

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Dasar**

Agar penulisan penelitian ini dapat berjalan dengan baik, maka penulis membutuhkan berupa teori-teori yang telah ada sebagai landasan bagi penelitian. Dalam penelitian ini akan menjelaskan mengenai kecerdasan buatan serta salah satu jenis Kecerdasan Buatan yaitu, Sistem Pakar.

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

Kecerdasan buatan yang berasal dari bahasa inggris "*Artificial Intelligence*" atau disingkat AI, yaitu *intellegence* merupakan kata sifat yang artinya cerdas, sedangkan *artificial* yang berarti buatan. Kecerdasan buatan bertujuan sebagai pengacu seperti mesin yang berpikir, yang mampu bertindak seperti manusia. Alan Turing, ahli matematika berkebangsaan Inggris menetapkan definisi *Artificial Intelligence* "apabila komputer tidak bisa dibedakan dengan manusia, maka dapat disebutkan komputer tersebut memiliki kepintaran" (Sutojo, 2011, p. 1, 2)

Kecerdasan alami apabila dibandingkan dengan kecerdasan buatan memiliki keuntungan komersial sebagai berikut, dalam Sutojo, dkk. (2011: 10-11):

1. Kecerdasan buatan bersifat lebih tetap.
2. Kecerdasan buatan lebih mudah disebar dan ditiru.
3. Kecerdasan buatan lebih murah dibandingkan kecerdasan alami.
4. Kecerdasan buatan bersifat lebih konsisten.

5. Kecerdasan buatan bisa didokumentasi.
6. Kecerdasan buatan dapat melakukan pekerjaan lebih cepat dan baik dibandingkan dengan kecerdasan alami.

Hal utama penerapan kecerdasan buatan ialah ilmu (*knowledge*), yang memiliki pemahaman dari beberapa bidang spesialis yang diperoleh melalui pelatihan dan pengalaman. Bagian utama yang dibutuhkan tersebut yaitu antara lain:

- Basis pengetahuan (*knowledge base*)

Basis Pengetahuan adalah Informasi yang mengatur menganalisa dengan tujuan mempermudah dalam penerapannya untuk pemecahan suatu masalah dan pengambilan keputusan. Informasi terdiri dari fakta, pemikiran, teori dan prosedur yang berhubungan satu sama lain (Victor Amrizal, 2013: p. 12).

- Motor inferensi (*inference engine*)

Motor inferensi merupakan pengguna menyediakan pengolahan fakta dalam mencari hubungan antara fakta serta aturan dalam penyimpanan basis pengetahuan.

Adapun kelebihan dan kelemahan Kecerdasan Buatan (Victor Amrizal, 2013: p.30):

**Tabel 2. 1** Tabel Kelebihan dan Kekurangan Kecerdasan Buatan

Kelebihan Kecerdasan Buatan	Kelemahan Kecerdasan Buatan
<p>1. Komputer masa depan akan memberikan kemudahan, kenyamanan, dan kenikmatan yang lebih bagi penggunaannya.</p> <p>2. Dapat berkomunikasi dengan komputer dengan bahasa alami/bahasa manusia sehari-hari.</p> <p>3. Akan terbebas dari keharusan belajar bahasa pemrograman dan sistem operasi.</p> <p>4. Para pengguna komputer yang tidak terbiasa sama sekalipun mampu menciptakan karya yang bermanfaat dengan menggunakan komputer.</p> <p>5. Menggunakan <i>computer</i> jauh akan lebih mudah dibanding menggunakan pesawat telepon.</p>	<p>1. Mahalnya dalam hal pengembangan dan riset.</p> <p>2. Waktu yang diperlukan dalam pengembangan aplikasi kecerdasan buatan adalah hal yang sulit.</p> <p>3. Perangkat lunak yang khusus untuk kecerdasan buatan masih sedikit. Padahal dengan perangkat lunak khusus ini, pekerjaan pembuatan serta pengembangan perangkat lunak Kecerdasan Buatan menjadi lebih mudah dan cepat.</p> <p>4. Belum adanya antarmuka (<i>interface</i>) bahasa alami khusus untuk Kecerdasan Buatan.</p>

Sumber: Victor Amrizal (2013: p.30)

## 2.1.2 Sistem Pakar

### 1. Definisi Sistem Pakar

Sistem pakar merupakan sebuah program komputer yang dibuat dalam mengatasi suatu permasalahan yang bekerja layaknya seorang ahli (*human expert*). Pembuatan sistem pakar ini dibuat sedemikian rupa sehingga dapat memecahkan suatu masalah dengan mencontoh cara kerja dari seorang ahli di bidangnya yang dapat dipahami dalam bentuk pengetahuan yang dibagikan dari seorang ahli (sumber pengetahuan lainnya) kedalam komputer, informasi yang ada selanjutnya disimpan kedalam memori komputer sehingga pengguna dapat menggunakannya untuk kepentingan tertentu. (Victor Amrizal, 2013: p.35).

Sistem pakar memiliki 4 elemen manusia, antara lain:

#### 1. Pakar (*The Expert*)

Pakar adalah seorang yang memiliki kemampuan mengenai bidang tertentu serta mempunyai pengetahuan mengenai metode tertentu dan menggunakan keterampilan tersebut sebagai saran atau nasehat guna pemecahan dari suatu masalah, pakar juga mampu memahami mana yang penting atau tidak mengenai suatu fakta.

#### 2. Perakayasa Pengetahuan (*Knowledge Engineer*)

Perakayasa pengetahuan merupakan seorang yang membantu ahli menciptakan area masalah serta mempraktikkan dan menghubungkan jawaban ahli dalam sebuah pertanyaan, merancang analogi, dan mengklarifikasi kesulitan konseptual.

### 3. pengguna (*user*)

Beberapa kelas pemakai dalam sistem pakar, antar lain:

- a. Pengguna bukan ahli. Dalam hal tersebut, sistem pakar bertindak sebagai konselor atau penasehat.
- b. Siswa atau pelajar, disini sistem pakar mempunyai peran sebagai tutor.
- c. Mengembangkan sistem pakar yang ingin menambah dan melengkapi basis pengetahuan, dengan kata lain sistem pakar bekerja sebagai rekan kerja.
- d. Pakar, dalam hal ini sistem pakar bekerja sebagai rekan atau asisten dan elemen lainnya.

### 4. Pembangun sistem (*system builder*)

Mendukung dalam menghubungkan suatu sistem pakar dengan sistem komputer lainnya.

## 2. Kategori Permasalahan Sistem Pakar

Sistem pakar saat ini dibuat untuk memecahkan masalah yang berbeda di setiap bidang tertentu. Berikut merupakan permasalahan umum pada aplikasi sistem pakar (Sutojo, 2011, p. 162):

1. Interpretasi, berlandaskan pada data *input* dapat dibuat gambaran kondisi.
2. Prediksi, berdasarkan situasi yang ada, konsekuensi yang mungkin terjadi dapat dinilai.
3. Diagnosis, berdasar pada gejala yang diberikan, kondisi dapat disimpulkan.
4. Desain, membuat desain berdasar pada batasan yang disertakan.
5. Planning, suatu perencanaan tindakan-tindakan yang akan dikerjakan.

6. *Monitoring*, proses membandingkan hasil observasi dengan proses perencanaan.
7. *Debugging*, penentuan penyelesaian dari sebuah kesalahan sistem.
8. Reparasi, melakukan rencana pemulihan.
9. *Instruction*, Melaksanakan instruksi, diagnostik, pemecahan masalah dan petunjuk pembaharuan.
10. Kontrol, melakukan kontrol, mengelola hasil interpretasi, diagnosis, pemecahan masalah, pemantauan dan perbaikan perilaku sistem.

Adapun beberapa Kelebihan dan kekurangan yang ada pada sistem pakar, sistem pakar memiliki kelebihan yang dapat dilihat pada tabel berikut:

**Tabel 2. 2** Kelebihan Sistem Pakar

Kelebihan Sistem Pakar
<ol style="list-style-type: none"> <li>1. Memperkenankan orang baru mampu melakukan pekerjaan para ahli.</li> <li>2. Dapat melakukan proses secara berulang secara otomatis.</li> <li>3. Menyimpan pengetahuan dan keahlian para pakar.</li> <li>4. Dapat mencari serta mempertahankan kemampuan spesialis.</li> <li>5. Dapat dioperasikan di lingkup yang berpotensi bahaya.</li> <li>6. Mempunyai keahlian dalam bekerja dengan informasi yang tidak lengkap serta tidak pasti.</li> <li>7. Tidak membutuhkan anggaran jika tidak di pakai sedangkan tenaga ahli membutuhkan anggaran harian.</li> <li>8. Bisa diperbanyak sesuai kebutuhan dalam waktu sesingkat mungkin.</li> <li>9. Mampu menyelesaikan masalah dengan cepat dari kemampuan manusia.</li> <li>10. Menghemat waktu dalam pengumpulan putusan.</li> </ol>

Sumber: Victor Amrizal (2013: p.38)

Adapun beberapa kelemahan dari sistem pakar yaitu sebagai dalam tabel berikut:

**Tabel 2. 3** Kekurangan Sistem Pakar

Kekurangan Sistem Pakar
1. Anggaran pembuatan, pemeliharaan, dan pengembangannya sangat tinggi.
2. Susah untuk ditingkatkan, sangat erat hubungannya pada penyediaan tenaga ahli di bagiannya.
3. Pendekatan setiap pakar terhadap sebuah kondisi atau permasalahan mungkin berbeda, sekalipun keduanya benar.
5. Penyampaian informasi dapat bersifat subyektif.
6. Kurangnya keyakinan pengguna dapat mencegah pemakaian sistem pakar.

Sumber: Victor Amrizal (2013: p.38)

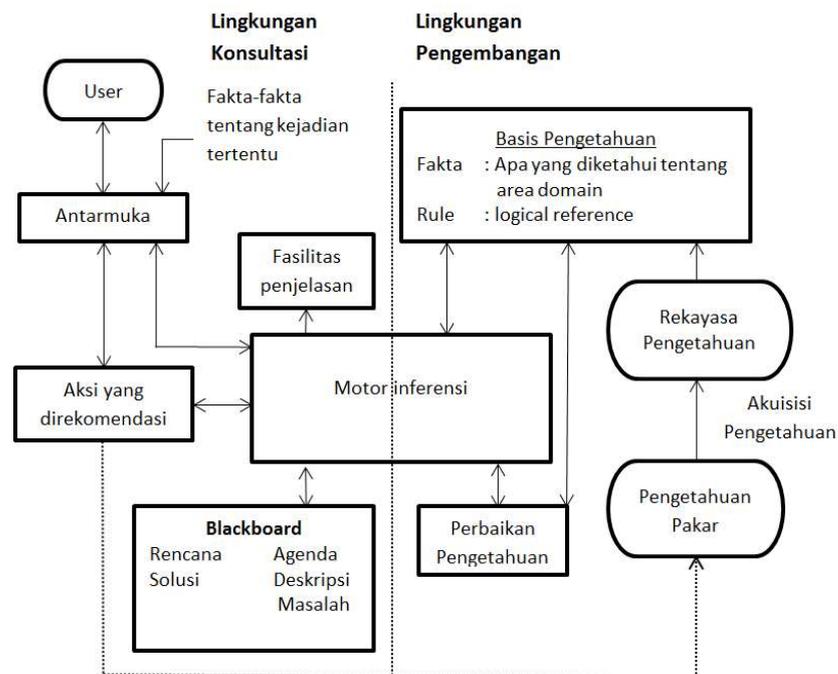
Sistem pakar seperti program yang dapat meniru manusia dalam mengerjakan suatu hal (Hartati & Sari, 2008, pp. 3–4). Elemen-elemen yang akan digunakan untuk membuat sistem yaitu:

1. Basis pengetahuan (*knowledge base*),
2. Antarmuka pengguna (*user interface*),
3. Memori kerja (*working memory*).
4. Mekanisme inferensi (*inference machine*),

### 3. Struktur Sistem Pakar

Sistem pakar mempunyai dua unsur penting, lingkungan pengembangan dan lingkungan konsultasi. Pembangun sistem pakar menggunakan lingkungan pengembangan untuk membuat komponen pengetahuan dan membawa informasi ke dalam basis data. Pengguna menggunakan lingkungan konsultasi untuk

berkonsultasi, sehingga pengguna dapat menerima informasi dan saran tentang sistem pakar (Hartati & Sari, 2008, pp. 3–4). berikut ini menampilkan elemen-elemen penting dalam sistem pakar.



**Gambar 2.1** Elemen-elemen sistem pakar  
Sumber: Sutojo, dkk. (2011: 166)

Penjelasan:

#### 1. Dasar Pengetahuan (*Knowledge Base*)

Dasar pengetahuan berisikan pengetahuan yang dibutuhkan dalam mengetahui dan menyimpulkan permasalahan. Basis pengetahuan terdiri dari dua dasar, yaitu:

- a. Fakta, misalnya situasi, kondisi atau permasalahan yang ada.
- b. *Rule* (Aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

2. Pengumpulan informasi

Tahapan ini bertujuan untuk memasukkan informasi dari seorang ahli dengan menggunakan ilmu teknik maka dapat diproses oleh komputer dan ditempatkan dalam database dalam bentuk tertentu.

3. Mesin Inferensi (*Inference Engine*)

Merupakan program yang mengontrol proses menyimpulkan sebuah keadaan berdasarkan basis pengetahuan yang ada, memanipulasi dan memverifikasi aturan, pola, dan fakta yang disimpan dalam database untuk sampai pada solusi atau kesimpulan.

4. Area Kerja (*Blackboard*)

Sistem membutuhkan area kerja yaitu, area memori yang berfungsi sebagai database, untuk menyimpan hasil rata-rata yang digunakan sebagai keputusan dan untuk menjelaskan masalah yang sedang terjadi saat ini.

5. Antarmuka Pengguna (*User Interface*)

Dipakai sebagai alat komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling baik jika disajikan dalam bahasa alami dan disertai dengan grafik, menu, dan formulir elektronik.

6. Subsisted Penjelasan (*Explanation Subsystem / Justifier*)

Menjelaskan kepada pengguna bagaimana kesimpulan dapat ditarik. Keterampilan ini sangat penting bagi pengguna untuk mengetahui transfer pengetahuan dan proses pemecahan masalah.

#### 7. Sistem Pengolahan Data

Fungsi peningkatan pengetahuan seorang pakar (knowledge refinement system) dibutuhkan dalam menganalisa pengetahuan dan selanjutnya meningkatkan pengetahuan agar bisa digunakan di masa depan.

#### 8. Pemakai (*User*)

Biasanya pemakai sistem bukanlah ahli yang memerlukan jalan keluar, kritik atau pelatihan untuk berbagai masalah yang ada.

### 4. Representasi Pengetahuan

Pemrosesan yang dikerjakan oleh sistem adalah pemrosesan informasi, bukan pemrosesan data seperti halnya pemrograman tradisional, yang banyak dilakukan oleh unsur informasi (Hartati & Sari, 2008, p. 18).berikut objek yang menghubungkan struktur kaidah *IF-THEN* yaitu:

1. *IF* premis *THEN* konklusi
2. *IF* input *THEN* output
3. *IF* keadaan *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnosa

Asumsi atau perkiraan mengarah terhadap realita yang pasti benar sebelum kesimpulan tertentu bisa ditarik. Inputan berarti informasi yang wajib tersaji sebelum hasil dapat dicapai. Keadaan mengarah terhadap situasi yang harus ada

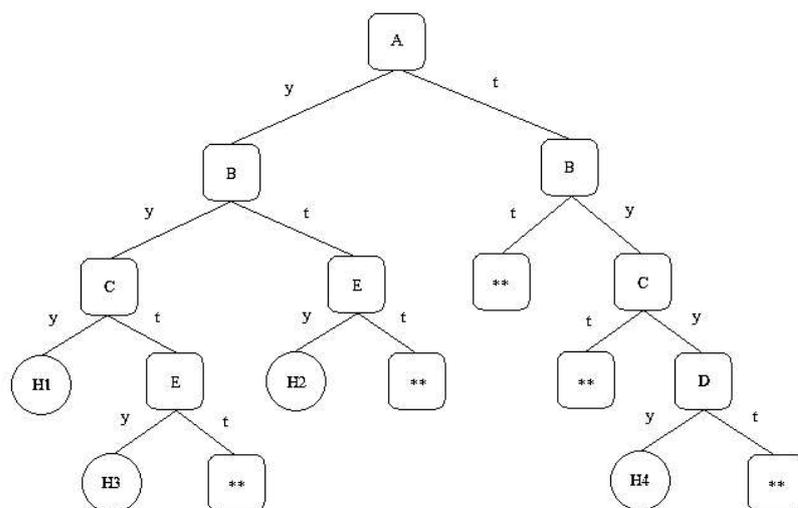
sebelum sebuah aksi dapat dilaksanakan. Contoh mengarah pada kondisi yang terjadi sebelum konsekuensi dapat dicermati. Data mengarah pada informasi yang harus tersedia untuk mencapai hasil. Gejala mengarah pada kondisi berbahaya atau kondisi spesifik yang membutuhkan pengecekan atau diagnosa.

Sebelum masuk ke dalam bentuk aturan produksi, informasi yang diperoleh dari suatu area disediakan pada bentuk tabel putusan dan selanjutnya dibuat pohon keputusan. Di bawah berikut merupakan gambaran representasi dalam tabel keputusan dan pohon keputusan (Hartati & Sari, 2008, pp. 26–39).

**Tabel 2. 4** Tabel Keputusan

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

Sumber: Hartati dan Iswanti (2008: 32)



**Gambar 2.2** Pohon Keputusan

Sumber: Hartati dan Iswanti, 2008: 33

Penjelasan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, \*\* = tidak mendapatkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Berdasarkan gambar diatas dapat dipahami apabila memenuhi *evidence* A, B, dan C maka hipotesa H1 terpenuhi. Apabila memiliki *evidence* A dan *evidence* E maka Hipotesa H2 terpenuhi. Apabila memiliki *evidence* A, B, dan E maka Hipotesa H3 akan terpenuhi. Apabila memenuhi *evidence* B, C, dan D maka Hipotesa H4 akan dihasilkan. Notasi “y” berisi arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak memenuhi.

Berdasarkan pohon keputusan pada gambar 2.2 masalah bisa muncul di awal sesi konsultasi, yaitu disaat sistem pakar bertanya "Apakah ada bukti A?". Masalahnya ialah setiap jawaban pengguna adalah "ya" atau "tidak", sistem meminta bukti B. Artinya sistem tidak terpengaruh oleh jawaban pengguna. Model penyajian informasi aturan produksi banyak dipakai pada aplikasi sistem pakar karena bentuk penyajian ini mudah dimengerti, bermakna serta mudah diinterpretasikan sesuai dengan metode pemecahan masalah pikiran manusia.

## 5. Forward Chaining (Runut Maju)

Runut maju merupakan teknik penelusuran yang diawali dengan realita yang dipahami, selanjutnya menggabungkan bukti tersebut dengan bagian *IF* dari *rules IF-THEN*. Jika fakta tersebut cocok dengan bagian *IF*, maka aturannya berfungsi. Saat aturan berjalan, fakta baru ditambahkan ke database (bagian *THEN*). Setiap

kali cocok, diawali dengan aturan paling atas. Masing-masing aturan dapat dilakukan satu kali saja. Tahap penyamaan berakhir ketika tidak lagi ada aturan yang harus dilakukan.

Konsep ini juga bisa disebut sebagai pencarian berbasis data. runut maju melakukan penalaran (*inferensi*) dimulai pertama dengan premis atau laporan input (*IF*) dan selanjutnya dengan kesimpulan atau informasi turunan (*THEN*).

Sebab laporan *inputan* dalam bentuk pengamatan serta resolusi mungkin dalam bentuk diagnosa, bisa dikatakan bahwa penalaran berkembang dari pengamatan terhadap diagnosa. Dengan tahap ini, sistem bukan mengasumsi atas kesimpulan, melainkan sistem memperoleh semua fakta yang disarankan oleh pemakai, memeriksanya dan menerangkannya ke sebuah resolusi yang tepat (Sutojo, 2011, p. 171).

## **2.2 Variabel**

### **2.2.1 Penyakit Lambung**

Lambung adalah bagian tubuh yang mempunyai ruang besar yang mirip dengan saluran antara kerongkongan dengan usus mini yang bertanggung jawab dalam sistem pengolahan makanan dalam tubuh manusia. Kelebihan asam dalam lambung seringkali menyebabkan sakit perut, suatu kondisi pada saluran cerna terutama pada lambung yang disebabkan oleh tingginya kadar asam lambung yang dapat mengiritasi saluran cerna. Dalam keadaan kosong, lambung menyamai tabung bentuk J, apabila penuh, berbentuk seperti buah pir besar. Lambung terbentuk dari *antrum kardia* (yang menerima *esofagus*), *fundus* besar seperti kubah, badan utama atau *korpus* dan *pylorus* (Price & Wilson, 2006).

## 2.2.2 Beberapa Jenis Penyakit Lambung

### 1. Gerd

Gerd merupakan refluks berulang dengan atau tanpa penyakit mukosa, namun dapat mempengaruhi kegiatan kehidupan seseorang. Pada Gerd, peradangan dan nyeri pada kerongkongan terjadi akibat asam lambung dan enzim yang mengalir kembali ke kerongkongan dari lambung.

### 2. Mahg (*gastritis*)

Mahg adalah peradangan pada lambung yang disebabkan oleh peningkatan sekresi asam berlebih dan mengiritasi lambung. Maag bisa mengakibatkan sakit perut dan mules berlebih akibat kebiasaan tidak teratur makan.

### 3. Tukak Lambung (*peptic ulcer*)

Merupakan iritasi pada lambung maupun usus dua belas jari yang disebabkan oleh ketidaksetaraan aspek agresif seperti *enzim* dalam getah lambung, sekresi asam dan radang *Helicobacter pylori*. Gejala ini umumnya sudah sangat parah karena kondisi lambung yang sudah luka berat.

## 2.2.3 Gejala Penyakit Asam Lambung

Setiap orang dapat mengalami gejala asam lambung yang meningkat, terutama setelah mengonsumsi makanan dalam jumlah banyak dan setelah mengonsumsi makanan yang sulit dicerna atau produksi asam lambung yang meningkat, beberapa ciri gejala dari penyakit lambung yaitu:

1. Perut buncit/ kembung.
2. Mulas/muak.
3. Rasa nyeri di perut bagian kiri.

4. Di dada terasa panas.
5. Tenggorokan terasa ada hambatan/ganjalan.
6. Terasa pahit atau asam di mulut.
7. Sering bersendawa.

Adapun cara penanggulangan dari penyakit lambung antara lain:

1. Dalam mengkonsumsi makanan kurangi konsumsi olahan terlampau pedas, produk bersantan, makanan asam, olahan bertepung dan gorengan serta minuman bersoda.
2. Mengatur pola makan, makanlah dalam waktu seperdelapan setiap hari, dan jangan dalam porsi besar sekaligus, serta jangan langsung berbaring 30 menit setelah makan. Dan saat mengunyah sebaiknya dikunyah hingga 23 kali, agar kudapan makin lembut dan gampang diolah, dan tidak boleh menunda-nunda sarapan dan makan.
3. Tidak terlampau memikirkan semua hal karena depresi dapat meningkatkan asam berlebih, jadi depresi perlu dikendalikan.
4. Penyembuhan dengan obat-obatan, yaitu *antasida* sangat membantu dalam atasi asam lambung jika kondisinya bukan kronis. Pola makan dijaga, makan dan minum olahan yang baik dikonsumsi Misalnya, makanan tidak terlalu pedas dan asam, olahan berkafein, konsumsi air mineral serta istirahat yang cukup.

## **2.3 Software Pendukung**

### **2.3.1 UML (*Unified Modelling Language*)**

*UML* umumnya dipergunakan dalam perancangan awal dari suatu sistem yang akan dibangun. *Unified Modelling Language (UML)* merupakan suatu “frasa” yang sudah sebagai patokan dalam industri untuk visualisasi, membangun dan menyimpan sistem *software*. *UML* memberikan suatu patokan dalam membangun model suatu prosedur/sistem. (Yasin & Verdi, 2012, p. 194).

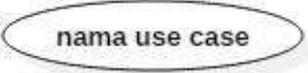
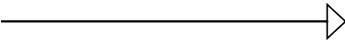
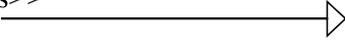
*UML* memiliki macam-macam diagram yaitu antara lain:

1. *Use case diagram*
2. *Class diagram*
3. *Activity diagram*
4. *Sequence diagram*

#### **1. Use Case Diagram**

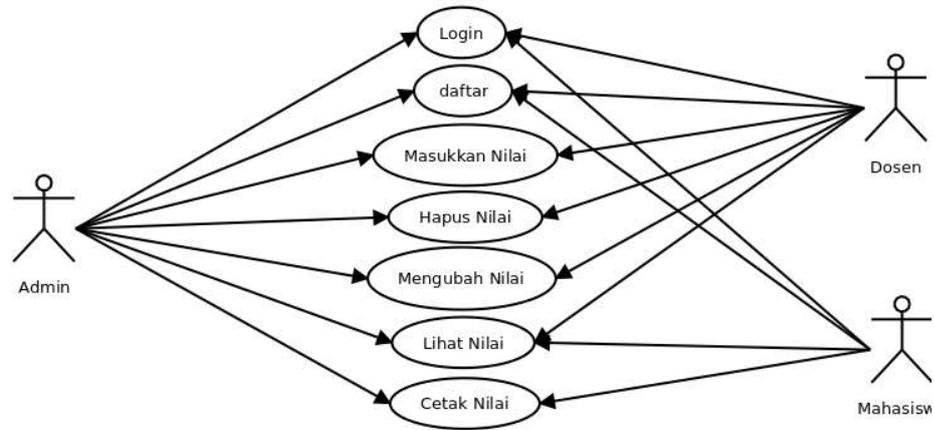
*Use case diagram* gambaran dari kegunaan yang diinginkan dari sistem. Penekanannya ialah pada "apa" yang dilakukan sistem, bukan "bagaimana". *Use case diagram* memiliki tugas khusus yakni menerangkan koneksi atau interaksi diantara aktor dan sistem. (Yasin & Verdi, 2012, p. 198).

Tabel 2. 5 Lambang *Use Case Diagram*

Lambang	Keterangan
<p><i>Use case</i></p> 	Fungsi yang ada pada sistem yaitu menjadi unsur-unsur yang saling berganti informasi diantara unit atau aktor.
<p>Aktor/<i>actor</i></p> 	Aktor tidak harus orang, dan biasanya dilambangkan dengan mengawali frase nama aktor dengan kata benda.
<p>asosiasi/<i>association</i></p> 	Aktor yang berpartisipasi dan berkomunikasi dalam sebuah <i>use case</i> atau mempunyai koneksi dengan aktor.
<p>Ekstensi/<i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	Hubungan <i>use case</i> ekstra dengan dirinya sendiri, dimana <i>use case</i> yang diekstra juga bisa mandiri tanpa <i>use case</i> lain.
<p>generalisasi/<i>generalization</i></p> 	Koneksi (umum-spesifik) diantara dua kasus penggunaan antara satu fitur lebih umum daripada yang lain.
<p>Menggunakan/<i>include/uses</i></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	Merupakan syarat agar <i>use case</i> dapat berjalan. Panah menunjuk ke kasus dimana penggunaan yang ditambahkan.

Sumber: A.S. dan Shalahuddin (Rosa & M.Shalahuddin, 2013, p. 162)

Contoh dari *use case diagram* dapat dilihat pada gambar:



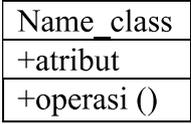
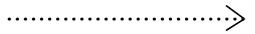
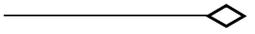
**Gambar 2.3** Contoh *Use Case Diagram*

Sumber: <https://www.pinhome.id/blog/contoh-use-case-diagram/>

## 2. *Class Diagram*

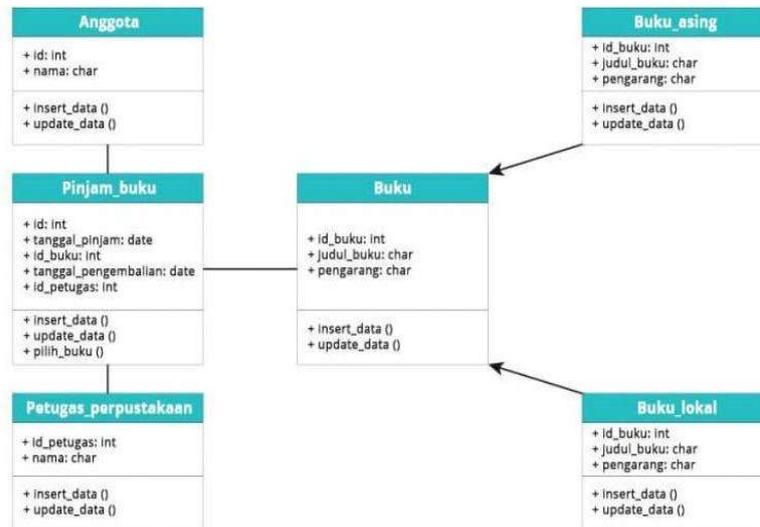
Diagram kelas ialah spesifikasi yang ketika dipakai dapat membuat objek dan merupakan inti dari desain dan pengembangan berorientasi objek. Kelas menggambarkan keadaan sistem (atribut/properti) dan menyajikan layanan untuk memanipulasi keadaan ini (metode/fungsi).

Tabel 2. 6 Lambang *Class Diagram*

Lambang	Keterangan
Class 	Kelas atas bentuk sistem.
Antarmuka / <i>interface</i>  <b>Name_ <i>interface</i></b>	Hal yang serupa berlaku untuk antarmuka dalam perancangan berfokus pada objek.
Asosiasi / <i>association</i> 	Hubungan umum diantara kelas, sebagian besar mengandung multiplisitas.
Asosiasi berarah / <i>directed Association</i> 	Relasi diantara kelas serta semantik kelas dipakai kelas lain, bersifat multiplisitas dalam asosiasinya.
Generalisasi 	Relasi antara kelas dalam hal umum-spesifik.
Keterkaitan / <i>dependency</i> 	Hubungan antara kelas menurut keterlibatan antar kelas
Penghimpunan / <i>aggregation</i> 	Pengelompokan semua bagian (lengkap-sebagian)

Sumber: (Rosa & M.Shalahuddin, 2013, pp. 123–124)

Berikut merupakan contoh gambar dari *class diagram*:



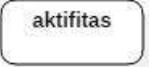
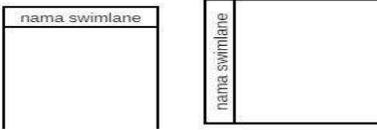
**Gambar 2.4** Contoh *Class Diagram*

Sumber: <https://www.dicoding.com/blog/memahami-class-diagram-lebih-baik/>

### 3. Activity Diagram

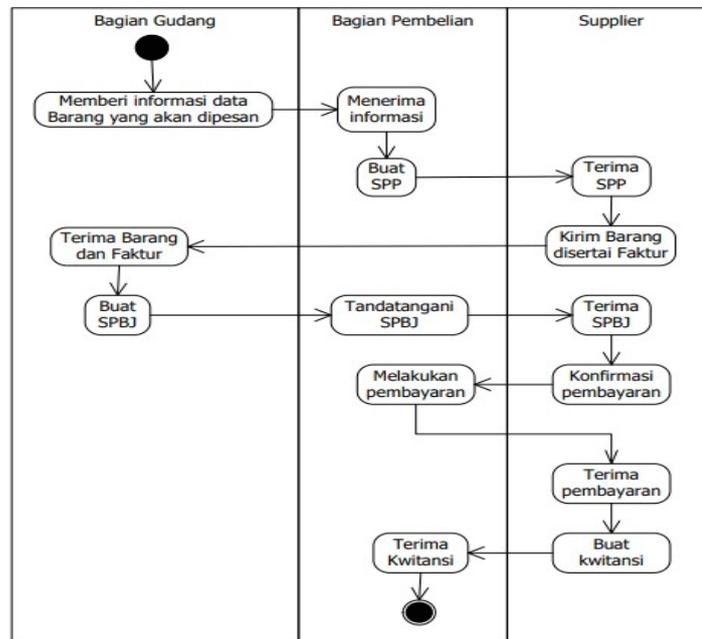
Diagram aktivitas menerangkan alur aktivitas yang berbeda dari sistem yang direncanakan, bagaimana setiap aliran dimulai, keputusan apa yang dapat terjadi, dan bagaimana akhirnya. Diagram aktivitas juga dapat mewakili proses paralel yang dapat terjadi dalam banyak eksekusi (Yasin & Verdi, 2012, p. 201).

Tabel 2. 7 Lambang *Activity Diagram*

Lambang	Keterangan
kondisi awal 	Keadaan awal operasi sistem, skema operasi mempunyai keadaan awal.
Aktivitas 	Aktivitas yang dikerjakan pada sistem Biasanya aktivitas yang diawali dengan frase kerja.
Percabangan/ <i>decision</i> 	Gabungan industri dengan lebih dari satu peluang operasi.
Penggabungan/ <i>join</i> 	Penggabungan dimana lebih dari satu kegiatan dihubungkan menjadi satu.
Status akhir 	Keadaan akhir berdasarkan sistem, diagram fungsional mempunyai keadaan akhir.
<i>Swimlane</i> 	Pemisahan organisasi bisnis yang bertanggung jawab atas kegiatan yang berlangsung.

Sumber:A.S. dan Shalahuddin (Rosa & M.Shalahuddin, 2013, pp. 162–163)

Contoh *activity diagram* dapat dilihat pada gambar berikut:



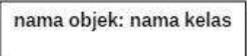
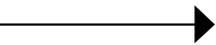
**Gambar 2.5** Contoh *Activity Diagram*

Sumber: <https://www.dicoding.com/blog/apa-itu-activity-diagram/>

#### 4. *Sequence Diagram*

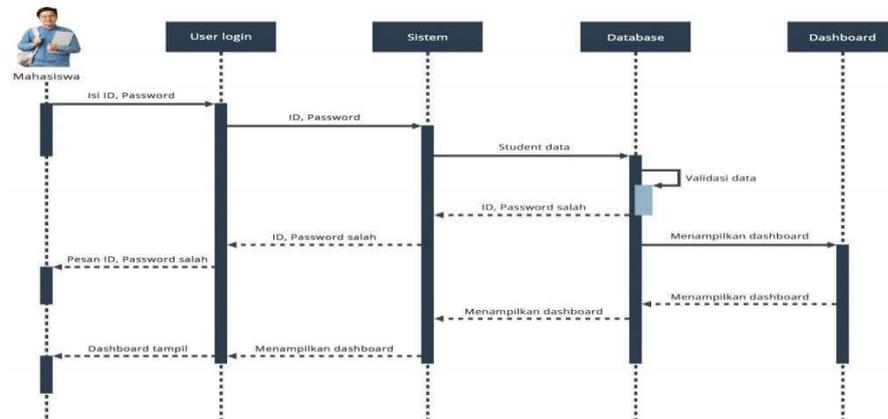
*Sequence diagram* merupakan gambaran hubungan antara objek di dalam serta di sekitar sistem yakni pengguna, layar, dan lainnya. Dalam hal pesan yang dijelaskan dalam waktu. *Sequence diagram* terbentuk dari dimensi vertikal (waktu) dan dimensi horizontal (objek berkaitan). *Diagram urutan* sering dipakai dalam menyimpulkan output tertentu (Yasin & Verdi, 2012, p. 201).

Tabel 2. 8 Lambang *Sequence Diagram*

Lambang	Keterangan
<p>Aktor/<i>actor</i></p> 	Orang, tahap ataupun sistem lain yang berhubungan bersama sistem informasi yang mau dikerjakan di luar sistem informasi itu sendiri.
<p>Garis hidup/<i>lifeline</i></p> 	interaksi antar aktor yang terlibat dalam use case serta berinteraksi langsung terhadap aktor.
<p>Objek</p> 	Mendeklarasikan objek yang berkomunikasi dengan pesan.
<p>Waktu aktif</p> 	Mendeklarasikan objek dan dapat berkomunikasi, serta segala sesuatu yang berhubungan bersama waktu aktif merupakan langkah yang dikerjakan di dalamnya.
<p>Tipe pesan <i>create</i></p> <p>&lt;&lt;create&gt;&gt;</p> 	Mendeklarasikan objek menciptakan objek lain. Arah panah menunjuk ke objek yang dibangun.

Sumber: A.S. dan Shalahuddin (Rosa & M.Shalahuddin, 2013, pp. 166–167)

Berikut merupakan contoh gambar dari *sequence diagram*:



**Gambar 2.6** Contoh *Sequence Diagram*

Sumber: <https://www.dicoding.com/blog/apa-itu-sequence-diagram/>

### 2.3.2. HTML (*Hypertext Markup Language*)

HTML adalah format hypertext sederhana untuk menampilkan berbagai informasi, membuat halaman web, digunakan oleh browser web Internet, dan ditulis dalam file ASCII. Bahasa dasar yang digunakan untuk membuat dan menghasilkan tampilan terintegrasi, Menurut wikipedia.



**Gambar 2.7** Logo HTML

Sumber: <https://www.google.com/search?q=logo+html&oq=logo+html&aqs>

### 2.3.3 PHP (*Hypertext Preprocessor*)

*PHP* merupakan bahasa perancangan yang dipakai dalam mengubah basis kode program menjadi kode mesin yang mampu dimengerti sisi server dan menambahkan HTML ke dalamnya serta untuk menjadikan website dinamis yang mampu berkomunikasi dengan pengunjung dan pengguna (Wardana, 2016:1).



**Gambar 2.8** Logo PHP

Sumber: <https://www.pngdownload.id/download/php-logo>.

### 2.3.4 CSS (*Cascading Syle Sheet*)

CSS merupakan bahasa pemrograman sistem yang dirancang khusus dalam mengendalikan dan membuat berbagai komponen sistem agar web terlihat lebih rapi, lebih terstruktur, dan lebih konsisten (Solichin, 2016:10). CSS merupakan bahasa yang membantu perancangan desain tata letak situs web dan membuat efek animasi yang bagus (Jasakom, 2012, pp. 27–28).



**Gambar 2.9** Logo CSS

Sumber: <https://www.freepik.com/free-photos-vectors/logo-css>

### **2.3.5 XAMPP (*X Apache MySQL PHP Perl*)**

XAMPP merupakan *software*, mendorong banyak sistem operasi, serta merupakan komplikasi dari banyak program yang bekerja sebagai server independen (localhost) yang terdiri dari program Apache *HTTP* Server, database *MY SQL* dan *Perl*. Tersedia di bawah Lisensi Publik Umum *GNU* dan gratis, program ini adalah server *web* yang gampang dipergunakan yang mampu menampilkan halaman web dinamis (Alan Nur, 2011, p. 16).



**Gambar 2.10** Logo XAMPP

Sumber: [https://id.m.wikipedia.org/wiki/Berkas:Xampp\\_logo.svg](https://id.m.wikipedia.org/wiki/Berkas:Xampp_logo.svg)

### 2.3.6 MySQL Database

*MySQL* merupakan database SQL open source yang berkembang pada saat ini. Pada database *MySQL* mendorong banyak fungsi seperti *multithreading*, banyaknya pengguna dan sistem manajemen database SQL (*DBMS*). Basis data ini mempertimbangkan sistem basis data yang cepat, andal, dan mudah dipakai (Madcoms, 2016, p. 2).



**Gambar 2.11** Logo MySQL  
Sumber: <https://1000logos.net/mysql-logo/>

### 2.3.7 JavaScript

*JavaScript* adalah bahasa scripting yang disebarakan oleh *Netscape*, Dalam mengoperasikan skrip yang ditulis dalam *JavaScript*, yang memerlukan *browser* dalam mengaktifkan *JavaScript*, yaitu browser yang dapat mengeksekusi *JavaScript* (Jasakom, 2012, p. 3).



**Gambar 2.12** Logo JavaScript

Sumber: <https://www.freepnglogos.com/pics/javascript>

### **2.3.8** *Notepad++*

*Notepad++* merupakan pengeditan teks yang guna untuk semua orang dan terkhusus untuk editor-editor saat pembuatan aplikasi. *Notepad++* terdiri dari komponen *Scintilla* untuk dapat melihat dan mengedit file teks dan kode menghasilkan berbagai bahasa pemrograman yang berlangsung di sistem operasi Windows. Selanjutnya kelebihan dan *skill* ketika mengerjakan berbagai bahasa pemrograman, *Notepad++* dilisensikan dengan alat gratis. siapapun yang membuka aplikasi tersebut tidak perlu membayar software ini sebab *sourceforge.net* menyediakannya sebagai layanan untuk memudahkan *Notepad++*. (Madcoms, 2016, p. 15).



**Gambar 2.13** Logo Notepad++

Sumber: <https://www.computerhope.com/jargon/n/notepad-plus-plus.htm>

## 2.4 Penelitian Terdahulu

Penelitian yang pertama dibuat oleh Nanda Jarti, d.k.k (Jarti, 2017) dengan judul **“Sistem Pakar Diagnosa Penyakit Alergi pada Anak Berbasis Web dengan Metode *Forward Chaining* di Kota Batam”** Tujuan dari penelitian ini merupakan membuat *website* dengan konsep sistem pakar untuk mendiagnosa penyakit alergi pada anak dengan menggunakan metode *forward chaining*, yang akan memudahkan dalam mendiagnosa alergi anak tersebut oleh pengguna.

Penelitian yang kedua dibuat oleh Marla Nur Assyifa (Marla Nur Assyifa, 2019) dengan judul **“Aplikasi Sistem Pakar Berbasis Android Untuk Diagnosis Penyakit *Gastroesophageal Reflux Disease (Gerd)* Dengan Metode *Certainty Factor*”** Riset ini membuat sistem pakar dalam mendiagnosa *gastroesophageal reflux disease (GERD)* dengan menerapkan metode *certainty factor*. Sistem pakar ini dipakai dalam mendiagnosa *GERD* yang terjadi pada penderita dengan tepat dan jelas dengan berdasar pada ciri-cirinya.

Penelitian ketiga dilaksanakan oleh Rahmad Dian, d.k.k (Rahmad Dian, 2020) dengan judul **“Sistem Pakar dalam Identifikasi Kerusakan Gigi pada**

**Anak dengan Menggunakan Metode *Forward Chaining* dan *Certainty Factor***".

Pada riset ini membuat sistem pakar yang mendeteksi bagaimana rusaknya gigi pada balita, sistem ahli dalam pengenalan sepuluh macam rusaknya gigi pada balita dengan ciri-ciri yang terjadi.

Penelitian keempat ditulis oleh Rizky Ardiansyah, d.k.k (Rizky Ardiansyah, 2019) dengan judul "**Sistem Pakar Untuk Diagnosa Awal Penyakit Lambung Menggunakan Metode *Dempster-Shafer* Berbasis Web**" Pada riset ini, sistem pakar mengajarkan pemahaman mengenai tentang *GERD*, *gastroparesis*, *dispepsia*, serta penyakit tukak lambung.

Penelitian kelima ditulis oleh Muhammad Ruswin Nasution, d.k.k (Muhammad Ruswin Nasution, 2021) dengan judul "**Perancangan Sistem Pakar Mendiagnosa Penyakit Covid-19 Dengan Metode *Backward Chaining* Berbasis Online**" Dalam riset ini, sistem pakar diagnosa gejala COVID-19 yang dipakai oleh pakar ataupun *user* dalam mendapatkan berita tentang ciri-ciri pertama saat orang terpapar COVID-19.

Penelitian keenam ditulis oleh Abdul Muis Alfatah, d.k.k (Alfatah, 2018) dengan judul "***Implementation of Decision Tree and Dempster Shafer on Expert System for Lung Disease Diagnosis***" pada riset tersebut dikatakan "*Dempster Shafer can optimize the resulting diagnostics because the system is not only rule-based, it also has value. The level of confidence is more accurately supported by decision trees as an aid for disease prediction. Dempster Shafer uses density values or expert-derived weights for each known fact. The weights of these facts are combined to produce a known combination of actual density values*".

(*Dempster Shafer* dapat mengoptimalkan diagnostik yang menerangkan bahwa sistem bukan dari berbasis aturan, namun tak memiliki nilai. Dilihat dari nilai kepercayaan tentang pohon keputusan yang lebih terperinci. bantuan untuk diagnosa penyakit. *Dempster Shafer* akan mengetahui dalam setiap realita yang menghasilkan nilai yang menitik beratkan dari seorang pakar. Bobot realita dipadukan untuk membuat perpaduan angka densitas aktual yang dipahami).

Penelitian ketujuh yang ditulis oleh Muhammad Ilham Insani, d.k.k (Insani, 2018) dengan judul “***Implementation of Expert System for Diabetes Diseases using Naïve Bayes and Certainty Factor Methods***” pada penelitian dikatakan,” *The naive Bayes method has 3 steps, namely the previous survey, looking for probability values, and checking the probabilities after receiving the classification results from the posterior naive bayes calculations, making the selected class from the naive bayes classification process. calculate the safety factor to find the safety value. After calculating the confidence factor, the confidence value of the selected class is obtained in the Naive Bayes calculation*”.

(Metode naive bayes memiliki 3 langkah yaitu survey sebelumnya, mencari nilai probabilitas, dan memeriksa probabilitas setelah menerima yang mengkategorikan dalam menghitung posterior, dibuat berdasarkan kategori kelas naive bayes. menghitung faktor keamanan untuk mencari nilai keamanan. Setelah dilakukan perhitungan faktor kepercayaan maka didapatkan nilai kepercayaan dari kelas yang dipilih pada perhitungan Naive Bayes.)

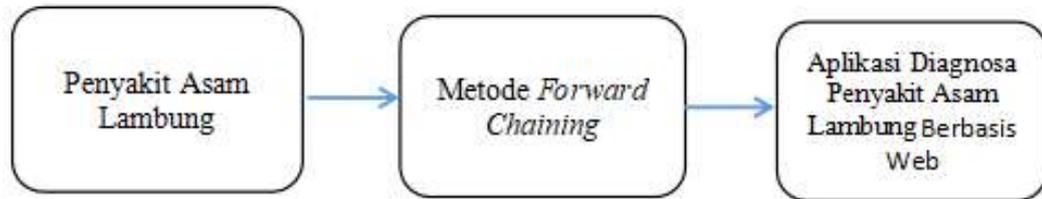
## 2.5 Kerangka Pemikiran

Framework merupakan bentuk yang konseptual mengenai teori yang berkaitan dengan beberapa faktor yang teridentifikasi dengan isu-isu penting. Dengan teori, cara berpendapat yang benar menerangkan aturan setiap variabel yang diperiksa (Sudaryono, 2015, p. 21)

menunjukkan bahwa keadaan pikiran yang baik meliputi:

1. Variabel yang akan diperiksa harus diterangkan.
2. Pembahasan mengenai penalaran pemikiran wajib bisa memperlihatkan dan menerangkan koneksi antara variabel yang diperiksa dengan materi yang melatarbelakanginya.
3. Pembahasan memperlihatkan serta menerangkan bagaimana koneksi antar variabel bersifat baik atau buruk, harmonis, sebab akibat atau aksi (resiprokal atau reaksi).
4. Cara berasumsi berikut harus diungkapkan pada bentuk skema (desain riset) agar orang lain mengerti cara berasumsi yang disajikan pada riset.

Berikut merupakan bentuk dari kerangka berfikir dalam riset ini:



**Gambar 2.14** Kerangka Pemikiran  
Sumber: Desain penelitian (2022)

Berdasarkan gambar diatas dapat diuraikan sebagai berikut ini:

1. Menu input yang menjelaskan tentang adanya penyakit asam lambung yaitu *Gerd*, Magh akut, Magh kronis dan Tukak lambung.
2. Menu proses yang menjelaskan tentang bagaimana inputan diproses ke dalam suatu sistem web dengan menggunakan metode *forward chaining* dan menggunakan software pendukung seperti *xampp*, *notepad++* dan lain sebagainya.
3. Menu output menjelaskan tentang tampilan aplikasi berbasis web diagnosis penyakit asam lambung yang sederhana namun mudah dipahami.