

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Pakar

2.1.1 Pengertian Sistem Pakar

Sistem Pakar adalah salah satu bidang dalam sebuah *Artificial Intelligence* (AI) yang berfokus pada implementasi aplikasi komputer yang akan dapat meniru aktivitas manusia yang berhubungan dengan pemikiran, keterampilan dan pengalaman. Sistem Pakar ini dapat digunakan untuk mendiagnosis, analisis dan membantu manajemen dalam hal pengambilan keputusan. Sistem Pakar (*Expert System*) dapat diartikan sebagai sebuah model yang saling berkaitan satu dengan yang lainnya dalam sebuah domain tertentu, yang mana tingkat keahlian Sistem Pakar dapat dibandingkan dengan keahlian seorang yang pakar di bidang yang sama dengan Sistem Pakar [5].

Sistem pakar adalah salah satu cabang kecerdasan buatan yang menggunakan pengetahuan – pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu. Sistem pakar dibuat hanya pada domain pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia disalah satu bidang saja. Sistem pakar mencoba mencari penyelesaian yang memuaskan, yaitu sebuah penyelesaian yang cukup bagus agar pekerjaan dapat berjalan walaupun itu bukan penyelesaian yang optimal. Tujuan utama sistem pakar bukan untuk mengganti kedudukan seorang pakar, tetapi hanya memasyarakatkan pengetahuan dan pengalaman seorang pakar [6].

2.1.2 Komponen-Komponen Sistem Pakar

Berikut ini penjelasan tentang komponen-komponen arsitektur sistem pakar adalah [7]:

1. Basis pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

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

2. Basis Data (*Databases Spreadsheet*)

Digunakan sebagai media yang berfungsi untuk menampung fakta-fakta, kondisi yang diperoleh dari basis pengetahuan untuk disimpan dan diproses oleh komputer.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi merupakan program yang memiliki fungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

4. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus jika dalam bahasa alami dan dilengkapi dengan *graphic*, *menu* dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

5. Fasilitas Penjelasan (*Explanation Subsystem/Justifier*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan ini penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

6. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*nonexpert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

7. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu. Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen multimedia, basis data, laporan riset khusus, dan informasi yang terdapat di web.

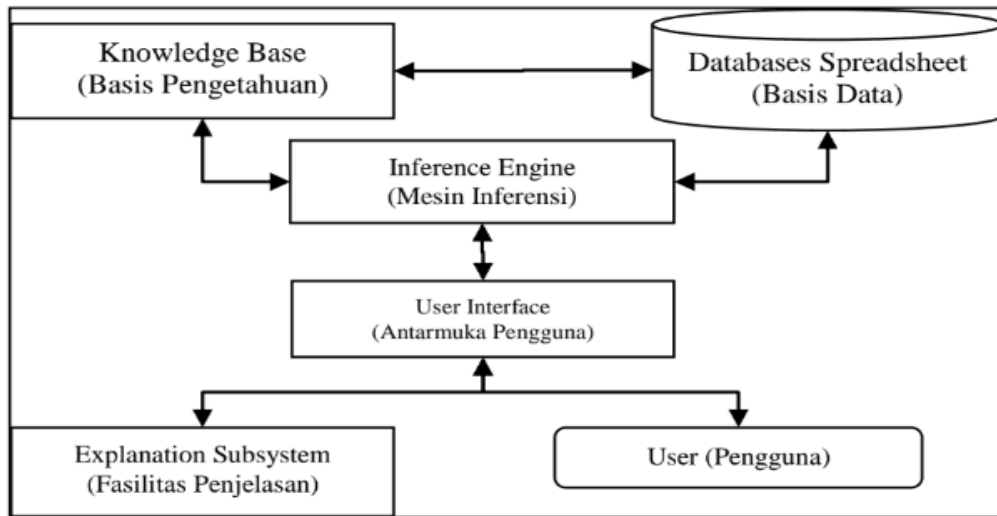
8. Daerah Kerja (*Blackboard*)

Merekam hasil sementara yang akan dijadikan sebagai kepusan dan untuk menjelaskan sebuah masalah yang sering terjadi, sistem pakar membutuhkan miring, yaitu area pada memori yang berfungsi sebagai basis data.

9. Sistem Perbaikan Pengetahuan

Kemampuan memperbaiki pengetahuan (*Knowledge Refining System*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat digunakan pada masa mendatang. Kemampuan evaluasi diri seperti itu

diperlukan oleh program agar dapat menganalisis alasan-alasan kesuksesan dan kegagalannya dalam mengambil keputusan.



Gambar 2.1 Komponen-Komponen Arsitektur Sistem Pakar

2.1.3 *Forward Chaining*

Algoritma *forward-chaining* adalah satu dari dua metode utama *reasoning* (pemikiran) ketika menggunakan *inference engine* (mesin pengambil keputusan) dan bisa secara logis dideskripsikan sebagai aplikasi pengulangan dari modus ponens (satu set aturan inferensi dan argumen yang valid). *Forward-chaining* mulai bekerja dengan data yang tersedia dan menggunakan aturan-aturan inferensi untuk mendapatkan data yang lain sampai sasaran atau kesimpulan didapatkan. Mesin inferensi yang menggunakan *forward-chaining* mencari aturanaturan inferensi sampai menemukan satu dari *antecedent* (dalil hipotesa atau klausa IF - THEN) yang benar. Ketika aturan tersebut ditemukan maka mesin pengambil keputusan dapat membuat kesimpulan, atau konsekuensi (klausa THEN), yang menghasilkan informasi tambahan yang baru dari data yang disediakan. Mesin akan mengulang melalui proses ini sampai sasaran ditemukan [8].

Mekanisme dari sistem forward chaining dimulai dengan memasukkan sekumpulan fakta yang diketahui ke dalam memori kerja, 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 atau sudah mencapai goal atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui.

Bentuk representasi Rule Based Reasoning digunakan karena memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan pakar dapat menyelesaikan masalah tersebut secara sistematis dan berurutan. Representasi berbasis aturan yang mempunyai pola IF kondisi/premis THEN aksi/konklusi pada suatu tabel pakar memberikan keuntungan pada berbagai aspek, di antaranya mudah dalam memodifikasi, baik perubahan data, penambahan data atau penghapusan data. Dalam hal ini IF bisa direpresentasikan sebagai gejala-gejala yang dirasakan saat mengendarai mobil dan Then berupa solusi-solusi yang dicapai [9].

Contoh pelacakan forward chaining:

Rule-rule yang diberikan :

R1 : Jika A dan C, maka E

R2 : Jika D dan C maka F

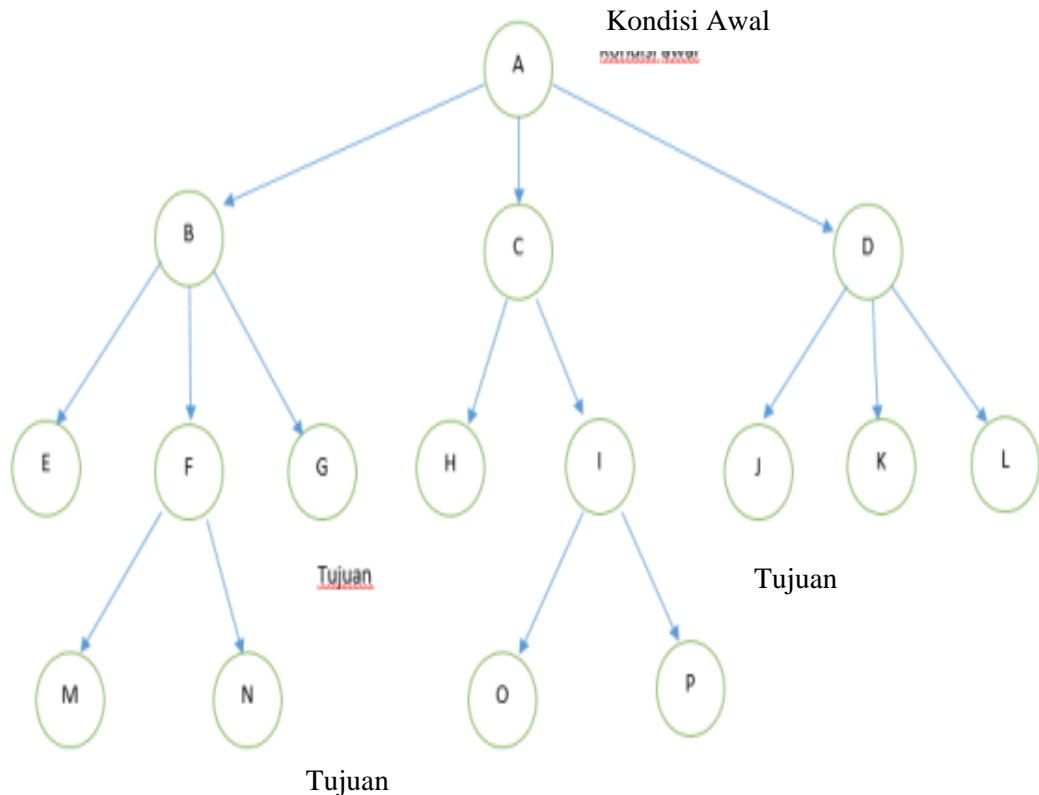
R3 : Jika B dan E maka F

R4 : Jika B maka C

R5 : Jika F maka G

Fakta yang ada: A benar dan B benar

Dalam Forward Chaining pencarian dimulai dengan fakta yang diketahui dan mengambil fakta baru menggunakan aturan yang telah diketahui pada sisi Jika. Karena diketahui A dan B benar, sistem pakar mulai dengan mengambil fakta baru menggunakan aturan yang memiliki A dan B pada sisi Jika. Dengan menggunakan R4, sistem pakar mengambil fakta baru C dan menambahkannya ke dalam assertion base sebagai benar. Sekarang R1 fire (karena A dan C benar) dan menyatakan E sebagai benar dalam assertion base sebagai benar. Karena B dan E keduanya benar (berada dalam assertion base), R3 fire dan menetapkan F sebagai benar dalam assertion base. Sekarang R5 fire (karena F berada dalam sisi Jika), yang menetapkan G sebagai benar, jadi hasilnya adalah G.



Gambar 2.2 Contoh Pelacakan *Forward Chaining*

2.1.4 Penyakit Gigi

Gigi merupakan organ pengunyah yang terdiri dari gigi-gigi pada rahang bawah, lidah dan saluransaluran penghasil air ludah. Fungsi utama dari gigi adalah merobek dan mengunyah makanan pada sistem pencernaan sehingga secara tidak langsung gigi berpengaruh pada kondisi kesehatan seseorang [10]. Sebuah gigi mempunyai mahkota, leher, dan akar. Mahkota gigi menjulang di atas gusi, lehernya dikelilingi gusi dan akarnya berada di bawahnya. Gigi dibuat dari bahan yang sangat keras, yaitu dentin. Di dalam pusat strukturnya terdapat rongga pulpa.

Manusia memiliki susunan gigi primer dan sekunder, yaitu [11]:

- a. Gigi primer, dimulai dari tuang diantara dua gigi depan yang terdiri dari 2 gigi seri, 1 taring, 3 geraham dan untuk total keseluruhan 20 gigi.
- b. Gigi sekunder, terdiri dari 2 gigi seri, 1 taring, 2 premolar dan 3 geraham untuk total keseluruhan 32 gigi.

Fungsi gigi adalah dalam proses matrikasi (pengunyah). Mengunyah ialah menggigit dan menggiling makanan di antara gigi atas dan bawah. Gerakan lidah dan pipi membantu dengan memindahmindahkan makanan lunak ke palatum keras ensit gigi-gigi.

Jenis-jenis penyakit gigi adalah

- a. Gingivitis merupakan suatu inflamasi yang mengenai jaringan gingiva. Pada gingivitis, gusi tampak eritematosa, edema, dan mungkin berdarah saat diperiksa. Gingivitis disebabkan oleh biofilm bakteri (plak gigi) yang terakumulasi pada gigi yang berdekatan dengan gingiva. Jika tidak segera diobati, maka akan menyebabkan periodontitis dan gigi goyah [12].








- b. Nekrosis Pulpa merupakan suatu kondisi medis dimana infeksi bakteri menyebabkan timbulnya peradangan pada gigi, yang menyebabkan jaringan hidup yang ada dalam pulpa gigi akhirnya mengalami kematian.
- c. Abses Periodontal merupakan infeksi yang terletak di sekitar poket periodontal dan dapat menyebabkan kerusakan ligamentum atau jaringan periodontal dan tulang rahang.
- d. Abses Peripikal merupakan pembentukan nanah atau pus diujung akar gigi yang merupakan kelanjutan dari infeksi pulpa gigi bermula dari karies atau lubang yang dalam yang nekrosis pulpa sehingga pulpa menjadi mati atau nekrosis.
- e. Abrasi Gigi merupakan keadaan abnormal dimana terdapat lapisan pada gigi yaitu email yang hilang dan terkikis, terkadang lapisan paling dalam yaitu dentin, penyebabnya yaitu salah cara menyikat.
- f. Kalkulus (Karang Gigi) merupakan kotoran dalam mulut yang menempel pada gusi dalam jangka waktu lama, sehingga lama-kelamaan akan mengeras dan membatu sehingga sulit dibersihkan dengan hanya menggosok gigi
- g. Bruxism gigi merupakan kebiasaan menggeretakkan dan menggesekkan gigi yang dilakukan secara tidak sadar. Kebiasaan ini bisa dialami oleh siapa saja, mulai dari anak-anak hingga dewasa. Jika kebiasaan ini tidak ditangani, penderita bruxism berpotensi mengalami kerusakan berat pada giginya [13].

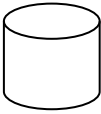
2.1.5 Alat Bantu Perancangan Sistem

2.1.5.1 Aliran Sistem Informasi (ASI)

Aliran sistem informasi sangat berguna untuk mengetahui permasalahan yang ada pada suatu sistem. Dari sini dapat diketahui apakah sistem informasi tersebut masih layak digunakan atau tidak, masih manual atau komputerisasi. Jika sistem informasinya tidak layak digunakan kembali maka perlu adanya perubahan dalam pengolahan datanya sehingga menghasilkan informasi yang cepat dan akurat serta keputusan yang lebih baik [14]. Berikut ini adalah gambar simbol-simbol dan keterangannya dari aliran sistem informasi (ASI):

Tabel 2.1 Simbol Aliran Sistem Informasi (ASI)

No	Nama	Simbol	Keterangan
1	Dokumen		Dokumen menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
2	Proses Manual		Proses manual menunjukkan pekerjaan manual.
3	Proses Komputerisasi		Proses menunjukkan kegiatan proses dari operasi program komputer.
4	Arsip		Arsip untuk menggambarkan penyimpanan data baik dalam bentuk arsip atau <i>file</i> komputer.
5	Penghubung		Penghubung menunjukkan penghubung pada halaman yang sama.
6	Aliran Sistem		Arus data menunjukkan arus dari proses.
7	Display		Penghubung menunjukkan penghubung pada halaman yang berbeda.

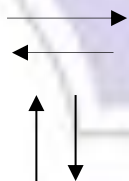


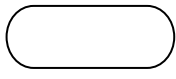
8	Basis Data		Basis data adalah simbol yang digunakan untuk basis data atau digunakan untuk melambangkan data yang disimpan dalam <i>hard drive</i> .
---	------------	---	---

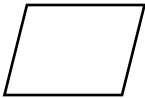

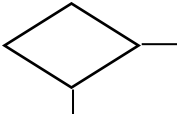

Sumber: Putra [15]

2.1.5.2 Flowchart

Flowchart adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman. Baik *flowchart* maupun algoritma bisa dibuat sebelum maupun setelah pembuatan program. *Flowchart* dan algoritma yang dibuat sebelum membuat program digunakan untuk mempermudah pembuat program untuk menentukan alur logika program, sedangkan yang dibuat setelah pembuatan program digunakan untuk menjelaskan alur program kepada orang lain [16]. Berikut ini adalah gambar simbol-simbol dan keterangannya dari *flowchart* yaitu:

Tabel 2.2 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		Arus (<i>Flow</i>)	Menyatakan arah sebuah proses. Symbol ini dapat mengarah ke kiri, kanan, atas dan bawah.
2		<i>On-page Connector</i>	Menunjukkan lanjutan proses ke symbol lain yang berada pada halaman yang sama.
3		<i>Off-Page Connector</i>	Mengarahkan sambungan proses ke symbol yang berada pada halaman berbeda.
4		Terminal	Menandakan permulaan dan berakhirnya suatu program.

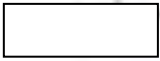
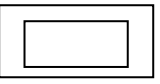
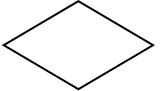
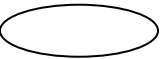
5		<i>Input/Output</i>	Mengindikasikan pernyataan membaca masukan dari <i>user</i> dan juga berfungsi sebagai simbol untuk menampilkan hasil dari proses tanpa menspesifikasikan alat.
6		Proses (<i>Process</i>)	Mempresentasikan tindakan yang dilakukan dalam sistem.
7		Keputusan (<i>Decision</i>)	Mengisyaratkan proses pilihan terhadap suatu kondisi yang mengarah kepada jawaban Ya/Benar atau Tidak/Salah.
8		<i>Display</i>	Menampilkan <i>output</i> ke layar.

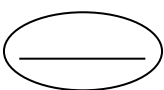

Sumber: Junus [17]

2.1.5.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan gambaran bentuk hubungan antara *file-file* yang ada, dimana *entity* adalah suatu kesatuan atau kesimpulan data yang memiliki karakteristik yang sama. *Entity* biasa berupa orang, tempat, benda, peristiwa atau konsep yang memberikan atau yang mengandung suatu informasi [18]. Berikut ini adalah simbol-simbol yang digunakan untuk membuat *Entity Relationship Diagram* yaitu:

Tabel 2.3 Simbol Entity Relationship Diagram

No	Simbol	Nama	Deskripsi
1		<i>Entity</i>	Objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data.
2		<i>Weak Entity</i>	Entitas yang keberadaannya sangat bergantung pada keberadaan entitas yang lainnya.
3		<i>Relationship</i>	Hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.
5		<i>Attribute</i>	Atribut adalah segala sesuatu yang ingin diketahui dari suatu <i>entity</i> .

6		<i>Attribute Primary Key</i>	Kunci attribute yang uni yang membedakan antara <i>record</i> yang satu dengan <i>record</i> yang lain.
7		<i>Attribute Multivalued</i>	Nilai dari suatu attribute yang mempunyai lebih dari satu.

Sumber: Herlina [19]

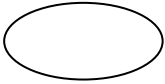


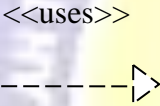
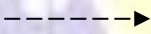

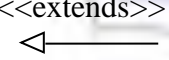
2.1.5.4 Unified Modeling Language (UML)

UML merupakan singkatan dari “*Unified Modeling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software [20]. Diagram UML yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Use Case Diagram*

Use case adalah deskripsi fungsi dari sebuah sistem dari sudut pandang para pengguna sistem. *Use case* mendefinisikan apa yang dilakukan oleh sistem dan elemen-elemennya, bukan bagaimana sistem dan elemen-elemennya saling berinteraksi. *Use case* bekerja dengan menggunakan *scenario* yaitu deskripsi urutan langkah yang menerangkan apa yang dilakukan penggunaan terhadap sistem maupun sebaliknya. *Use case diagram* mengidentifikasi fungsionalitas yang dimiliki oleh sistem (*use case*), *user* yang berinteraksi dengan sistem (*actor*) dan asosiasi atau keterhubungan antara *user* dengan fungsionalitas sistem [21]. Berikut ini adalah gambar simbol-simbol dan keterangannya dari *use case diagram*:

Tabel 2.4 Simbol *Use Case Diagram*


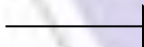
No	Simbol	Nama	Keterangan
1		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya digunakan menggunakan kata benda di awal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Include</i>	Relasi <i>use case</i> tambahan ke seluruh <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ni untuk menjalankan fungsi.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
6		<i>Generalization</i>	Hubungan generalisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
7		<i>Extends</i>	Relasi <i>use case</i> tambahan ke seluruh <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal arah panah mengarah pada <i>use case</i> yang ditambahkan.

Sumber: Mesran, dkk [22]

2. Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungannya satu sama lain seperti *containment*, pewaris, asosiasi dan lain-lain [21]. Berikut ini adalah gambar simbol-simbol dan keterangannya dari *class diagram*:

Tabel 2.5 Simbol Class Diagram

No	Simbol	Nama	Keterangan			
1	<table border="1"><tr><td>Nama Kelas</td></tr><tr><td>+atribut</td></tr><tr><td>+operasi()</td></tr></table>	Nama Kelas	+atribut	+operasi()	Kelas (<i>Class</i>)	Terdiri dari 3 (tiga) bagian yaitu nama <i>class</i> , atribut dan operasi(). Atribut ditaruh dibaris kedua. Ini dapat digunakan untuk mendeskripsikan daftar variabel. Operasi ditaruh dibaris ketiga. Dapat digunakan untuk mendaftarkan seluruh fungsi dan prosedur yang dipakai.
Nama Kelas						
+atribut						
+operasi()						
2		Asosiasi (<i>Association</i>)	Menghubungkan antara <i>class</i> yang satu dengan yang lain.			
3		Generalisasi (<i>Generalisation</i>)	Menghubungkan beberapa <i>class</i> yang khusus ke sebuah <i>class</i> yang lebih umum.			


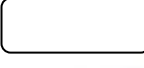

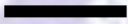



Sumber: Sutanto [23]

3. Activity Diagram

Activity diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang dimulai dari pandangan *business level* hingga ke *operational level*. Pada dasarnya, *activity diagram* merupakan variasi dari *statechart diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi

perbedaannya dengan *flowchart* adalah *activity diagram* bias mendukung perilaku parallel sedangkan *flowchart* tidak bisa [21]. Berikut ini adalah gambar simbol-simbol dan keterangannya dari *activity diagram*:

Tabel 2.6 Simbol Activity Diagram



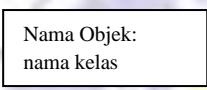

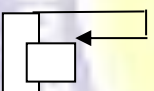
No	Simbol	Nama	Keterangan
1		Status Awal (<i>Start State</i>)	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas (<i>Activity</i>)	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		Percabangan (<i>Decision</i>)	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		Penggabungan (<i>Join</i>)	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		<i>Interaction</i>	Merupakan alur.
6		(Status Akhir) <i>End State</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
7		Partisi Peran (<i>Swimlane</i>)	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Mesran, dkk [22]

4. Sequence Diagram

Sequence diagram mendokumentasikan komunikasi atau interaksi antar *class*. Diagram ini menunjukkan sejumlah objek dan *message* (pesan) yang diletakkan di antara objek-objek di dalam *use case*. Perlu diingat bahwa di dalam diagram ini, kelas dan aktor diletakkan dibagian atas diagram dengan urutan dari kiri ke kanan dengan garis yang diletakkan secara vertical terhadap kelas dan aktor [21]. Berikut ini adalah gambar simbol-simbol dan keterangannya dari *sequence diagram*:

Tabel 2.7 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi, walaupun symbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda di <i>wal frase</i> nama aktor.
2		Garis Hidup/ <i>Life line</i>	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek berinteraksi pesan.
4		<i>Message Tipe Create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
5		<i>Message Tipe Call</i>	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.


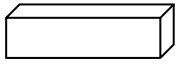


Sumber: Aldo dan Putra [24]

5. *Deployment Diagram*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* dapat digunakan pada bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur

fisik sebuah sistem [21]. Berikut ini adalah gambar simbol-simbol dan keterangannya *deployment diagram*:

Tabel 2.8 Simbol *Deployment Diagram*

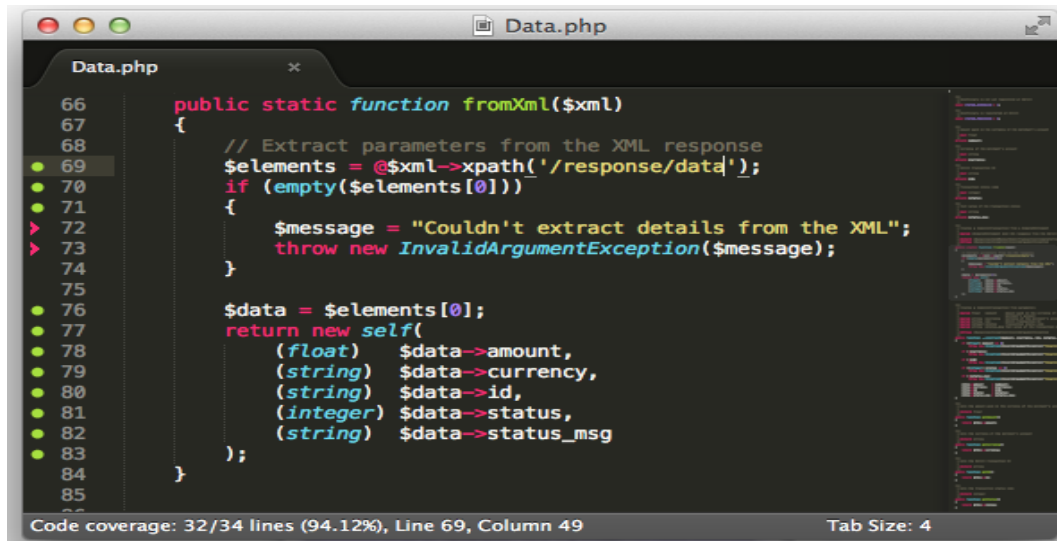
No	Simbol	Nama	Keterangan
1		<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih node.
2		<i>Node</i>	Biasanya mengacu pada perangkat keras, perangkat lunak, jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya.
3		<i>Dependency</i> / Kebergantungan	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4		<i>Link</i>	Relasi antar <i>node</i> .

Sumber: Aldo dan Putra [24]

2.1.6 Perangkat Lunak yang Digunakan

2.1.6.1 PHP

PHP adalah kependekan dari PHP: *Hypertext Preprocessor*. Sedangkan pengertian PHP adalah bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan [25].



```
66 public static function fromXml($xml)
67 {
68     // Extract parameters from the XML response
69     $elements = @$xml->xpath('/response/data');
70     if (empty($elements[0]))
71     {
72         $message = "Couldn't extract details from the XML";
73         throw new InvalidArgumentException($message);
74     }
75
76     $data = $elements[0];
77     return new self(
78         (float) $data->amount,
79         (string) $data->currency,
80         (string) $data->id,
81         (integer) $data->status,
82         (string) $data->status_msg
83     );
84 }
85
```

Code coverage: 32/34 lines (94.12%), Line 69, Column 49 Tab Size: 4

Gambar 2.3 PHP

2.1.6.2 Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Sublime Text mendukung berbagai Bahasa pemrograman dan mampu menyajikan fitur syntax highlight hamper semua Bahasa pemrograman yang di dukung ataupun dikembangkan oleh komunitas seperti C, C++, C=, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Phyton, R, Ruby, SQL, TCL, Textile, and XML [26].

Berikut beberapa fitur yang diunggulkan dari aplikasi *Sublime Text*:

1. *Goto Anything*

Fitur yang sangat membantu dalam membuka *file* ataupun menjelajahi isi dari *file* hanya dengan beberapa *keystrokes*.

2. *Multiple Selections*

Fitur ini memungkinkan *user* untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi *file* lebih cepat dari sebelumnya.

3. *Command Pallete*

Dengan hanya beberapa *keystorkes*, *user* dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

4. *Distraction Free Mode*

Bila *user* memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu *user* dengan memberikan tampilan layar penuh. *Split Editing* dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan *editing* perpecahan. Mengedit sisi *file* dengan sisi, atau mengedit dua lokasi di satu *file*. Anda dapat mengedit dengan banyak baris dan kolom yang *user* inginkan.

5. *Instant Project Switch*

Menangkap semua *file* yang dimasukkan kedalam *project* pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua *file* yang ada ataupun untuk beralih ke *file* dalam *project* lainnya dengan cepat.

6. *Plugin API*

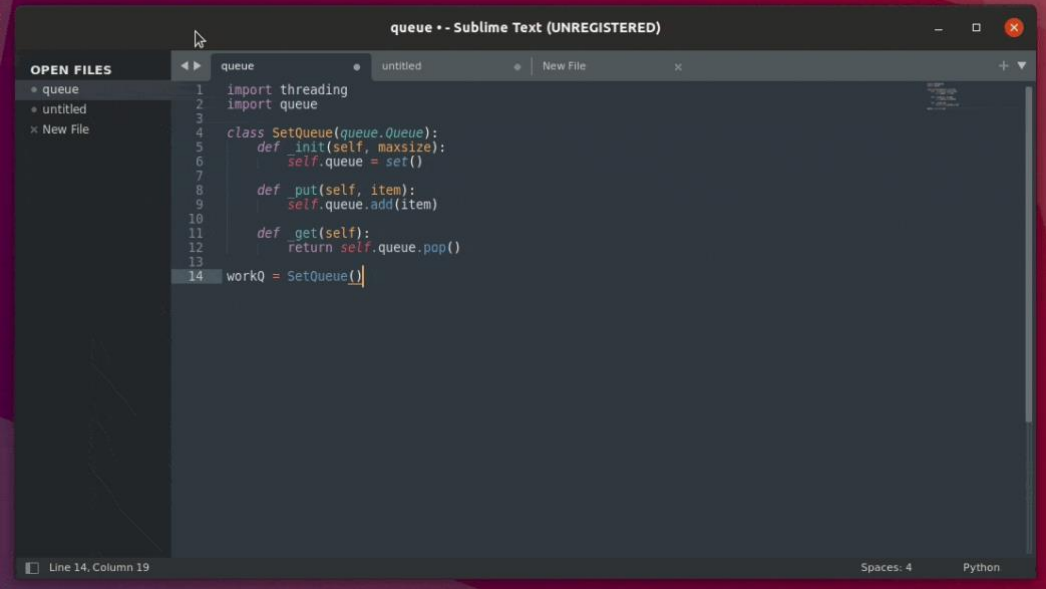
Dilengkapi dengan *plugin* API berbasis *Phyton* sehingga membuat aplikasi ini sangat tangguh.

7. *Customize Anything*

Aplikasi ini memberikan *user* fleksibilitas dalam hal pengaturan fungsional dalam aplkasi ini.

8. Cross Platform

Aplikasi ini dapat berjalan hampir disemua *operating system modern* seperti *Windows, OS X, dan Linux based operating system.*



The image shows a screenshot of the Sublime Text editor window titled "queue -- Sublime Text (UNREGISTERED)". The editor displays Python code for a class named `SetQueue` that inherits from `queue.Queue`. The code includes imports for `threading` and `queue`, and defines methods for `__init__`, `put`, and `get`. A variable `workQ` is instantiated at the bottom of the code block.

```
1 import threading
2 import queue
3
4 class SetQueue(queue.Queue):
5     def __init__(self, maxsize):
6         self.queue = set()
7
8     def put(self, item):
9         self.queue.add(item)
10
11     def get(self):
12         return self.queue.pop()
13
14 workQ = SetQueue()
```

Gambar 2.4 Sublime Text

2.1.6.3 XAMPP

XAMPP adalah kepanjangan dari X = lintas sistem operasi, A = Apache HTTP server, M = MySQL Database, PP = PHP dan Perl. XAMPP merupakan paket PHP berbasis Open Source yang dikembangkan oleh sebuah komunitas Open Source. Informasinya dapat diperoleh pada alamat <http://www.apachefriends.org>. Untuk paket dengan versi Windows, anda dapat memperolehnya dengan cara mengunduh langsung pada alamat website [27].

XAMPP memiliki kelebihan untuk bisa berperan sebagai *server web apache* untuk simulasi pengembangan *website*. *Tool* pengembang ini mendukung teknologi *web* populer seperti PHP, MySQL dan PERL. Melalui program ini, *programmer web* dapat menguji aplikasi *web* yang dikembangkan dan

mempresentasikan ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP dilengkapi fitur manajemen *database* PhpMyAdmin seperti pada *server hosting*, sehingga pengembang *web* dapat diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman web. XAMPP merupakan software gratis dan dapat berjalan di sistem operasi windows 2000/XP/Vista/7/8/10 dan sistem operasi lain [28].



Gambar 2.5 XAMPP

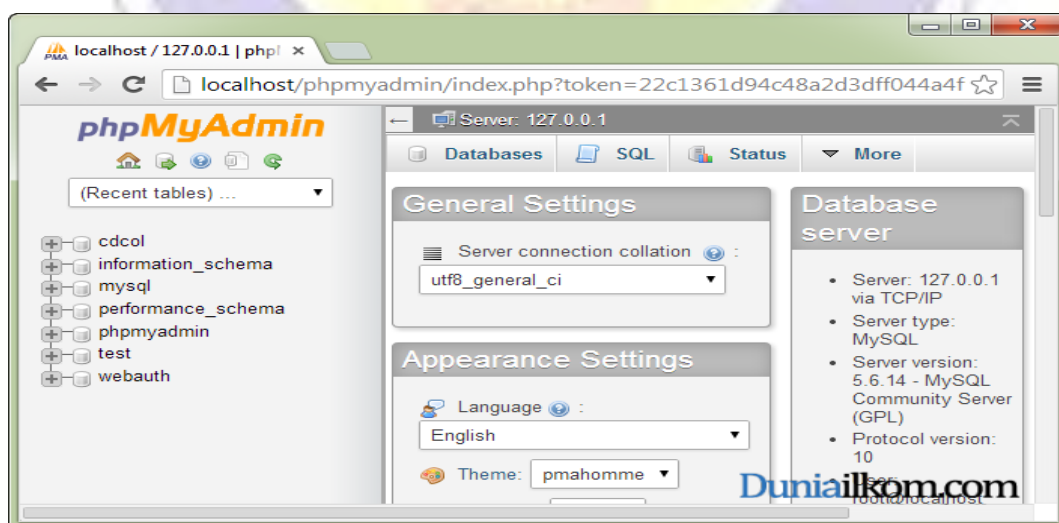
2.1.6.4 MySQL

Database merupakan media yang digunakan untuk menampung data. Ada beberapa macam database, antara lain *Oracle*, *Microsoft Access*, *Microsoft SQL Server*, *MySQL* dan lain-lain. *MySQL* adalah sebuah perangkat lunak sistem manajemen biasa data SQL (bahasa inggris: *database management system*) atau BDMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU/*General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaanya tidak cocok dengan pengguna GPL [29].

Fitur-fitur PHPMyAdmin adalah sebagai berikut [30]:

1. Antarmuka berbasis Web.
2. Dukungan banyak fitur MySQL seperti:
 - a. Menelusuri dan drop basis data (database), tabel, pandangan (view), bidang (fields), dan index.
 - b. Membuat, menyalin, drop, dan mengubah nama basis data, tabel, kolom.
 - c. Pemeliharaan server, basis data dan table, dengan server konfigurasi.
 - d. Melaksanakan, meng edit, dan petunjuk pernyataan-SQL, bahkan batchqueries.
 - e. Mengelola penggunaan MySQL bahkan dengan hak istimewa.
 - f. Mengelola prosedur penyimpanan.
 - g. Import data dari CSV dan SQL.
 - h. Export data ke berbagai format : CSV, SQL, XML, PDF, ISO/IEC 26300, OpenDocument text dan Spreadsheet, Word, Excel, LATEKS.
 - i. Membuat grafik PDF dari tampilan basis data anda.

Maka akan tampil halaman phpMyAdmin seperti gambar berikut:



Gambar 2.6 PHPMyAdmin

2.1.6.5 Web Browser

Web browser merupakan aplikasi yang bisa menjelajahi, menyajikan maupun mengambil konten yang ada di berbagai sumber informasi pada jaringan internet atau WWW. Pengertian dari *web browser* juga sering disebut dengan suatu perangkat lunak dengan fungsi yang dimilikinya sebagai penerima, pengakses, penyaji berbagai informasi di internet [31].

Web browser adalah aplikasi untuk mengakses *website*. *Web browser* berbeda dengan *search engine*, untuk mengakses *website* melalui *web browser* maka harus menulis alamat lengkap *website* misalnya www.namasitus.com. *Google chrome* menjadi *web browser* pilihan utama 70% pengguna internet desktop dan laptop. Contoh aplikasi *web browser* adalah *google chrome*, *Mozilla firefox*, *opera*, *safari*, *internet explorer* dan lainnya [32].



Gambar 2.7 Jenis-Jenis Web Browser

2.2 Tinjauan Penelitian

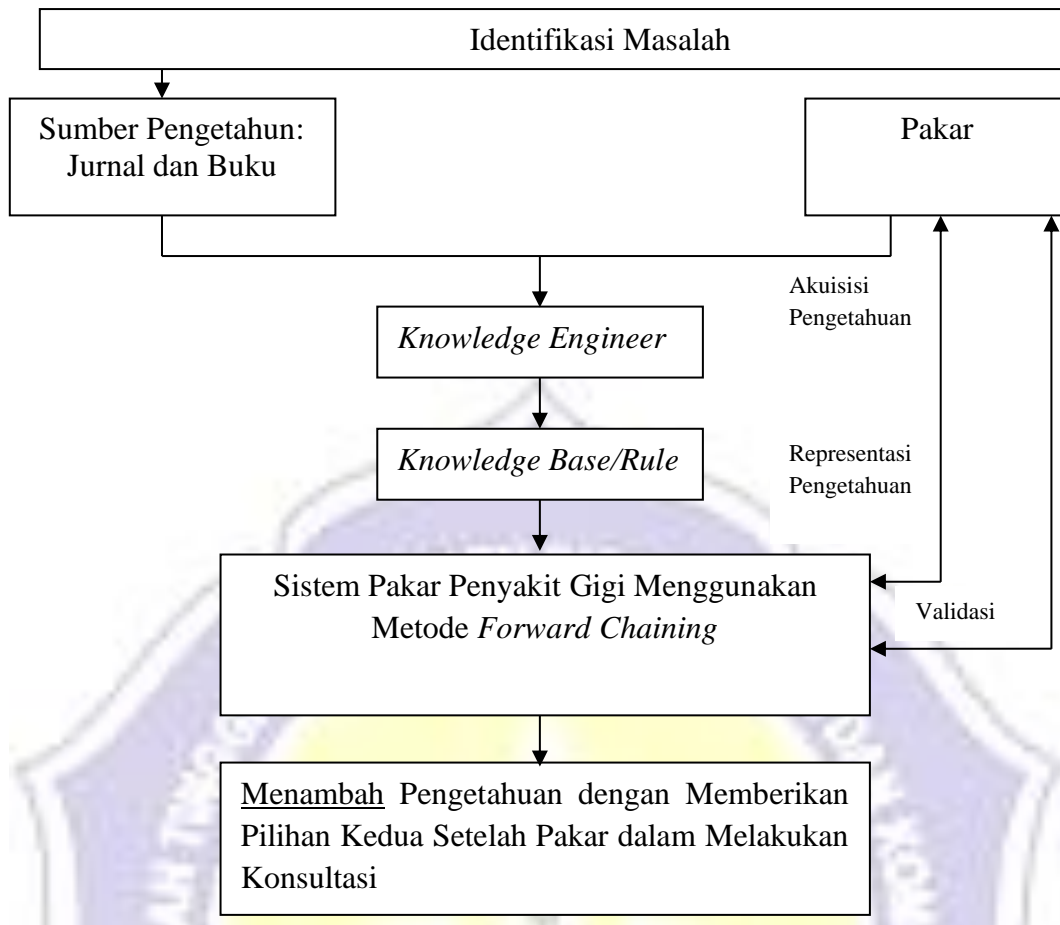
Berikut beberapa penelitian yang relevan dengan skripsi ini adalah:

1. Darsin, Dody Ahmad Kurniawan, dan Mira Febriana Sesunan (2022) melakukan penelitian dengan judul “Sistem Pakar untuk Mendiagnosa Penyakit Gigi dan Mulut Menggunakan Metode *Forward Chaining* (Studi Kasus di RSUD Menggala). Hasilnya adalah berhasil dirancang suatu sistem pakar untuk mendiagnosa penyakit gigi dan mulut dengan penerapan metode *forward chaining*. Aplikasi sistem pakar ini digunakan untuk membantu memudahkan asisten dokter dalam mendiagnosa pasien sehingga membantu dokter untuk mengambil keputusan [33].
2. Heru Purwanto dan Ratnasih (2022) melakukan penelitian dengan judul “Implementasi Metode *Forward Chaining* pada Sistem Pakar untuk Diagnosa Penyakit Gigi Berlubang”. Hasil diagnosa dan saran yang dihasilkan oleh Aplikasi sistem pakar, dapat dijadikan alternatif dalam penanganan dan pencegahan penyakit gigi. [34].
3. Muhammad Rizki Hidayat dan Andry Maulana (2022) melakukan penelitian dengan judul “Sistem Pakar Berbasis Android untuk Diagnosa Penyakit Gigi dengan Metode *Forward Chaining*”. Aplikasi sistem pakar untuk mendiagnosa penyakit gigi dibuat dengan menerapkan pengetahuan dokter gigi. Dengan menggunakan metode *forward chaining* pembuatan aplikasi ini dapat menghasilkan diagnosa yang akurat sehingga diharapkan dapat membuahkan solusi berupa informasi yang tepat sebagai langkah awal untuk menangani penyakit yang dialami oleh pasien [35].

4. Wahyuni Fithratul Zalmi (2023) melakukan penelitian dengan judul “Sistem Pakar Pendeteksi Penyakit Gigi dan Mulut dengan Menggunakan Metode *Forward Chaining*”. Aplikasi ini memudahkan masyarakat umum untuk mendiagnosa lebih dini jenis penyakit pada gigi dan mulut dimana saja sehingga penanganan lebih lanjut terhadap penyakit tersebut dapat dengan cepat dilakukan [36].
5. Yolanda Anggraini, Mohamad Indra, M. Khoirusofi, Ibnu Nur Azis, dan Perani Rosyani (2023) melakukan penelitian dengan judul “Systematic Literature Review: Sistem Pakar Diagnosa Penyakit Gigi Menggunakan Metode *Forward Chaining*”. Faktor yang mempengaruhi keberhasilan penerapan sistem pakar diagnosa penyakit gigi adalah kualitas data yang memadai serta pengembangan dan pemeliharaan sistem. Sistem pakar diagnosa penyakit gigi menggunakan metode forward chaining sangat berperan penting untuk dijadikan alternatif pilihan yang dapat digunakan untuk mengatasi kendala dalam pendiagnosaan penyakit gigi [37].

2.3 Kerangka Pemikiran

Kerangka pemikiran ini merupakan langkah-langkah yang akan dilakukan dalam penyelesaian masalah yang akan dibahas. Demi menjelaskan dari proses penelitian yang akan dilakukan dengan judul “**Sistem Pakar Penyakit Gigi pada Klinik Drg. Nurmuhayanah, Sp. Ort metode FC**” berdasarkan teori sistem pakar, maka dijelaskan dalam kerangka teori berikut ini:

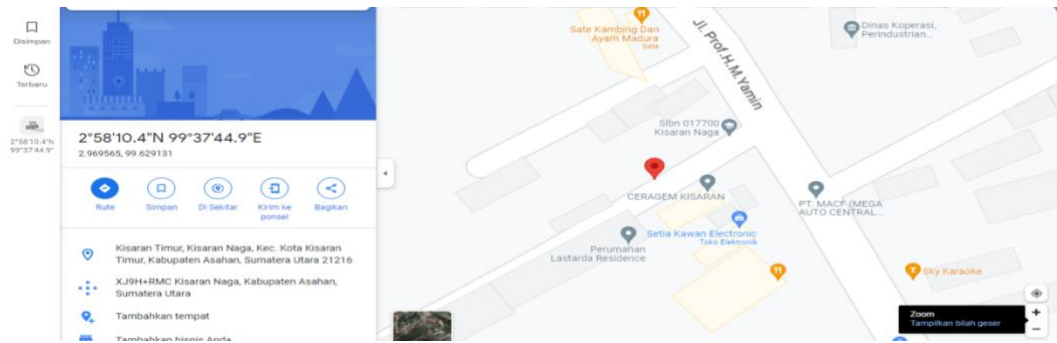


Gambar 2.8 Kerangka Pemikiran

2.4 Tinjauan Umum Perusahaan

2.4.1 Klinik drg. Nurmuhayanah, Sp. Ort

Klinik drg. Nurmuhayanah, Sp. Ort merupakan usaha yang bergerak dibidang jasa pelayanan kesehatan khusus perawatan gigi dan mulut yang sudah berdiri sejak tahun 2018. Klinik drg. Nurmuhayanah, Sp. Ort beralamat di Jalan Prof. H. M. Yamin, No. 89 C, Kelurahan Kisaran Naga, Kecamatan Kota Kisaran Timur, Kabupaten Asahan. Berikut ini adalah gambar lokasi Klinik drg. Nurmuhayanah, Sp. Ort yang dilihat dari *Google Maps*:



Gambar 2.9 Lokasi Klinik drg. Nurmuhayanah, Sp. Ort



Gambar 2.10 Klinik drg. Nurmuhayanah, Sp. Ort

2.4.2 Visi dan Misi Klinik drg. Nurmuhayanah, Sp. Ort

Adapun visi dan misi dari Klinik drg. Nurmuhayanah, Sp. Ort adalah:

1. Visi

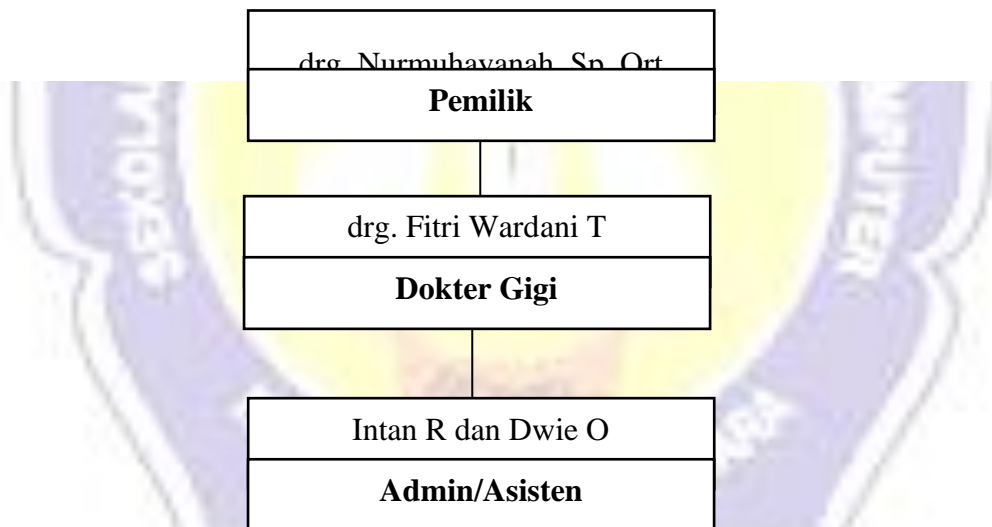
Melaksanakan pelayanan kesehatan gigi yang berfokus pada bidang estetik dan pelayanan secara holistik untuk pasien dan keluarga.

2. Misi

- a. Menyediakan layanan yang personal sesuai dengan kebutuhan dan kondisi pasien dengan tenaga medis yang professional.
- b. Memberikan perawatan dengan fokus keselamatan, kenyamanan dan kepuasan pasien.
- c. Menyediakan layanan kesehatan rongga mulut secara holistik berorientasi pada kebutuhan pasien dan keluarga pasien.

2.4.3 Struktur Organisasi Klinik drg. Nurmuhayanah, Sp. Ort

Berikut ini adalah gambar struktur organisasi dari Klinik drg. Nurmuhayanah, Sp. Ort adalah:



Gambar 2.11 Struktur Organisasi drg. Nurmuhayanah, Sp. Ort

2.4.4 Uraian Tugas Struktur Organisasi Klinik drg. Nurmuhayanah, Sp. Ort

Berikut ini adalah uraian tugas dari struktur organisasi dari Klinik drg. Nurmuhayanah, Sp. Ort:

1. Pemilik
 - a. Memantau perkembangan klinik.
 - b. Bertanggung jawab atas kegiatan yang ada di klinik.
 - c. Dalam memberikan konsultasi harus professional sesuai standar.
2. Dokter Gigi
 - a. Mendiagnosa penyakit pasien dan memberikan terapi.
 - b. Melakukan tindakan medik gigi dan mulut umum
 - c. Melakukan tindakan darurat medik gigi dan mulut.
3. Admin/Asisten
 - a. Mengevaluasi tugas administrasi dan pelayanan kesehatan.
 - b. Mendampingi tugas dokter saat praktek.
 - c. Menjaga kebersihan klinik.

2.5 Hipotesis

Hipotesis merupakan jawaban sementara terhadap rumusan masalah yang harus diuji kebenarannya. Adapun hipotesis dalam penelitian ini adalah sistem pakar mendiagnosa penyakit gigi dengan menggunakan metode *Forward Chaining* berbasis *web* dapat memberikan informasi mengenai gejala serta solusi dari penyakit yang diderita oleh pasien dan dapat digunakan kapanpun serta tidak membutuhkan waktu yang lama dalam mendiagnosanya.

