

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi telah berkembang sangat pesat dan memberikan dampak signifikan dalam berbagai bidang kehidupan, termasuk dalam dunia kesehatan. Kebutuhan akan kecepatan akses informasi serta kemudahan dalam mendapatkan layanan kesehatan mendorong pemanfaatan teknologi sebagai alat bantu dalam proses Diagnosis dan penanganan penyakit. Salah satu bentuk implementasi teknologi di bidang kesehatan adalah sistem pakar (*expert system*).

Sistem pakar adalah salah satu cabang dari kecerdasan buatan yang dirancang untuk meniru cara berpikir dan pengambilan keputusan seorang pakar dengan memanfaatkan basis pengetahuan yang disusun dalam bentuk aturan atau kaidah tertentu. Sistem ini memungkinkan untuk menganalisis data, menarik kesimpulan, serta memberikan solusi terhadap suatu permasalahan secara logis dan sistematis, layaknya seorang ahli di bidangnya. Penerapan sistem pakar dalam berbagai bidang, termasuk kesehatan, bertujuan untuk membantu pengguna dalam menyelesaikan permasalahan atau mendiagnosis suatu kondisi dengan cara mengadopsi pengalaman dan pengetahuan dari pakar, sehingga proses pengambilan keputusan menjadi lebih cepat, akurat, dan efisien (Pramudita & Hibatullah, 2024).

Penyakit Telinga, Hidung, dan Tenggorokan (THT) merupakan salah satu jenis penyakit yang cukup sering dialami oleh masyarakat. Penyakit Telinga, Hidung, dan Tenggorokan (THT) adalah gangguan kesehatan yang menyerang organ telinga, hidung, dan tenggorokan, yang umumnya disebabkan oleh infeksi kuman. Namun, selain infeksi, penyakit THT juga dapat disebabkan oleh kelainan pertumbuhan sel tubuh yang abnormal, yang berpotensi berkembang menjadi tumor atau bahkan kanker. Penyakit THT sering kali dianggap sepele, padahal jika tidak ditangani dengan baik dapat menimbulkan komplikasi serius pada organ pernapasan dan pendengaran (Sari & Desiani, 2023).

Penyakit THT dapat menyerang berbagai kalangan usia dan jika tidak ditangani dengan cepat dan tepat, dapat menimbulkan komplikasi serius yang berdampak pada kesehatan organ lain. Namun, keterbatasan jumlah dokter spesialis THT, terutama di daerah pedesaan atau wilayah terpencil, seringkali menjadi hambatan bagi masyarakat dalam mendapatkan pelayanan kesehatan yang optimal.

Selain keterbatasan tenaga medis, masih banyak masyarakat yang kurang memahami gejala-gejala awal penyakit THT sehingga sering mengabaikan keluhan-keluhan ringan yang sebenarnya memerlukan penanganan dini. Kurangnya pengetahuan ini menyebabkan masyarakat baru memeriksakan diri ke dokter setelah kondisi penyakit sudah memasuki tahap serius. Oleh karena itu,

diperlukan sebuah sistem yang dapat membantu masyarakat dalam melakukan Diagnosis awal secara mandiri agar dapat segera mengambil langkah-langkah pencegahan yang tepat.

Metode *Case Based Reasoning* (CBR) merupakan salah satu pendekatan yang sesuai untuk diterapkan dalam perancangan sistem pakar Diagnosis penyakit THT. Metode ini bekerja dengan cara menyimpan kasus-kasus lama yang pernah terjadi dan memanfaatkannya untuk menyelesaikan permasalahan baru yang memiliki karakteristik serupa. Dengan pendekatan ini, sistem mampu memberikan solusi berdasarkan pengalaman kasus sebelumnya, sehingga proses Diagnosis menjadi lebih akurat dan relevan (Dona et al., 2021).

Perancangan sistem pakar Diagnosis penyakit THT berbasis web dipilih karena platform web memiliki keunggulan dalam hal aksesibilitas yang luas, mudah diakses kapan saja dan di mana saja selama terkoneksi dengan jaringan internet. Hal ini akan sangat membantu masyarakat, terutama yang berada di wilayah dengan akses terbatas terhadap fasilitas kesehatan, untuk memperoleh informasi Diagnosis secara cepat dan efisien. Selain itu, sistem berbasis web memungkinkan pengelolaan data yang lebih terstruktur dan memudahkan proses update pengetahuan dalam sistem pakar.

Berdasarkan uraian di atas, penulis tertarik untuk melakukan penelitian dan pengembangan dengan judul **“Sistem Pakar Diagnosis Penyakit THT Menggunakan Metode *Case Based Reasoning* Berbasis Web”**. Penelitian ini

diharapkan dapat memberikan kontribusi dalam membantu masyarakat untuk mengetahui kondisi kesehatannya secara mandiri, serta mendukung tenaga medis dalam proses Diagnosis awal secara efektif dan efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem pakar yang mampu mendiagnosis penyakit Telinga, Hidung, dan Tenggorokan (THT) sehingga mudah digunakan oleh pengguna dan dapat memberikan informasi Diagnosis beserta langkah penanganannya?
2. Bagaimana mengimplementasikan metode *Case Based Reasoning* (CBR) ke dalam sistem pakar sehingga mampu mencocokkan kasus baru dengan kasus terdahulu untuk menghasilkan solusi Diagnosis yang akurat dan relevan?

1.3 Ruang Lingkup Permasalahan

Ruang lingkup permasalahan dari penelitian ini adalah sebagai berikut :

1. Sistem pakar yang dibangun menggunakan metode *Case Based Reasoning* (CBR) yang bekerja dengan mencocokkan kasus baru berdasarkan kemiripan gejala dengan kasus-kasus sebelumnya.

2. Jenis penyakit THT yang menjadi variabel dalam penelitian ini terdiri dari 11 penyakit, yaitu: *Otitis Eksterna*, *Otitis Media*, *Rhinitis* (alergi musiman), *Sinusitis*, Polip Hidung, Kanker *Nasofaring*, *Faringitis* (radang tenggorokan), *Tonsilitis* (radang amandel), *Laringitis* (radang pita suara), *Acute Upper Respiratory Infection* (infeksi saluran pernapasan), dan *Acute Nasopharyngitis* (common cold).
3. Metode penelitian yang digunakan adalah studi literatur, wawancara dengan Dokter Sp. THT, dan observasi terhadap data gejala-gejala penyakit THT yang relevan.
4. Sistem pakar dirancang berbasis web dengan menggunakan bahasa pemrograman PHP dan database MySQL, agar dapat diakses dengan mudah oleh pengguna.

1.4 Tujuan Penelitian

Tujuan dari Penelitian ini adalah sebagai berikut :

1. Untuk merancang dan membangun sistem pakar berbasis web yang dapat membantu proses Diagnosis penyakit Telinga, Hidung, dan Tenggorokan (THT) dengan menggunakan metode *Case Based Reasoning* (CBR).
2. Untuk mengimplementasikan metode *Case Based Reasoning* (CBR) dalam sistem pakar guna mencocokkan gejala penyakit baru dengan kasus-kasus yang pernah terjadi sebelumnya sehingga menghasilkan Diagnosis yang akurat dan relevan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Bagi Mahasiswa

- a. Memberikan referensi dan pengetahuan bagi mahasiswa tentang penerapan metode *Case Based Reasoning* (CBR) dalam perancangan sistem pakar, khususnya pada kasus Diagnosis penyakit.
- b. Menjadi contoh nyata penerapan teknologi informasi di bidang kesehatan, sehingga dapat memperluas wawasan mahasiswa mengenai integrasi kecerdasan buatan dalam penyelesaian masalah di dunia nyata.
- c. Sebagai bahan rujukan untuk penelitian lanjutan atau tugas akhir yang berkaitan dengan sistem pakar, metode CBR, maupun pengembangan aplikasi berbasis web di bidang kesehatan.

2. Bagi Masyarakat

- a. Membantu masyarakat dalam memperoleh informasi awal mengenai kemungkinan jenis penyakit THT yang dialami berdasarkan gejala yang dirasakan.
- b. Mempermudah masyarakat untuk melakukan Diagnosis awal secara mandiri dengan akses sistem berbasis web yang praktis dan dapat diakses kapan saja.

- c. Meningkatkan kesadaran masyarakat akan pentingnya melakukan pemeriksaan lebih lanjut ke tenaga medis profesional apabila gejala yang dirasakan mengarah pada kondisi serius.

1.6 Metode Pengumpulan Data

Tahapan pengumpulan data yang akan dilakukan adalah sebagai berikut :

1. Wawancara

Wawancara untuk penelitian adalah suatu metode pengumpulan data yang melibatkan interaksi langsung antara peneliti dan responden dengan tujuan untuk mendapatkan informasi yang relevan dengan topik penelitian. Wawancara dalam konteks penelitian dapat dilakukan dengan berbagai metode dan pendekatan tergantung pada jenis penelitian dan tujuan yang ingin dicapai.

2. Studi Pustaka

Studi pustaka adalah suatu kegiatan atau proses yang melibatkan pengumpulan, analisis, dan sintesis literatur atau bahan pustaka yang relevan dengan topik penelitian atau topik tertentu. Studi pustaka dilakukan untuk mendapatkan pemahaman menyeluruh tentang kerangka teoritis, konsep-konsep, temuan penelitian sebelumnya, dan perkembangan pengetahuan terkait dengan suatu bidang atau disiplin ilmu.

1.7 Sistematika Penulisan

Sistematika penulisan Skripsi ini disusun untuk memberikan gambaran umum tentang perangkat lunak yang akan dibuat. Sistematika penulisan Skripsi ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Berisi tentang deskripsi umum dari Skripsi ini yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan Skripsi, manfaat Skripsi dan sistematika penulisan.

BAB II. LANDASAN TEORI

Berisi mengenai dasar-dasar teori yang digunakan sebagai landasan dalam pembuatan aplikasi.

BAB III. METODOLOGI PENELITIAN

Memaparkan tentang metodologi penelitian, kerangka kerja penelitian dan analisis permasalahan.

BAB IV. ANALISIS DAN PERANCANGAN

Pada bab ini akan menganalisis cara kerja dan merancang aplikasi yang akan dibuat.

BAB V. TESTING DAN IMPLEMENTASI SISTEM

Bab ini berisi tentang bagaimana mengimplementasikan aplikasi berdasarkan analisis dan perancangan pada bab sebelumnya.

BAB VI. PENUTUP

Bab ini berisi tentang kesimpulan yang membahas hasil penelitian pada bab sebelumnya serta saran yang diambil untuk menghasilkan pemecahan masalah yang sudah dituangkan dalam perancangan.

BAB 2

LANDASAN TEORI

2.1 Sistem Pakar

Manusia sering dihadapkan pada permasalahan yang memerlukan pengetahuan dan pengalaman khusus untuk menemukan solusi yang tepat. Namun, tidak semua orang memiliki akses langsung kepada pakar atau ahli di bidangnya. Kebutuhan akan solusi yang cepat dan akurat mendorong perkembangan teknologi yang mampu membantu menyelesaikan permasalahan tersebut, salah satunya melalui penerapan sistem pakar. Sistem ini dirancang untuk mempermudah proses pengambilan keputusan dengan cara mengadopsi pengetahuan dan pengalaman seorang ahli ke dalam bentuk sistem yang dapat digunakan oleh siapa saja.

Sistem pakar merupakan suatu aplikasi komputerisasi yang berusaha menirukan proses penalaran dari seorang ahli dalam memecahkan masalah spesifik dan membuat suatu keputusan atau kesimpulan karena pengetahuannya disimpan didalam basis pengetahuan untuk diproses pemecahan masalah. Dasar dari sistem pakar bagaimana memindah kan pengetahuan yang dimiliki oleh seorang pakar ke komputer dan bagaimana membuat keputusan serta mengambil kesimpulan berdasarkan pengetahuan itu (Fahmi Ginting et al., 2023).

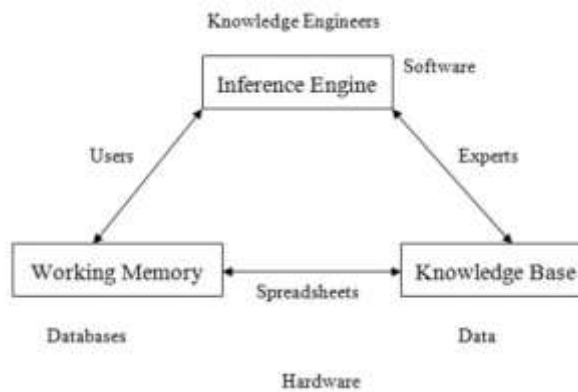
Menurut R. E. Putri et al. (2024), Sistem pakar adalah sebuah sistem berbasis teknologi yang dirancang untuk meniru kemampuan seorang pakar dalam menganalisis dan menyelesaikan permasalahan di bidang tertentu. Pakar yang dimaksud adalah individu dengan keahlian khusus yang tidak dimiliki oleh orang awam, seperti dokter, teknisi, atau profesional lainnya. Pengetahuan yang dimiliki pakar kemudian dikumpulkan dan disimpan dalam basis pengetahuan yang menjadi acuan bagi sistem untuk memberikan solusi. Karena berisi pengetahuan spesifik dari bidang tertentu, sistem pakar hanya dirancang untuk menangani jenis permasalahan yang sesuai dengan keahlian pakarnya.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa sistem pakar merupakan sistem berbasis komputer yang dirancang untuk meniru cara berpikir dan penalaran seorang ahli dalam menyelesaikan suatu permasalahan tertentu. Pengetahuan yang dimiliki oleh pakar dikumpulkan dan disimpan dalam basis pengetahuan, kemudian diproses oleh sistem untuk memberikan solusi, keputusan, atau kesimpulan terhadap masalah yang dihadapi. Dengan demikian, sistem pakar dapat membantu pengguna dalam memperoleh keputusan yang mendekati hasil analisis seorang ahli, khususnya pada bidang yang membutuhkan keahlian khusus.

2.2 Struktur Sistem Pakar

Struktur dasar dari sebuah sistem pakar umumnya terbagi ke dalam dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan

digunakan oleh pengembang sistem untuk memasukkan, mengelola, dan memodifikasi pengetahuan yang berasal dari pakar ke dalam sistem. Sementara itu, lingkungan konsultasi dirancang agar dapat digunakan oleh pengguna yang bukan ahli untuk memperoleh solusi atau informasi yang bersumber dari pengetahuan pakar yang telah dimasukkan ke dalam sistem tersebut.



Gambar 2.1 Struktur Sistem Pakar (Sumber : brainkart.com)

Menurut Alam et al. (2021), sistem pakar terdiri dari tiga komponen utama, yaitu *User Interface*, *Inference Engine*, dan *Knowledge Base*. *User Interface* merupakan antarmuka yang berfungsi sebagai penghubung antara pengguna dengan sistem pakar. Melalui *interface* ini, pengguna dapat memberikan input data, serta menerima informasi atau solusi yang disajikan oleh sistem. *User interface* memegang peran penting dalam memudahkan proses interaksi agar pengguna dapat mengikuti alur sistem dengan baik.

Komponen kedua adalah *Inference Engine*, yaitu bagian dari sistem yang berfungsi sebagai mesin penalaran. *Inference Engine* bertugas menganalisis data

masukan yang diberikan oleh pengguna dan mencocokkannya dengan aturan-aturan yang tersimpan dalam basis pengetahuan. Proses ini dilakukan secara sistematis, mengikuti pola urutan tertentu, hingga sistem dapat memberikan kesimpulan atau solusi. Terdapat dua pendekatan yang sering digunakan dalam inference engine, yaitu metode penelusuran maju (*forward chaining*) dan penelusuran mundur (*backward chaining*).

Komponen terakhir adalah *Knowledge Base* yang menjadi inti dari sistem pakar. Basis pengetahuan ini berisi kumpulan informasi, baik dalam bentuk fakta maupun aturan (rules), yang diperoleh dari pengalaman dan pengetahuan seorang pakar di bidang tertentu. *Knowledge Base* dapat dikembangkan dengan berbagai metode representasi, salah satunya adalah model berbasis aturan (*rule-based system*), di mana sistem akan memproses fakta-fakta yang dimasukkan oleh pengguna untuk menghasilkan kesimpulan baru berdasarkan aturan yang telah ditetapkan sebelumnya.

Berdasarkan uraian tersebut, dapat disimpulkan bahwa struktur sistem pakar terdiri dari beberapa komponen utama yang saling mendukung agar sistem dapat bekerja secara optimal dalam memberikan solusi terhadap suatu permasalahan. Sistem pakar memiliki dua lingkungan utama, yaitu lingkungan pengembangan yang digunakan untuk memasukkan serta mengelola pengetahuan dari pakar, dan lingkungan konsultasi yang digunakan oleh pengguna untuk memperoleh solusi dari sistem. Selain itu, sistem pakar juga didukung oleh tiga komponen penting yaitu User Interface, Inference Engine, dan Knowledge Base. User Interface berfungsi

sebagai media interaksi antara pengguna dengan sistem, Inference Engine berperan sebagai mesin penalaran yang memproses data masukan berdasarkan aturan yang ada, sedangkan Knowledge Base menjadi tempat penyimpanan seluruh pengetahuan pakar. Dengan adanya ketiga komponen tersebut, sistem pakar mampu meniru cara berpikir seorang ahli dalam menganalisis masalah dan menghasilkan keputusan yang tepat.

2.3 Diagnosa

Proses penentuan jenis penyakit berdasarkan gejala yang dialami pasien menjadi langkah awal yang sangat penting sebelum diberikan penanganan lebih lanjut dalam dunia kesehatan. Proses ini tidak hanya membutuhkan ketelitian, tetapi juga keahlian dalam menganalisis berbagai kemungkinan kondisi medis yang terjadi. Oleh karena itu, diagnosis memegang peran krusial dalam memastikan pasien mendapatkan pengobatan yang tepat sesuai dengan permasalahan kesehatan yang dihadapinya. Seiring berkembangnya teknologi, proses diagnosis kini tidak hanya dilakukan secara manual oleh tenaga medis, tetapi juga dapat dibantu dengan sistem berbasis kecerdasan buatan untuk meningkatkan ketepatan dan efisiensi.

Secara etimologi, diagnosis berasal dari bahasa Yunani, yaitu *gnosis* yang berarti ilmu pengetahuan. Sedangkan secara terminologi, pengertian diagnosis adalah penetapan suatu keadaan yang menyimpang atau keadaan normal melalui dasar pemikiran dan pertimbangan ilmu pengetahuan (Kusumah et al., 2022). Namun menurut Fuad et al. (2022), pengertian diagnosis merupakan suatu penetapan

keadaan yang menyimpang atau juga keadaan normal dengan melalui dasar pemikiran serta juga pertimbangan ilmu pengetahuan. Maksudnya, Tiap-tiap penyimpangan dari keadaan normal tersebut dikatakan ialah sebagai suatu keadaan abnormal/anomali/kelainan.

Sinurat (2021), memaparkan bahwa Diagnosis memiliki beberapa manfaat, antara lain:

1. Untuk menemukan atau mengidentifikasi kelemahan atau penyakit (*weakness, disease*) apa yang dialami seseorang.
2. Untuk menemukan karakteristik atau kesalahan-kesalahan atas gejala-gejala atau fakta tentang suatu hal.
3. Sebagai pertimbangan dalam upaya pengendalian penyakit di lapangan.
4. Salah satu upaya untuk mencegah dan menanggulangi penyebaran suatu penyakit atau wabah

2.4 Penyakit Telinga, Hidung, Tenggorokan (THT)

Penyakit yang menyerang Telinga, Hidung, dan Tenggorokan (THT) menjadi salah satu masalah kesehatan yang sering dialami oleh masyarakat dari berbagai kalangan usia. Ketiga organ ini memiliki keterkaitan erat dalam sistem pernapasan dan pendengaran, sehingga gangguan pada salah satunya sering memengaruhi fungsi organ lainnya. Faktor lingkungan, pola hidup, serta kebersihan yang kurang terjaga menjadi pemicu utama timbulnya berbagai keluhan di area THT, mulai dari infeksi

ringan hingga kondisi yang lebih serius. Oleh karena itu, penanganan yang tepat dan diagnosa yang cepat sangat dibutuhkan untuk mencegah komplikasi lebih lanjut.

Telinga, Hidung, dan Tenggorokan merupakan bagian tubuh manusia yang memiliki fungsi vital. Telinga adalah organ yang kompleks dengan fungsi sebagai alat pendengaran sekaligus menjaga keseimbangan tubuh. Hidung berfungsi sebagai bagian terluar dari sistem pernapasan dan berperan sebagai indra penciuman. Sementara itu, tenggorokan merupakan jalur yang menjadi tempat pertemuan antara saluran suara, jalur menelan, dan saluran pernapasan. Gangguan fungsi pada bagian-bagian tersebut dikenal sebagai penyakit Telinga, Hidung, dan Tenggorokan (THT). Penyakit THT mengakibatkan terganggunya fungsi penting seperti mendengar, bernapas, mencium aroma, berbicara, dan menelan makanan maupun minuman (Nurhayati et al., 2022).

Menurut Hutagalung & Sriani (2024), Penyakit THT merupakan penyakit yang disebabkan oleh gangguan pada organ telinga, hidung, dan tenggorokan. Ketiga organ ini memiliki peranan yang sangat vital bagi tubuh manusia, sehingga gangguan pada salah satu atau ketiganya dapat berdampak serius bagi kesehatan. Penyakit THT dapat menyerang siapa saja, mulai dari bayi, anak-anak, hingga orang dewasa. Telinga berfungsi sebagai organ pendengaran sekaligus menjaga keseimbangan tubuh. Hidung berperan sebagai indra penciuman dan merupakan bagian terluar dari sistem pernapasan. Sementara itu, tenggorokan memiliki fungsi penting sebagai jalur pita suara, saluran menelan, dan saluran pernapasan.

Penyakit THT sering kali disebabkan oleh infeksi kuman, namun ada pula yang diakibatkan oleh kelainan perkembangan sel tubuh yang dapat berkembang menjadi tumor atau kanker. Meskipun termasuk jenis penyakit yang dapat berdampak serius, banyak masyarakat yang masih menganggap remeh penyakit THT. Kurangnya pengetahuan dan informasi mengenai gejala serta penanganan yang tepat menyebabkan banyak kasus penyakit THT tidak ditangani dengan baik, sehingga berisiko menimbulkan komplikasi yang lebih parah.

Seringkali, seseorang sudah merasakan adanya gangguan kesehatan dari gejala yang muncul, namun tidak mengetahui secara pasti jenis penyakit apa yang sedang dialaminya dan bagaimana cara menanganinya. Hal ini menunjukkan pentingnya ketersediaan informasi yang mudah diakses mengenai penyakit THT, agar masyarakat dapat memahami kondisi kesehatannya secara dini dan mendapatkan penanganan yang sesuai.

Jenis penyakit Telinga, Hidung, dan Tenggorokan (THT) yang menjadi variabel dalam penelitian ini terdiri dari 11 penyakit, yaitu: *Otitis Eksterna*, *Otitis Media*, *Rhinitis* (alergi musiman), *Sinusitis*, *Polip Hidung*, *Kanker Nasofaring*, *Faringitis* (radang tenggorokan), *Tonsilitis* (radang amandel), *Laringitis* (radang pita suara), *Acute Upper Respiratory Infection* (infeksi saluran pernapasan atas), dan *Acute Nasopharyngitis (common cold)*. Masing-masing penyakit tersebut memiliki gejala khas yang dapat dijadikan indikator dalam proses diagnosis . Secara umum, gejala-gejala yang sering muncul meliputi nyeri pada telinga, penurunan pendengaran,

hidung tersumbat, keluarnya cairan dari hidung atau telinga, nyeri tenggorokan, kesulitan menelan, suara serak, batuk, demam, dan rasa penuh di sekitar wajah. Gejala-gejala tersebut akan dianalisis dalam sistem pakar untuk membantu menentukan jenis penyakit THT yang diderita oleh pengguna berdasarkan data kasus yang pernah terjadi sebelumnya.

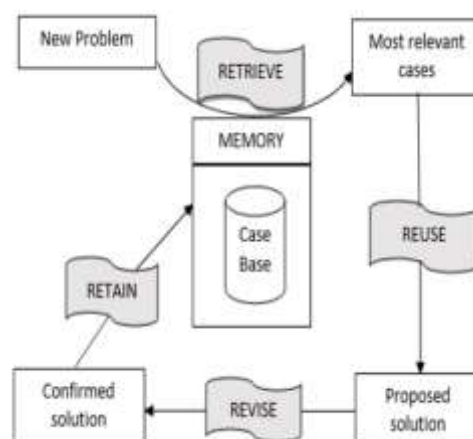
2.5 *Case Based Reasoning (CBR)*

Manusia sering menyelesaikan permasalahan dengan mengingat kembali pengalaman yang pernah dialami sebelumnya. Pola berpikir ini menjadi dasar munculnya pendekatan *Case Based Reasoning (CBR)*, di mana pengalaman dari kasus-kasus terdahulu dijadikan acuan dalam menangani permasalahan baru yang memiliki kemiripan. Pendekatan ini dianggap lebih fleksibel dan adaptif, karena tidak memerlukan aturan yang kaku, melainkan bergantung pada kemiripan antar kasus yang ada. Oleh karena itu, CBR banyak diterapkan dalam sistem pakar untuk membantu proses pengambilan keputusan dengan cepat dan efisien.

Case Based Reasoning (CBR) adalah suatu model penalaran yang menyelesaikan permasalahan dengan cara mengingat dan memanfaatkan pengalaman dari kasus-kasus yang pernah terjadi sebelumnya. CBR bekerja dengan mencari kemiripan antara masalah baru yang dihadapi dengan kasus lama yang sudah ada di dalam sistem, lalu menggunakan solusi dari kasus tersebut untuk menyelesaikan permasalahan baru. Proses ini tidak hanya mengandalkan pencarian kasus yang relevan, tetapi juga melibatkan tahapan adaptasi, yaitu penyesuaian solusi agar sesuai

dengan kondisi permasalahan saat ini. Selain itu, CBR juga memiliki kemampuan untuk menyimpan pengalaman baru ke dalam basis kasus (case base), sehingga semakin banyak kasus yang tersimpan, semakin kaya pula pengetahuan yang dimiliki oleh sistem. Konsep CBR banyak digunakan dalam perancangan sistem pakar di berbagai bidang, karena mampu memadukan proses pemecahan masalah, pembelajaran, dan pengolahan memori secara bersamaan (Harahap et al., 2022).

Case Based Reasoning (CBR) adalah model penalaran yang menggabungkan proses pemecahan masalah, pemahaman, dan pembelajaran dengan memanfaatkan pengalaman dari kasus-kasus sebelumnya yang telah disimpan dalam sistem. Konsep utama dari CBR adalah menyelesaikan permasalahan baru dengan mencari kemiripan dari kasus-kasus yang pernah terjadi, kemudian menggunakan pengetahuan tersebut untuk memberikan solusi yang sesuai. Proses dalam CBR terdiri dari siklus yang dibagi menjadi empat tahap, yaitu *Retrieve*, *Reuse*, *Revise*, dan *Retain* (Nur et al., 2022).



**Gambar 2.2 Tahapan Proses Case Based Reasoning (CBR) (sumber :
Nur et al. (2022))**

1. Retrieve

Retrieve adalah proses pencarian kasus lama yang memiliki kemiripan dengan kasus baru yang sedang dihadapi. Kasus-kasus tersebut diambil dari basis kasus (case base) untuk dijadikan referensi solusi. Beberapa metode yang umum digunakan dalam proses retrieval antara lain algoritma pohon keputusan (decision tree) dan algoritma nearest neighbor.

2. Reuse

Reuse merupakan tahap penggunaan kembali solusi dari kasus lama untuk diterapkan pada kasus baru. Karena kasus baru biasanya tidak sepenuhnya identik dengan kasus yang ada di basis data, maka solusi dari kasus lama perlu disesuaikan agar relevan dengan kondisi kasus baru.

3. Revise

Revise adalah proses evaluasi dan perbaikan terhadap solusi yang dihasilkan. Pada tahap ini, solusi yang diadaptasi akan dikaji ulang, baik melalui tanggapan dari pakar maupun melalui pengujian secara langsung. Jika terdapat ketidaksesuaian, solusi akan diperbaiki dengan metode adaptasi, seperti substitution (penggantian), transformation (transformasi), atau reinstantiation (mengkopi dan menyesuaikan solusi dari kasus sebelumnya).

4. Retain

Retain adalah tahap penyimpanan solusi baru yang telah berhasil divalidasi ke dalam case base. Proses ini bertujuan agar solusi dari kasus baru tersebut dapat menjadi referensi untuk menangani kasus serupa di masa mendatang.

Harahap et al. (2022), memaparkan bahwa pada saat terjadi permasalahan baru, pertama-tama sistem akan melakukan proses *retrieve*. Proses *retrieve* akan melakukan dua langkah pemrosesan, yaitu pengenalan masalah dan pencarian persamaan masalah pada database. Setelah proses *retrieve* selesai dilakukan, selanjutnya sistem akan melakukan proses *reuse*. Di dalam proses *reuse*, sistem akan menggunakan informasi permasalahan sebelumnya yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru. Pada proses *reuse* akan menyalin, menyeleksi, dan melengkapi informasi yang akan digunakan.

Selanjutnya pada proses *revise*, informasi tersebut akan dikalkulasi, dievaluasi, dan diperbaiki kembali untuk mengatasi kesalahan-kesalahan yang terjadi pada permasalahan baru. Pada proses terakhir, sistem akan melakukan proses *retain*. Proses *retain* akan mengindeks, mengintegrasikan, dan mengekstrak solusi yang baru tersebut ke dalam database. Selanjutnya, solusi baru itu akan disimpan ke dalam basis pengetahuan (knowledge-base) untuk menyelesaikan permasalahan yang akan datang. Tentunya, permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan dengannya.

2.6 Alat Bantu Perancangan

2.6.1 *Unified Modeling Language (UML)*

Dalam proses pengembangan perangkat lunak, pemodelan sistem menjadi langkah penting untuk menggambarkan rancangan dan alur kerja dari sistem yang akan dibangun. Pemodelan ini tidak hanya mempermudah komunikasi antara pengembang dan pemangku kepentingan, tetapi juga membantu memahami struktur dan perilaku sistem secara menyeluruh sebelum diimplementasikan. Salah satu pendekatan yang sering digunakan untuk memvisualisasikan desain sistem secara terstruktur dan standar adalah dengan menggunakan *Unified Modeling Language* (UML), yang mampu merepresentasikan kebutuhan fungsional maupun non-fungsional dari sebuah sistem.

Unified Modeling Language (UML) merupakan sebuah bahasa pemodelan yang divisualisasikan dalam bentuk gambar atau grafik, yang berfungsi untuk memberikan gambaran dan spesifikasi dalam proses pembangunan serta dokumentasi perancangan sistem berbasis objek (object-oriented). UML menyediakan standar dalam pembuatan blueprint sistem yang mencakup berbagai aspek, seperti perancangan proses bisnis, pembuatan class yang dapat diimplementasikan ke dalam bahasa pemrograman tertentu, desain basis data, hingga komponen-komponen penting lainnya yang dibutuhkan dalam proses perancangan sistem (Siska Narulita et al., 2024).

UML merupakan singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Saat ini, UML banyak digunakan dalam perancangan sistem berbasis objek karena mampu mempermudah proses analisis dan desain secara

terstruktur. UML menyediakan seperangkat diagram yang berfungsi untuk menggambarkan berbagai sudut pandang dari sistem yang sedang dikembangkan, sehingga memudahkan dalam mendokumentasikan kebutuhan, proses, dan rancangan sistem secara visual (Arianti et al., 2022).

UML merupakan kumpulan dari berbagai jenis diagram yang digunakan untuk memodelkan sistem secara visual. Setiap diagram berperan dalam menggambarkan permasalahan yang dihadapi serta solusi yang dirancang untuk menyelesaikannya. Secara umum, UML memiliki sembilan jenis diagram yang dapat digunakan dalam proses perancangan sistem. Namun, dalam penelitian ini hanya akan difokuskan pada empat jenis diagram yang dianggap paling relevan untuk menggambarkan alur dan struktur sistem yang dibangun.

1. *Use Case*

Use Case Diagram adalah diagram yang digunakan untuk menggambarkan hubungan antara aktor (pengguna atau sistem lain) dengan fungsi-fungsi (*use case*) yang disediakan oleh sistem. Diagram ini menunjukkan bagaimana interaksi terjadi antara aktor dan sistem dari sudut pandang pengguna, serta membantu menentukan ruang lingkup sistem melalui batasan (*system boundary*) yang memisahkan aktivitas di dalam dan di luar sistem (Wayahdi & Ruziq, 2023).

Tabel 2.1 Simbol Use Case (sumber : Indah Nurlita & Reni Anggraini,(2023))

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara

			objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2. Activity Diagram

Activity Diagram adalah sebuah diagram yang digunakan untuk menggambarkan alur kerja atau aliran aktivitas dalam suatu sistem, baik

berupa aliran data maupun aliran kontrol. Diagram ini merepresentasikan urutan aktivitas yang terjadi, mulai dari awal hingga proses berakhir, termasuk kondisi percabangan, perulangan, serta parallel proses yang mungkin terjadi di dalam sistem. *Activity Diagram* memvisualisasikan langkah-langkah proses bisnis atau proses sistem secara rinci dan terstruktur, sehingga mempermudah dalam memahami bagaimana suatu proses berjalan dan bagaimana aktivitas-aktivitas tersebut saling berinteraksi dalam sebuah aliran kerja yang logis (Arianti et al., 2022).

Tabel 2.2 Simbol *Activity Diagram* (sumber: Indah Nurlita & Reni Anggraini, 2023)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

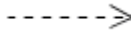
(sumber: Indah Nurlita & Reni Anggraini, 2023)

3. *Class Diagram*

Class Diagram adalah sebuah diagram yang digunakan untuk memodelkan struktur statis dari suatu sistem dengan menunjukkan hubungan antar class yang ada di dalamnya. Setiap class merepresentasikan blueprint dari objek, yang berisi atribut (ciri-ciri atau data) dan fungsi (operasi atau metode) yang dimiliki oleh objek tersebut. *Class Diagram* memvisualisasikan bagaimana class saling berhubungan melalui asosiasi, pewarisan (*inheritance*), agregasi, atau komposisi, sehingga membantu dalam memahami struktur data dan interaksi antar komponen dalam sistem yang sedang dikembangkan (Arianti et al., 2022).

Tabel 2.3 Simbol *class diagram* (sumber: Indah Nurlita & Reni Anggraini, 2023))

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<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
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

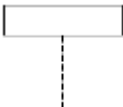
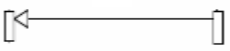
(sumber: Indah Nurlita & Reni Anggraini, 2023)

4. *Sequence Diagram*

Sequence Diagram adalah diagram yang digunakan untuk menggambarkan alur interaksi antar objek dalam suatu *use case* secara

berurutan dari waktu ke waktu. Diagram ini menunjukkan objek-objek yang terlibat dan bagaimana pesan atau message dikirimkan di antara objek tersebut selama proses berlangsung. *Sequence Diagram* memodelkan hubungan antar objek dan memperlihatkan urutan komunikasi yang terjadi dalam rangka menjalankan sebuah fungsi atau skenario tertentu dalam sistem (Annisa Tri Hidayati et al., 2023).

Tabel 2.4 Simbol *Sequence Diagram* (sumber: Indah Nurlita & Reni Anggraini, 2023))

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

(sumber: Indah Nurlita & Reni Anggraini, 2023)

2.7 Bahasa Pemrograman

2.7.1 *Hyper Text Markup Language (HTML)*

HTML (*Hyper Text Markup Language*) adalah bahasa markah standar yang digunakan untuk membuat dan menyusun halaman web. Pengembangan dan penggunaan HTML memiliki latar belakang yang penting dan beragam aspek yang telah mempengaruhi evolusi web seperti yang kita kenal sekarang. Menurut Permatasari & Suhendi (2020), HTML merupakan bahasa web standar yang dikelola oleh World Wide Web Consortium (W3C) dalam bentuk tag yang mengatur elemen-elemen sebuah situs web.

Selanjutnya menurut Sama & Hartanto (2021), HTML (*Hypertext Markup Language*) merupakan gabungan dua istilah: *Hypertext* dan *Markup Language*. *Hypertext* adalah dokumen yang berisi link yang memungkinkan pengguna terhubung ke halaman lain. Bahasa *markup* adalah bahasa komputer yang terdiri dari set kode untuk mengatur struktur dan merepresentasikan informasi. Namun menurut Mira Orisa et al. (2023), HTML adalah dasar dari seluruh konten yang ditampilkan di World Wide Web (WWW) dan terdiri dari dua bagian penting seperti konten, informasi dan serangkaian instruksi yang memberi tahu komputer cara menampilkan konten ini.

2.7.2 *Cascading Style Sheets (CSS)*

CSS (*Cascading Style Sheets*) adalah teknologi penting dalam pengembangan web yang digunakan untuk mengontrol tampilan dan tata letak halaman web. Sejak diperkenalkan, CSS telah berkembang menjadi alat yang sangat diperlukan bagi pengembang web untuk menciptakan desain yang menarik, konsisten, dan responsif. Menurut Permatasari & Suhendi (2020), CSS adalah singkatan dari *Cascading Style Sheet*, sebuah dokumen web yang mengatur elemen HTML dengan properti berbeda sehingga dapat ditampilkan dalam gaya berbeda yang Anda inginkan.

Selanjutnya Kurniawan et al. (2023) memaparkan bahwa CSS adalah merupakan sebuah gaya bahasa *sheet* yang dipakai guna membuat pengaturan terhadap tampilan dari sebuah dokumen yang ditulis menggunakan bahasa markup. Menurut Mira Orisa et al. (2023) CSS merupakan bahasa pemrograman markup (HTML) yang sering digunakan untuk membuat halaman web menjadi lebih menarik.

2.7.3 Javascript

Dalam proses pembuatan aplikasi berbasis web, diperlukan teknologi yang mampu menghadirkan tampilan dan fungsi yang interaktif agar pengguna mendapatkan pengalaman yang lebih dinamis saat berinteraksi dengan sistem. Salah satu teknologi yang sering digunakan untuk memenuhi kebutuhan tersebut adalah *JavaScript*. Melalui *JavaScript*, pengembang dapat menambahkan berbagai fitur interaktif pada halaman web, seperti validasi form, animasi, hingga proses komunikasi data secara real-time tanpa harus memuat ulang halaman.

JavaScript merupakan bahasa script yang mehidupkan halaman-halaman HTML. *JavaScript* dapat dijalankan hampir pada semua platform. *JavaScript* sangat identik dengan browser, sehingga sangat terintegrasi dengan HTML. Ketika browser memuat suatu halaman, server akan mengirimkan konten utuh dari dokumen termasuk HTML dan statemen-statemen dari *JavaScript*. Konten HTML kemudian dibaca baris demi baris hingga sampai pada pembacaan *JavaScript*, pada saat itu interpreter *JavaScript* akan mengambil alih. Ketika tag penutup *JavaScript* diraih, pemrosesan HTML kemudian berlanjut (Putawa, 2022).

JavaScript pertama kali digunakan oleh Netscape pada tahun 1995. Pada awalnya, bahasa ini dinamakan "LiveScript" dan digunakan sebagai bahasa untuk browser Netscape Navigator 2. *JavaScript* adalah bahasa yang terdiri dari kumpulan kode yang berfungsi untuk dieksekusi pada dokumen HTML. Seiring berjalannya sejarah internet, *JavaScript* menjadi bahasa skrip pertama yang diimplementasikan untuk pengembangan web. Selain itu, bahasa pemrograman ini memungkinkan penggunaan perintah peristiwa untuk meningkatkan kemampuan bahasa HTML, memberikan fleksibilitas dan interaktivitas yang lebih besar dalam pengembangan aplikasi web (Christian & Voutama, 2024).

2.7.4 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah salah satu bahasa pemrograman yang paling populer dan banyak digunakan untuk pengembangan web di seluruh dunia. Sejak pertama kali diperkenalkan, PHP telah mengalami berbagai evolusi dan

perbaikan yang menjadikannya sebagai salah satu bahasa server-side yang andal dan fleksibel. Menurut Yoraeni et al. (2020), PHP yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan HTML.

Selanjutnya menurut Suli & Nirsal (2023), PHP merupakan pemograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. Nemun menurut Waidah (2024), PHP merupakan salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. ketika di panggil dari web browser, program yang ditulis dengan PHP akan diparsing di dalam web server oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser.

2.7.5 Structured Query Language (SQL)

Structured Query Language atau yang disingkat SQL adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen data. SQL merupakan bahasa pemrograman khusus yang digunakan untuk memanajemen data dalam RDBMS. SQL biasanya berupa perintah sederhana yang berisi instruksi-instruksi untuk manipulasi dan pengambilan

data pada relational database atau database yang terstruktur. Perintah SQL ini sering juga disingkat dengan sebutan 'query' (Kalsum Siregar et al., 2024).

SQL merupakan bahasa standar yang terstruktur untuk digunakan dalam basis data dalam menyimpan, memanipulasi dan mengambil data. SQL menjadi standar dalam *American National Standards Institute* pada tahun 1986, serta dalam *International Organization of Standardization* tahun 1987. SQL dapat menjadi alat untuk analisis data karena kemampuannya dalam mengolah data yang banyak dalam hitungan detik (Maringka et al., 2021).

2.8 Alat Bantu Pemrograman

2.8.1 *My Structured Query Language (MySQL)*

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang sangat populer dan banyak digunakan di seluruh dunia. Dikembangkan oleh *Oracle Corporation*, MySQL menyediakan platform yang kuat untuk penyimpanan dan pengelolaan data dalam berbagai aplikasi, baik itu untuk skala kecil hingga besar. Menurut Lim & Silalahi (2023) MySQL merupakan software RDBMS (*database server*) yang memungkinkan *database* dikelola dengan sangat cepat dan dapat menampung data dalam jumlah besar, sehingga dapat diakses dan disinkronkan oleh banyak pengguna.

Selanjutnya menurut Alit et al. (2020), MySQL adalah perangkat lunak sistem manajemen DBMS multi-pengguna. MySQL merupakan *database* yang sangat

populer di kalangan pemrogram web karena sifat yang open source dan kemampuannya menampung kapasitas yang sangat besar. MySQL juga merupakan program server *database* yang memungkinkan Anda mengirim dan menerima data untuk banyak pengguna dengan sangat cepat menggunakan perintah SQL (*Structured Query Language*) standar. Namun menurut Suhartini et al. (2020) MySQL adalah database SQL multi-threaded dan multi-pengguna atau perangkat lunak DBMS atau perangkat lunak sistem manajemen. MySQL sebenarnya berasal dari salah satu konsep terpenting dalam database untuk seleksi atau pemilihan dan pemasukan data, sehingga pengoperasian data dapat dilakukan dengan mudah dan otomatis.

2.8.2 Laragon

Laragon merupakan solusi ideal bagi pengembang yang membutuhkan lingkungan server lokal yang ringan dan fleksibel. Laragon adalah perangkat lunak yang bersifat open source (terbuka) yang dapat mendukung banyak sekali sistem operasi di mana Laragon bertugas sebagai server virtual atau sering disebut sebagai localhost. Selain itu, Laragon adalah perangkat lunak gratis yang mencakup beberapa sistem operasi sebagai localhost atau server mandiri. Laragon menyediakan banyak layanan, alat, dan fitur termasuk Apache, PHP Server, PhpMyAdmin, MySQL, Memcached, Redis, Composer, Xdebug, Cmdre, dan Laravel. Dengan fitur-fitur ini, Laragon menjadi pilihan populer bagi pengembang untuk membangun dan menguji aplikasi secara efisien (Zahra et al., 2024).

Laragon adalah perangkat lunak yang bersifat open source (terbuka) yang dapat mendukung berbagai sistem operasi, di mana Laragon berfungsi sebagai server virtual atau yang sering disebut sebagai localhost. Laragon juga memungkinkan pengguna untuk menggunakan domain sesuai keinginan, atau dikenal dengan istilah pretty URL's. Aplikasi ini sangat baik digunakan untuk pengelolaan dan pengembangan aplikasi berbasis *Website* secara efisien (Andarsyah et al., 2022).

2.8.3 Website

Website adalah suatu halaman atau kumpulan halaman yang terdiri dari konten digital yang dapat diakses melalui jaringan internet. Sejak pertama kali ditemukan pada tahun 1990-an, *Website* telah mengalami perkembangan yang signifikan dan menjadi salah satu bagian integral dari kehidupan digital modern. Menurut Suhartini et al. (2020) *Website* merupakan sebuah kumpulan halaman-halaman web beserta file-file pendukungnya, seperti file gambar, video, dan file digital lainnya yang disimpan pada sebuah web server yang umumnya dapat diakses melalui internet. Atau dengan kata lain, *Website* adalah sekumpulan folder dan file yang mengandung banyak perintah dan fungsi fungsi tertentu, seperti fungsi tampilan, fungsi menangani penyimpanan data dan sebagainya.

Menurut Arthalita & Prasetyo (2020) *Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *Website* disebut dengan web page dan link dalam *Website* memungkinkan pengguna bisa berpindah dari satu page ke page lain (hyper text), baik diantara page

yang disimpan dalam server yang sama maupun server diseluruh dunia. Selanjutnya menurut Permatasari & Suhendi(2020) *Website* dapat didefinisikan sebagai kumpulan halaman yang berisi informasi data digital berupa teks, gambar, animasi, suara dan video, atau kombinasi keduanya, yang disediakan dan dapat diakses melalui koneksi Internet bahkan dari sudut pandang orang-orang di seluruh dunia.

2.8.4 *Visual Studio Code*

Visual Studio Code adalah kode editor sumber terbuka yang dikembangkan oleh Microsoft untuk Windows, Linux, dan macOS. *Visual Studio Code* memudahkan dalam penulisan kode dengan mendukung berbagai jenis bahasa pemrograman serta memberikan variasi warna yang sesuai dengan fungsi dari setiap bagian kode tersebut. Selain itu, *Visual Studio Code* memiliki fitur penambahan ekstensi, sehingga pengembang dapat menambahkan fitur tambahan yang tidak tersedia secara default di dalam aplikasi. *Visual Studio Code* bersifat open source, yaitu aplikasi dengan source code yang dapat dilihat dan dikembangkan oleh siapa saja. Kode sumber *Visual Studio Code* juga dapat diakses melalui GitHub, menjadikan aplikasi ini memiliki banyak kontributor dan penggemar dalam pengembangan fitur-fitur barunya di masa depan (Wilyanto et al., 2023).

Visual Studio Code dipilih sebagai alat pengembangan karena kemudahan penggunaannya serta dukungannya terhadap berbagai bahasa pemrograman web seperti HTML, CSS, JavaScript, dan PHP, yang

memungkinkan pembuatan sistem enkripsi dan dekripsi secara efisien dan interaktif (Ermawita et al., 2025).

2.8.5 Model View Controller (MVC)

Model View Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk*, *Trygve Reenskaug*, untuk mengenkapsulasi data beserta pemrosesannya (*model*), dan mengisolasinya dari proses manipulasi (*controller*) serta tampilan (*view*) yang direpresentasikan melalui *User interface*. MVC menjadi salah satu konsep yang cukup populer dalam pembangunan aplikasi web. Konsep ini memisahkan pengembangan aplikasi berdasarkan komponen utama yang menyusun sebuah aplikasi, yaitu manipulasi data (*model*), antarmuka pengguna (*view*), dan bagian yang mengatur alur aplikasi (*controller*) (Sihombing & Rukmana Wahab, 2021).

Menurut Salam et al. (2023), ada tiga jenis komponen utama dalam aplikasi berbasis *Model View Controller* (MVC), yaitu:

1. Model

Model adalah kelas yang berisi logika proses dan perangkat lunak terkait. Model merupakan objek yang tidak memiliki informasi tentang antarmuka pengguna. Model juga berisi metode atau fungsi yang digunakan untuk menyimpan data dan aturan bisnis yang relevan dengan sistem.

2. View

View adalah kumpulan kelas yang mewakili elemen antarmuka pengguna. Tampilan (*view*) memiliki nama yang digunakan untuk mengidentifikasi file skrip tampilan ketika dipanggil oleh fungsi rendering. Nama *view* biasanya sama dengan nama file skrip *view*-nya.

3. *Controller*

Controller adalah kelas yang berfungsi sebagai penghubung antara model dan *view*. *Controller* bertugas mengelola komunikasi dan aliran data dari model ke *view*, serta sebaliknya. *Controller* memiliki fungsi standar yang dijalankan ketika permintaan pengguna tidak secara spesifik menentukan fungsi tertentu. Dengan demikian, *controller* menjadi pusat kendali alur kerja dalam pola arsitektur MVC.

2.9 Laravel

Laravel merupakan salah satu *framework* berbasis bahasa pemrograman PHP yang saat ini memiliki komunitas dan pengguna yang terus berkembang. Laravel mengusung konsep *Model View Controller* (MVC), yaitu konsep modern yang memisahkan bagian tampilan (*front-end*) dengan bagian pengelolaan data atau kontrol aplikasi (*back-end*). Laravel memiliki berbagai kelebihan, salah satunya adalah fitur routing yang berfungsi untuk mengatur rute permintaan (*request*) sesuai kebutuhan aplikasi. Selain itu, Laravel menyediakan query builder dan ORM (*Object Relational Mapping*) yang memudahkan dalam pengelolaan database dan mendukung berbagai jenis database (Fauzi & Darmawan, 2023).

Laravel juga telah terintegrasi dengan *Composer*, yaitu *tools* manajemen dependensi yang mempermudah pengembang untuk menambahkan modul atau *library* tambahan sesuai kebutuhan, sekaligus dapat memperbarui modul tersebut secara otomatis. Kelebihan lainnya, Laravel memiliki komunitas yang sangat aktif sehingga banyak tersedia *library* tambahan yang dapat diunduh dan diintegrasikan dengan aplikasi. Selain fitur-fitur tersebut, Laravel masih menyediakan banyak kemudahan lainnya yang mendukung proses perancangan dan pengembangan aplikasi secara lebih efisien.

2.10 *Black Box Testing*

Metode *Black box Testing* adalah metode pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, di mana tester mendefinisikan kumpulan kondisi input dan melakukan pengujian terhadap spesifikasi fungsional program. Proses *Black box Testing* dilakukan dengan cara mencoba program yang telah dibuat, kemudian memasukkan data pada setiap form untuk memastikan apakah program berjalan sesuai dengan kebutuhan. Pengujian ini penting dilakukan untuk memastikan bahwa perangkat lunak berfungsi sebagaimana mestinya dan memenuhi kebutuhan pengguna atau perusahaan (Shadiq et al., 2021).

Metode *Black Box Testing* merupakan pengujian yang bertujuan untuk menemukan kesalahan pada sistem aplikasi, seperti kesalahan pada fungsi sistem maupun menu aplikasi yang tidak berjalan atau hilang. *Black Box Testing* berfokus pada pengujian fungsionalitas sistem aplikasi tanpa memerhatikan struktur internal

dari program. Pengujian dilakukan dengan memberikan masukan data secara acak untuk memastikan hasil keluaran yang sesuai. Dikatakan pasti apabila data input yang salah akan ditolak oleh sistem informasi atau tidak disimpan ke dalam database. Sebaliknya, apabila data input yang dimasukkan benar, maka sistem akan menerima dan menyimpannya ke dalam database dengan benar (Uminingsih et al., 2022).

Pengujian menggunakan metode *Blackbox Testing* memiliki dua teknik, yaitu *Equivalence Partitioning* dan *Boundary Value Analysis*. Pada penelitian ini, teknik yang digunakan adalah *Equivalence Partitioning*, di mana proses pengujiannya meliputi penentuan *test case*, penentuan kriteria, pendefinisian partisi, pembuatan data uji, pembuatan kasus uji, serta diakhiri dengan tahap pengujian dan evaluasi.

2.11 Penelitian Terdahulu

Tabel 2.5 Penelitian Terdahulu

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
1	Sistem Pakar Identifikasi Masalah Kulit Wajah Menggunakan Metode <i>Case based</i>	Zahrotul Mustafidah, Annisaa Utami (2024)	Masalah Kulit Wajah (MCM Beauty Center Ajibarang)	<i>Case Based Reasoning</i> (CBR), Nearest Neighbor	Sistem mampu membantu diagnosis kulit wajah secara cepat dengan akurasi 93%

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
	<i>reasoning</i>				dari 30 data uji, mempermudah konsultasi tanpa tatap muka langsung.
2	Sistem Pakar dalam Mendiagnosis Gizi Buruk pada Balita dengan Menggunakan Metode CBR	Sandi Alam, Gunadi Widi Nurcahyo (2022)	Gizi Buruk pada Balita	<i>Case Based Reasoning</i> (CBR)	Sistem mampu mendiagnosis 8 jenis penyakit gizi buruk dengan akurasi sangat baik, mempermudah identifikasi gejala secara cepat dan akurat.

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
3	Penerapan Metode <i>Case Based Reasoning</i> (CBR) Dalam Sistem Pakar Untuk Menentukan Diagnosa Penyakit Pada Tanaman Hidroponik	Irfan Nugraha, Muhammad Siddik (2020)	Penyakit Tanaman Hidroponik	<i>Case Based Reasoning</i> (CBR), k- Nearest Neighbor	Sistem pakar berhasil membantu mendiagnosa penyakit pada tanaman hidroponik dengan solusi yang akurat dan konsisten, mempermudah identifikasi penyakit tanpa harus bergantung pada pakar secara langsung.

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
4	Sistem Pakar Diagnosa Penyakit Flutd (Gangguan Saluran Kemih) Pada Kucing Menggunakan Metode <i>Case based reasoning</i>	Kiki Adilianti, Novriyenni, Hermansyah Sembiring (2024)	Kucing (penyakit FLUTD)	<i>Case Based Reasoning</i> (CBR)	Sistem pakar berhasil membantu mendiagnosis penyakit FLUTD pada kucing berdasarkan gejala. Dibangun menggunakan PHP dan MySQL.
5	Sistem Pakar Menggunakan Metode <i>Case Based Reasoning</i> dalam Akurasi	Surya Aulia Rahman, Sumijan (2021)	Penyakit pada manusia akibat bakteri Staphylococcus aureus	<i>Case Based Reasoning</i> (CBR)	Sistem pakar mempermudah pasien dan dokter dalam mendiagnosis penyakit ini,

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
	Penyakit Disebabkan oleh Bakteri Staphylococcus Aureus				dengan bobot gejala ringan- sedang-berat
6	Sistem Pakar Diagnosa Penyakit Kambing Menggunakan Metode <i>Case Based Reasoning</i> untuk Kesehatan Ternak	Hikmah Tiar Alamsyah, Intan Nur Farida, Danang Wahyu Widodo (2024)	Kambing	<i>Case Based Reasoning</i> (CBR), Waterfall	Sistem berhasil memberikan diagnosis cepat dan akurat untuk penyakit kambing dan membantu manajemen kesehatan ternak.
7	Sistem Pakar Untuk Mendiagnosa	Rozi Eka Putri, Indra Kanedi,	Angular Cheilitis	<i>Case Based Reasoning</i>	Sistem memberikan kemudahan

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
	Penyakit Angular Cheilitis Menggunakan Metode <i>Case Based Reasoning</i> (CBR)	Yupianti (2024)		(CBR), Waterfall	dalam diagnosa awal Angular Cheilitis dengan hasil berupa persentase kemiripan kasus.
8	Sistem Pakar Mendiagnosa Penyakit Trombosis Vena Dalam Menggunakan Metode <i>Case based reasoning</i>	Riska Gustiara, Hendra Jaya, Ismawardi Santoso (2022)	Deep Vein Thrombosis (DVT)	<i>Case Based Reasoning</i> (CBR)	Sistem pakar menghasilkan diagnosa cepat dan akurat berdasarkan gejala dengan bantuan algoritma Nearest Neighbor dan similarity

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
					score.
9	Sistem Pakar Diagnosis Kerusakan Amplifier dengan Metode <i>Case Based Reasoning</i> (CBR)	Mohamad Inung Pangestu, Danar Putra Pamungkas, Resty Wulaningrum (2021)	Kerusakan pada amplifier	<i>Case Based Reasoning</i> (CBR)	Tingkat keberhasilan diagnosa mencapai 80% dalam 15 pengujian, memberikan solusi berdasarkan gejala kerusakan.
10	Sistem Pakar Memilih Tipe Belajar Anak Retardasi Mental Menggunakan Metode <i>Case based</i>	Khairunnisa Samosir, Dasril Aldo, Yeyi Gusla Nengsih (2021)	Anak dengan retardasi mental	<i>Case Based Reasoning</i> (CBR)	Sistem memberikan rekomendasi tipe belajar berdasarkan similarity terhadap kasus sebelumnya,

NO	Judul	Peneliti (Tahun)	Objek	Metode	Hasil Penelitian
	<i>reasoning</i>				hasil berupa persentase kemiripan.

BAB 3

METODE PENELITIAN

3.1 Pendahuluan

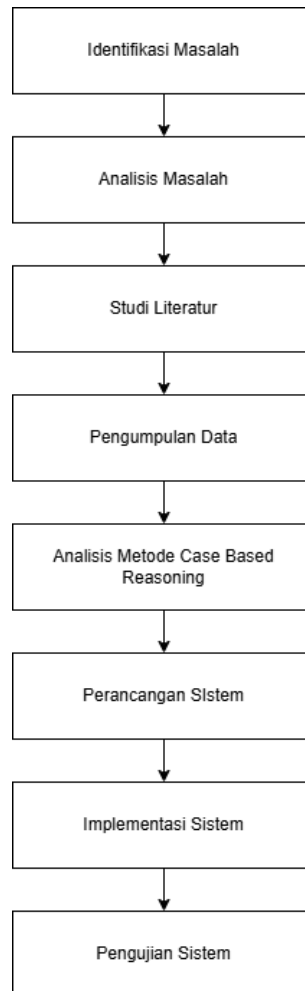
Penelitian ini bertujuan untuk merancang dan membangun sistem pakar berbasis web yang mampu mendiagnosis penyakit THT (Telinga, Hidung, Tenggorokan) menggunakan metode *Case Based Reasoning* (CBR). Sistem ini dikembangkan sebagai solusi untuk membantu masyarakat dalam melakukan diagnosis awal terhadap keluhan gejala THT, sehingga dapat mengurangi ketergantungan terhadap konsultasi langsung ke dokter yang terkadang memerlukan waktu dan biaya lebih.

Metode *Case Based Reasoning* dipilih karena kemampuannya dalam mengidentifikasi dan menyelesaikan masalah baru berdasarkan kasus-kasus lama yang mirip. Dalam konteks ini, gejala-gejala yang dialami pengguna akan dicocokkan dengan data gejala dari kasus penyakit THT yang sudah tersimpan di basis data, sehingga sistem dapat memberikan hasil diagnosa berupa jenis penyakit dan rekomendasi penanganannya.

Sistem ini dirancang berbasis web agar mudah diakses kapan saja dan di mana saja oleh pengguna menggunakan perangkat yang terkoneksi internet, serta memanfaatkan bahasa pemrograman PHP dan database MySQL sebagai teknologi utama pengembangannya..

3.2 Kerangka Kerja Penelitian

Pada bab 3 akan diuraikan metodologi penelitian dan kerangka kerja penelitian, Kerangka kerja ini merupakan langkah-langkah yang akan dilakukan dalam penyelesaian masalah yang akan di bahas. Adapun kerangka kerja penelitian ini dapat di gambarkan pada gambar 3.1.



Gambar 3.1 Kerangka Kerja Penelitian

Berdasarkan kerangka diatas, maka masing-masing langkahnya dapat diuraikan sebagai berikut.

3.2.1 Identifikasi Masalah

Tahap pertama dalam penelitian ini adalah mengidentifikasi permasalahan yang menjadi dasar pengembangan sistem pakar. Permasalahan utama yang diangkat adalah rendahnya kesadaran masyarakat terhadap gejala-gejala awal penyakit THT (Telinga, Hidung, Tenggorokan), serta keterbatasan akses ke tenaga medis secara cepat dan efisien. Banyak orang menunda konsultasi ke dokter karena berbagai alasan seperti lokasi yang jauh, biaya, antrean panjang, atau minimnya pengetahuan medis dasar.

Dengan kondisi tersebut, dibutuhkan suatu sistem bantu yang mampu memberikan informasi awal dan diagnosis sederhana mengenai penyakit THT berdasarkan gejala yang dialami pengguna. Sistem ini diharapkan dapat menjadi alat bantu yang mudah diakses dan dimengerti oleh masyarakat luas, serta mampu memberikan informasi awal secara cepat sehingga pengguna dapat mengambil tindakan medis lanjutan dengan lebih tepat..

3.2.2 Analisis Permasalahan

Setelah masalah berhasil diidentifikasi, dilakukan analisis terhadap penyebab, dampak, dan ruang lingkup permasalahan tersebut. Salah satu penyebabnya adalah kurangnya sistem informasi berbasis teknologi yang mampu membantu masyarakat

melakukan deteksi dini terhadap penyakit THT. Selain itu, proses diagnosis oleh dokter pun biasanya membutuhkan waktu dan ketergantungan pada interaksi langsung, yang tidak selalu bisa dilakukan oleh semua kalangan.

Analisis ini membantu peneliti menyusun kebutuhan sistem yang akan dikembangkan, termasuk fitur utama yang harus tersedia, jenis data medis yang dibutuhkan, serta metode diagnosis yang sesuai. Dengan analisis masalah yang mendalam, sistem pakar yang akan dibuat diharapkan benar-benar menjawab kebutuhan nyata dan dapat dimanfaatkan secara optimal oleh pengguna.

a. Analisis Kebutuhan Proses

Kebutuhan proses menggambarkan tahapan-tahapan fungsional yang harus dilakukan oleh sistem agar dapat membantu pengguna dalam proses diagnosis penyakit. Sistem pakar ini harus mampu melakukan pencocokan antara gejala yang diinputkan pengguna dengan basis data kasus penyakit yang telah tersimpan. Pencocokan tersebut dilakukan menggunakan pendekatan metode *Case Based Reasoning* (CBR) yang terdiri dari empat tahapan utama yaitu:

1. *Retrieve*, yaitu tahap pengambilan kasus lama yang paling mirip dengan kasus baru berdasarkan gejala yang dimasukkan pengguna;
2. *Reuse*, yaitu tahap penggunaan solusi dari kasus lama yang telah diambil sebelumnya;

3. *Revise*, yaitu tahap revisi atau penyesuaian terhadap solusi yang diperoleh, jika diperlukan;
4. *Retain*, yaitu tahap penyimpanan kasus baru ke dalam basis data sebagai pengetahuan baru.

Selain proses diagnosis , sistem juga harus menyediakan fitur pengelolaan data oleh admin, seperti pengelolaan data gejala, penyakit, relasi gejala-penyakit, dan solusi penanganan awal. Hal ini diperlukan untuk menjaga keakuratan dan kelengkapan basis pengetahuan dalam sistem.

b. Analisis Kebutuhan Masukan

Masukan dalam sistem ini berasal dari pengguna yang ingin mengetahui jenis penyakit THT berdasarkan gejala yang dirasakan. Kebutuhan masukan utama antara lain adalah:

1. Data gejala yang dipilih oleh pengguna (bisa berupa checkbox atau form input gejala)
2. Informasi pengguna (opsional), seperti nama, usia, atau jenis kelamin, yang dapat digunakan untuk pencatatan atau laporan
3. Masukan dari admin berupa data-data berikut:
 - a) Daftar gejala penyakit THT
 - b) Daftar penyakit THT
 - c) Relasi antara gejala dan penyakit

- d) Solusi atau tindakan medis awal
- e) Bobot kemiripan antara gejala

Masukan ini menjadi dasar untuk proses perhitungan similarity dalam metode CBR, sehingga harus diatur dalam format dan struktur yang sesuai dengan kebutuhan sistem.

c. Analisis Kebutuhan Keluaran

Keluaran dari sistem ini merupakan informasi hasil diagnosis yang ditampilkan kepada pengguna setelah proses perhitungan dilakukan. Keluaran utama dari sistem ini adalah:

1. Jenis penyakit THT yang paling memungkinkan diderita pengguna berdasarkan gejala yang dipilih.
2. Nilai tingkat kemiripan (similarity) dalam bentuk persentase antara gejala yang diinput pengguna dengan kasus yang ada dalam database.
3. Solusi atau tindakan awal yang direkomendasikan untuk dilakukan oleh pengguna.
4. Saran untuk konsultasi lebih lanjut, apabila gejala yang dialami memerlukan pemeriksaan oleh tenaga medis profesional.

Keluaran dari sistem harus disajikan dengan tampilan antarmuka yang jelas, ringkas, dan mudah dipahami oleh pengguna awam. Sistem juga harus

mampu menampilkan keluaran berupa laporan diagnosis jika dibutuhkan, baik untuk pengguna maupun untuk pengelolaan data oleh admin.

3.2.3 Studi Literatur

Studi literatur dilakukan untuk mengkaji teori-teori dan penelitian terdahulu yang berkaitan dengan sistem pakar, penyakit THT, serta metode *Case Based Reasoning* (CBR). Literatur yang digunakan mencakup jurnal ilmiah, buku, artikel, dan dokumentasi sistem serupa yang pernah dikembangkan sebelumnya. Fokus utama studi ini adalah menggali cara kerja sistem pakar dan bagaimana metode CBR dapat diterapkan secara efektif dalam proses diagnosis penyakit.

Melalui studi literatur, peneliti memperoleh dasar teori yang kuat untuk pengembangan sistem, termasuk struktur CBR (*retrieve, reuse, revise, retain*), model diagnosis penyakit, dan teknologi pendukung yang sesuai (seperti PHP dan MySQL). Tahapan ini sangat penting karena menjadi landasan konseptual dari seluruh proses perancangan dan implementasi sistem..

3.2.4 Pengumpulan Data

Pengumpulan data merupakan tahapan penting dalam pembangunan sistem pakar diagnosis penyakit THT berbasis web. Data yang diperoleh akan digunakan sebagai dasar dalam penyusunan basis pengetahuan dan pengujian sistem. Adapun jenis data yang dikumpulkan dalam penelitian ini dibagi menjadi dua, yaitu data primer dan data sekunder. Data yang dibutuhkan dalam penelitian ini yaitu :

1. Data Primer

Data primer diperoleh secara langsung dari sumber pertama melalui metode observasi dan wawancara. Dalam penelitian ini, data primer dikumpulkan melalui:

- a. Wawancara dengan dokter THT, guna memperoleh informasi mengenai jenis-jenis penyakit THT, gejala yang menyertainya, serta penanganan awal yang biasa dilakukan.
- b. Observasi terhadap proses diagnosa penyakit THT di fasilitas layanan kesehatan sebagai referensi untuk membangun alur diagnosis dalam sistem.
- c. Data primer ini digunakan untuk membangun relasi antara gejala dan penyakit, serta menentukan bobot gejala yang relevan untuk proses pencocokan kasus dalam metode *Case Based Reasoning* (CBR).

2. Data Sekunder

Data sekunder diperoleh dari sumber-sumber yang telah tersedia sebelumnya, baik dalam bentuk dokumen tertulis maupun digital. Adapun sumber data sekunder dalam penelitian ini meliputi:

- a. Buku dan jurnal ilmiah yang membahas penyakit THT, sistem pakar, dan metode *Case Based Reasoning* (CBR).

- b. Website kesehatan resmi seperti Kementerian Kesehatan RI, WHO, atau platform medis lainnya yang menyediakan informasi ilmiah tentang penyakit THT dan gejalanya.
- c. Penelitian terdahulu yang relevan dengan topik sistem pakar berbasis web atau diagnosis penyakit menggunakan pendekatan kecerdasan buatan.

Data sekunder digunakan sebagai referensi pendukung dalam validasi data primer dan sebagai bahan perbandingan untuk menentukan struktur dan metode yang tepat dalam membangun sistem..

3.2.5 Penerapan metode *Case Based Reasoning* (CBR)

Metode CBR digunakan karena mampu menyelesaikan masalah diagnosis berdasarkan pengalaman sebelumnya. Dalam CBR, ketika pengguna memasukkan gejala yang dialami, sistem akan mencari kasus lama yang memiliki tingkat kemiripan tinggi, dan menggunakan solusi dari kasus tersebut sebagai dasar diagnosis . CBR terdiri dari empat tahap utama: *retrieve*, *reuse*, *revise*, dan *retain*.

Tahap *retrieve* digunakan untuk mengambil kasus yang mirip dari basis data. Selanjutnya, *reuse* digunakan untuk menerapkan solusi dari kasus lama pada kasus baru. Jika diperlukan, solusi tersebut dapat disesuaikan pada tahap *revise*. Terakhir, pada tahap *retain*, kasus baru dan solusinya disimpan kembali dalam basis data jika terbukti valid, sehingga sistem menjadi lebih pintar seiring waktu.

3.2.6 Perancangan Sistem

Setelah data dan metode dianalisis, tahap berikutnya adalah merancang sistem pakar yang akan dibangun. Desain sistem mencakup struktur database, antarmuka pengguna (UI), dan diagram alur sistem (seperti UML). Sistem dirancang agar *User-friendly* dan dapat diakses melalui peramban (*browser*), baik melalui komputer maupun perangkat mobile.

Rancangan sistem mempertimbangkan alur konsultasi pengguna, yaitu dari proses input gejala, pencocokan gejala menggunakan metode CBR, hingga output berupa hasil diagnosis dan saran penanganan awal. Setiap fitur dirancang untuk memberikan pengalaman penggunaan yang mudah dipahami oleh pengguna non-teknis sekalipun.

3.2.7 Implementasi Sistem

Pada tahap ini, hasil rancangan sistem diimplementasikan ke dalam bentuk aplikasi nyata menggunakan bahasa pemrograman web (PHP) dan sistem manajemen basis data MySQL. Implementasi mencakup pembuatan halaman input gejala, proses perhitungan similarity menggunakan CBR, hingga halaman hasil diagnosis dan saran penanganan.

Sistem ini dibangun berbasis web agar dapat diakses oleh pengguna kapan saja dan di mana saja melalui jaringan internet. Implementasi dilakukan secara bertahap, dimulai dari modul input gejala, modul pengolahan data, dan akhirnya

modul output hasil diagnosis . Uji coba awal juga dilakukan untuk memastikan sistem berjalan sesuai dengan fungsinya.

3.2.8 Pengujian Sistem

Tahap terakhir adalah melakukan pengujian terhadap sistem pakar yang telah dibangun. Pengujian dilakukan menggunakan metode black box testing, yaitu dengan menguji setiap fungsi sistem berdasarkan input dan output tanpa melihat struktur kode internal. Selain itu, pengujian akurasi sistem dilakukan dengan membandingkan hasil diagnosis sistem terhadap data uji atau konsultasi dengan pakar THT.

Tujuan dari tahap ini adalah memastikan bahwa sistem dapat memberikan hasil diagnosis yang relevan, akurat, dan sesuai dengan data gejala yang diberikan. Jika ditemukan kekurangan, maka sistem akan diperbaiki dan disesuaikan kembali. Dengan pengujian yang menyeluruh, sistem pakar diagnosis THT berbasis CBR ini diharapkan dapat digunakan secara efektif oleh masyarakat luas sebagai alat bantu konsultasi kesehatan awal.