

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).

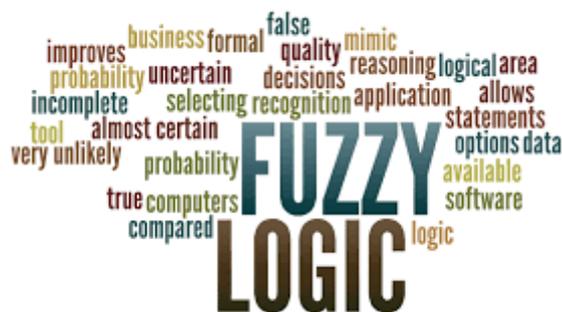
1. Fuzzy Logic

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*.



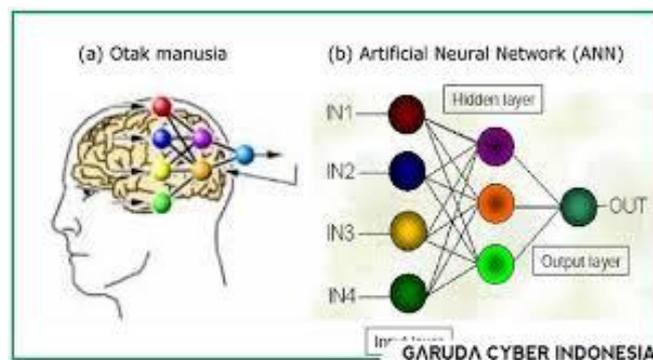
Gambar 2.2 logika fuzzy

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 dipKti 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 tepat 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

Sistem Normal	Sistem Pakar
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 antesenden 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. *Antesenden* 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.

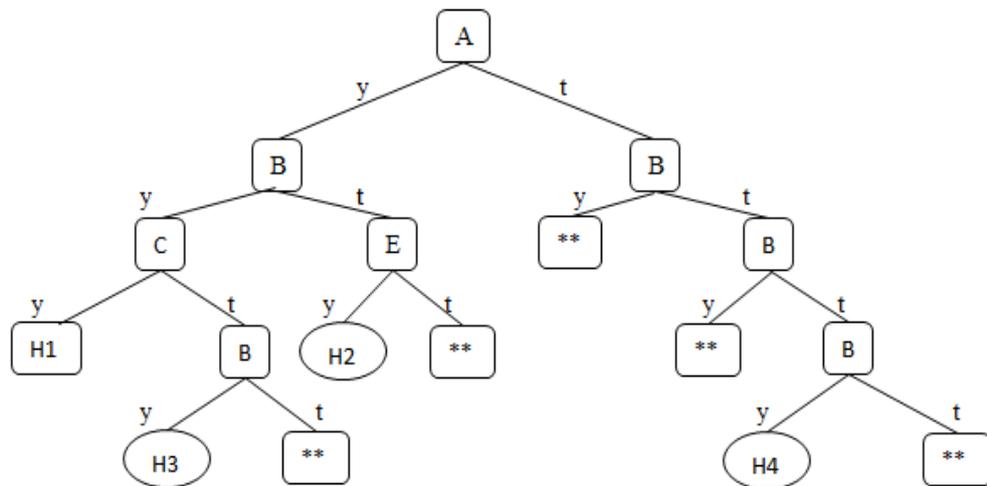
Sebelum sampai pada bentuk kaidah produksi, terdapat langkah-langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut adalah menyampaikan pengetahuan yang sudah berhasil didapatkan dalam bentuk tabel keputusan (*decision table*) kemudian dari tabel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

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
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

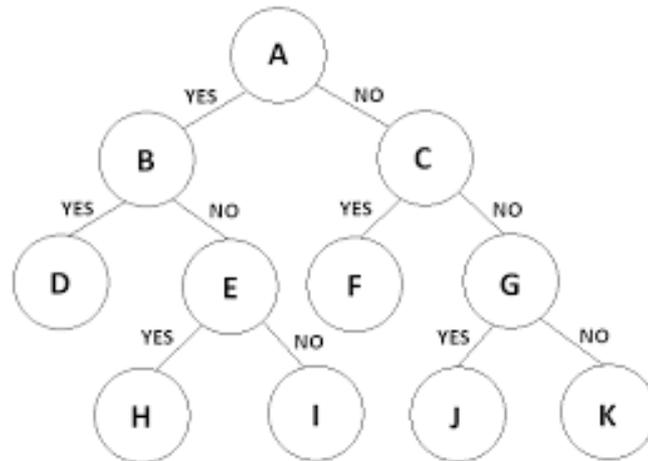
Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
Bukti A	Ya	Ya	Ya	Tidak
Bukti D	Tidak	Tidak	Tidak	Ya
Bukti B	Ya	Tidak	Ya	Ya
Bukti C	Ya	Tidak	Tidak	Ya
Bukti E	Tidak	Ya	Ya	Tidak

Sumber:(Zulfian, 2017)

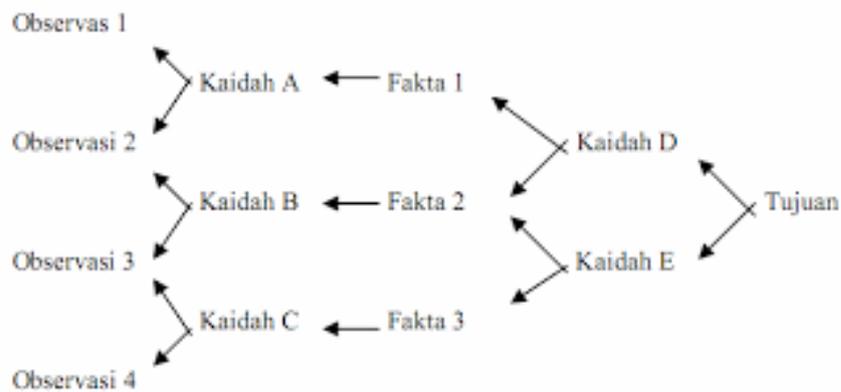
2.1.3 *Forward Chaining*

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:

- a. 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.
- b. 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

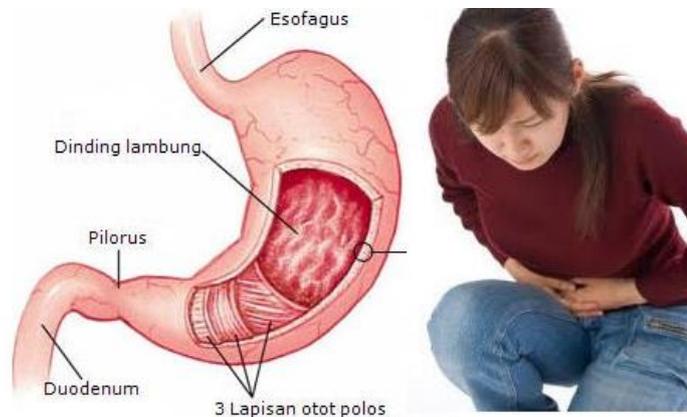
yang disebut dengan browser web (web browser), Internet Explorer dan Netscape Navigator merupakan browser web yang populer. (Gani, 2017)

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 abdomen 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)

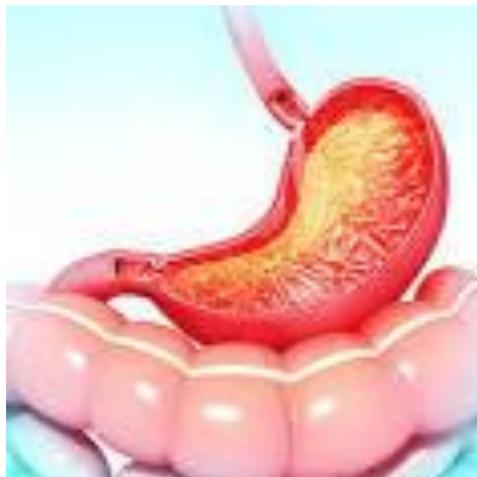
Penyakit lambung merupakan penyakit yang tidak bisa dianggap remeh, karena jika dibiarkan terus menerus dapat menyebabkan kematian jika tidak segera ditangani. Penyakit lambung dapat disebabkan oleh pola makan yang tidak sesuai, beban pikiran dan juga infeksi yang disebabkan bakteri. Beberapa penyakit

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, reflus empedu, zat korosif (cuka, lada) kemoterapi

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 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 hari, 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

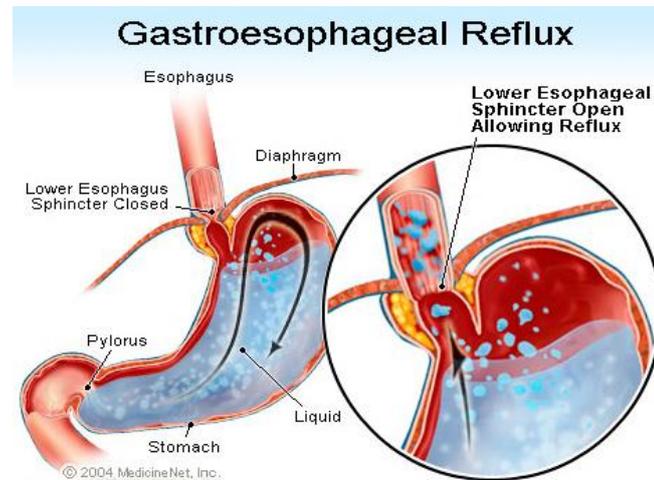
Melaksanakan pola hidup yang sehat, mengkonsumsi makanan yang bergizi dan sehat, mengkonsumsi buah dan sayur

4. Diagnosis

Pencatatan riwayat medis, Pemeriksaan fisik, Teknik-teknik diagnostik

4. GERD

Gastroesophageal Reflux Disease (GERD) merupakan suatu keadaan patologis yang merupakan akibat dari *reflux* kandungan 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.

Kedaaan *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 reflusat 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

Indikator	Gejala
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 (Jubliee, 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 (Jubliee, 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* (*timestanp*).
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 (ekspesi) {
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 pemrograman *PHP*, dimaksudkan untuk menangani administrasi *MySQL* melalui *Web*, berikut logo dari *phpMyAdmin*.



Gambar 2.17 Logo phpMyAdmin
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).

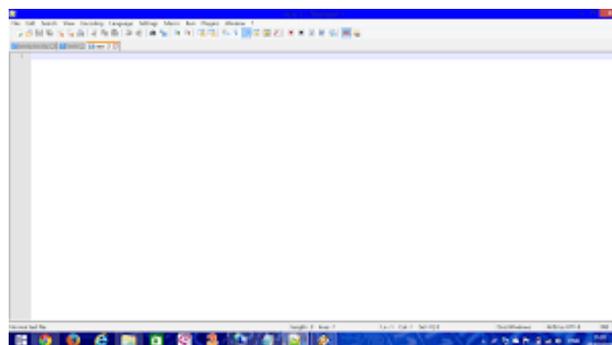
4. Fleksibel, interaktif, tampilan lebih menarik dan nyaman dipandang.
5. Lebih kecil ukuran *file* sehingga *bandwith* yang digunakan juga otomatis menjadi lebih kecil.
6. Dapat digunakan pada semua *web browser*.

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: (Jubliee, 2018)



Gambar 2.21 Tampilan awal notepad++
Sumber: (Jubliee, 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: (Jubliee, 2018)



Gambar 2.26 Tampilan awal Star UML
Sumber: (Jubliee, 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*.

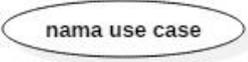
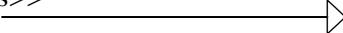
Menurut (A.S & Shalahuddin, n.d.) *use case* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:

1. *Use Case Diagram*

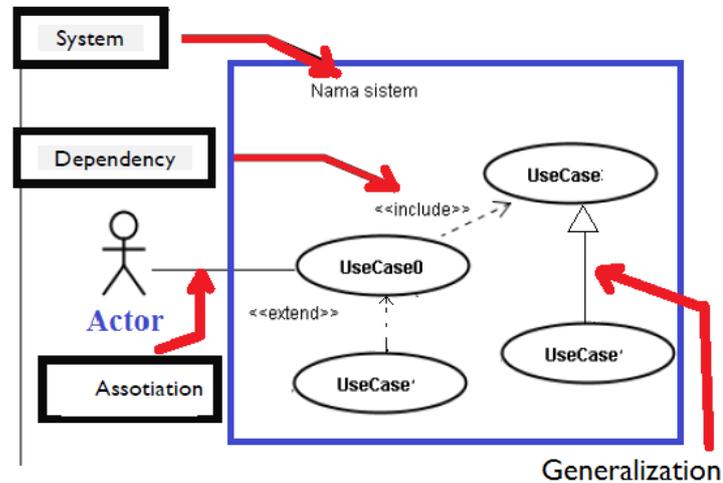
Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang 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 data-bbox="507 416 619 443">Use case</p> 	<p data-bbox="826 432 1345 611">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p data-bbox="323 647 480 674"><i>Aktor/actor</i></p> 	<p data-bbox="826 647 1345 898">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 data-bbox="323 925 580 952"><i>asosiasi/association</i></p> 	<p data-bbox="826 936 1345 1077">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 data-bbox="323 1120 528 1146"><i>Ekstensi/extend</i></p> <p data-bbox="323 1173 480 1200"><<extend>></p> 	<p data-bbox="826 1120 1345 1335">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 data-bbox="323 1361 671 1388"><i>generalisasi/generalization</i></p> 	<p data-bbox="826 1373 1345 1624">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 data-bbox="323 1664 676 1691"><i>Menggunakan/include/uses</i></p> <p data-bbox="323 1718 491 1744"><<include>></p>  <p data-bbox="323 1823 453 1850"><<uses>></p> 	<p data-bbox="826 1664 1345 1915">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. Susunan 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.

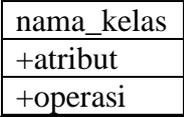
3. Kelas yang diambil dari pendefinisian *usse case*

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

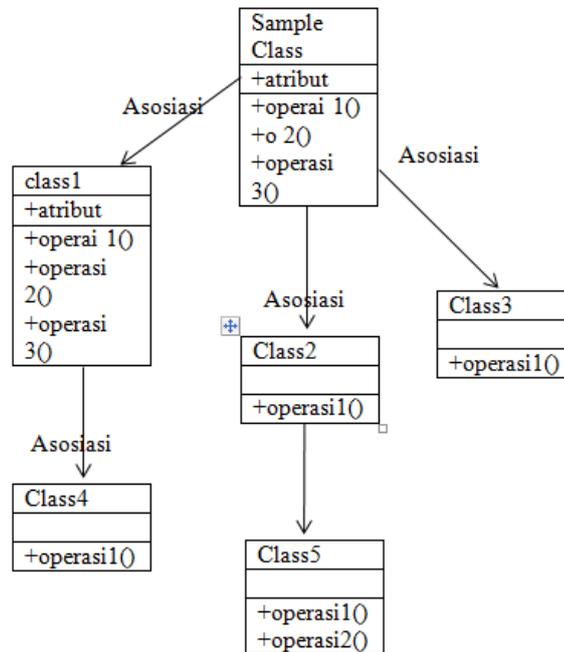
4. Kelas yang diambil dari pendefinisian data

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

Tabel 2.6 Simbol Class Diagram

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka/<i>interface</i></p> 	Sama dengan konsep interface dalam pemrograman dalam berorientasi objek
<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
<p>Kebergantungan / <i>defedency</i></p> 	Relasi anata kelas dengan makna kebergantungan antar kelas
<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

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

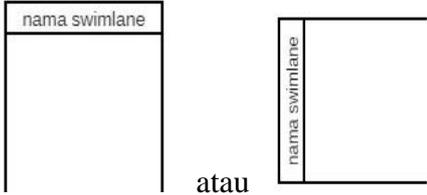


Gambar 2.28 Class Diagram
Sumber: (A.S & Shalahuddin, n.d.)

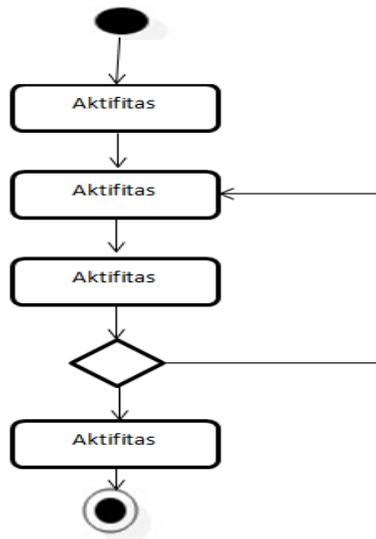
3. Activity Diagram

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)



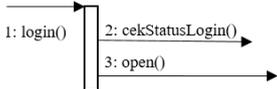
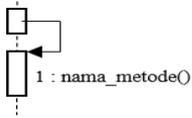
Gambar 2.29 Usecase Diagram
Sumber: (A.S & Shalahuddin, 2018)

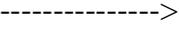
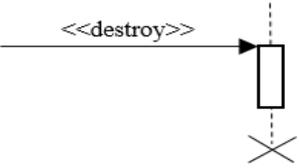
4. Sequence Diagram

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.

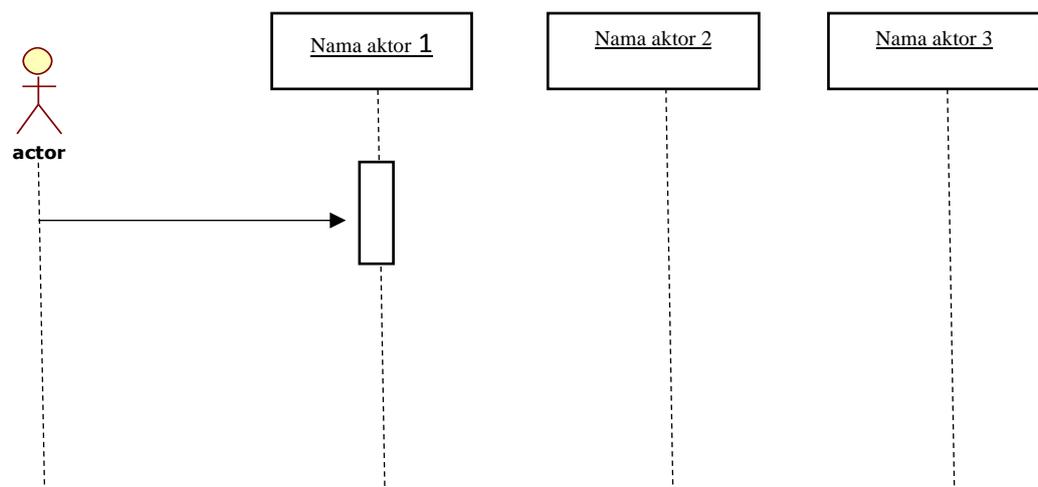
Tabel 2.8 Simbol *Sequennce* Diagram

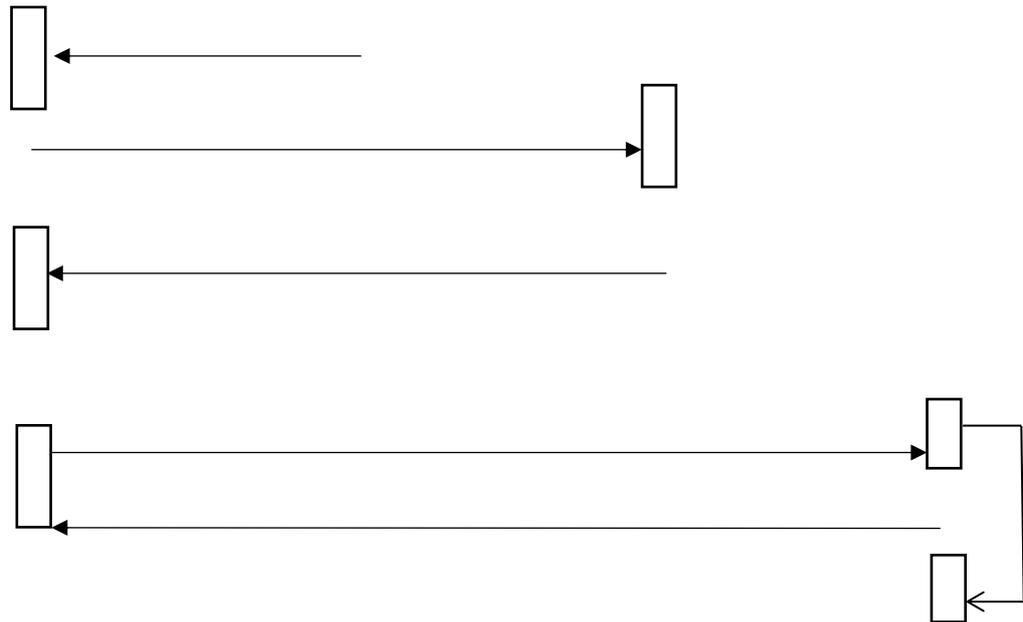
Simbol	Deskripsi
<p data-bbox="300 1518 379 1547">Aktor</p>  <p data-bbox="427 1682 491 1711">Atau</p> <div data-bbox="327 1760 539 1868" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p data-bbox="363 1787 502 1816">Nama aktor</p> </div> <p data-bbox="300 1973 539 2002">Tanpa waktu aktif</p>	<p data-bbox="641 1518 1359 1843">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; width: fit-content; margin: 10px auto;"> <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><code><<create>></code></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>→</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 : masukan 	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 diangkat 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 kurangnya informasi dan pengetahuan 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 fakta bahwa, 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 fakta bahwa, Suatu gejala penyakit dapat merupakan indikasi dari suatu penyakit yang akan diderita. Setiap orang wajib menjaga kesehatannya masing-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.

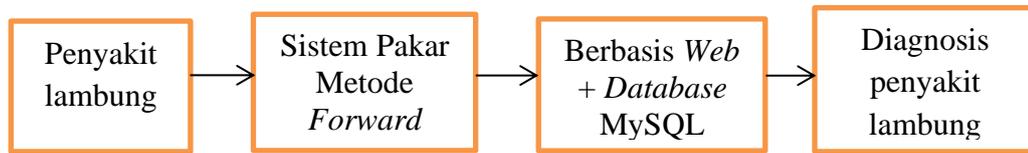
4. (Ritonga, 2013) **Sistem Pakar Mendiagnosa Penyakit Lambung Menggunakan Metode Certainty Factor**di peroleh fakta bahwa, Penyakit lambung antara lain adalah sakit maag (*gastritis*), dispepsia dan *gastroesophageal reflux disease* (GERD). Penyakit maag diakibatkan oleh asam lambung yang berlebihan, 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** diperoleh fakta bahwa, Kemampuannya untuk mendinginkan ruangan telah menarik minat penggunanya 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** diperoleh 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 pemrograman 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 pemrograman 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.