

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Teori dasar merupakan kumpulan ide juga konsep yang digunakan untuk menjelaskan fenomena sosial dengan menghubungkan berbagai fakta secara teratur dengan tujuan membuat fakta-fakta tersebut lebih mudah di pahami (Madrallis, 2023). Pada penelitian ini, teori dasar mencakup konsep Sistem Pakar, *Forward Chaining*, UML, database, gangguan pencernaan pada anak, dan *tools* yang digunakan.

2.1.1 Expert System

Expert System merupakan program komputer yang di rancang untuk meniru cara berpikir dan pengambilan keputusan seorang ahli manusia dalam menyelesaikan masalah pada domain tertentu. Sistem ini memungkinkan pengguna profesional dan umum mendapatkan solusi atas suatu masalah berdasarkan pengetahuan dan pengalaman yang telah ditanamkan dalam sistem tersebut oleh para pakar (Miftaviana, 2022). *Expert system* merupakan bagian dari cabang ilmu *Artificial Intelligence* yang mulai dikembangkan sejak tahun 1960-an hingga 1970-an, *Expert System* mengalami perkembangan cukup signifikan pada era 90-an yang ditunjukkan dengan semakin luasnya implementasi pada beberapa bidang secara komersil, khususnya pada bidang kedokteran, teknik dan ekonomi (Kesumaningtyas & Handayani, 2020).

Tujuan utama dari *Expert System* bukanlah untuk menggantikan peran seorang ahli, tetapi untuk mendukung dan membantu proses pengambilan keputusan, memberikan saran, atau melakukan diagnosis awal berdasarkan pengetahuan (Kesumaningtyas & Handayani, 2020).

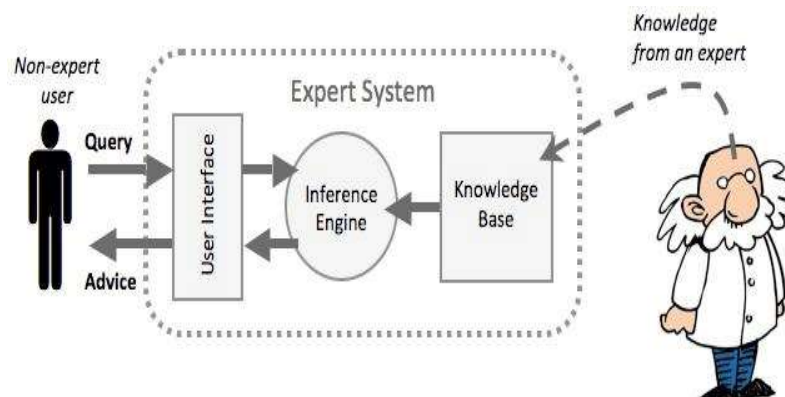
Sistem pakar memiliki sejumlah keunggulan yang menjadikannya bermanfaat dalam berbagai bidang. Sistem ini mampu meniru proses berfikir dan pengambilan keputusan layaknya seorang pakar manusia, serta dapat di akses kapan saja, bahkan ketika pakar tidak tersedia. Selain itu, sistem pakar mampu memberikan kesimpulan meskipun data yang di berikan tidak lengkap, dengan penjelasan logis yang menyertainya. Keunggulan lainnya termasuk peningkatan efisiensi kerja, percepatan proses analisis, serta kemampuan untuk digunakan secara otomatis dalam proses yang berulang seperti pelatihan dan pengambilan keputusan. Sistem pakar juga memberikan keputusan berbasis aturan logis yang konsisten (Maukar et al., 2023).

Namun demikian, sistem juga memiliki keterbatasan. Sistem pakar hanya berfungsi sebagai alat bantu awal dan tidak dapat menggantikan pemeriksaan fisik secara langsung. Akurasi hasil sangat bergantung pada kualitas data yang dimasukkan, dan sistem hanya efektif untuk keahlian yang telah dirancang secara spesifik. Selain itu, pengembangan dan pemeliharaan sistem pakar membutuhkan waktu, biaya, dan keterlibatan pakar secara intensif. Dalam bidang kesehatan khususnya, basis pengetahuan perlu di perbarui secara berkala agar tetap relevan. Meskipun sistem dapat menangani data yang tidak lengkap, tetap ada keterbatasan

dalam mengelola ketidakpastian ekstrem atau kasus baru yang belum terwakili dalam basis pengetahuan (Raenida & Zukhri, 2019).

Untuk memahami bagaimana sistem pakar bekerja dan memberikan solusi layaknya seorang ahli, perlu di pahami beberapa aspek fundamental yang membentuk dan mendukung sistem ini. Adapun aspek-aspek tersebut meliputi :

a. Komponen Utama *Expert System*



Gambar 2. 1 Komponen *Expert System*
Sumber : (Huang et al., 2021)

Menurut (Huang et al., 2021), *Expert System* memiliki dari beberapa komponen utama yang saling berintegrasi, yaitu :

a) *Knowledge Base*

Sebuah basis pengetahuan yang terdiri dari representasi informasi secara deklaratif, yang dapat di wujudkan dalam bentuk aturan *if-then*, jaringan semantik, atau representasi berbasis frame. Komponen ini memuat data mengenai jenis penyakit, gejala, metode, pengobatan, serta struktur hierarkis antar penyakit. Dalam sistem pakar ini, basis pengetahuan berfungsi sebagai

sumber informasi utama yang digunakan oleh mesin inferensi dalam proses penentuan diagnosis.

b) Inference Engine

Sering di kenal dengan sebutan mesin inferensi, ini merupakan bagian inti dari sistem yang menjalankan proses penalaran berdasarkan data dalam basis pengetahuan dan input pengguna yang tersimpan dalam memori kerja. Sistem ini menerapkan pendekatan untuk menarik kesimpulan dari gejala yang diberikan pengguna. Selain itu, mesin ini juga dapat menghasilkan berupa pohon keputusan dan menunjukkan tingkat keyakinan dari hasil diagnosis yang diberikan.

c) User Interface

Dikenal dengan antarmuka pengguna yang dirancang secara visual untuk memudahkan interaksi antara sistem dan pengguna. Melalui antarmuka ini, pengguna dapat melihat proses pengambilan keputusan, seperti pohon diagnosis dan tingkat kepercayaan, sehingga membantu dalam memahami alur penalaran yang dilakukan sistem. Pengguna juga dapat memasukkan gejala yang dialami dan memperoleh hasil diagnosis dengan cara yang interaktif dan mudah dipahami.

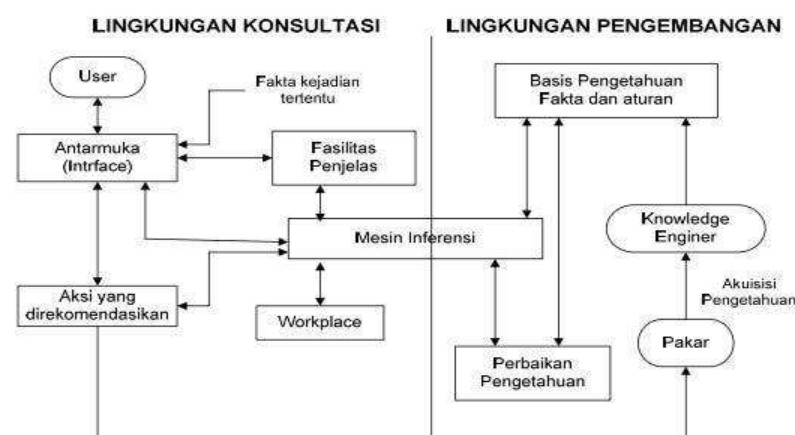
Sistem pakar yang di implementasikan untuk diagnosis penyakit, maka komponen-komponen sistem pakar saling berintegrasi untuk menerima output juga input melalui antar muka yang kemudian di proses *inference system* sehingga hasil diagnosis dapat di tampilkan kembali secara inovatif dan

interaktif. Integrasi ketiga komponen memungkinkan sistem pakar meniru berfikir seorang pakar medis, sehingga dapat memberikan diagnosis dan rekomendasi secara otomatis dan efisien (Huang et al., 2021).

b. *Arsitektur Expert System*

konsultasi di rancang untuk di gunakan oleh pengguna non-ahli dalam memperoleh informasi atau solusi dari sistem. Proses kerja sistem mengikuti siklus “pengamatan hingga tindakan”, di mana sistem akan menelusuri aturan aturan yang sesuai dari memori kerja, memilih salah satu aturan yang paling relevan, arsitektur sistem pakar memiliki 2 bagian utama, yaitu lingkungan pengembang dan lingkungan konsultasi. Lingkungan pengembangan berfungsi untuk mengintegrasikan pengetahuan yang di peroleh dari pakar ke dalam sistem, sedangkan lingkungan konsultasi melaksanakan tindakan berdasarkan strategi penyelesaian konflik yang telah di tentukan (Revaldo et al., 2023).

Berikut gambar yang akan memperlihatkan arsitektur *Expert System* secara umum yang dibagi menjadi 2 bagian utama, yaitu :



Gambar 2. 2 Arsitektur *Expert System*

Sumber : (Revaldo et al., 2023)

1. Lingkungan Konsultasi

Bagian ini digunakan oleh pengguna, terutama yang bukan pakar, untuk berinteraksi langsung dengan sistem guna mendapatkan solusi atau rekomendasi.

- a) Pengguna : Pengguna memasukkan data atau pertanyaan (misalnya gejala) melalui antarmuka yang disediakan.
- b) Fakta kejadian tertentu : Data atau fakta yang di masukkan oleh pengguna yang berguna sebagai informasi awal yang akan di proses.
- c) Mesin inferensi (*inference engine*) : Komponen inti yang melakukan proses penalaran. Mesin ini mencari aturan yang sesuai dari basis pengetahuan berdasarkan fakta yang tersedia, lalu menentukan solusi atau diagnosis.
- d) *Workplace* (memori kerja) : Bagian yang sementara untuk menyimpan fakta dan hasil penelusuran selama proses penalaran berlangsung.
- e) Fasilitas penjelasan (*Explanation Facility*) : Menyediakan penjelasan kepada pengguna mengenai bagaimana sistem sampai pada kesimpulan tertentu.

2. Lingkungan pengembangan

Bagian ini ditujukan untuk membangun dan memperbarui pengetahuan dalam sistem pakar.

- a) Pakar (*Expert*) : Sumber utama pengetahuan yang akan di masukkan ke dalam sistem.
- b) *Knowledge Engineer* : Pakar yang bertugas mengumpulkan, merumusan dan menerjemahkan pengetahuan dari pakar ke dalam bentuk yang bisa di

pahami oleh komputer yang ditulis dengan struktur *if-then*.

- c) Akuisisi pengetahuan : Proses perolehan pengetahuan dari pakar oleh *knowledge engineer*.
- d) Basis pengetahuan (*Knowledge Base*) : menyimpan semua pengetahuan dalam bentuk aturan dan fakta yang digunakan oleh mesin inferensi untuk menyelesaikan masalah.
- e) Perbaikan pengetahuan : proses pembaruan terhadap basis pengetahuan apabila ada informasi baru, perubahan aturan atau koreksi dari pakar.

Secara keseluruhan, gambar tersebut menjelaskan bagaimana sistem pakar bekerja :

- a) Di satu sisi, pengguna berkonsultasi dengan sistem melalui antarmuka untuk mendapatkan solusi.
- b) Di sisi lain, pakar dan *knowledge engineer* terus mengembangkan dan menyempurnakan basis pengetahuan agar sistem tetap akurat dan relevan.

Intinya, kedua lingkungan ini saling berhubungan dan menjadi fondasi dari sistem pakar yang cerdas dan adaptif.

c. Mekanisme Kerja Sistem Pakar

Alur kerja sistem pakar di mulai dari tahap akuisisi pengetahuan, dimana proses pengambilan informasi dari pakar atau sumber lainnya yang kemudian dikonversi dan disimpan dalam basis pengetahuan. Setelah itu, sistem menjalankan siklus pengenalan dan tindakan, yakni dengan menelusuri aturan-

aturan yang relevan dari memori kerja, memilih satu aturan menggunakan strategi tertentu untuk mengatasi konflik, lalu mengeksekusi tindakan yang sesuai. Dalam hal ini, mesin inferensi berperan sebagai pusat pemrosesan utama yang melakukan penalaran dan pengambilan keputusan berdasarkan kumpulan aturan dan fakta tersedia. Sistem juga dilengkapi dengan fitur yang jelas untuk memberikan informasi kepada pengguna terkait alasan di balik keputusan yang dihasilkan (Revaldo et al., 2023).

Mekanisme kerja sistem pakar dimulai dari *input* fakta atau data oleh pengguna. Selanjutnya, sistem menggunakan metode inferensi seperti *Forward Chaining* untuk mencocokkan fakta tersebut dengan aturan *if-then* yang ada di basis pengetahuan. Jika ditemukan kecocokan, sistem akan menjalankan aturan tersebut dan menambahkan informasi baru ke dalam basis data. Proses ini berlangsung berulang hingga tidak ada lagi aturan yang bisa dijalankan atau hingga sistem mencapai kesimpulan akhir (Purnama et al., 2022).

d. Penerapan *Expert System*

Sistem pakar telah banyak diterapkan dalam berbagai bidang untuk membantu pengambilan keputusan secara otomatis berdasarkan pengetahuan pakar. Pemanfaatan sistem pakar dalam bidang kesehatan, khususnya untuk diagnosis penyakit pada anak, berhasil dibuat dalam bentuk aplikasi interaktif berbasis web. Sistem ini menggunakan metode *Certainty Factor* (CF) sebagai dasar inferensi, dan pengetahuan yang dimasukkan ke dalam sistem berasal dari dokter spesialis anak sebagai pakar. Setiap gejala diberikan nilai CF dalam rentang 0 hingga 1, sehingga memudahkan pengguna dalam memilih gejala

yang sesuai dengan kondisi anak. Hasil diagnosis akan ditampilkan berdasarkan tingkat kepercayaan secara menurun. Selain itu, sistem ini memungkinkan pembaruan basis pengetahuan secara berkala, sesuai dengan perkembangan informasi medis (Maulina, 2020).

Pemanfaatan sistem pakar dalam bidang teknologi telah di terapkan untuk mendeteksi kerusakan ada perangkat laptop. Sistem ini di kembangkan sebagai aplikasi berbasis web yang menggunakan metode *Forward Chaining* untuk membantu teknisi maupun pengguna umum dalam mengidentifikasi gangguan pada komponen seperti RAM, LCD, Baterai, dan Charger. Informasi kerusakan yang di masukkan oleh pengguna akan di proses oleh sistem, kemudian di tampilkan bersama solusi. Pendekatan ini membuat proses diagnosis menjadi lebih cepat, efisien serta memudahkan perbaikan tanpa harus bergantung sepenuhnya pada teknisi professional (Alindi et al., 2023).

Dalam bidang otomotif, sistem pakar telah di kembangkan untuk membantu mendiagnosis kerusakan pada sepeda motor, khususnya tipe matic dan injeksi. Sistem ini menerapkan metode *Forward Chaining* untuk menelusuri gejala yang di berikan pengguna dan mencocokkannya dengan aturan yang ada dalam basis pengetahuan guna menentukan jenis kerusakan. Aplikasi ini di rancang dalam *platform android* agar mudah di akses kapan saja dan di mana saja. Proses pengembangannya mencakup tahap pengumpulan informasi, perancangan sistem, hingga pengujian, dengan hasil berupa sistem yang mampu memberikan diagnosis kerusakan serta solusi praktis dan efisien (Taufiq et al., 2021).

Sistem pakar juga di manfaatkan dalam bidang pelatihan kerja, khususnya untuk meningkatkan keterampilan teknis karyawan bengkel motor. Salah satu implementasinya pengembangan aplikasi dengan metode *Forward Chaining* untuk membantu proses pelatihan dalam mengenali kerusakan pada komponen seperti V-belt dan aki. Sistem ini berperan sebagai sumber referensi bagi karyawan magang dalam memahami gejala kerusakan serta menentukan langkah perbaikan yang sesuai. Penerapan sistem pakar ini terbukti meningkatkan efisien pelatihan dan mempercepat adaptasi teknis karyawan terhadap permasalahan kendaraan bermotor (Sihombing & Sitio, 2021).

Di bidang infrastruktur, sistem pakar di gunakan untuk mendeteksi jenis dan tingkat kerusakan jalan berdasarkan data *survei* lapangan. Sistem ini memanfaatkan data *input* dari *surveyor* terkait ciri fisik kerusakan jalan, yang menghasilkan diagnosis secara otomatis. Proses *inferensi* dalam sistem ini di bantu oleh penggunaan bagan alur dan algoritma pemograman yang menelusuri ciri-ciri kerusakan yang telah di validasi oleh tim teknis. Hasil akhirnya diagnosis yang mampu memberikan informasi secara sistematis dan akurat mengenai kondisi jalan (Jordan et al., 2021).

2.1.2 *Forward Chaining*

Forward Chaining adalah teknik inferensi berbasis data (*data driven inference*) dalam sistem pakar, di mana penalaran dimulai dari fakta atau bukti awal yang telah tersedia, kemudian secara bertahap menerapkan aturan *if-then* untuk menarik kesimpulan baru hingga tidak ada aturan yang tersisa (Naima et al., 2025). Dalam konteks diagnostik, metode ini sangat cocok di gunakan ketika gejala (fakta)

sudah di ketahui dan tujuan sistem adalah menemukan diagnosis atau rekomendasi yang sesuai.

Metode *Forward Chaining* memiliki berbagai penerapan yang dapat di temukan di berbagai bidang. Pengembangan sistem pakar untuk membantu menentukan kelayakan kenaikan pangkat bagi anggota POLRI. Sistem di rancang untuk mempermudah proses perhitungan dan verifikasi persyaratan kenaikan pangkat secara otomatis dan tepat. Dalam hal ini, metode *Forward Chaining* berperan sebagai teknik *inferensiasi* utama, di mana proses penalaran di mulai dari tujuan akhir, yaitu jenis pangkat yang di tuju dan menelusuri aturan aturan yang relevan untuk sampai pada kesimpulan yang sesuai (Gusmita et al., 2023).

Penerapan lainnya membahas pembangunan sistem pakar berbasis web untuk mendiagnosis penyakit pada tanaman karet, sistem ini bertujuan membantu petani mengenali gejala sejak dini, memahami penyebabnya dan menentukan tindakan pengendalian yang tepat. Proses *inferensi* dilakukan dengan metode *Forward Chaining* yang dimulai dari data gejala yang tersedia dan secara bertahap di teruskan hingga mencapai kesimpulan berupa jenis penyakit tanaman. Penggunaan metode ini mendukung pengambilan keputusan yang lebih tepat dalam bidang pertanian (Puspita, 2021).

Perancangan sistem pakar yang bertujuan menentukan tipe kepribadian seseorang dengan menerapkan metode *Forward Chaining* di rancang untuk mempermudah proses identifikasi kepribadian berdasarkan data karakteristik yang di peroleh, sehingga pengguna dapat memperoleh hasil diagnosis tanpa perlu

berkonsultasi langsung dengan psikolog. Metode *Forward Chaining* di gunakan sebagai landasan penalaran, di mana informasi dari hasil tes kepribadian di analisis secara bertahap hingga di peroleh kesimpulan mengenai tipe kepribadian individu (Darmansah et al., 2021).

Ketiga kasus tersebut menunjukkam kesamaan dalam penggunaan metode *Forward Chaining* yang di rancang untuk menghasilkan kesimpulan dari data atau fakta yang tersedia. Berdasarkan contoh kasus tersebut metode ini memiliki kelebihan dan kekurangan. Kelebihannya meliputi pendekatan berbasis data yang memulai penalaran dari fakta yang tersedia, sehingga cocok untuk diagnosis awal. Metode ini cepat, otomatis, mudah di terapkan dan fleksibel untuk berbagai bidang seperti kesehatan, psikologi, pertanian dan kepegawaian. Namun kekurangannya kemungkinan menghasilkan kesimpulan yang terlalu luas, kurang efisien pada kasus dengan banyak fakta dan aturan, bergantung pada kelengkapan data awal, serta kurang cocok untuk mencapai tujuan tertentu sejak awal (Naima et al., 2025).

Mekanisme kerja *Forward Chaining* yaitu yang memiliki berbagai langkah (Noviyani, 2023) :

- a. Inisialisasi Fakta : Sistem memuat semua fakta awal ke dalam *working memory*.
- b. *Matching* : *Inference engine* menelusuri setiap aturan di *rule base* dan mengidentifikasi aturan yang kondisinya terpenuhi.
- c. *Conflict Resolution* : Ketika ada lebih dari satu aturan yang cocok untuk situasi tertentu, sistem harus memilih salah satunya.

- d. *Execution* : Aturan terpilih di jalankan, bagian “*then*” menambahkan fakta baru atau menghasilkan keluaran.
- e. Iterasi : Proses *matching–resolution–execution* di ulang hingga tidak ada lagi aturan yang bisa di picu, dan inferensi berhenti.

Ada beberapa komponen penting dalam penerapan *Forward Chaining* yang di jelaskan oleh (Noviyani, 2023) :

1. Fakta : Fakta merupakan informasi dasar yang menjadi landasan sistem pakar untuk menarik kesimpulan. Fakta adalah data atau kondisi yang sudah di ketahui kebenarannya sebelum sistem menerapkan aturan-aturan untuk penalaran. Contohnya yaitu, “Pasien mengalami gatal di kulit”.
2. Operasional variabel : ini merupakan variabel yang di definisikan secara praktis dimana variabel itu adalah nama variabel dalam penelitian itu misalnya “penyakit kulit” dan indikatornya adalah jenis-jenis penyakit tersebut. Tabel di bawah ini adalah contoh pendefinisian dari operasional variabel :

Tabel 2. 1 Contoh tabel operasional variabel indikator

Variabel	Indikator
Nama variabel	Jenis indikator 1
	Jenis indikator 2
	Jenis indikator 3 dan seterusnya

Sumber : (Wayan et al., 2025)

Tabel 2.1 menunjukkan bentuk operasionalisasi variabel “penyakit” ke dalam beberapa indikator, yaitu jenis-jenis penyakit yang di gunakan dalam sistem pakar berbasis *Forward Chaining*.

Tabel 2. 2 Contoh tabel indikator, gejala, solusi

Indikator	Gejala	Solusi
indikator = Jenis penyakit 1	gejala indikator 1	solusi dari gejala
	gejala indikator 2	
	gejala indikator 3 dan seterusnya	
indikator = Jenis penyakit 2	gejala indikator 1	solusi dari gejala
	gejala indikator 2	
	gejala indikator 3 dan seterusnya	
indikator = Jenis penyakit 3 dan seterusnya ..	gejala indikator 1	
	gejala indicator 2	
	gejala indikator 3 dan seterusnya	solusi dari gejala

Sumber : (Wayan et al., 2025)

Pada tabel 2.2 menggambarkan hubungan antara indikator, gejala dan solusi dengan menunjukkan alur *inferensi Forward Chaining* serta mencocokkan gejala

yang di masukkan pengguna dengan indikator penyakit, lalu memberikan solusi yang sesuai.

- 1) Gejala : biasa di kenal dengan tanda-tanda atau indikasi yang menunjukkan kemungkinan adanya suatu penyakit. Dalam konteks sistem pakar ini, gejala-gejala berkaitan dengan penyakit dan di berikan kode tertentu yang memudahkan sistem mengenali dan menghubungkannya dengan penyakit yang mungkin terjadi. Contoh dari gejala yaitu, “Terjadi gatal pada kulit, kulit bersisik, ruam kemerahan, bercak putih di kulit”.
- 2) Perancangan Basis Pengetahuan : proses menyelidiki dan mengumpulkan informasi terkait variabel yang mencakup indikator, gejala dan solusi. Di implementasikan dalam bentuk tabel seperti berikut :

Tabel 2. 3 Contoh tabel kode indikator

No	Kode Indikator	Indikator
1	kode jenis penyakit 1	jenis penyakit 1
2	kode jenis penyakit 2	jenis penyakit 2
3	kode jenis penyakit 3	jenis penyakit 3, seterusnya...

Sumber : (Adilla Laela Tusifaiyah, 2022)

Tabel 2.3 menyajikan daftar kode indikator, yang merupakan representasi dari jenis-jenis penyakit yang digunakan dalam basis pengetahuan sistem pakar. Setiap jenis penyakit di berikan sebuah kode unik, misalnya “kode jenis penyakit” atau “PPA-01” agar lebih mudah di proses dan di organisasi dalam sistem.

Pemberian kode ini penting dalam tahap *inferensi* sistem pakar, khususnya ketika mencocokkan hasil analisis gejala dengan jenis penyakit yang relevan.

Tabel 2. 4 Contoh tabel kode gejala

No	Gejala	Kode Gejala
1	Gatal pada kulit	G01
2	deskripsi gejala	G02
3	deskripsi gejala dan seterusnya..	Seterusnya..

Sumber : (Adilla Laela Tusifaiyah, 2022)

Tabel 2.4 memperlihatkan daftar gejala yang di gunakan sebagai input dalam sistem pakar, lengkap dengan kode gejalanya. Setiap gejala memiliki deskripsi serta di beri kode, seperti “G01”, “G02” untuk menyederhanakan proses identifikasi dan pencocokan.

Tabel 2. 5 Contoh tabel keputusan

Kode Gejala	Kode indikator		
	kode 1	kode 2	kode 3
Kode-1	✓		
Kode -2	✓		
Kode-3	✓		
Kode-4	✓		

Kode-5	✓		
Kode-6	✓		
Kode-7		✓	
Kode-8		✓	
Kode-9		✓	
Kode-10		✓	
Kode-11		✓	
Kode-12		✓	
Kode-13			✓
Kode-14			✓
Kode-15			✓
Kode-16			✓
Kode-17			✓

Sumber : (Adilla Laela Tusifaiyah, 2022)

Tabel 2.5 menunjukkan tabel keputusan, yaitu struktur relasi antara kode gejala dan kode indikator. Tabel ini berfungsi sebagai bagian ini dari basis aturan dalam

sistem pakar. Tabel ini menjadi dasar pengambilan keputusan otomatis dalam sistem.

- 3) Aturan dan tabel aturan : Biasa di kenal dengan pernyataan bersyarat yang menjelaskan bagaimana sistem mengambil kesimpulan berdasarkan fakta yang tersedia. Berikut tabel data aturannya :

Tabel 2. 6 Tabel aturan

Kode rules	Rules
Kode rules 1	<i>IF</i> gejala-1 <i>AND</i> gejala-2 <i>AND</i> gejala-3 <i>AND</i> gejala-4 <i>AND</i> gejala-5 <i>AND</i> seterusnya <i>THEN</i> Indikator-1
Kode rules 2	<i>IF</i> gejala-6 <i>AND</i> gejala-7 <i>AND</i> gejala-8 <i>AND</i> gejala-9 <i>AND</i> gejala-10 <i>AND</i> seterusnya <i>THEN</i> Indikator-2
Kode rules 3	<i>IF</i> gejala-11 <i>AND</i> gejala-12 <i>AND</i> gejala-13 <i>AND</i> gejala-14 <i>AND</i> gejala-15 <i>AND</i> seterusnya <i>THEN</i> Indikator-3

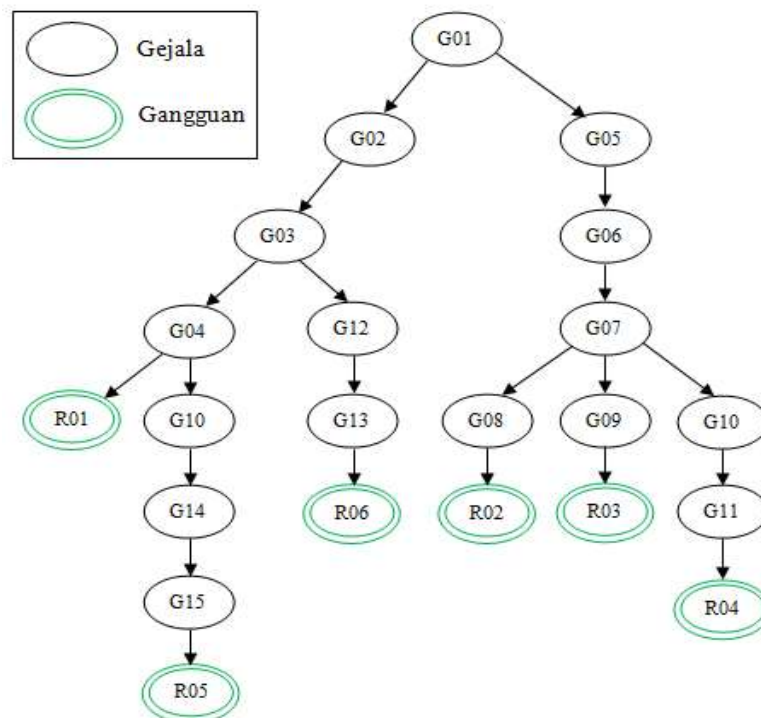
Sumber : (Adilla Laela Tusifaiyah, 2022)

Contoh pada tabel 2.6, jika gejala A dan B muncul, maka kemungkinan penyakit X terdiagnosis. Aturan-aturan ini menjadi dasar logika dalam sistem pakar untuk menentukan hasil diagnosis. Contohnya, “*IF* terjadi gatal pada kulit *AND* kulit bersisik *AND* ruam kemerahan *AND* bercak putih di kulit *THEN* infeksi jamur kulit.”

- 4) Pohon Keputusan : Pohon ini biasa digunakan untuk menggambarkan

langkah-langkah logis berdasarkan fakta dan aturan, serta dapat membantu menyederhanakan proses penentuan diagnosis dengan mengikuti alur dari gejala menuju kesimpulan.

Gambar 2.3 menunjukkan pohon keputusan yang menggambarkan alur logis dalam mendiagnosis gangguan berdasarkan kombinasi gejala-gejala tertentu. Setiap simpul merepresentasikan gejala (G) yang harus dipenuhi secara berurutan, dan cabang-cabangnya mengarahkan pada simpul akhir berupa jenis gangguan (R) yang terdiagnosis. Dengan struktur ini, pohon keputusan mempermudah proses identifikasi penyakit secara sistematis dan efisien, sehingga sangat bermanfaat untuk diterapkan dalam sistem pakar berbasis aturan.



Gambar 2. 3 Pohon Keputusan

Sumber : (Nyoman Irvianti Windaputri, Sri Widowati, 2020)

2.1.3 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) dikenal dengan bahasa visual standar yang di gunakan untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan komponen dalam sistem perangkat lunak. UML terdiri dari berbagai jenis diagram yang membantu memodelkan struktur dan perilaku sistem, seperti *use case* diagram, *activity* diagram, *class* diagram dan lainnya (Priscasindo & Anggia, 2020).

a. *Usecase Diagram*

Usecase diagram bisa dikatakan salah satu jenis diagram UML yang menggambarkan hubungan antara aktor dan sistem yang sedang di kembangkan. Diagram ini mengilustrasikan interaksi utama antara aktor dan sistem untuk mencapai tujuan tertentu (Sari & Putri, 2020). Misalnya, dalam sistem pakar mendiagnosis gangguan pencernaan pada anak, aktor dapat berupa “orang tua” atau dokter, sedangkan *usecase*-nya mencakup “memasukkan gejala anak”, “menerima hasil diagnosis” dan “melihat saran penanganan”.

Berikut gambar dan simbol *usecase* diagram:

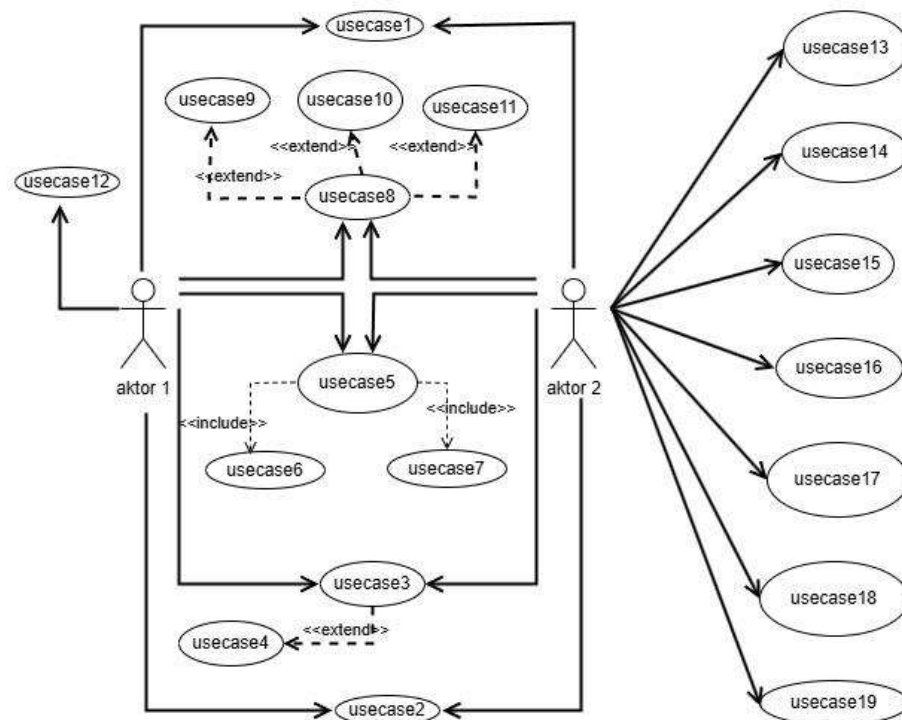
Tabel 2. 7 Keterangan simbol usecase diagram

Simbol	Makna	Keterangan
	Aktor	Berperan sebagai pengguna sistem (bisa manusia atau sistem lain).

	<i>UseCase</i>	Aktivitas atau layanan yang disediakan sistem untuk aktor.
	<i>Association</i>	Menunjukkan hubungan komunikasi antara aktor.
	<i>Include</i>	<i>Usecase</i> wajib yang di panggil oleh use case lainnya.
	<i>Extend</i>	<i>Usecase</i> opsional yang di tambahkan sesuai kondisi tertentu.

Sumber : (Septiara et al., 2025)

Simbol-simbol pada *usecase* membantu memvisualisasikan siapa melakukan apa, fungsi apa saja yang disediakan sistem, dan bagaimana hubungan antar komponen itu terbentuk



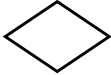
Gambar 2. 4 Contoh *usecase diagram*

Sumber : (Agusli et al., 2020)

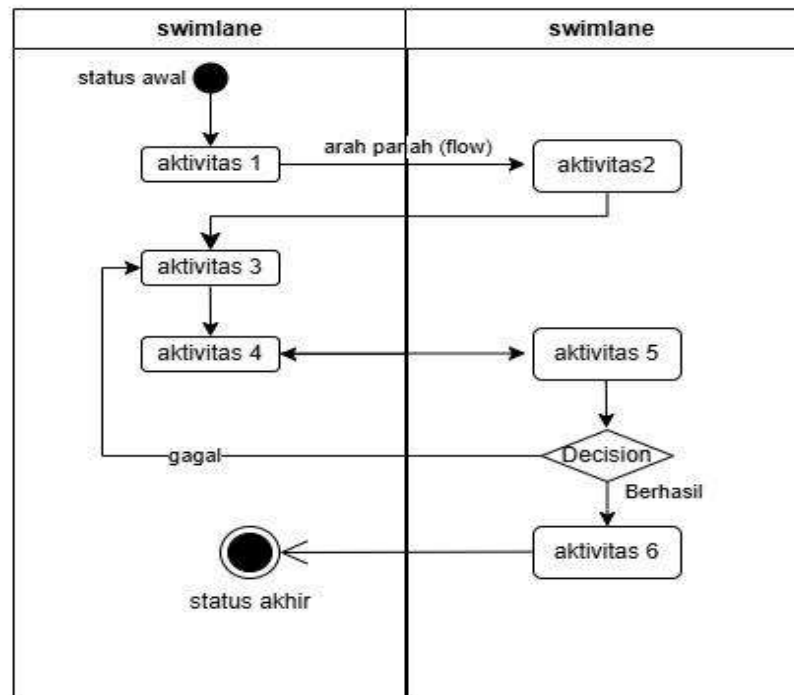
b. *Activity Diagram*

Activity Diagram boleh dikatakan salah satu dari beberapa jenis diagram UML yang digunakan untuk memvisualisasikan alur kerja atau rangkaian tindakan dalam proses menu pada perangkat lunak (Sari & Putri, 2020). Berikut contoh gambar dan tabel simbol yang digunakan pada *Activity Diagram* :

Tabel 1. 1 Simbol *Activity* Diagram

Simbol	Keterangan
	Status awal >> titik mulai proses (lingkaran hitam penuh)
	Aktivitas >> tindakan/proses sistem (persegi panjang sudut membulat berisi kata kerja)
	Decision (Keputusan) >> pilihan jalur proses (berhasil/gagal)
	Panah arah (flow) >> menunjukkan urutan alur aktivitas
	Status akhir >> titik akhir proses (lingkaran ganda)
	Swimlane >> pemisah aktivitas berdasarkan pelaku atau pihak (kolom admin/user/sistem)

Sumber : (Septiara et al., 2025)



Gambar 2. 5 Activity Diagram
Sumber : (Agusli et al., 2020)

c. *Class Diagram*

Diagram class merupakan jenis diagram UML yang menggambarkan struktur sistem dengan menampilkan kelas-kelas, atribut, metode, serta hubungan antar kelas juga digunakan untuk memodelkan struktur data dan relasi antar objek dalam sistem (Priscasindo & Anggia, 2020).

Simbol-simbol dalam *class diagram* di gunakan untuk merepresentasikan berbagai komponen dan hubungan dalam sistem. Berikut penjelasan setiap simbol dalam *class diagram* :

Tabel 2. 8 Tabel Class Diagram

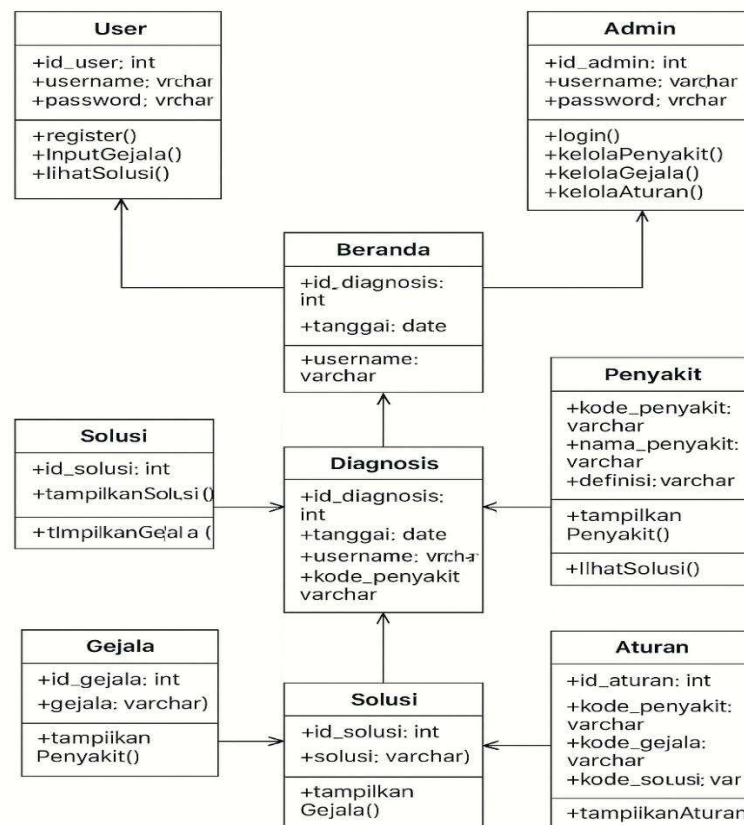
Simbol	Keterangan
--------	------------

<i>Class</i>	Kotak dengan 3 bagian : nama kelas (atas), atribut (tengah), dan operasi / metode (bawah)
<i>Atribut</i>	data yang di miliki kelas, di tulis di bagian tengah
<i>Operasi /Metode</i>	fungsi atau aksi kelas, terletak di bagian bawah
<i>Association</i>	Garis penghubung antar kelas, kadang dengan panah untuk arah hubungan
<i>Aggregation</i>	Garis berujung diamond kosong, menunjukkan hubungan keseluruhan bagian
<i>Composition</i>	Seperti aggregation, tetapi diamond diisi penuh, melambangkan bagian tak bisa terjadi sendiri
<i>Generalization</i>	Garis dengan ujung segitia, melambangkan pewarisan kelas anak ke induk
<i>Dependency</i>	panah putus-putus dari pengguna ke kelas yang di gunakan, menyatakan ketergantungan

Sumber : (Ramdany, 2024)

Tabel 2.8 menjelaskan simbol-simbol yang digunakan dalam *class diagram*, yang merupakan bagian dari *Unified Modelling Language* (UML) untuk memodelkan struktur sistem. Simbol-simbol tersebut merepresentasikan berbagai komponen dan hubungan antar kelas dalam sistem. Setiap simbol memiliki arti tertentu, seperti kotak untuk merepresentasikan *class*, garis penghubung untuk *association*, dan simbol lainnya terhadap simbol-simbol ini

penting untuk membaca dan merancang *class diagram* secara tepat dalam pengembangan sistem berbasis objek.



Gambar 2. 6 Class Diagram

Sumber : (Larasaty & Prasetyaningrum, 2024)

Class Diagram pada gambar 2.6 menggambarkan struktur sistem pakar diagnosis penyakit berdasarkan gejala pada anak. Diagram ini menunjukkan 18 kelas utama yaitu *user*, *admin*, *beranda*, *diagnosis*, *penyakit*, *solusi*, *gejala* dan *aturan*. Masing-masing kelas memiliki atribut dan metode yang mencerminkan fungsionalitas sistem, seperti proses *login*, *input* gejala, Kelola data penyakit, dan penampilan solusi.

Relasi antarkelas juga tergambar dengan jelas melalui garis asosiasi, menunjukkan hubungan seperti pengguna yang dapat melakukan diagnosis, serta diagnosis yang terkait dengan penyakit dan gejala tertentu. Diagram ini menjadi dasar penting dalam pengembangan sistem karena membantu memahami interaksi dan strukture logika antarkomponen dalam sistem pakar.

d. *Sequence Diagram*

Sequence diagram dijelaskan sebagai diagram yang memperlihatkan rangkaian interaksi antar objek secara berurutan berdasarkan alur waktu, serta menggambarkan tahapan-tahapan proses yang dapat dijalankan (NurJumala et al., 2022). *Sequence diagram* menggambarkan interaksi antara objek dalam suatu sistem melalui pengiriman pesan yang disusun sesuai urutan waktu, sebagaimana dilakukan oleh aktor ketika berinteraksi dengan sistem (Larasaty & Prasetyaningrum, 2024). Berikut komponen dan simbol penting dalam *sequence diagram*:

- 1) Aktor merupakan elemen eksternal yang berinteraksi dengan sistem, biasanya berupa pengguna atau sistem lain. Aktor digambarkan sebagai ikon manusia atau teks yang berada di sisi kiri atau kanan diagram.
- 2) Objek merepresentasikan komponen dalam sistem yang saling berinteraksi selama proses berlangsung, seperti antarmuka pengguna, mesin inferensi, basis pengetahuan, dan database. Objek ini digambarkan dengan persegi panjang yang dihubungkan dengan garis vertikal putus-putus yang menunjukkan keberadaannya selama siklus waktu berlangsung.
- 3) Garis Hidup adalah garis vertikal putus-putus yang ditarik dari setiap objek

dalam *Sequence diagram*. Garis ini menunjukkan waktu hidup dari suatu objek selama proses berlangsung dan menjadi jalur utama bagi pesan-pesan yang dikirim atau diterima objek tersebut.

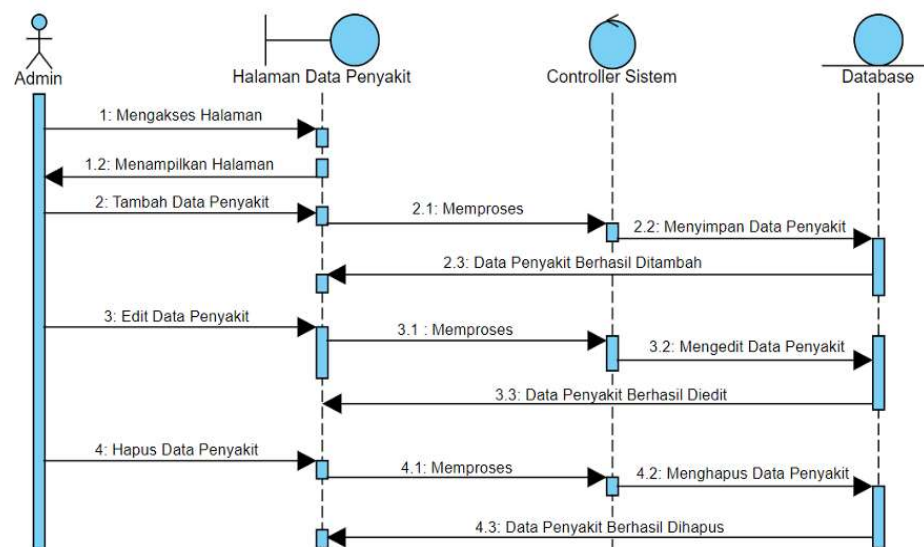
- 4) Pesan merupakan garis panah horizontal yang menghubungkan antar objek dan menunjukkan bentuk komunikasi antar entitas dalam sistem. Pesan bisa berupa permintaan, tanggapan, atau pemanggilan fungsi tertentu. Jenis pesan meliputi *synchronous* (panah penuh), *asynchronous* (panah terbuka), dan *return* (panah putus-putus).
- 5) Aktivasi ditampilkan sebagai persegi panjang tipis di atas garis hidup objek dan menggambarkan waktu di mana objek sedang mengeksekusi suatu aksi atau proses. Semakin panjang persegi ini, semakin lama objek tersebut aktif dalam proses tertentu.
- 6) Titik waktu merujuk pada susunan vertikal dalam diagram yang bergerak dari atas ke bawah, menggambarkan urutan kejadian secara kronologis. Setiap interaksi antar objek ditempatkan sesuai dengan urutan waktu terjadinya agar alur logika proses sistem mudah dipahami.
- 7) *Fragment* digunakan untuk merepresentasikan struktur kontrol logika dalam *Sequence diagram*, seperti percabangan (*alt*), kondisi opsional (*opt*), dan pengulangan (*loop*). Simbol ini berbentuk kotak besar yang mencakup bagian interaksi tertentu dan membantu dalam memodelkan skenario kompleks dalam sistem.

Tabel 2. 9 Komponen *Sequence Diagram*

Komponen	Contoh dalam Sistem
----------	---------------------

Aktor	<i>User, Admin</i>
Objek	<i>Form Diagnosa, Sistem Pakar, Mesin Inferensi, Database</i>
Pesan	<i>submitGejala(), cariAturan(), tampilkanHasil()</i>
Loop	Mengulang semua gejala yang diinput
Alt	Jika cocok → diagnosa; Jika tidak cocok → minta input ulang

Sumber : (Larasaty & Prasetyaningrum, 2024)



Gambar 2. 7 *Sequence Diagram*

Sumber : (Larasaty & Prasetyaningrum, 2024)

Sequence diagram pada gambar 2.7 menggambarkan alur interaksi aktor admin, halaman data penyakit, *controller sistem*, dan *database* dalam proses pengelolaan data penyakit. Proses diawali ketika admin mengakses halaman

data penyakit, kemudian sistem menampilkan halaman tersebut. Admin dapat melakukan tiga aksi utama yaitu menambah, mengedit, dan menghapus data penyakit. Setiap aksi dikirim sebagai pesan ke *controller sistem* untuk diproses, lalu *controller* berinteraksi dengan *database* untuk menyimpan, mengedit atau menghapus data. Hasil dari setiap aksi dikirimkan kembali ke admin sebagai konfirmasi keberhasilan proses. Diagram ini menunjukkan urutan waktu dan logika interaksi sistem secara terstruktur, memudahkan pemahaman tentang aliran kerja fungsional pada sistem pengelolaan data penyakit.

2.1.4 Database

Database berupa kumpulan data atau informasi yang di atur dan di simpan secara sistematis dalam perangkat komputer dengan tujuan mempermudah proses pencarian, pemeriksaan dan pengelolaan data secara terstruktur (Polii & Arnomo, 2022). Skema database menggambarkan susunan data terorganisir dengan baik, di mana setiap tabel memiliki *primary key* sebagai elemen penting untuk menjaga konsistensi dan integritas antar tabel, serta menghindari duplikasi data (Umairah & Dasa, 2023). Berikut contoh tabel rancangan basis data yang akan di implementasikan secara teknis pada bab 3 :

Tabel 2. 10 Tabel Database

Nama Atribut	Tipe Data (umum)	Deskripsi
user_id	Integer	ID unik pengguna (Primary Key)

name	Varchar	Nama pengguna
email	Varchar	Email pengguna
password	Varchar	Kata sandi (terenkripsi)
role	Enum	Peran pengguna: admin atau user
created_at	Datetime	Waktu pendaftaran pengguna

Sumber : (Umairah & Dasa, 2023)

2.1.5 Gangguan Pencernaan Pada Anak

Gangguan pencernaan pada anak adalah kondisi abnormal yang di sebabkan oleh masalah dalam sistem pencernaan. Gangguan ini memiliki potensi untuk menghambat pertumbuhan optimal dan perkembangan kognitif anak. Sebagian besar kasus bersifat fungsional dan melibatkan masalah di lambung atau organ pencernaan lainnya, yang dapat berdampak pada kesehatan dan perkembangan anak secara keseluruhan (Syahreni et al., 2024). Berikut beberapa penjelasan mengenai gangguan pencernaan pada anak:

a. Diare



Gambar 2. 8 Diare

Sumber : (Amukti et al., 2024)

Diare merupakan salah satu gangguan sistem pencernaan yang paling sering dialami oleh anak-anak (Setyaningsih & Diyono, 2020). Gejalanya ditandai dengan peningkatan frekuensi buang air besar lebih dari 3x per hari dengan konsistensi tinja yang encer atau lembek (Jayanto et al., 2020). Jika diare tidak segera ditangani, anak bisa mengalami dehidrasi, kekurangan gizi, bahkan kematian, terutama pada balita (Sasmito et al., 2023). Penyebab diare terbagi menjadi dua kategori utama, yaitu (Rendang Indriyani & Putra, 2020):

- Diare yang bersumber dari infeksi umumnya disebabkan oleh virus, bakteri, *parasit* atau *protozoa* yang masuk ke tubuh melalui makanan atau minuman yang tercemar.
- Penyebab non-infeksi meliputi konsumsi obat-obatan tertentu, reaksi alergi, gangguan penyerapan nutrisi, kekurangan vitamin, dan faktor psikologi seperti stress.

Secara fisiologis, diare terjadi akibat terganggunya mekanisme penyerapan dan pengeluaran cairan dalam usus. Kondisi ini dapat disebabkan oleh peningkatan tekanan *osmotik* atau gangguan gangguan pada sistem *ion* di usus. Akibatnya, terjadi kelebihan cairan dalam saluran cerna yang memicu diare. Kondisi ini dapat menimbulkan risiko serius bagi anak, terutama karena mengakibatkan kehilangan cairan dan elektrolit dalam jumlah besar (Rendang Indriyani & Putra, 2020).

Selain faktor yang telah dijelaskan diatas, aspek lingkungan juga berkontribusi terhadap kejadian diare. Penggunaan air bersih yang tidak layak konsumsi, kebiasaan mencuci tangan yang kurang baik, serta tidak tersedianya jamban yang memadai merupakan faktor risiko utama yang meningkatkan kemungkinan terjadinya infeksi penyebab diare (Ribek et al., 2020). Orang tua dianjurkan memberikan larutan rehidrasi oral (*oralit*) sebanyak 10 ml/kg berat badan setiap kali anak buang air besar. Selain itu, asupan cairan dan makanan tetap harus dianjurkan untuk menjaga kondisi tubuh (Sasmito et al., 2023).

Pemberian *zinc* selama 10 hingga 14 hari, penggunaan *probiotik*, serta edukasi mengenai cara pemberian *oralit* dan menghindari obat-obatan yang tidak dianjurkan seperti antidiare dan *antiemetik* menjadi bagian dari penatalaksanaan yang disarankan untuk mempercepat proses penyembuhan dan mencegah diare kambuh (Rendang Indriyani & Putra, 2020).

b. Konstipasi



Gambar 2. 9 Konstipasi

Sumber : (Kadim, 2021)

Konstipasi pada anak merupakan salah satu jenis gangguan pencernaan yang kerap tidak mendapatkan perhatian, padahal kondisi ini dapat menurunkan kualitas hidup anak dan menambah beban biaya dalam kesehatan (Kadim, 2021). Gejala konstipasi pada anak dapat berbeda tergantung pada usia-nya. Anak-anak yang berusia di atas 4 tahun umumnya menunjukkan tanda-tanda tambahan, seperti kebiasaan menahan buang air besar dengan menyilangkan kaki atau bersembunyi dari orang lain. Perbedaan ini menunjukkan bahwa usia turut memengaruhi bentuk dan cara gejala konstipasi muncul pada anak (Kadim, 2021).

Gejala umum yang terjadi pada anak penderita konstipasi terbagi menjadi 2 (Maghfuroh et al., n.d.) :

1) Gejala Fisik

- Nyeri saat buang air besar
- Feses keras dan memanjang
- Bau tinja menyengat
- Perut kembung atau tegang
- Gerakan usus terlihat
- Feses terasa saat diraba
- Tanda malnutrisi dan anemia
- Risiko hemoroid

2) Gejala Psikologis dan Perilaku

- Rasa percaya diri rendah, anak merasa malu atau tidak percaya diri akibat kesulitan BAB.
- Ketakutan saat toilet training, anak enggan ke toilet karena trauma atau ketidaknyamanan.
- Menghindari buang air besar, anak menahan BAB karena takut sakit atau stress saat di kamar mandi.

Penanganan konstipasi pada anak, posisi orang tua sangat penting sebelum di lakukan intervensi medis untuk melakukan penanganan dini juga pencegahan. Beberapa penanganan dan upaya pencegahan yang dapat dilakukan (Rohman & Marmi, 2023) :

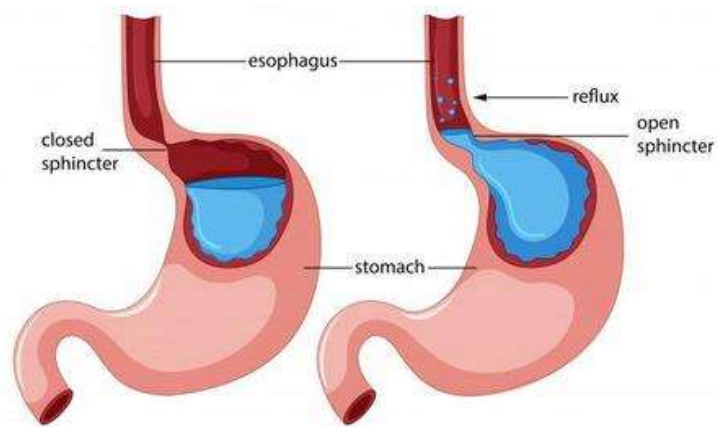
a) Penanganan konstipasi pada anak

- 1) Perbaiki pola makan, dengan memberikan anak makanan tinggi serat seperti buah, sayur dan biji-bijian serta pastikan asupan cairan cukup.
- 2) Buat rutinitas buang air besar, biasakan anak duduk di toilet secara rutin,

terutama setelah makan untuk membentuk kebiasaan sehat.

- 3) Kurangi stress, identifikasi dan kurangi sumber stress di rumah maupun di sekolah yang bisa memperburuk kondisi konstipasi.
 - 4) Dorong aktivitas fisik, ajak anak untuk aktif bergerak setiap hari karena aktivitas membantu kerja usus menjadi lebih optimal
 - 5) Ciptakan lingkungan yang nyaman, pastikan anak merasa nyaman saat di toilet termasuk memberikan privasi agar tidak merasa tertekan.
- b) Pencegahan konstipasi pada anak
- 1) Ajarkan manajemen stress : bantu anak mengenali dan mengelola stress sejak diri, baik di rumah maupun melalui program sekolah.
 - 2) Kenali faktor psikologi, tangani masalah seperti kecemasan dan depresi sejak dini untuk mencegah gangguan pencernaan yang di picu faktor emosional.
 - 3) Gunakan probiotik secara tepat, pemberian probiotik secara teratur dapat membantu menjaga kesehatan usus dan mencegah konstipasi.
 - 4) Pastikan anak terus mengonsumsi makanan tinggi serat dan cukup minum air setiap hari sebagai Langkah utama dalam mencegah konstipasi.

c. *Gastroesophageal Reflux Disease (GERD)*



Gambar 2. 10 GERD
Sumber : (Amukti et al., 2024)

Gastroesophageal Reflux Disease (GERD) merupakan gangguan pencernaan yang disebabkan oleh naiknya asam lambung ke *esofagus* sehingga menimbulkan rasa tidak nyaman terutama di bagian dada dan perut (Amukti et al., 2024). Gejala GERD pada anak-anak bervariasi tergantung pada kelompok usia, pada anak usia 2 tahun GERD biasanya di tandai dengan muntah yang sering dan kerewelan setelah menyusui. Sementara itu, pada anak di atas usia 2 tahun, gejala umum meliputi *heartburn*, sensasi terbakar di dada dan ketidaknyamanan pada perut bagian atas. Komplikasi GERD dapat mencakup gangguan pernapasan, kesulitan menambah berat badan, serta peradangan di kerongkongan (Jannata et al., 2021).

Faktor-faktor yang memicu GERD pada anak antara lain pola makan yang kurang sehat, seperti konsumsi makanan pedas, asam, berlemak atau berkarbonasi, serta kebiasaan berbaring setelah makan. Selain itu, gaya hidup tidak sehat seperti merokok, stres, dan obesitas juga dapat meningkatkan risiko GERD (Jodi, 2024). Penanganan GERD pada anak mencakup perubahan gaya hidup, edukasi tentang

pola makan sehat, serta penggunaan obat-obatan seperti antasida dan penghambat pompa proton (PPI) untuk mengurangi asam lambung dan mengatasi gejalanya (Waluyo & Solikah, 2023).

2.1.6 Tools

Penelitian ini mengandalkan sejumlah perangkat lunak dan bahasa pemrograman utama untuk membangun sistem pakar berbasis web :

a. Visual Studio Code



Gambar 2. 11 Visual studio code
Sumber : Penelitian, 2025

VS Code adalah editor kode gratis dan *open source* dari indikator yang populer di kalangan pengembangan karena ringan, fleksibel dan mendukung berbagai bahasa pemrograman seperti Java, Python dan JavaScript yang cocok untuk pengembangan web dan aplikasi modern. Fitur utamanya mencakup ekstensi yang kaya, *integrasi git*, *intellisense*, debugging bawaan dan lainnya (Kaur et al., 2023).

b. Draw.io



Gambar 2. 12 Draw.io
Sumber : (Pokhrel, 2024)

Draw.io merupakan aplikasi berbasis web yang di rancang untuk memudahkan pembuatan diagram *Unified Modelling Language* (UML) secara *online*. Dengan antarmuka yang responsif dan integrasi dengan Google Drive, aplikasi ini memungkinkan pengguna membuat diagram UML dengan cepat dan efisien. Draw.io di manfaatkan sebagai alat bantu bagi mahasiswa untuk memvisualisasikan kebutuhan pengembangan perangkat lunak, menganalisis kebutuhan, dan mendesain diagram UML (Noneng Marthiawati et al., 2024).

c. Figma



Gambar 2. 13 Figma
Sumber : (Yohnes Madawara et al., 2023)

Figma merupakan alat untuk merancang antarmuka pengguna (*UI*) dan pengalaman pengguna (*UX*) aplikasi. Figma di pilih karena kemampuannya

membuat *prototipe* interaktif yang memudahkan pengujian dan memungkinkan pengguna mencoba konsep desain untuk memastikan tampilan dan fungsi aplikasi berjalan baik. Pada tahap *prototyping*, Figma di gunakan untuk menciptakan interaksi antar tampilan, sehingga pengguna dapat merasakan pengalaman menggunakan aplikasi sebelum implementasi. Fitur seperti Figma *Mirror* juga di gunakan selama pengujian, meskipun beberapa pengguna memerlukan panduan untuk mengoperasikannya (Yohnes Madawara et al., 2023).

d. XAMPP



Gambar 2. 14 Xampp
Sumber : (Akuntansi, 2022)

Xampp merupakan aplikasi yang berfungsi untuk membuat *web server local* di komputer, yang dapat menyederhanakan proses instalasi dan pengoperasian *server* berbasis PHP dan MySql tanpa membutuhkan konfigurasi yang rumit. Xampp dapat mendukung aktivitas pengembangan dan pengujian website berbasis PHP (Akuntansi, 2022).

2.2 Penelitian Terdahulu

Penelitian terdahulu dapat menjadi acuan kajian pustaka untuk memahami penelitian serupa, metode, data, memunculkan ide baru serta memvalidasi keaslian dengan tujuan mengisi celah penelitian dan menyusun hipotesis lebih baik (Ridwan et al., 2021). Ada beberapa penelitian terdahulu yang terkait dengan aplikasi sistem pakar ini adalah:

1. Penelitian “Implementasi *Rule-Based Reasoning* dalam Pencegahan *Stunting* (Studi Kasus: Dinas Kesehatan Kabupaten Klaten)” bertujuan untuk membantu tenaga kesehatan dalam menentukan status gizi balita dan intervensi yang tepat, serta memetakan wilayah dengan tingkat *stunting* tinggi. Penelitian ini menggunakan metode *Rule-Based Reasoning*, yang mampu memberikan hasil status gizi (BB/U dan TB/U) serta rekomendasi intervensi berbasis aturan. Pengembangan sistem dilakukan melalui tahapan *Agile Development*. Sistem di uji di Dinas Kesehatan Kabupaten Klaten, dan menunjukkan akurasi 84% untuk BB/U, 100% untuk TB/U, dan akurasi keseluruhan sebesar 92%. Selain itu, *usability testing* menghasilkan skor kepuasan pengguna sebesar 89,8%, menunjukkan bahwa sistem mudah digunakan dan efektif dalam mendukung pencegahan *stunting* (Klaten, 2024).
2. Penelitian “Penerapan Metode *Forward Chaining* dalam Sistem Pakar Mendiagnosa Penyakit Diare pada Anak” bertujuan untuk membantu orang tua dalam mendiagnosa penyakit diare pada anak secara mandiri melalui sistem pakar berbasis web. Sistem ini menggunakan metode *forward*

chaining, yang bekerja dengan menelusuri gejala yang di alami anak dan mencocokkannya dengan basis pengetahuan dalam sistem untuk menentukan kemungkinan jenis diare. Penelitian ini membangun sistem dengan pendekatan inferensi berbasis aturan, sistem ini menghasilkan diagnosis berdasarkan *rule* yang paling sesuai dan menyediakan solusi awal yang dapat di lakukan sebelum mengunjungi dokter. Sistem juga ditujukan melalui indikator gejala yang di input pengguna, dan hasilnya menunjukkan tingkat kecocokan tertinggi dengan diare akut bakteri. Sistem di kembangkan menggunakan model UML, mencakup *use case diagram* serta interaksi pengguna dengan sistem, mulai dari *login*, *input* data, memilih gejala, hingga mendapatkan hasil diagnosis dan solusi. Hasil penelitian menunjukkan bahwa sistem ini cukup efektif sebagai alat bantu awal penyakit diare pada anak, meskipun masih perlu pengembangan lebih lanjut dalam memperluas jenis gejala, solusi, serta peningkatan spesifikasi diagnosis (Fachri et al., 2022).

3. Penelitian “Penerapan Metode *Forward Chaining* untuk Sistem Pakar Diagnosis Penyakit Tanaman Mentimun Berbasis Web” bertujuan membantu petani mendiagnosa penyakit mentimun melalui gejala yang dimasukkan ke sistem. Sistem ini dikembangkan menggunakan metode *Forward Chaining* dan pendekatan *Rational Unified Process* (RUP), serta di implementasikan dalam bentuk aplikasi web berbasis PHP dan Bootstrap. Hasil pengujian menunjukkan sistem berjalan sesuai harapan, dengan tingkat kepuasan pengguna sebesar 83,8% dan 90% dari admin, yang

tergolong dalam kategori sangat setuju menurut skala Likert. Sistem ini efektif dalam mengenali enam jenis penyakit mentimun, serta menyediakan hasil diagnosis dan solusi berdasarkan aturan *if-then* dari gejala yang di input (Puspita, 2021).

4. Penelitian “Pendekatan Metode *Forward Chaining* pada Penentuan Menu Makanan Sehat Pasien Paru Obstruktif Kronis” bertujuan membantu petugas gizi dalam menyusun menu makanan sehat bagi pasien PPOK (Paru Obstruktif Kronis). Sistem pakar berbasis web di kembangkan menggunakan metode *forward chaining* untuk menghasilkan rekomendasi makanan berdasarkan input umur, berat badan, dan jenis kelamin pasien. Sistem ini dirancang untuk menggantikan metode manual yang memakan waktu dan bergantung pada buku pedoman. Hasil pengujian menunjukkan akurasi sistem mencapai 100% dalam 20 kali pengujian, serta berhasil menghasilkan menu yang sesuai dengan kriteria pasien. Dengan penerapan sistem ini, petugas gizi dapat lebih cepat dan tepat dalam menentukan menu pasien PPOK (Mubarak et al., 2025).
5. Penelitian “Pengembangan Sistem Pakar untuk Diagnosa Penyakit Diabetes Melitus Menggunakan Metode *Forward Chaining*” bertujuan untuk membantu tenaga medis dalam mendiagnosis diabetes secara cepat dan akurat melalui sistem berbasis web. Sistem ini di rancang menggunakan metode *forward chaining*, yang bekerja berdasarkan input gejala pasien untuk menghasilkan rekomendasi diagnosis. Pengembangan sistem di lakukan melalui studi literatur, wawancara, dan kuesioner, dengan hasil akurasi sistem mencapai 85%. Sistem di nilai mudah di gunakan dan mampu

mendeteksi gejala secara efisien, sehingga dapat mempercepat proses diagnosis dan mengurangi kesalahan manual. Implementasi sistem ini diharapkan dapat meningkatkan deteksi dini serta kualitas layanan kesehatan, khususnya di Klinik Nugraha (Supiyanto, 2025).

2.3 Kerangka Pemikiran



Gambar 2. 15 Kerangka Pemikiran

Sumber : Data Penelitian, 2025

Masalah gangguan pencernaan pada anak menjadi perhatian serius dalam dunia kesehatan, karena dapat memengaruhi tumbuh kembang dan kualitas hidup mereka. Berbagai jenis gangguan pencernaan pada anak sering kali disertai gejala yang beragam, seperti diare, konstipasi, dan Gerd yang membutuhkan diagnosis cepat dan penanganan yang tepat. Namun, proses diagnosis sering kali memerlukan waktu dan keahlian khusus dari tenaga medis, sehingga diperlukan solusi inovatif untuk mendukung proses tersebut, terutama dalam memberikan diagnosis awal secara akurat dan efisien.

Untuk mengatasi kebutuhan tersebut, penelitian ini mengembangkan sistem pakar berbasis web menggunakan metode *Forward Chaining*, di mana mekanisme inferensi dilakukan dengan menelusuri aturan berdasarkan gejala yang dimasukkan oleh pengguna. Sistem ini dirancang menggunakan indikator *if-then* yang

diterapkan dalam basis aturan. Pengembangan sistem melibatkan pembuatan basis data yang mencakup gejala dan jenis penyakit, serta pengolahan data menggunakan MySQL. Komponen teknis lainnya meliputi penggunaan Visual Studio Code sebagai editor, bahasa pemrograman PHP untuk logika sistem, dan HTML, CSS, serta JavaScript untuk antarmuka pengguna yang interaktif. Perancangan sistem ini juga didukung dengan diagram UML untuk memvisualisasikan alur dan struktur data secara sistematis.

Hasil akhir dari penelitian ini adalah sebuah aplikasi berbasis web yang dirancang untuk membantu proses diagnosis awal gangguan pencernaan pada anak. Aplikasi ini memungkinkan pengguna, seperti orang tua atau tenaga medis, untuk menginput gejala yang dialami anak. Sistem akan memproses informasi tersebut melalui mekanisme inferensi dan menampilkan diagnosis berupa jenis gangguan pencernaan yang paling sesuai. Selain itu, aplikasi ini juga menyediakan informasi tambahan tentang gejala umum dari penyakit terkait dan saran penanganan awal, sehingga dapat menjadi alat bantu yang efektif sebelum konsultasi lebih lanjut dengan dokter spesialis.