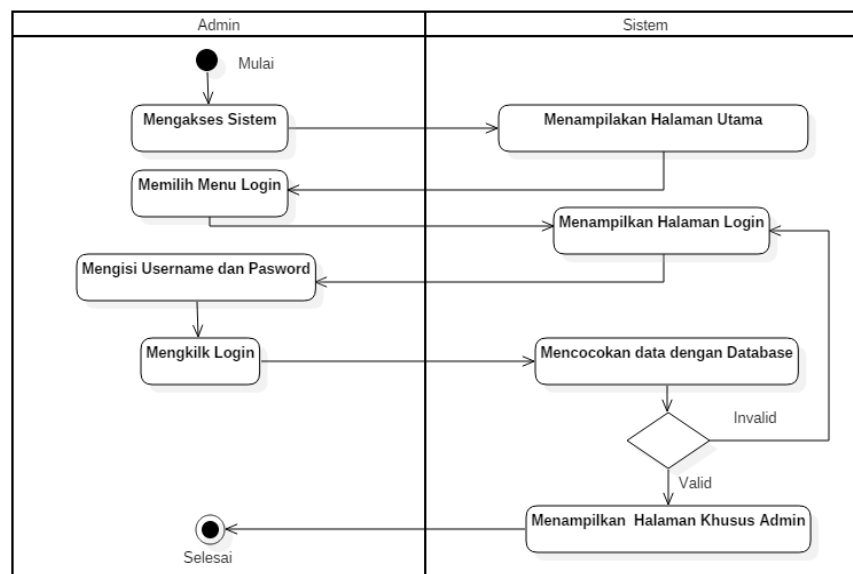
Admin dan user kedua ini disebut aktor. Mengelola home merupakan intersai sistem yang dilakukan admin, mengelola data penyakit dan solusi, mengelola data gejala, mengelola data relasi, melihat relasi dan *logout*. Sedangkan *user* berinteraksi dengan sistem yaitu dapat melihat menu *home* atau informasi penyakit lambung dan melakukan diagnosis mengenai penyakit lambung dan dapat melihat hasil dari diagnosis yang dilakukan yaitu berupa solusi. Semua interaksi dapat dilakukan setelah admin melakukan *Login* pada menu *Login*.

## 2. Activity Diagram

Berikut ini adalah diagram *activity* (kegiatan) yang dirancang dalam penelitian ini:

### a. Activity Diagram Login Admin

*Activity diagram Login admin* merupakan *UML* yang menggambarkan kegiatan pengguna pada halaman khusus *admin*.



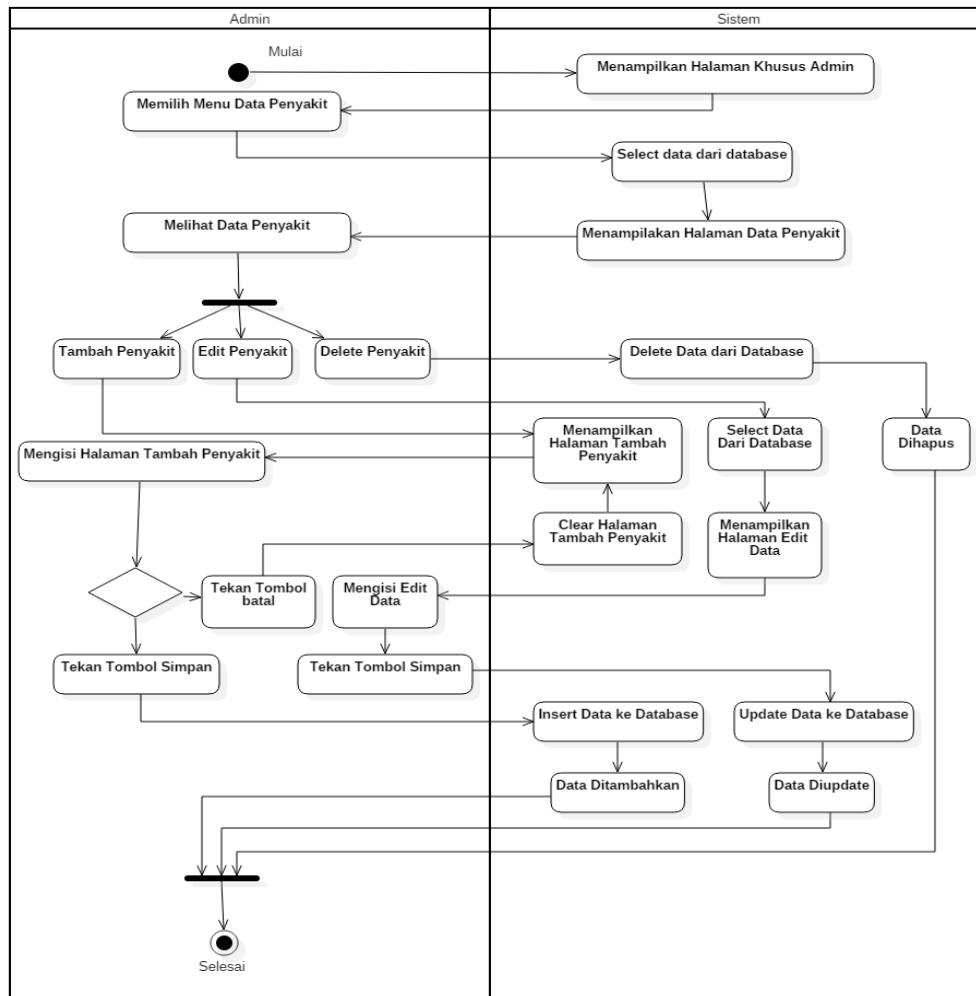
**Gambar 3.4** Activity Diagram Login

**Sumber:** Data Penelitian (2019)

Pada **Gambar 3.4** diatas, Jika username dan *password* tidak sesuai dengan yang ada di database maka sistem akan menampilkan pesan kesalahan dan sistem kembali menampilkan halaman menu Login, apabila benar maka sistem akan menampilkan halaman khusus sistem. proses Login admin adalah admin mengakses sistem, kemudian sistem akan melakukan halaman utama. Kemudian admin akan memilih menu admin dan sistem akan menampilkan halaman khusus admin. Admin akan mengisi username dan password pada menu Login, kemudian klik tombol Login. Maka sistem akan mengecek username dan password kemudian dicocokkan dengan data yang ada di database.

b. *ctivitydiagram* Mengelola Menu Penyakit

*Activity diagram* mengelola data jenis penyakit dan solusi yang digunakan menu penyakit merupakan kegiatan *admin* dalam menu jenis penyakit. Berikut ini gambar *activity diagram* mengelola menu penyakit (**Gambar 3.5**):



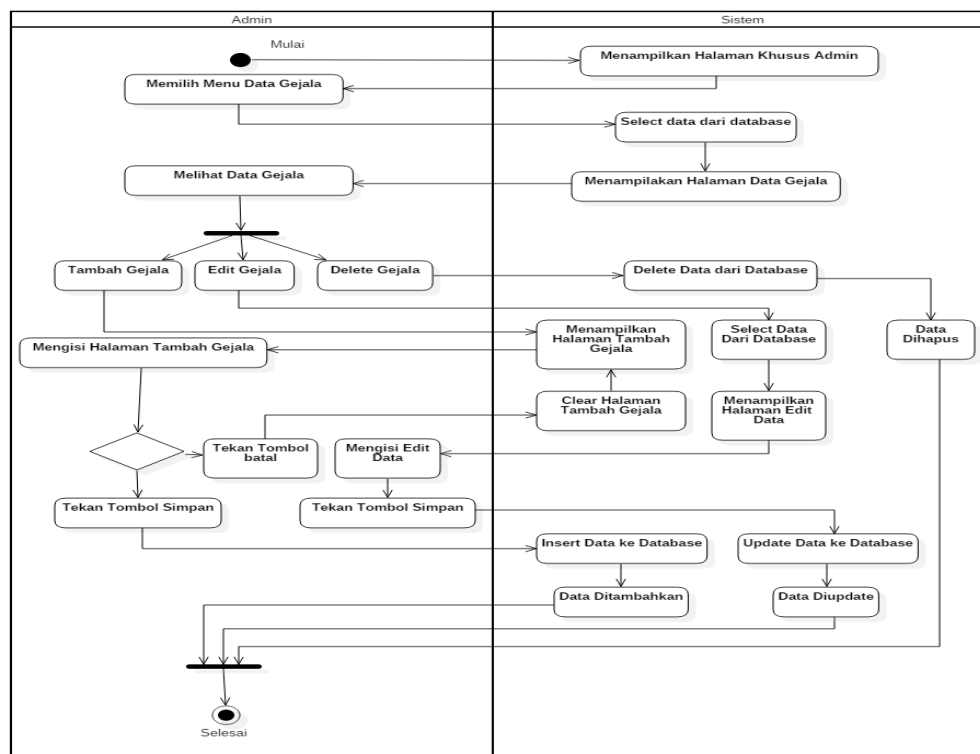
**Gambar 3.5 Activity Diagram**  
**Sumber:** Data Penelitian (2019)

Pada **Gambar 3.5** diatas, setelah *admin* melakukan login maka halaman admin akan terbuka halaman khusus *admin*. Sistem akan menampilkan menu-menu pada halaman khusus *admin*, kemudian *admin* memilih menu penyakit. Sistem memanggil data dari database dan menampilkan halaman data penyakit. *Admin* melihat 3 pilihan yaitu tambah, *edit* , dan *delete*. Jika *admin* mengklik tombol tambah maka sistem akan menampilkan halaman sistem penyakit, kemudian admin mengisi data penyakit dan solusi. Kemudian memilih tombol

simpan maka data akan memasukkan ke *database* kemudian data ditambahkan di *database*, jika *admin* menekan tombol batal maka sistem akan membersihkan halaman tambah penyakit. Jika *admin* memilih tombol *edit*, maka sistem akan mengambil data dari *database*. *Admin* mengedit data kemudian klik tombol simpan. Sistem akan melakukan *update database*. Jika *admin* menekan tombol *delete* maka data yang ada di *database* akan terhapus maka proses selesai.

### c. Activity Diagram Mengelola Gejala

*Activity diagram* mengelola menu gejala merupakan kegiatan *admin* untuk mengontrol gejala penyakit yang digunakan. Berikut ini gambar *activity diagram* mengelola menu gejala (**Gambar 3.6**):



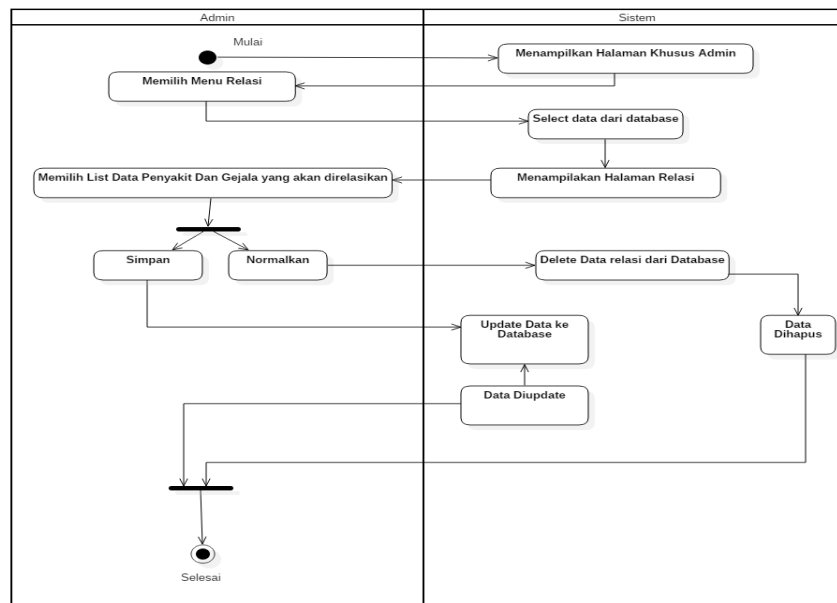
**Gambar 3.6** Activity Diagram Mengelola Gejala

Sumber: Data Penelitian (2019)

Pada **Gambar 3.6**, Sistem akan menampilkan menu-menu pada halaman khusus *admin*, Sistem memanggil data dari database dan menampilkan halaman data gejala halaman khusus *admin* yang terbuka setelah *admin* melakukan *Login*. kemudian *admin* memilih menu gejala.. *Admin* melihat 3 pilihan yaitu tambah, *edit* , dan *delete*. Jika *admin* mengklik tombol tambah maka sistem akan menampilkan halaman tambah gejala, kemudian admin mengisi data gejala. Kemudian memilih tombol simpan maka data akan dimasukkan ke *database* kemudian data ditambahkan di *database*, jika *admin* menekan tombol batal maka sistem akan membersihkan halaman tambah gejala. Jika *admin* memilih tombol *edit*, maka sistem akan mengambil data dari *database*. *Admin* mengedit data kemudian klik tombol simpan. Sistem akan melakukan *updatedatabase*. Jika *admin* menekan tombol *delete* maka data yang ada di *database* akan terhapus maka proses selesai.

#### d. Activity Diagram Mengelola Menu Relasi

*Activity diagram* mengelola menu relasi atau desai kontrol merupakan diagram *UML* yang menggambarkan tindakan *administrator* dalam manajemen data relasi sesuai dengan data aturan penelitian ini. Berikut ini gambar *activity diagram* mengelola data relasi (**Gambar 3.7**):



**Gambar 3.7** Activity Diagram Mengelola Menu Relasi

**Sumber:** Data Penelitian (2019)

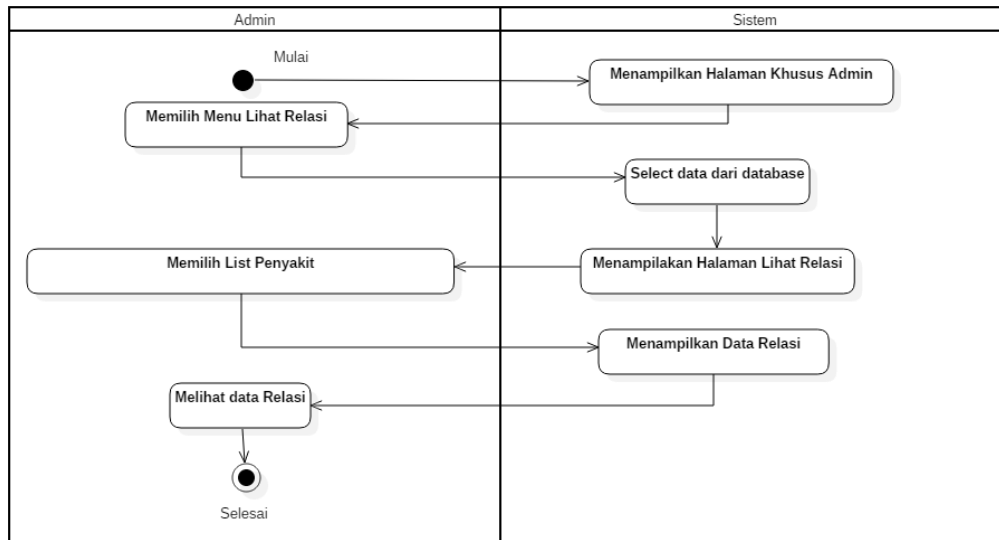
Pada **Gambar 3.7** diatas, admin dapat memilih jenis penyakit dan gejala yang akan direlasikan. *admin* mulai mengakses sistem dan sistem menampilkan halaman khusus admin. *admin* memilih menu relasi kemudian sistem menampilkan halama menu relasi. *Admin* dapat melakukan 2 pilihan yaitu simpan dan normalkan. Pertama, jika *admin* menekan tombol simpan maka sistem akan mengupdate data relasi ke *database*. Kedua, jika Admin menekan tombol normalkan , maka sistem akan menghapus data relasi antar data penyakit dan gejala dan data relasi akan dihapus dari *database* maka proses selesai.

e. ActivityDiagramMengelola Menu Lihat Relasi

*Activity diagram* dalam data relasi atau aturan dari *rule forward chaining*, mengelola menu lihat relasi merupakan diagram *UML* yang menggambarkan



kegiatan *admin*. Berikut ini gambar *activity diagram* mengelola menu lihat relasi (Gambar 3.8):



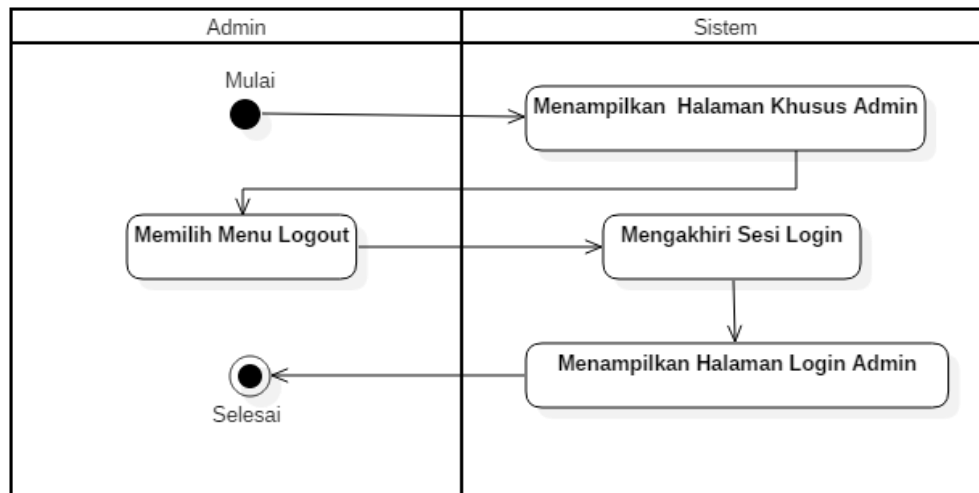
**Gambar 3.8** Activity Diagram Mengelola Menu Lihat Relasi

**Sumber:** Data Penelitian (2019)

Pada **Gambar 3.8** diatas, admin mulai dengan akses ke sistem dan sistem menampilkan halaman dengan hanya administrator. Administrator memilih menu *Relation View*. Sistem mengambil data dari database dan menampilkan halaman *Relationship View*. Administrator dapat melihat daftar penyakit yang dilihat hubungan, setelah itu sistem menampilkan halaman data dari hubungan yang telah dibuat. Administrator akan memeriksa apakah data hubungan sesuai dengan aturan penyelidikan ini, maka prosesnya selesai.

*f. Activity diagram menu logout*

*Activity diagram* atau diagram *UML* yang menggambarkan kegiatan admin dalam menggunakan menu *logout*. Berikut ini gambar *activity diagram* menu *logout* (Gambar 3.9):



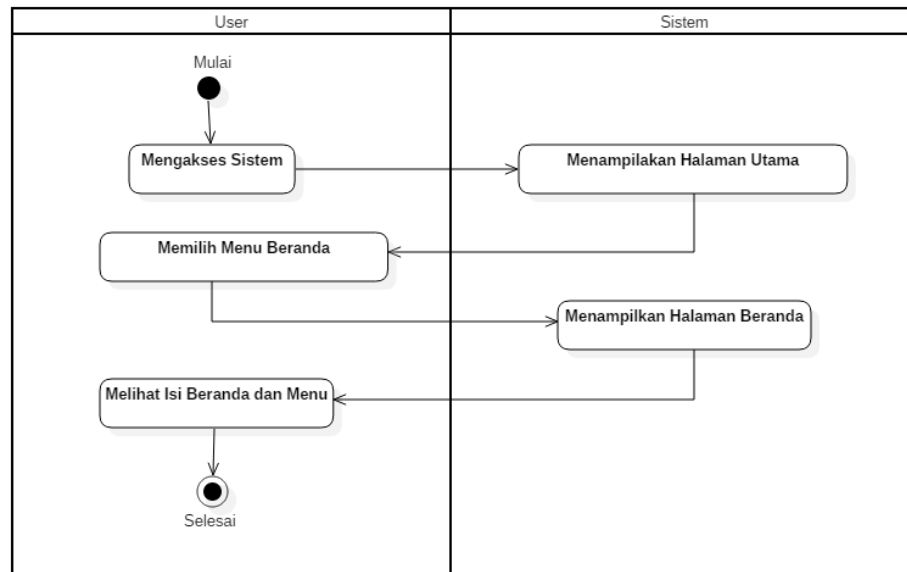
**Gambar 3.9** Activity Diagram Logout

**Sumber:** Data Penelitian (2019)

Pada **Gambar 3.9** diatas, *admin* mulai dengan mengakses sistem dan sistem menampilkan halaman khusus admin. *Admin* memilih menu *logout* kemudian sistem akan mengakhiri sesi *Login admin* dan sistem akan menampilkan halaman menu *Login*.

g. Activity Diagram Melihat Menu Beranda

Activity diagram berikut ini melihat menu beranda merupakan diagram UML yang menggambarkan kegiatan *user* melihat menu beranda. Dibawah gambar melihat menu beranda (**Gambar 3.10**):

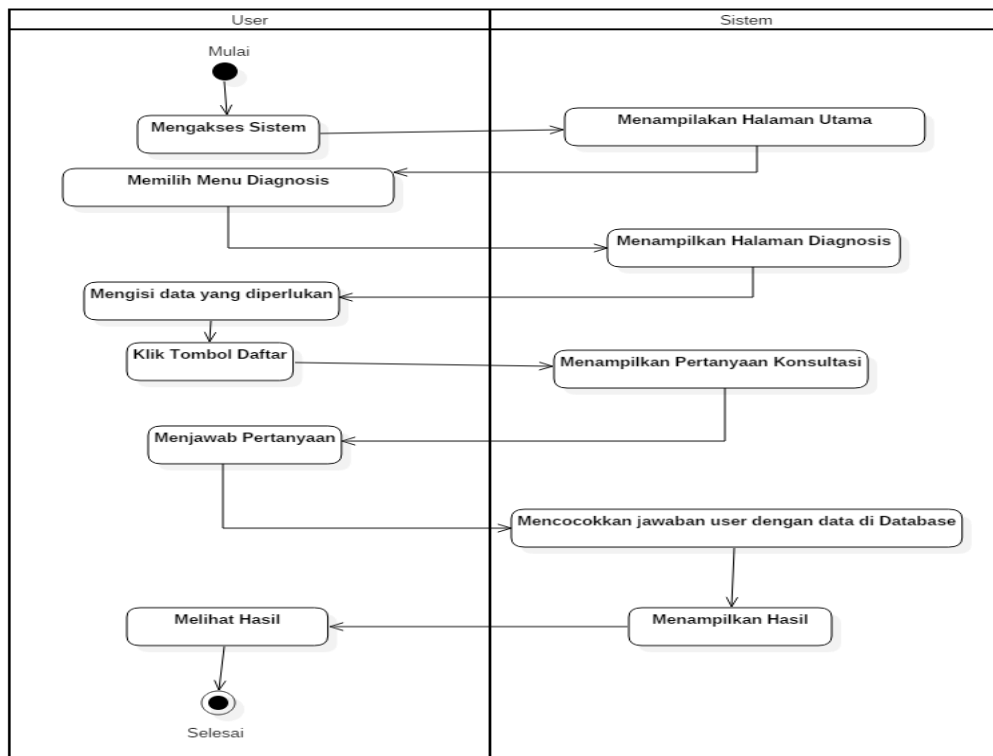


**Gambar 3.10** Activity *Diagram Melihat Menu Beranda*  
**Sumber:** Data Penelitian (2019)

Pada **Gambar 3.10** diatas, Pengguna mulai mengakses sistem dan sistem menampilkan beranda. Pengguna memilih menu mulai dan sistem menampilkan halaman beranda. Pengguna akan dapat melihat informasi tentang penyakit lambung yang tersedia di sistem, dan kemudian mengakhiri proses melihat menu awal samapai selesai.

#### h. Activity diagram Menu Diagnosis

*Activity diagram* diagnosis merupakan diagram UML yang menggambarkan kegiatan pengguna melakukan diagnosis penyakit lambung. Berikut ini gambar diagram aktivitas diagnosis menu diagnosis (**Gambar 3.11**):



**Gambar 3.11** Activity Diagram Menu Diagnosis

Sumber: Data Penelitian (2019)

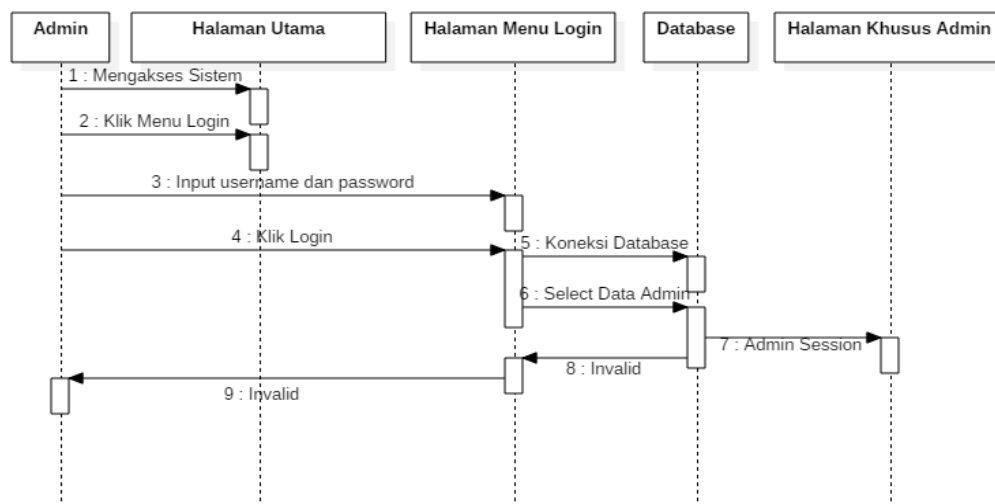
Pada **Gambar 3.11** diatas, *User* akan mengisi data terlebih dahulu dan menekan tombol daftar, *user* mulai dengan mengakses sistem dan sistem menampilkan halaman utama. *User* memilih menu diagnosis kemudian sistem menampilkan halaman menu diagnosis. Sistem akan menampilkan halaman pertanyaan diagnosis. *User* akan menjawab pertanyaan sesuai dengan apa yang terjadi pada dirinya. Kemudian sistem membandingkan respons dengan data dalam database dan sistem menampilkan hasil atau solusinya. *User* dapat melihat solusi dari gejala yang terjadi pada pasien yang terkena penyakit lambung. maka proses melakukan diagnosis selesai.

### 3. Sequence Diagram

*Diagram Sequence* menunjukkan perilaku objek jika digunakan dengan menjelaskan waktu hidup objek dan pesan yang dikirim dan diterima di antara objek. (A.S & Shalahuddin).

#### a. Sequenced agram Login Admin

Urutan login administrator adalah urutan waktu aktivitas administrator saat login. Berikut ini gambar *sequence diagram Login admin* (**Gambar 3.12**):



**Gambar 3.12** Sequence Diagram Login Admin

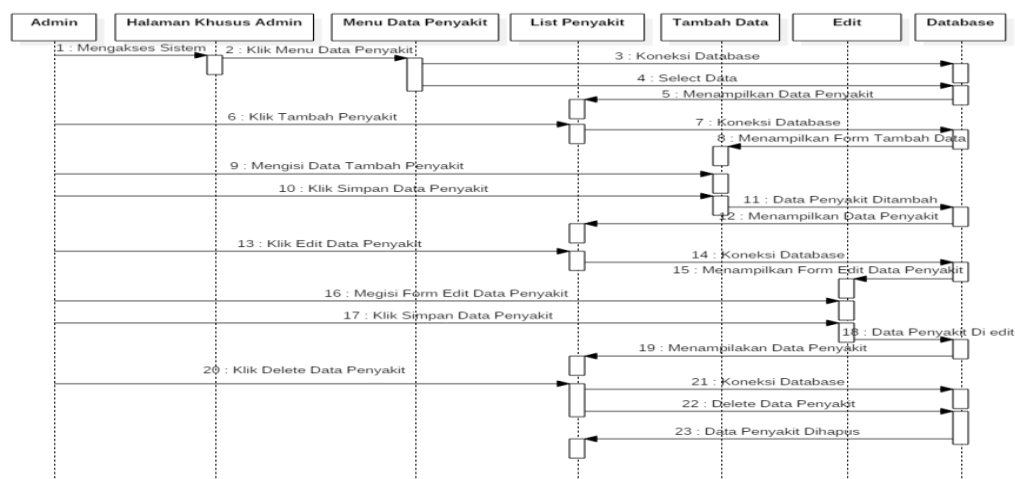
**Sumber:** Data Penelitian (2019)

*Admin* mengakses sistem lalu sistem menampilkan halaman utama. Dari Halaman *Login Admin*, sistem akan mengecek *username* dan *password* yang sudah dimasukkan kemudian sistem akan terhubung dengan *database*, setelah itu *username* dan *password* akan dicocokkan oleh sistem. *Admin* mengklik menu admin, maka sistem akan menampilkan halaman *Login admin*. *Admin* akan

memasukkan *username* dan *password* ke menu *Login* kemudian *admin* melakukan klik tombol *Login*. Jika *username* dan *password* valid maka halaman khusus admin akan ditampilkan. Jika *username* dan *password* invalid (tidak sesuai dengan *database*) maka sistem akan menampilkan pesan gagal di halaman menu *Login*.

#### b. *Sequence Diagram* Mengelola Menu Penyakit

*Sequence diagram* Diagram urutan untuk mengelola menu penyakit adalah urutan tindakan administratif untuk mengelola data dan solusi penyakit. Berikut ini gambar *sequence diagram* mengelola menu penyakit (**Gambar 3.13**)



**Gambar 3.13** *Sequence Diagram Mengelola Menu Penyakit*

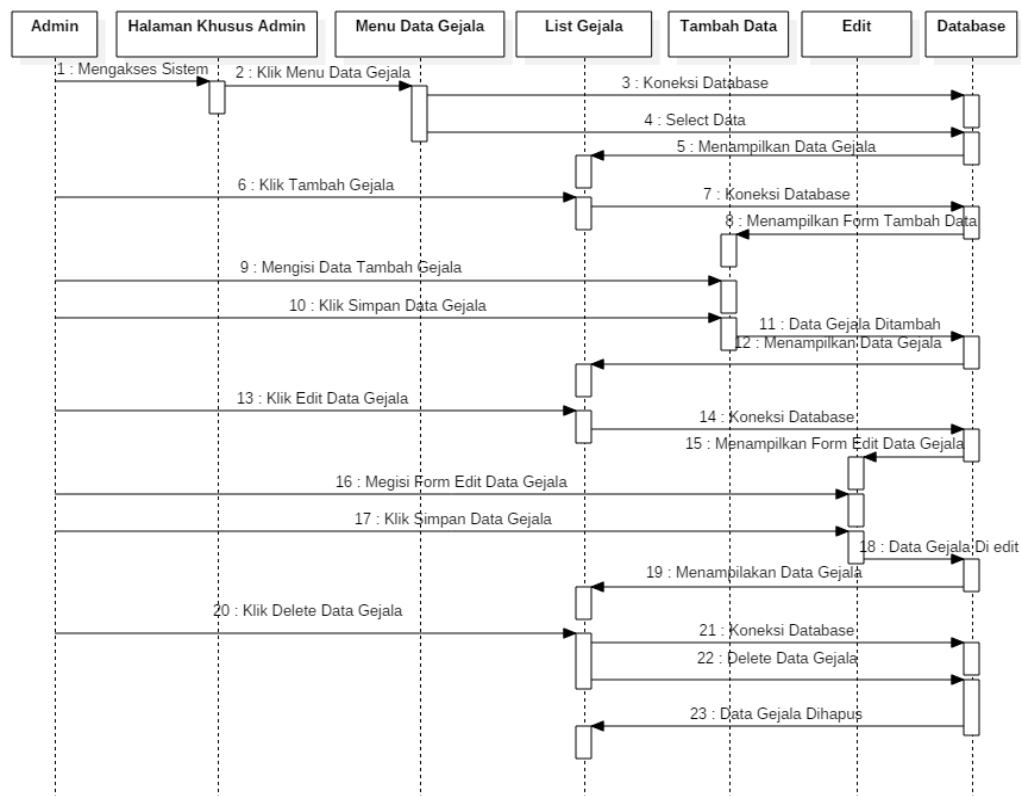
**Sumber:** Data Penelitian (2019)

*Admin* mengakses sistem, kemudian menampilkan halaman khusus admin. Lalu *admin* mengklik menu penyakit, *admin* menekan tombol tambah penyakit. Sistem akan menampilkan form tambah data. Kemudian admin akan mengisi form tambah penyakit lalu admin mengklik tombol simpan. Data ditambahkan ke *database* dan sistem akan menampilkan kembali menu penyakit. Kemudian admin

klik *edit* data penyakit dan sistem akan menampilkan form edit data, kemudian admin akan mengisi form edit data penyakit lalu mengklik tombol simpan. Data user diupdate pada database, kemudian sistem akan menampilkan form menu data penyakit kembali. Admin mengklik tombol *delete*, sistem pun akan mendelete data penyakit pada *database*.

### c. Sequence Diagram Mengelola Gejala

*Sequence diagram* urutan waktu kegiatan mengelola gejala merupakan *admin* saat melakukan pengelolaan gejala penyakit. Berikut ini gambar *sequence diagram* mengelola data gejala (**Gambar 3.14**):



**Gambar 3.14** Sequence Diagram Mengelola Gejala

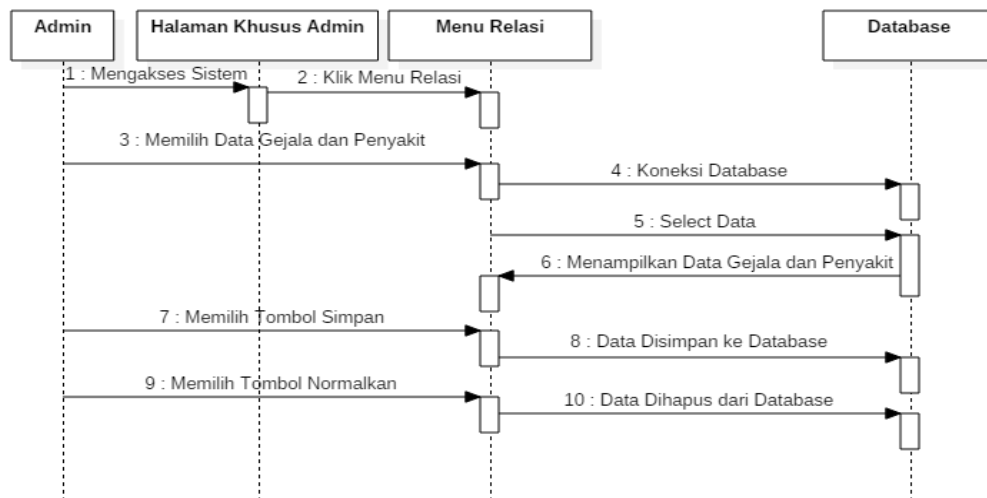
**Sumber:** Data Penelitian (2019)

Administrator mengakses sistem dan kemudian menampilkan halaman hanya untuk *administrator*. Kemudian *administrator* mengklik menu gejala, administrator mengklik tombol untuk menambahkan gejala. Sistem akan menampilkan informasi formulir yang ditambahkan. Kemudian *administrator* akan mengisi formulir dan menambahkan gejala, dan kemudian *administrator* mengklik tombol simpan. Data ditambahkan ke database dan sistem akan menampilkan kembali menu gejala. Kemudian administrator akan mengklik untuk mengedit data gejala dan sistem akan menampilkan formulir edit data, kemudian administrator akan mengisi formulir untuk mengedit data gejala dan kemudian klik tombol simpan. Data pengguna diperbarui dalam database, dan kemudian sistem akan menampilkan kembali formulir di menu informasi gejala. Administrator mengklik tombol hapus, sistem akan menghapus gejala dari data database.

d. *Sequence Diagram* Mengelola Menu Relasi

*Sequence diagram* yang mengatur hubungan menu relasi merupakan urutan waktu kegiatan *admin* saat mengelola data relasi sesuai aturan. Berikut ini gambar *sequence diagram* mengelola menu relasi (**Gambar 3.15**):





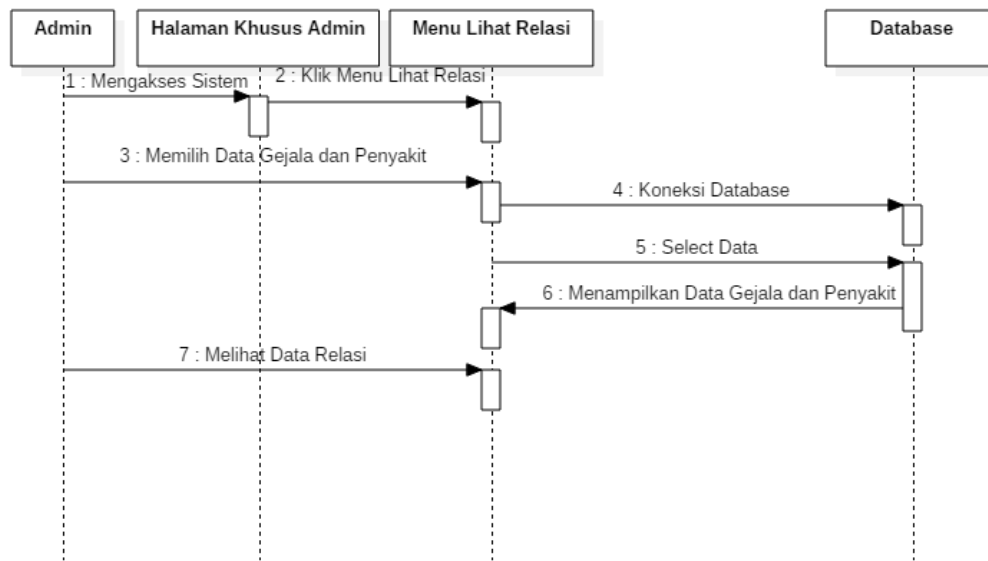
**Gambar 3.15** Sequence Diagram Mengelola Menu Relasi

**Sumber:** Data Penelitian (2019)

Sistem akan menampilkan halaman menu relasi dan admin akan memilih data penyakit yang akan direlasikan ke data gejala kemudian sistem terkoneksi dengan *database* dan mengambil data, kemudian menampilkan data gejala dan penyakit. *Admin* mengakses sistem dan menampilkan halaman khusus admin. kemudian admin mengklik menu relasi,. Jika *admin* menekan tombol simpan maka sistem akan menambahkan data ke database. Bila *admin* memilih tombol normalkan maka data gejala dan penyakit yang direlasikan akan dihapus dari *database*.

e. *Sequence Diagram* Mengelola Menu Lihat Relasi

*Sequence diagram* manajemen menu lihat relasi merupakan urutan waktu kegiatan administrator saat melihat relasi yang telah sesuai dengan aturan rule pada sistem. Berikut ini gambar *sequence diagram* mengelola menu lihat relasi (**Gambar 3.16**):



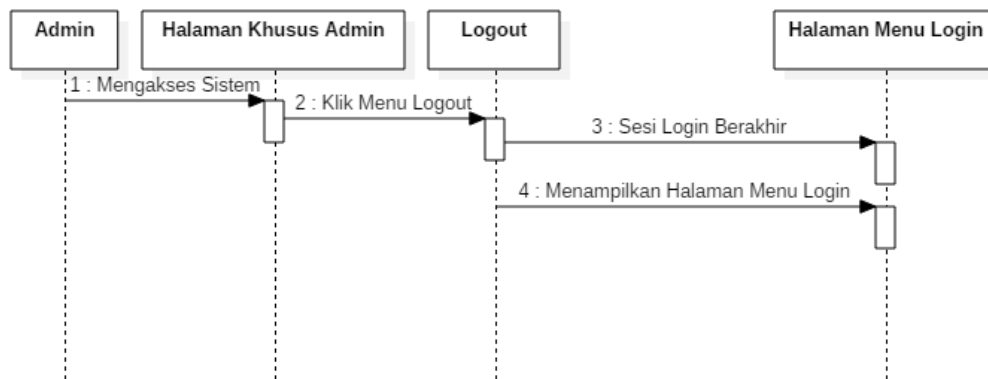
**Gambar 3.16** Sequence diagram Mengelola Menu Lihat Relasi

**Sumber:** Data Penelitian (2019)

*Admin* mengakses sistem dan hanya menampilkan halaman khusus admin. kemudian admin mengklik menu lihat relasi, kemudian sistem menampilkan data relasi yang telah dibuat pada menu relasi. *Admin* memilih data gejala dan penyakit yang akan dilihat relasinya. Sistem akan terkoneksi ke *database* dan sistem akan menampilkan data gejala dan penyakit yang telah direlasikan.

f. *Sequence diagram logout admin*

*Sequence diagram logout admin* adalah urutan tindakan administrasi ketika mereka keluar dari sistem. Berikut ini gambar *sequence diagram logout admin* (**Gambar 3.17**):



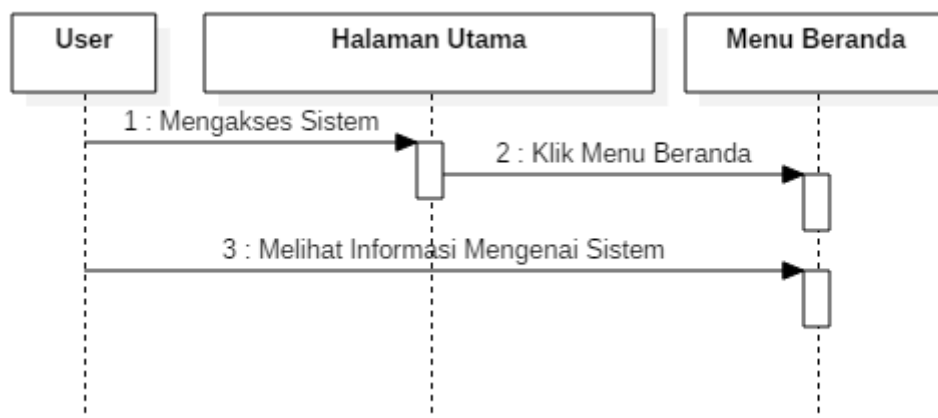
**Gambar 3.17** Sequence *diagram* logout *admin*

**Sumber:** Data Penelitian (2019)

*Admin* pada sistem pun berakhir kemudian akan menampilkan halaman menu *Login*. *Admin* mengakses halaman khusus admin lalu mengklik menu *logout* dan sesi *Login*

g. *Sequence diagram* Melihat Menu Beranda

*Sequence diagram* (urutan) melihat menu beranda sebagai urutan dari aktivitas pengguna ketika mengakses sistem dan informasi tentang sistem pakar. Berikut ini gambar *sequence diagram* melihat menu beranda (**Gambar 3.18**):

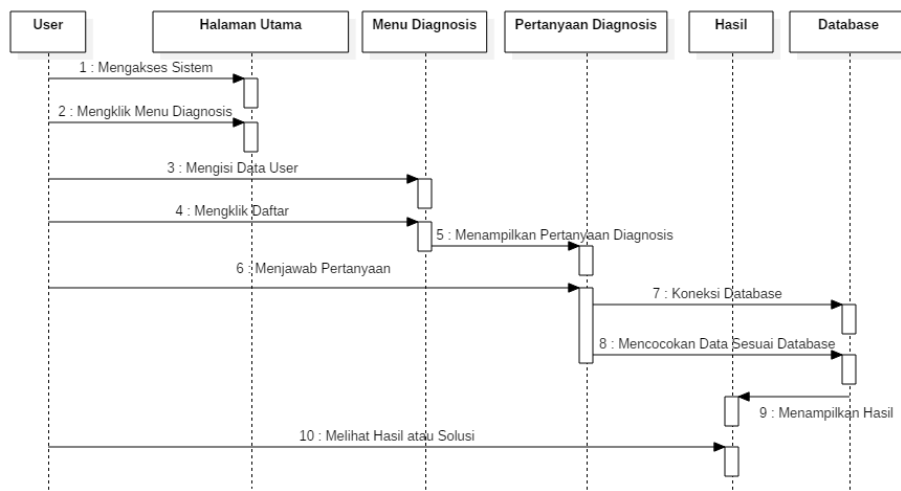


**Gambar 3.18** Sequence *diagram* Melihat Menu Beranda

**Sumber:** Data Penelitian (2019)

#### h. *Sequence Diagram* Menu Diagnosis

*Sequence diagram* urutan waktu kegiatan pengguna (*user*) menu diagnosis merupakan saat melakukan diagnosis penyakit lambung. Berikut ini gambar *sequence diagram* menu diagnosis (**Gambar 3.19**):



**Gambar 3.19** *Sequence Diagram* Menu Diagnosis

**Sumber:** Data Penelitian (2019)

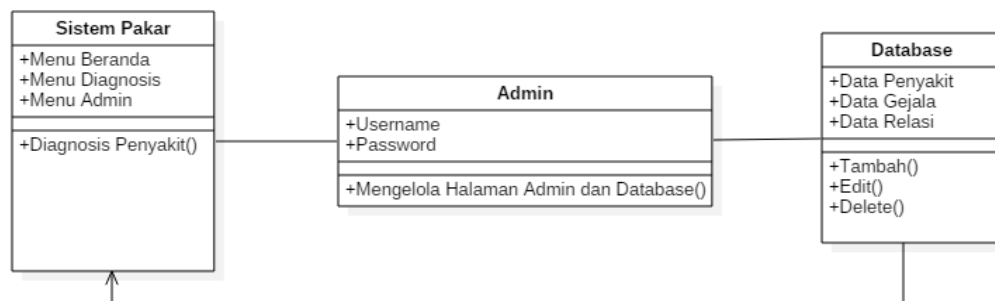
Pengguna mengakses sistem dan mengklik menu pencarian di halaman sistem utama. Sistem akan menampilkan tab menu pencarian. Pengguna akan melengkapi informasi yang mereka butuhkan untuk buku tamu yang disimpan di panel database dengan mengklik tombol registrasi. Sistem akan menunjukkan tanda-tanda tentang penyakit lambung. Sistem akan mengintegrasikan infrastruktur data agar sesuai dengan jawaban untuk setiap pertanyaan yang dijawab pengguna, kemudian sistem akan menampilkan hasil pencarian dengan cara yang dapat dilihat pengguna.

#### 4. Class Diagram

Hanya dibuat 2 macam class diagram dalam penelitian ini yaitu class diagram user dan class diagram admin. Berikut ini adalah gambar-gambar class diagram yang digunakan dalam sistem pakar pada penelitian ini:

##### a. Class Diagram Admin

*Class diagram admin* merupakan serangkaian tindakan admin ketika mengakses sistem pakar diagnosis penyakit lambung. Berikut adalah gambar diagram kelas admin.



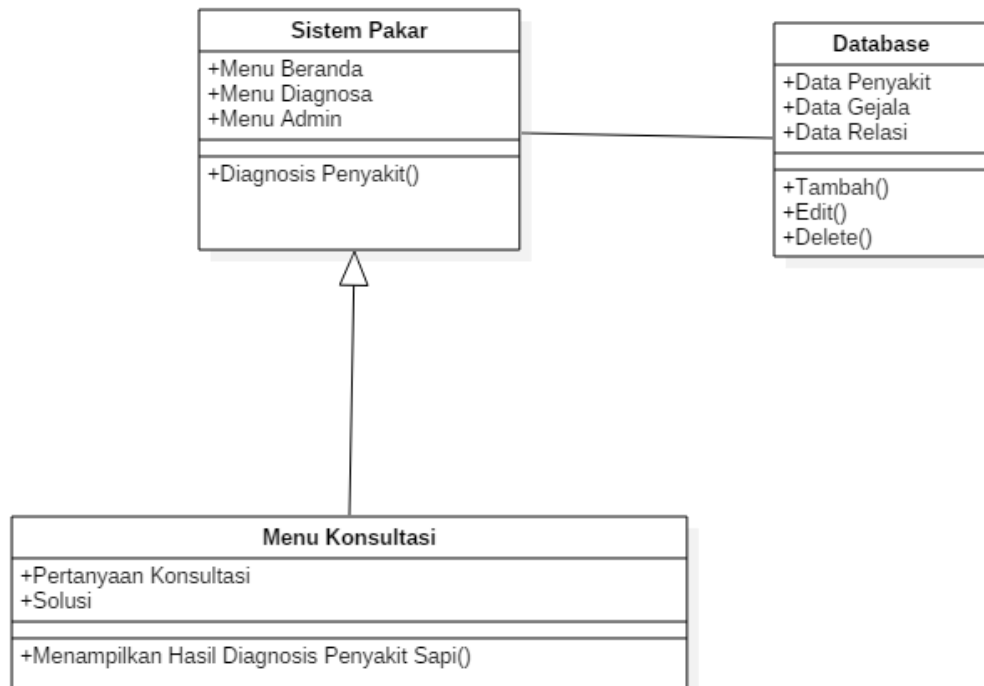
**Gambar 3.20** Class Diagram Admin

**Sumber:** Data Penelitian (2019)

**Gambar 3.20** Struktur *class diagram admin* diatas, ini menunjukkan bahwa pada sistem pakar memiliki atribut untuk menu *admin*, menu beranda, dan menu diagnosis, yang memiliki fungsi yaitu mendiagnosis penyakit lambung. Sistem pakar terhubung dengan *database* yang memiliki atribut data penyakit, data gejala, data relasi serta memiliki fungsi tambah, *edit* dan *delete*, dan melakukan *Login* terlebih dahulu pada sistem.

##### b. Class Diagram User

Desain ruang kelas adalah seperangkat pengalaman pengguna ketika datang ke sistem profesional dalam diagnosis penyakit lambung. Berikut adalah gambar desain lapisan pengguna.



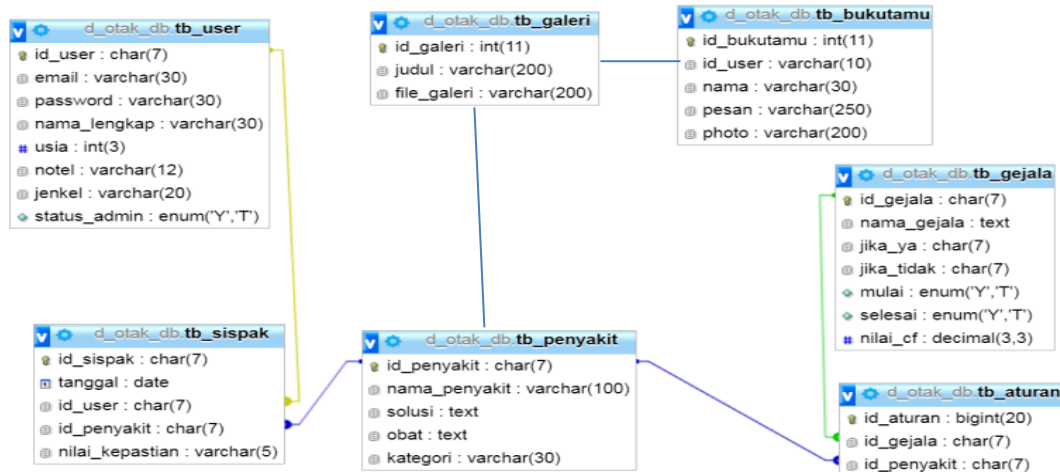
**Gambar 3.21** Class Diagram User

**Sumber:** Data Penelitian (2019)

Struktur *class diagram* (antarmuka) pengguna pada Gambar 3.21 di atas menunjukkan bahwa sistem pakar memiliki fitur dalam bentuk daftar beranda, menu diagnosis, dan menu admin yang memiliki fitur dalam fungsi penyakit lambung. Sistem pakar mengintegrasikan database dengan data penyakit, gejala, hubungan data dan mengintegrasikan, memodifikasi (*edit*), dan menghapus (*delete*) operasi.

Menu diagnosis yang dapat diakses memberikan pertanyaan dan solusi kepada pengguna untuk mendiagnosis penyakit lambung.

### 3.4.5. Desain Database



**Gambar 3.22** Gambar Desain Database

Sumber: Data Penelitian (2019)

*PDM (Physical Data Model)* adalah model yang menggunakan beberapa tabel untuk menghubungkan data serta hubungan antar data-data tersebut. PDM merupakan konsep yang menerangkan detail bagaimana data disimpan di dalam *database*. (A.S. & Shalahuddin). Berikut ini adalah konsep database yang dibuat dalam penelitian.

### 3.4.2. Desain Antarmuka

Berikut ini adalah desain antarmuka yang akan dibuat dalam sistem pakar untuk mendeteksi penyakit lambung:

#### 1. Halaman Beranda

Halaman beranda menunjukkan informasi tentang aplikasi sistem pakar

mendiagnosis penyakit lambung.

Header			Logo
Beranda	Diagnosis	Admin	Search <input type="text"/> <input type="button" value="Search"/>
Content			
Footer			

**Gambar 3.23** Halaman Beranda

**Sumber:** Data Penelitian (2019)

## 2. Halaman Diagnosis Daftar

Halaman Diagnosis diaktifkan bagi pengguna untuk memasukkan data pribadi yang awalnya menggunakan sistem dan akan diarahkan ke pertanyaan tentang diagnosis penyakit lambung.

Header			Logo
Beranda	Diagnosis	Admin	Search <input type="text"/> <input type="button" value="Search"/>
MASUKAN DATA Nama <input type="text"/> Kelamin <input type="radio"/> Pria <input type="radio"/> Wanita Alamat <input type="text"/> Jenis Sapi <input type="text"/> <input type="button" value="Daftar"/>			
Footer			

**Gambar 3.24** Halaman Diagnosis

**Sumber:** Data Penelitian (2019)



### 3. Halaman Diagnosis Pertanyaan

Halaman Pertanyaan Diagnostik memungkinkan pengguna untuk mendiagnosis kondisi lambung. Pengguna akan menjawab pertanyaan yang ditampilkan oleh sistem.

Header			Logo
Beranda	Diagnosis	Admin	
<p>JAWABLAH PERTANYAAN BERIKUT : Pertanyaan ?</p> <p><input type="radio"/> Benar (YA)   <input type="radio"/> Salah (TIDAK)</p> <p><input type="button" value="Jawab"/></p>			<p>Search</p> <p><input type="text"/></p> <p><input type="button" value="Search"/></p>
Footer			

**Gambar 3.25** *Halaman Diagnosis Pertanyaan*

**Sumber:** Data Penelitian (2019)

### 4. Halaman Diagnosis Solusi

Solusi tersebut berasal dari database sistem. Halaman Diagnosis Solusi ini menampilkan solusi dari setiap pertanyaan yang telah user jawab.

**Gambar 3.26** *Halaman Diagnosis Solusi*  
**Sumber:** Data Penelitian (2019)

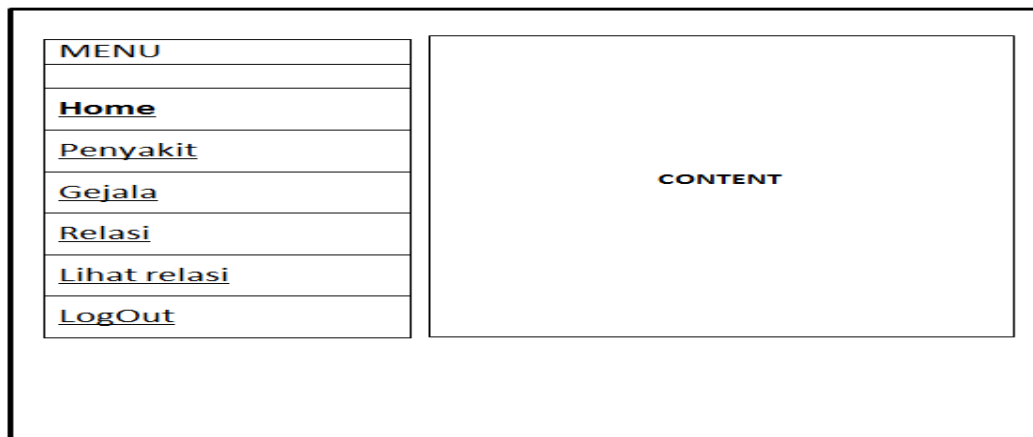
## 5. Halaman Admin

Admin akan melakukan pengeditan data sesuai data yang didapat dari Pakar. Halaman Admin berfungsi untuk menambahkan data gejala penyakit dan solusi penyaki lambung.

**Gambar 3.27** *Halaman Admin*  
**Sumber:** Data Penelitian (2019)

## 6. Halaman Home Admin

Halaman beranda administrator adalah halaman yang menampilkan informasi tentang halaman administrator.

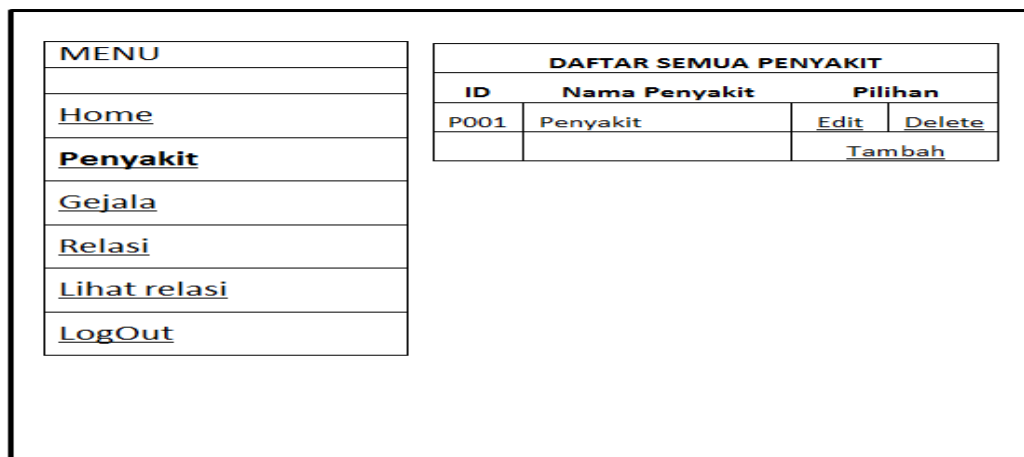


**Gambar 3.28** Halaman Home admin

**Sumber:** Data Penelitian (2019)

## 7. Halaman Menu Penyakit

Halaman menu berfungsi untuk menambah, mengedit, dan menghapus data penyakit lambung dalam database.



**Gambar 3.29** Halaman Menu Penyakit

**Sumber:** Data Penelitian (2019)

### 8. Halaman Tambah Penyakit

Fungsi halaman menu menambah, mengedit, dan menghapus data penyakit lambung di atas dalam data admin.

MENU	ENTRY DATA
<a href="#">Home</a>	ID : <input type="text" value="P001"/>
<b><a href="#">Penyakit</a></b>	Penyakit : <input type="text"/>
<a href="#">Gejala</a>	Penyebab : <input type="text"/>
<a href="#">Relasi</a>	Keterangan : <input type="text"/>
<a href="#">Lihat relasi</a>	Solusi : <input type="text"/>
<a href="#">LogOut</a>	<input type="button" value="Simpan"/> <input type="button" value="Batal"/>

**Gambar 3.30** *Halaman Tambah Penyakit*  
**Sumber:** Data Penelitian (2019)

### 9. Halaman Menu Gejala

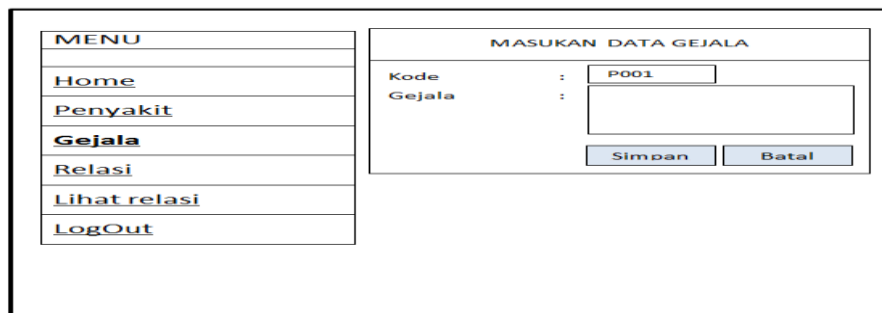
Halaman Menu Gejala ini berfungsi untuk administrator untuk menambah dan menghapus data untuk gejala penyakit lambung.

MENU	DAFTAR SEMUA GEJALA												
<a href="#">Home</a>	<table border="1"> <thead> <tr> <th>ID</th> <th>Nama Gejala</th> <th colspan="2">Pilihan</th> </tr> </thead> <tbody> <tr> <td>G001</td> <td>Gejala</td> <td><a href="#">Edit</a></td> <td><a href="#">Delete</a></td> </tr> <tr> <td colspan="4" style="text-align: center;"><a href="#">Tambah</a></td> </tr> </tbody> </table>	ID	Nama Gejala	Pilihan		G001	Gejala	<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Tambah</a>			
ID	Nama Gejala	Pilihan											
G001	Gejala	<a href="#">Edit</a>	<a href="#">Delete</a>										
<a href="#">Tambah</a>													
<a href="#">Penyakit</a>													
<b><a href="#">Gejala</a></b>													
<a href="#">Relasi</a>													
<a href="#">Lihat relasi</a>													
<a href="#">LogOut</a>													

**Gambar 3.31** *Halaman Menu Gejala*  
**Sumber:** Data Penelitian (2019)

#### 10. Halaman Tambah Gejala

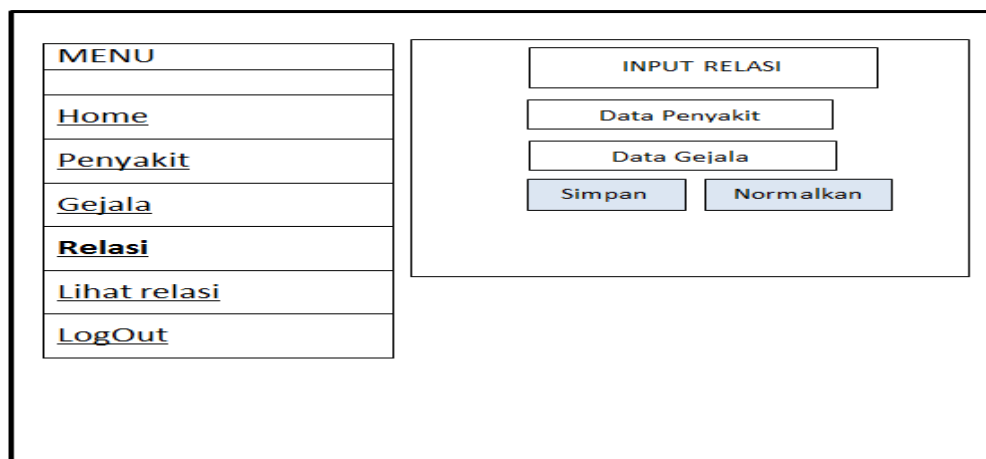
Halaman gejala tambahan ini berfungsi bagi administrator untuk menambahkan pengeditan dan menghapus data gejala-gejala lambung.



**Gambar 3.32** *Halaman Tambah Gejala*  
**Sumber:** Data Penelitian (2019)

#### 11. Halaman Menu Relasi

Halaman hubungan (relasi) adalah administrator halaman untuk menautkan data penyakit ke data gejala sesuai dengan tabel keputusan.



**Gambar 3.33** *Halaman Menu Relasi*  
**Sumber:** Data Penelitian (2019)

#### 12. Halaman Menu Lihat Relasi

Halaman Tampilan Daftar relasi (hubungan) adalah halaman yang menunjukkan hubungan antara data penyakit dan data gejala penyakit.

<b>MENU</b>	
<a href="#">Home</a>	
<a href="#">Penyakit</a>	
<a href="#">Gejala</a>	
<a href="#">Relasi</a>	
<a href="#">Lihat relasi</a>	
<a href="#">LogOut</a>	

<b>TAMPILKAN GEJALA PER PENYAKIT</b>	
<b>Penyakit</b>	List Daftar Penyakit
	<input type="button" value="Tampil"/>

NAMA PENYAKIT :

<b>DAFTAR GEJALA</b>		
<b>Penyakit</b>	<b>Gejala</b>	<b>Nama Gejala</b>
P001	G001	

**Gambar 3.34** Rancangan Halaman Menu Konsultasi

**Sumber:** Data Penelitian (2019)

### 3.5. Lokasi dan Jadwal Penelitian

#### 3.5.1. Lokasi Penelitian

Klinik Nira Medika Batam, Batu Aji, Kota Batam-Kepulauan Riau Penelitian ini dilakukan. Alasan mengapa peneliti memilih ini sebagai tempat penelitian adalah:

1. Ketersediaan data untuk melakukan penelitian.
2. Mudah mendapatkan data yang dibutuhkan.
3. Lokasi mudah untuk dijangkau.
4. Efisiensi biaya dan waktu

#### 3.5.2. Jadwal Penelitian

Sebuah rencana penelitian harus dibuat menggambarkan kapan dan berapa lama setiap fase penelitian akan dilakukan. Selain itu, jadwal untuk penelitian ini juga merupakan tujuan (batas waktu) di mana peneliti yang bersangkutan dapat

melakukan dan menyelesaikan penelitian. Berikut ini adalah jadwal untuk kegiatan yang dilakukan selama penelitian.

**Tabel 3.7** Tabel Jadwal Penelitian

No	Kegiatan	Jadwal																					
		September 2019				Oktober 2019				November 2019				Desember 2019				Januari 2019				Februari 2019	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
1	Pemilihan Judul																						
2	Pengajuan Judul																						
3	Pengumpulan Data																						
4	Penyusunan Bab I																						
5	Penyusunan Bab II																						
6	Penyusunan Bab III																						
7	Penyusunan Bab IV																						
8	Penyusunan Bab V, Daftar Pustaka, Lampiran																						

**Sumber:** (Data penelitian, 2019)