

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Teori dasar merupakan landasan dalam memahami ide-ide pokok, hubungan antar variabel, serta penjelasan metodologis terhadap suatu fenomena. Pemahaman yang mendalam mengenai teori dasar membantu dalam menerapkan konsep-konsep tersebut pada berbagai situasi serta mendukung proses analisis yang lebih terstruktur dan menyeluruh.

2.1.1 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence/AI*) merupakan cabang ilmu komputer yang berfokus pada pengembangan sistem atau mesin yang mampu meniru kemampuan kognitif manusia, seperti belajar, berpikir, memahami, dan mengambil keputusan. AI memungkinkan mesin untuk melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia, seperti pengenalan suara, pengolahan bahasa alami, dan pengambilan keputusan kompleks. Menurut (Simanjuntak et al., 2024), AI memiliki potensi besar dalam meningkatkan efisiensi dan efektivitas dalam berbagai bidang, termasuk keamanan jaringan, dengan kemampuannya dalam menganalisis pola data yang kompleks dan mendeteksi anomali yang mungkin menunjukkan adanya serangan.

2.1.2 Logika Fuzzy

Logika fuzzy yaitu sebuah metode matematika yang digunakan untuk menangani ketidakpastian dan ambiguitas dalam pengambilan keputusan atau

pemrosesan informasi. Berbeda dengan logika klasik yang hanya mengenal dua nilai (benar atau salah, 1 atau 0), *logika fuzzy* memungkinkan adanya rentang nilai kebenaran yang kontinu antara 0 dan 1. Ini berarti bahwa *logika fuzzy* dapat menangani keadaan di mana suatu kondisi tidak sepenuhnya benar atau salah, melainkan berada di antara kedua nilai tersebut.(Hafiz & Sriani, 2023)

2.1.3 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network (ANN)* merupakan model komputasi yang terinspirasi oleh struktur dan fungsi jaringan saraf biologis. JST terdiri dari unit-unit pengolahan yang disebut neuron, yang saling terhubung melalui sinapsis. Masing-masing neuron menerima sinyal, memprosesnya, dan meneruskan hasilnya ke neuron lain. Proses ini memungkinkan JST untuk belajar dari data, mengenali pola, dan membuat prediksi atau keputusan(Suahati et al., 2022).

2.1.4 Sistem Pakar

Sistem pakar yaitu sebuah sistem komputer yang dirancang untuk meniru kemampuan pemecahan masalah dan pengambilan keputusan dari seorang ahli dalam bidang tertentu. Sistem ini menggunakan basis pengetahuan (*knowledge base*) yang berisi informasi, aturan, atau fakta terkait dengan masalah yang dapat diselesaikan, serta mesin inferensi untuk menganalisis informasi tersebut dan menghasilkan solusi atau rekomendasi(Lutfi et al., 2022).

Karakteristik utama dari sistem pakar adalah sebagai berikut:

1. Basis Pengetahuan

Merupakan kumpulan informasi yang relevan dengan domain tertentu. Basis pengetahuan ini berisi fakta, aturan, dan hubungan yang digunakan oleh sistem pakar untuk memecahkan masalah.

2. Mesin Inferensi (*Inference Engine*)

Alat yang digunakan untuk menganalisis dan memproses data dari basis pengetahuan. Mesin inferensi ini bekerja dengan cara menarik kesimpulan atau solusi berdasarkan aturan dan fakta yang ada.

3. Antarmuka Pengguna (*User Interface*)

Fasilitas yang memungkinkan pengguna untuk berinteraksi dengan sistem. Pengguna dapat memberikan input atau menerima hasil dari sistem pakar melalui antarmuka ini.

4. Kemampuan Meniru Ahli

Sistem pakar dirancang untuk meniru cara berpikir dan pengambilan keputusan seorang ahli dalam domain tertentu, meskipun sistem ini tidak memiliki kecerdasan seperti manusia.

5. Penalaran Berbasis Aturan

Sebagian besar sistem pakar menggunakan aturan berbentuk "jika-maka" (*if-then*) untuk menyelesaikan masalah dan membuat keputusan.

6. Penalaran Deduktif

Sistem pakar sering kali melakukan penalaran deduktif, yang berarti menarik kesimpulan dari informasi yang sudah diketahui untuk memecahkan masalah baru.

7. Pembelajaran dan Adaptasi

Beberapa sistem pakar memiliki kemampuan untuk belajar dan beradaptasi berdasarkan pengalaman atau data baru, meskipun ini tidak selalu ada pada semua sistem pakar.

2.1.5 Manfaat Sistem Pakar

Manfaat dari sistem pakar sangat beragam dan signifikan dalam berbagai bidang kehidupan. Sistem ini mampu memberikan solusi terhadap masalah yang kompleks dengan tingkat ketelitian yang tinggi, sebagaimana halnya seorang pakar manusia. Salah satu manfaat utamanya adalah efisiensi waktu dan biaya. Dengan adanya sistem pakar, pengguna tidak perlu selalu bergantung pada kehadiran pakar manusia yang mungkin sulit dijangkau atau memerlukan biaya tinggi. Selain itu, sistem pakar dapat bekerja secara konsisten tanpa dipengaruhi oleh faktor kelelahan, emosi, atau subjektivitas pribadi, sehingga pengambilan keputusan dapat dilakukan secara lebih objektif dan stabil.

Sistem pakar juga dapat diakses kapan saja, selama 24 jam, sehingga sangat membantu dalam situasi darurat atau ketika keputusan harus diambil secara cepat. Tak hanya itu, sistem pakar juga dapat dimanfaatkan sebagai sarana pembelajaran, khususnya bagi individu yang ingin memahami cara kerja pengambilan keputusan dalam suatu bidang tertentu. Oleh karena itu, kehadiran sistem pakar sangat

bermanfaat dalam mendukung efisiensi kerja, meningkatkan akses terhadap pengetahuan, serta membantu proses pembelajaran dan pengambilan keputusan secara cerdas.

2.1.6 Kelebihan dan Kelemahan Sistem Pakar

Sistem pakar memiliki sejumlah keunggulan dalam membantu pengambilan keputusan, namun juga tidak lepas dari berbagai keterbatasan. Berikut ini adalah beberapa kelebihan dan kelemahannya:

1. **Konsistensi dalam Pengambilan Keputusan** Sistem pakar mampu memberikan keputusan atau solusi secara konsisten karena tidak dipengaruhi oleh emosi, kelelahan, atau faktor subjektif lainnya seperti yang biasa terjadi pada manusia.
2. **Efisiensi Waktu dan Biaya** Dengan adanya sistem pakar, proses pemecahan masalah dapat dilakukan dengan lebih cepat dan tanpa harus selalu mengandalkan kehadiran fisik seorang pakar, sehingga dapat menghemat waktu dan biaya operasional.
3. **Ketersediaan 24 Jam** Sistem pakar dapat diakses kapan saja, bahkan di luar jam kerja, karena bersifat otomatis dan tidak memerlukan istirahat. Hal ini sangat berguna dalam situasi darurat atau yang memerlukan penanganan segera.
4. **Transfer dan Penyimpanan Pengetahuan Pakar** Pengetahuan dari seorang ahli bisa ditransfer dan disimpan dalam sistem, sehingga tetap dapat digunakan meskipun pakar tersebut tidak lagi tersedia. Ini memungkinkan keberlanjutan keahlian dalam jangka panjang.

5. Mendukung Pembelajaran dan Pelatihan Sistem pakar dapat digunakan sebagai media belajar bagi pengguna yang ingin memahami proses pengambilan keputusan dalam suatu bidang tertentu, karena sistem sering kali dilengkapi dengan penjelasan atau logika di balik setiap solusi.
6. Meningkatkan Produktivitas Karena dapat menangani berbagai kasus dengan cepat dan tepat, sistem pakar berkontribusi dalam meningkatkan produktivitas kerja, terutama dalam lingkungan kerja yang membutuhkan keputusan rutin atau teknis.

Kelemahan Sistem Pakar:

1. Keterbatasan dalam Menangani Kasus Subjektif atau Kompleks Sistem pakar sulit menangani masalah yang memerlukan pertimbangan subjektif, intuisi, atau penilaian moral, karena sistem hanya beroperasi berdasarkan data dan aturan yang sudah ditentukan.
2. Ketergantungan pada Basis Pengetahuan yang Tersedia Akurasi dan kualitas keputusan yang dihasilkan sangat bergantung pada kelengkapan dan keakuratan basis pengetahuan. Jika basis pengetahuan tidak lengkap atau usang, sistem dapat memberikan solusi yang salah.
3. Proses Pengembangan yang Kompleks dan Mahal Untuk membangun sistem pakar yang baik, dibutuhkan waktu, biaya, dan sumber daya yang cukup besar, terutama untuk menggali pengetahuan dari pakar dan mengimplementasikannya ke dalam sistem.

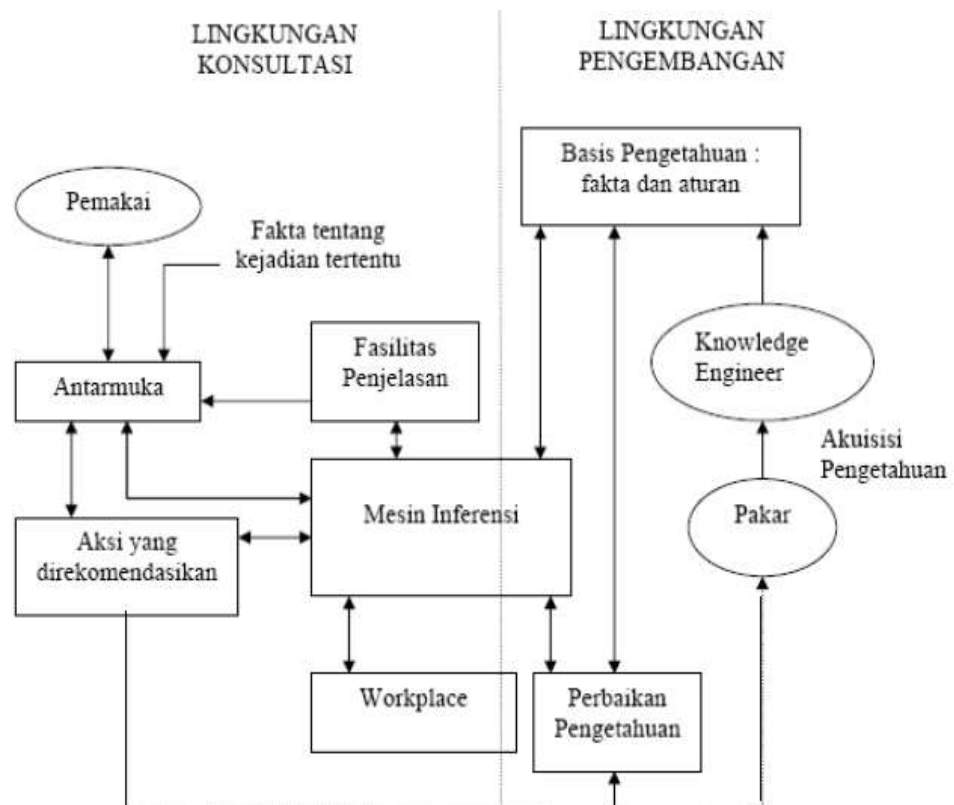
4. Keterbatasan dalam Menghadapi Situasi Baru Sistem pakar biasanya kesulitan dalam menghadapi situasi baru atau belum pernah ditangani sebelumnya, karena sistem tidak memiliki kemampuan belajar secara otomatis seperti manusia (kecuali jika dikombinasikan dengan kecerdasan buatan yang lebih kompleks).
5. Tidak Memiliki Kreativitas atau Intuisi Meskipun mampu menyelesaikan masalah dengan logika, sistem pakar tidak memiliki intuisi atau kreativitas dalam menemukan solusi baru di luar aturan yang telah ditentukan.

2.1.7 Struktur Sistem Pakar

Struktur sistem pakar terdiri dari beberapa komponen utama yang saling berinteraksi untuk meniru cara kerja seorang pakar dalam memecahkan masalah. Komponen pertama yang paling mendasar adalah basis pengetahuan (*knowledge base*), yaitu kumpulan fakta, teori, dan aturan-aturan logika yang berasal dari pakar manusia. Basis pengetahuan ini menjadi inti dari sistem karena seluruh proses penalaran dan pengambilan keputusan mengacu pada informasi yang tersimpan di dalamnya. Komponen kedua adalah mesin inferensi (*inference engine*), yang berfungsi sebagai pengolah informasi. Mesin ini bekerja dengan cara melakukan penalaran berdasarkan data yang diberikan oleh pengguna dan informasi yang ada di basis pengetahuan. Dalam prosesnya, mesin inferensi dapat menggunakan metode *forward chaining* (penalaran maju) atau *backward chaining* (penalaran mundur), tergantung pada arah analisis yang dibutuhkan.

Selanjutnya, terdapat basis fakta (*fact base*) yang berisi informasi atau data yang diinput oleh pengguna. Data ini akan digunakan oleh mesin inferensi untuk dicocokkan dengan aturan-aturan dalam basis pengetahuan guna menghasilkan suatu kesimpulan atau rekomendasi. Komponen keempat adalah antarmuka pengguna (*user interface*), yaitu media yang memungkinkan interaksi antara sistem dengan pengguna. Melalui antarmuka ini, pengguna dapat memberikan input berupa gejala atau masalah, dan sistem akan memberikan output berupa solusi, penjelasan, atau saran. Selain itu, sistem pakar juga dilengkapi dengan modul penjelasan (*explanation facility*), yang berfungsi untuk memberikan penjelasan kepada pengguna mengenai proses penalaran sistem, seperti mengapa suatu kesimpulan diambil atau bagaimana sistem sampai pada keputusan tersebut.

Hal ini penting untuk meningkatkan kepercayaan dan pemahaman pengguna terhadap sistem. Komponen terakhir yang juga tak kalah penting adalah modul akuisisi pengetahuan (*knowledge acquisition module*). Modul ini memungkinkan penambahan atau pembaruan pengetahuan dalam sistem secara berkala. Pengetahuan bisa diperoleh dari pakar manusia, buku, dokumen, atau sumber informasi lainnya, kemudian dimasukkan ke dalam sistem agar tetap relevan dan akurat. Dengan adanya struktur yang terdiri dari komponen-komponen tersebut, sistem pakar dapat bekerja secara efektif dan efisien dalam meniru proses berpikir seorang ahli, serta memberikan solusi atau rekomendasi terhadap masalah yang dihadapi pengguna.



Gambar 2. 1 Struktur Sistem Pakar

Sumber: Penelitian 2025

2.1.8 Mesin Inferensi

Mesin inferensi merupakan komponen utama dalam sistem pakar yang berfungsi sebagai "otak" atau pusat pengambilan keputusan. Mesin ini bertugas untuk melakukan proses penalaran atau inferensi berdasarkan informasi yang diberikan oleh pengguna dan pengetahuan yang tersimpan dalam basis pengetahuan. Dengan kata lain, mesin inferensi menghubungkan fakta-fakta yang tersedia dengan aturan-aturan yang relevan untuk menghasilkan kesimpulan atau rekomendasi. Proses penalaran ini biasanya dilakukan melalui dua metode utama, yaitu *forward chaining* (penalaran maju) dan *backward chaining* (penalaran mundur).

Pada metode *forward chaining*, sistem memulai dari fakta-fakta yang diketahui dan secara bertahap menelusuri aturan-aturan untuk mencapai suatu kesimpulan. Sedangkan dalam metode *backward chaining*, sistem memulai dari tujuan atau hipotesis tertentu dan bekerja mundur untuk mencari fakta-fakta pendukung. Mesin inferensi memastikan bahwa sistem pakar dapat berpikir secara logis, sistematis, dan mendekati cara berpikir seorang pakar manusia dalam menyelesaikan suatu permasalahan.

Ada beberapa metode inferensi yang penting dalam *expert system*, yaitu *forward chaining*, *backward chaining*, *case based reasoning*, *certainty factor* metode yang saya gunakan untuk penelitian ini.

a) *Forward chaining*

Forward chaining yaitu salah satu metode penalaran yang dimulai dengan fakta atau informasi yang sudah ada, kemudian bergerak maju untuk menemukan kesimpulan atau informasi baru berdasarkan aturan yang berlaku. Misalnya, dalam sistem pakar, jika kita memiliki fakta tentang cuaca yang cerah, sistem akan mencari aturan yang berlaku seperti "Jika cuaca cerah, maka bisa pergi ke luar rumah." Dengan begitu, kesimpulan yang diperoleh adalah kita bisa pergi keluar rumah karena cuaca mendukung.

b) *Backward chaining*

Berbeda dengan *forward chaining*, *backward chaining* bekerja dengan cara yang berlawanan. Dalam metode ini, kita mulai dari suatu hipotesis atau tujuan yang ingin dicapai, dan mencari fakta atau bukti yang dapat mendukung hipotesis

tersebut. Jadi, backward chaining lebih bersifat goal-driven. Misalnya, dalam sistem pakar medis, jika kita ingin mengetahui apakah seseorang mengidap flu, sistem akan memulai dengan hipotesis "Penderita flu" dan mencari bukti yang mendukungnya, seperti gejala demam atau batuk.

c) Case-Based Reasoning

Case-Based Reasoning (CBR) ialah pendekatan yang mengandalkan pengalaman masa lalu untuk menyelesaikan masalah saat ini. Sistem ini akan mencari kasus-kasus yang serupa dengan masalah yang dihadapi dan mencoba menyelesaikan masalah baru dengan cara yang sama seperti kasus sebelumnya. CBR sangat berguna dalam situasi di mana data atau aturan eksplisit mungkin tidak lengkap, tapi ada cukup banyak pengalaman sebelumnya yang bisa digunakan untuk memberi solusi.

d) certainty factor

Dalam konteks penalaran berbasis aturan, *certainty factor* digunakan untuk mengukur seberapa yakin sistem terhadap suatu kesimpulan. Ketika sistem menghasilkan kesimpulan berdasarkan aturan, *certainty factor* memberikan nilai numerik yang menunjukkan tingkat keyakinan terhadap kesimpulan tersebut. Untuk meningkatkan ketepatan diagnosis, metode Certainty Factor (CF) digunakan dalam sistem pakar. Metode ini memungkinkan sistem memberikan hasil diagnosis beserta tingkat keyakinannya, yang sangat bermanfaat ketika data yang tersedia tidak pasti atau tidak lengkap. Certainty Factor banyak digunakan dalam sistem

pakar medis, termasuk untuk hewan, karena kemampuannya dalam menangani ketidakpastian. (Eva Fatayatul Mufidah et al., 2023)

Rumus dasar untuk *certainty factor* dalam *MYCIN* adalah sebagai berikut :

$$CF(h, e) = MB(h, e) - MD(h, e)$$

Rumus 2. SEQ Rumus_2. *
ARABIC 1 MYCIN Certainty Factor

Keterangan :

$CF(h, e)$ = *Certainty factor* (ketidakpastian) dalam suatu hipotesis h dalam gejala e

$MB(h, e)$ = *measure of believe* (Ukuran Kepercayaan) dalam suatu hipotesis h dalam gejala e

h = hipotesis

e = *evidence* (gejala)

Skala Nilai Certainty Factor dalam Sistem Pakar

Dalam Penelitian ini, nilai *Certainty Factor* (CF) ditetapkan dalam 5 tingkatan utama, yaitu 0.2, 0.4, 0.6, 0.8 dan 1.0. Skala ini dipilih untuk mempermudah pakar dalam memberikan nilai keyakinan terhadap hubungan antara gejala dan penyakit.

Adapun penjelasan masing-masing nilai adalah sebagai berikut:

- 0.2 (Rendah) → Gejala sangat lemah hubungannya dengan penyakit, hanya kemungkinan kecil.
- 0.4 (Cukup Rendah) → Gejala terkadang muncul, namun tidak terlalu berpengaruh.

- 0.6 (Sedang) → Gejala Cukup sering muncul pada penyakit tertentu, tingkat keyakinan sedang.
- 0.8 (Kuat) → Gejala Khas, sangat sering muncul pada penyakit tertentu, keyakinan kuat.
- 1.0 (Sangat Kuat/Pasti) → Gejala utama yang secara pasti menunjukkan penyakit tertentu.

2.2 Variabel

Variabel dalam penelitian merupakan suatu faktor atau atribut yang dapat berubah atau bervariasi, yang dapat diukur atau diamati. Variabel digunakan untuk menggambarkan elemen-elemen yang ingin dianalisis atau diuji dalam sebuah penelitian. Variabel bisa mempengaruhi hasil penelitian dan digunakan untuk menjelaskan hubungan antara fenomena yang satu dengan fenomena lainnya (Ayumi & Budiatmo, 2021).

a) Kura-kura adalah salah satu jenis reptil yang memiliki ciri utama berupa tempurung keras yang berfungsi sebagai pelindung tubuh. Kura-kura bernapas dengan paru-paru dan hidup di lingkungan darat maupun air. Seiring dengan perawatan dan kondisi lingkungan yang tidak optimal, kura-kura dapat mengalami beberapa jenis penyakit, antara lain:

1. Penyakit Saluran Pernapasan Penyakit ini menyerang sistem pernapasan kura-kura dan dapat mengganggu aktivitas normal seperti makan dan bergerak.

2. *Shell Rot* (Busuk Tempurung) *Shell rot* merupakan penyakit infeksi pada karapas (tempurung) kura-kura yang menyebabkan kerusakan pada permukaan tempurung, seperti luka atau bercak yang membusuk.
3. *Hipovitaminosis A*, *Hipovitaminosis A* merupakan kondisi kekurangan vitamin A yang berdampak pada kesehatan mata, kulit, dan sistem pernapasan kura-kura.

b) Ular merupakan reptil yang memiliki tubuh panjang dan tidak berkaki. Hewan ini bergerak dengan cara melata dan menggunakan lidahnya sebagai alat penciuman. Ular termasuk karnivora dan sebagian besar hidup secara soliter. Beberapa penyakit yang umum ditemukan pada ular antara lain:

1. Stomatitis (*Mouth Rot*) Stomatitis adalah penyakit yang menyerang rongga mulut ular, ditandai dengan luka, pembengkakan, dan gangguan saat makan.
2. *Metabolic Bone Disease* (MBD) MBD merupakan penyakit metabolik yang menyerang sistem tulang ular, menyebabkan tulang menjadi rapuh, bengkok, atau mengalami deformitas.
3. Infeksi Parasit Internal Infeksi ini disebabkan oleh parasit yang menyerang organ dalam seperti usus, menyebabkan gangguan pencernaan, berat badan menurun, dan kondisi tubuh yang melemah.

c) Iguana adalah reptil yang berasal dari daerah tropis, dikenal sebagai hewan herbivora dan sering dipelihara sebagai hewan eksotik. Iguana memiliki tubuh

yang panjang dan kuat, serta cakar tajam. Beberapa penyakit yang sering menyerang iguana antara lain:

1. *Abscess* (Bisul) *Abscess* merupakan benjolan berisi nanah yang muncul akibat infeksi, biasanya terlihat sebagai benjolan keras di bawah kulit.
2. Penyakit Ginjal (Gagal Ginjal Kronis) Penyakit ginjal kronis merupakan gangguan yang menyerang fungsi ginjal, menyebabkan penumpukan zat-zat sisa di dalam tubuh yang berdampak buruk pada kondisi fisik iguana.
3. Dehidrasi

Dehidrasi merupakan kondisi kekurangan cairan tubuh yang berdampak pada kesehatan kulit, organ dalam, dan tingkat energi iguana.

2.3 Software Pendukung

2.3.1 Aplikasi

Aplikasi merupakan program yang dapat digunakan yang dapat menjalankan perintah dari pengguna aplikasi untuk mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi, dimana aplikasi mencoba menyelesaikan masalah dengan menggunakan teknik komputer dari aplikasi biasanya dikendalikan oleh komputer oleh komputer yang digunakan. seseorang berharap atau menunggu dan menunggu (Zalukhu et al., 2023).

2.3.2 Android

Android merupakan suatu sistem pembedahan buat fitur mobile berbasis linux yang mencakup sistem pembedahan, middleware, serta aplikasi. Android merupakan sistem pembedahan buat telepon seluler yang berbasis linux. Android

sediakan platform terbuka untuk para pengembang buat membuat aplikasi mereka sendiri (Tri Sulistyorini et al., 2022).

Tabel 2. 1 *Android versi*

No	Versi	Tahun
1	1.0	Desember 2014
2	1.1	Februari 2015
3	1.2	April 2015
4	1.3	Juli 2015
5	1.4	September 2015
6	1.5	November 2015
7	2.0	April 2016
8	2.1	April 2016
9	2.2	September 2016
10	2.3	Maret 2017
11	3.0	Oktober 2017
12	3.1	Maret 2018
13	3.2	September 2018
14	3.3	Januari 2019
15	3.4	April 2019 ^[18]
16	3.5	Agustus 2019
17	3.6	Februari 2020
18	4.0	Mei 2020

19	4.1	Oktober 2020 ^[19]
20	4.2	Mei 2021 ^[20]
21	Arctic Fox (2020.3.1)	Juli 2021 ^[21]
22	Bumblebee (2021.1.1)	Januari 2022 ^[22]
23	Chipmunk (2021.2.1)	Mei 2022 ^[23]
24	Dolphin (2021.3.1)	September 2022 ^[24]
25	Electric Eel (2022.1.1)	Januari 2023 ^[25]
26	Flamingo (2022.2.1)	April 2023 ^[26]
27	Giraffe (2022.3.1)	Juli 2023 ^[27]

Sumber: Penelitian 2025

2.3.3 Android studio

Android studio yang merupakan sebuah *Integrated Development Environment* (IDE) yang paling umum banyak dipakai untuk membuat aplikasi android. Android Studio menyediakan berbagai fitur yang akan mempermudah proses pembuatan aplikasi, seperti debugger, emulator, dan lain-lain(Chan et al., 2022)

1. Kode Editor yang Cerdas (*Intelligent Code Editor*)

Android Studio menyediakan editor kode yang dilengkapi dengan fitur auto-completion, highlight sintaks, dan real-time error checking. Fitur ini memudahkan pengembang dalam menulis kode secara lebih cepat dan mengurangi kemungkinan kesalahan penulisan kode.

2. *Layout Editor*

Layout Editor adalah fitur visual untuk mendesain antarmuka pengguna (UI). Dengan fitur ini, pengembang dapat menggunakan metode drag and drop untuk menambahkan komponen UI seperti tombol, teks, gambar, dan lainnya. Layout Editor juga mendukung preview antarmuka pada berbagai ukuran layar dan versi Android.

3. Android Emulator

Android Emulator merupakan fitur yang memungkinkan pengembang menjalankan dan menguji aplikasi tanpa memerlukan perangkat fisik. Emulator ini dapat dikonfigurasi untuk mensimulasikan berbagai jenis perangkat Android, termasuk smartphone, tablet, dan perangkat dengan spesifikasi tertentu.

4. Sistem Build Berbasis Gradle

Gradle adalah sistem build yang digunakan oleh Android Studio untuk mengelola proses kompilasi, dependency, dan distribusi aplikasi. Gradle memungkinkan pengembang untuk membuat berbagai varian build, seperti versi debug, release, maupun versi khusus seperti “pro” dan “free”.

5. Fitur Apply Changes

6. Fitur ini memungkinkan pengembang untuk melakukan perubahan kode dan langsung melihat hasilnya di emulator atau perangkat fisik tanpa harus me-restart aplikasi sepenuhnya. Hal ini mempercepat proses debugging dan pengujian.

7. Android Profiler

Android Studio menyediakan alat profiler untuk menganalisis performa aplikasi. Profiler ini mencakup pemantauan penggunaan CPU, memori, jaringan, dan energi. Dengan adanya profiler, pengembang dapat mengidentifikasi potensi masalah performa dan mengoptimalkan aplikasinya.

8. Lint dan Analisis Kode

Android Studio memiliki alat analisis statis (lint) yang digunakan untuk mendeteksi potensi kesalahan dalam kode, seperti masalah kompatibilitas versi Android, penggunaan API yang tidak efisien, atau potensi masalah keamanan.

9. Dukungan Bahasa Pemrograman

Android Studio mendukung beberapa bahasa pemrograman, yaitu:

Kotlin (bahasa utama yang direkomendasikan oleh Google)

Java (masih banyak digunakan secara luas)

C++ (untuk pengembangan native Android melalui NDK)

10. Integrasi Firebase

Android Studio memiliki integrasi langsung dengan Firebase, yaitu platform layanan backend dari Google. Pengembang dapat dengan mudah menambahkan layanan seperti *Authentication*, *Realtime Database*, *Cloud Messaging*, dan Crashlytics ke dalam aplikasi.

11. Versi Kontrol Terintegrasi

Android Studio mendukung integrasi dengan sistem kontrol versi seperti Git, GitHub, dan SVN, yang memungkinkan kolaborasi pengembangan secara tim dan pelacakan perubahan kode secara historis.



Gambar 2. 2 Logo Android Studio

Sumber: Penelitian 2025

2.3.4 MySQL

MySQL merupakan salah satu perangkat lunak Relational Database Management System (RDBMS) yang bersifat open-source dan menggunakan bahasa standar Structured Query Language (SQL) untuk mengelola basis data. MySQL awalnya dikembangkan oleh MySQL AB, sebuah perusahaan asal Swedia, yang kemudian diakuisisi oleh Sun Microsystems dan saat ini dimiliki oleh Oracle Corporation. Sebagai RDBMS, MySQL digunakan untuk menyimpan, mengatur, dan mengambil data dalam bentuk tabel yang saling berelasi. Pengguna dapat mengakses dan memanipulasi data dengan perintah-perintah SQL yang telah terstandarisasi secara internasional. MySQL banyak digunakan pada aplikasi berbasis web dan sering diintegrasikan dengan bahasa pemrograman seperti PHP, Python, dan Java(Suhartini et al., 2020)



Gambar 2. 3 Logo MySQL

Sumber: Penelitian 2025

2.3.5 UML (Unified Modeling Language)

Pada pertumbuhan metode pemrograman berorientasi objek, timbul lah suatu standarisasi bahasa pemodelan buat pembangunan fitur lunak yang dibentuk dengan memakai metode pemrograman berorientasi objek, ialah *Unified Modeling Language*(UML). UML timbul sebab terdapatnya kebutuhan pemodelan visual buat menspesifikasikan, menggambarkan, membangun, serta dokumentasi dari sistem fitur lunak. UML ialah bahasa visual buat pemodelan serta komunikasi menimpa suatu sistem dengan memakai diagram serta teks- teks pendukung(Abdillah, 2021).

UML cuma berperan buat melaksanakan pemodelan. Jadi pemakaian UML tidak terbatas pada metodologi tertentu, walaupun pada realitas UML sangat banyak

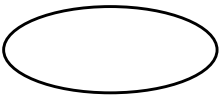
digunakan pada metodologi berorientasi objek. Semacam yang kita tahu kalau banyak perihal di dunia sistem data yang tidak bisa dibakukan, seluruh bergantung kebutuhan, area serta konteksnya. Begitu pula dengan pertumbuhan pemakaian UML tergantung tingkat abstraksi penggunaannya. Jadi belum pasti pemikiran yang berbeda dalam pemakaian UML merupakan sesuatu yang salah, tetapi butuh ditelaah dimakah UML digunakan serta perihal apa yang mau divisualkan. Secara analogi bila dengan bahasa yang kita pakai tiap hari, belum pasti penyampaian bahasa dengan puisi merupakan perihal yang salah. Sistem data tidaklah ilmu tentu, hingga bila terdapat banyak perbandingan serta interpretasi di dalam bidang sistem data ialah perihal yang sangat normal.

UML memiliki berbagai macam jenis diagram yang dikelompokkan menjadi dua kategori utama, yaitu diagram struktur dan diagram perilaku. Diagram struktur digunakan untuk menunjukkan bagian-bagian statis dari sistem, seperti class diagram yang menggambarkan struktur kelas dan relasinya. Sementara itu. Diagram perilaku digunakan untuk memodelkan dinamika dan sistem, atau squence diagram yang memperlihatkan urutan pesan antar objek dalam skenario tertentu.

1. Use Case

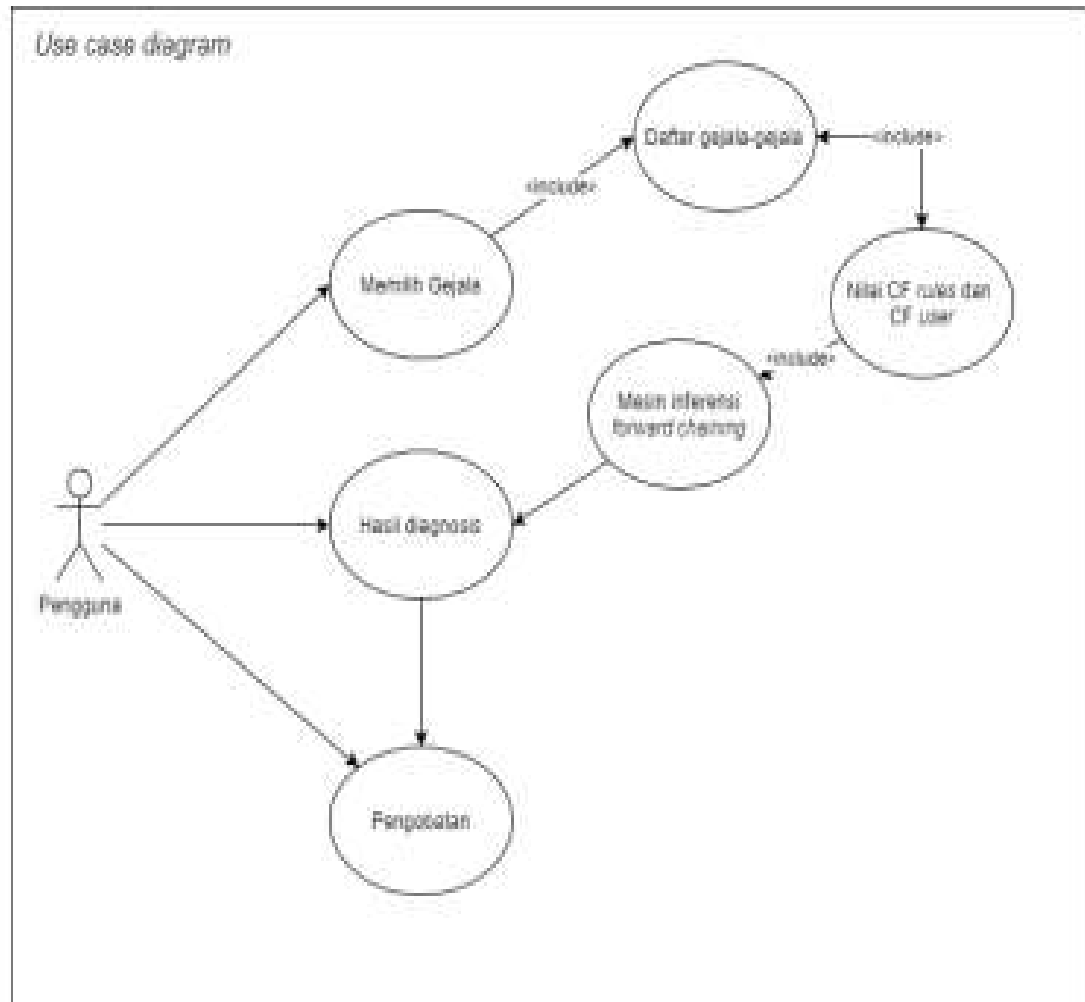
Tabel 2. 2 *Use Case*

NO	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menentukan sekumpulan peran yang dimainkan pengguna saat berinteraksi dengan use case

2		<i>Depedency</i>	Hubungan di mana perubahan elemen independen mempengaruhi elemen yang bergantung pada elemen non-independen
3		<i>Geralization</i>	Hubungan di mana objek anak berbagi perilaku struktur data objek di atas objek induk
4	<code><<include>></code>	<i>Include</i>	Secara eksplisit mendefinisikan kasus penggunaan sumber
5	<code><<extend>></code>	<i>Extend</i>	Menunjukkan bahwa kasus penggunaan target memperluas perilaku kasus penggunaan sumber pada titik tertentu
6		<i>Association</i>	Apa yang menghubungkan suatu objek dengan objek lain
7		<i>usecase</i>	Merupakan tindakan atau fungsi yang dapat dilakukan oleh aktor sistem atau aplikasi.

Sumber: Penelitian 2025

Sebagai gambaran awal, berikut ini merupakan contoh penerapan use case diagram yang menggambarkan interaksi antara pengguna dengan sistem serta fungsi- fungsi utama yang dapat dijalankan oleh pengguna.



Gambar 2. 4 Contoh Diagram Use Case

Sumber: Data Penelitian 2021

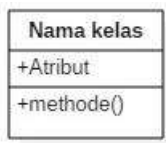
Dengan Menggunakan UML, proses perancangan sistem menjadi lebih terstruktur dan mudah dipahami oleh semua pihak yang terlibat.

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sistem. Penekanannya adalah pada "apa" yang dilakukan sistem dan bukan pada "bagaimana". (Nurdiana et al., 2022)

Diagram kelas menggambarkan hubungan antar kelas dan mewakili struktur sistem.

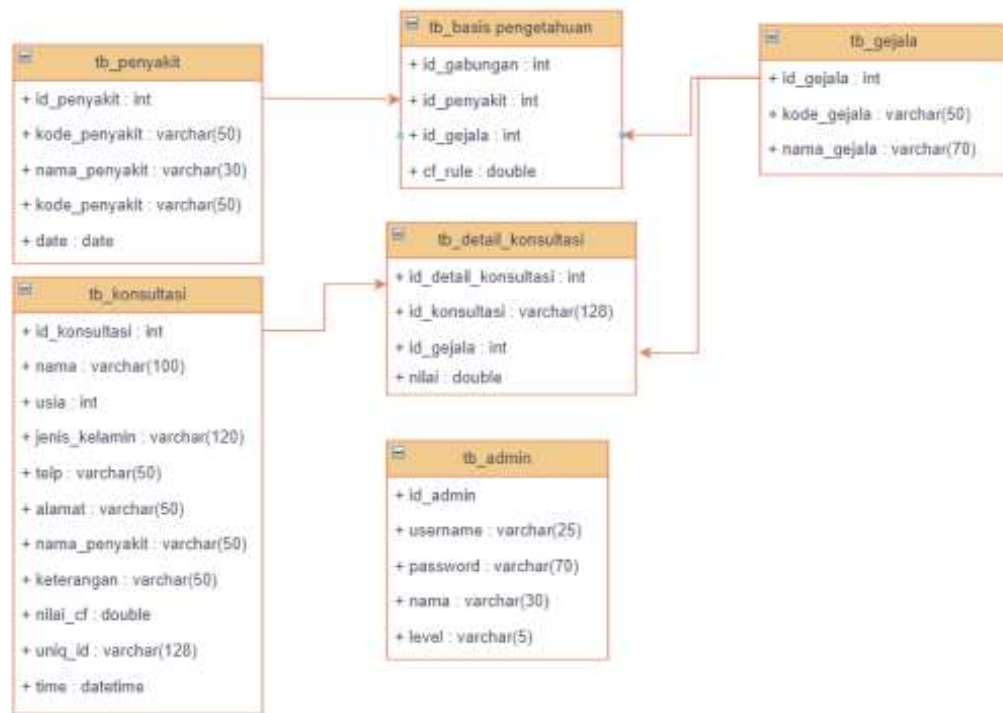
2. Diagram Calss

Tabel 2. 2 Diagram Class

NO	Gambar	Nama	Keterangan
1		Class	Menjelaskan kelas dari sistem
2		Association	Hubungan statis antar kelas
3		Agregation	Hubungan yang menyatakan bahwa satu kelas menjadi atribut dari kelas lain
4		Composition	Bentuk asosiasi khusus di mana kelas terkait dibuat setelah kelas dibuat
5		Generalization	Hubungan antar kelas menurut generalisasi spesialisasi
6		Directed Assocoation	Kelas lain menggunakan asosiasi dengan arti kelas

Sumber: Penelitian 2025

Untuk menjelaskan struktur data, berikut ini merupakan contoh penerapan class diagram yang menunjukkan struktur data dan hubungan antar kelas dalam sistem.



Gambar 2. 5 Contoh *Diagram Class*

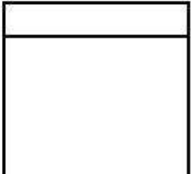
Sumber: Data Penelitian 2024

3. Diagram Activity

Diagram aktivitas menggambarkan aliran fungsional dari sistem yang baru dibuat.(Nurdiana et al., 2022).

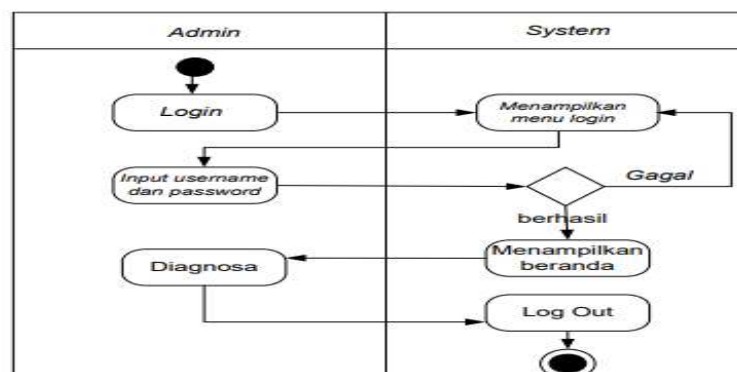
Tabel 2. 3 *Diagram Activity*

NO	Gambar	Nama	Keterangan
1	●	Initial	Titik awal untuk Memulai ktivitas
2	⦿	Final	Titik akhir untuk Mengakhiri aktivitas

3		Activity	Berarti tindakan
4		Decision	mengambil keputusan
5		Fork atau join	Ini digunakan untuk menunjukkan operasi parallel atau untuk menggabungkan dua operasi
6		Flow Final	Akhiri satu aliran
7		Swimlane	Untuk mengelompokkan activity berdasarkan aktor

Sumber: Penelitian 2025

Dalam menjabarkan alur aktivitas sistem, berikut ini merupakan contoh penerapan activity diagram yang menggambarkan alur aktivitas atau proses dalam sistem secara berurutan.



Gambar 2. 6 Contoh Diagram Activity

Sumber: Data Penelitian 2021

Pada sequence diagram berikut, user mengklik menu advice, kemudian memilih gejala, setelah form advice, memilih gejala dan mengklik tombol advice, kemudian web server memeriksa informasi rule di database dan menampilkan hasil dari advice tersebut.(Nurdiana et al., 2022)

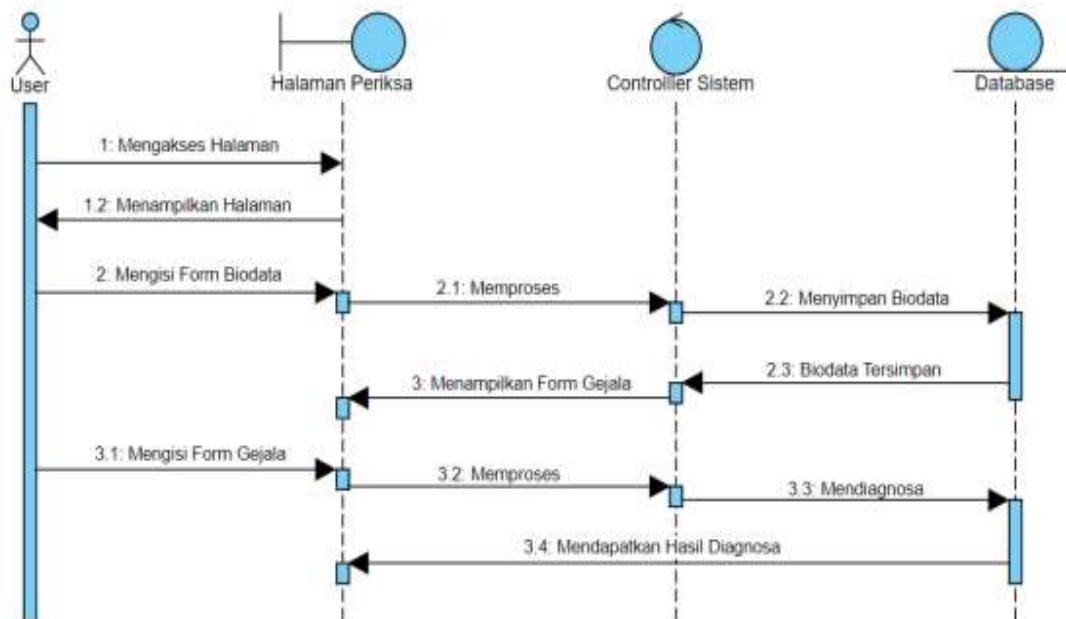
4. *Sequence Diagram*

Tabel 2. 4 *Sequence Diagram*

NO	Gambar	Nama	Keterangan
1		Actor	Orang yang berinteraksi dengan sistem
2		Boundary	Menggambarkan hubungan antara kegiatan yang akan dilakukan
3		Control	Menggambarkan penghubung antara boundary dengan tabel
4		Entity	Menggambarkan hubungan antara kegiatan yang akan dilakukan
5		Message	Mengidentifikasi komunikasi antar objek
6		Life Line	Identifikasi keberadaan objek dalam basis waktu

Sumber: Penelitian 2025

Unruk menjelaskan struktur data, berikut ini merupakan contoh penerapan class diagram yang menunjukkan struktur data dan hubungan antar kelas dalam sistem



Gambar 2. 7 Contoh Squence Diagram

Sumber: Data Penelitian 2024

2.4 Penelitian Terdahulu

Dalam menyusun penelitian ini, penulis merujuk pada beberapa penelitian terdahulu yang relevan sebagai dasar perbandingan serta untuk memperkuat landasan teori. Penelitian-penelitian tersebut memberikan gambaran mengenai metode, teknologi, serta pendekatan yang digunakan dalam studi sejenis. Adapun beberapa penelitian yang menjadi acuan adalah sebagai berikut:

1. mengembangkan sebuah sistem pakar berbasis Android untuk mendiagnosa penyakit mata menggunakan metode *Certainty Factor*. Sistem ini meniru kemampuan seorang pakar dalam menganalisis gejala penyakit mata berdasarkan nilai keyakinan (*certainty*) terhadap masing-masing gejala. Aplikasi ini dilengkapi antarmuka interaktif di mana pengguna dapat memilih gejala dan tingkat keyakinannya, yang kemudian dihitung untuk menghasilkan

persentase kemungkinan jenis penyakit mata yang diderita. Hasil penelitian menunjukkan bahwa sistem ini mampu memberikan informasi penyakit mata dengan cepat dan efisien serta memudahkan pengguna dalam mengenali jenis penyakit mata berdasarkan gejala yang dialaminya. (Sagat & Purnomo, 2021)

2. (Bere et al., 2021) Berdasarkan penelitian yang dilakukan, sistem pakar diagnosis penyakit pada ayam menggunakan metode Certainty Factor menunjukkan hasil yang signifikan dalam membantu peternak dalam mendiagnosis penyakit. Hasil pengujian menunjukkan bahwa sistem ini mampu memberikan diagnosis yang akurat dengan tingkat kesalahan 0%,
3. (Mardani & Adiguna, 2021) melakukan penelitian berjudul *"Sistem Pakar Mendiagnosa Penyakit Reptil Iguana dengan Metode Forward Chaining Berbasis Android"*. Penelitian ini mengembangkan sistem pakar berbasis aplikasi Android untuk membantu pengguna dalam mendiagnosis penyakit pada reptil jenis iguana. Metode yang digunakan adalah *forward chaining*, dengan basis pengetahuan yang berasal dari data gejala dan penyakit yang telah dikonsultasikan dengan ahli. Aplikasi ini dapat digunakan oleh pengguna awam sebagai alat bantu diagnosis awal terhadap kondisi iguana.
4. (Permadi et al., 2025) dalam penelitiannya yang berjudul *"Aplikasi Sistem Pakar Diagnosis Penyakit Ayam Berbasis Web dengan Teknik Forward Chaining dan Certainty Factor"* mengembangkan sistem berbasis web dengan pendekatan *forward chaining* yang dikombinasikan dengan metode *certainty factor* untuk menampilkan nilai keyakinan dari hasil diagnosis. Meskipun objek

penelitian adalah ayam, metode yang digunakan sangat relevan dan dapat diadaptasi untuk diagnosis penyakit pada reptil.

5. Dalam penelitian berjudul "*Pengembangan Sistem Pakar Diagnosis Penyakit Kelinci dengan Menggunakan Metode Case Based Reasoning*" menggunakan pendekatan *case based reasoning* (CBR), yaitu metode diagnosis berdasarkan kasus-kasus sebelumnya yang serupa. Meskipun metode yang digunakan berbeda, pendekatan ini memberikan gambaran tentang bagaimana sistem pakar dapat membantu diagnosis penyakit pada hewan peliharaan..(Affandi et al., 2021)
6. Pada penelitian berjudul "*Perancangan Sistem Pakar Diagnosis Penyakit pada Hewan Peliharaan Anjing Berbasis Web Menggunakan Metode Forward Chaining*" merancang sistem pakar berbasis web yang bertujuan untuk membantu diagnosis penyakit pada anjing peliharaan. Sistem ini menggunakan metode *forward chaining* untuk menarik kesimpulan diagnosis dari data gejala yang dimasukkan oleh pengguna.(Lumbantoruan & Yandra Niska, 2024).
7. Penelitian terdahulu jurnal internasional yang berjudul "Application of the Certainty Factor Method for Mobile-Based Identification of Freshwater Fish Diseases" Penelitian ini membahas pengembangan sistem pakar berbasis android untuk mendiagnosis penyakit pada ikan air tawar menggunakan metode certainty factor. Tujuan utama dari penelitian ini adalah memberikan kemudahan bagi peternak atau pemilik ikan dalam mengidentifikasi penyakit berdasarkan gejala-gejala yang ditunjukkan oleh ikan. Metode certainty factor digunakan untuk menghitung tingkat keyakinan terhadap suatu penyakit,

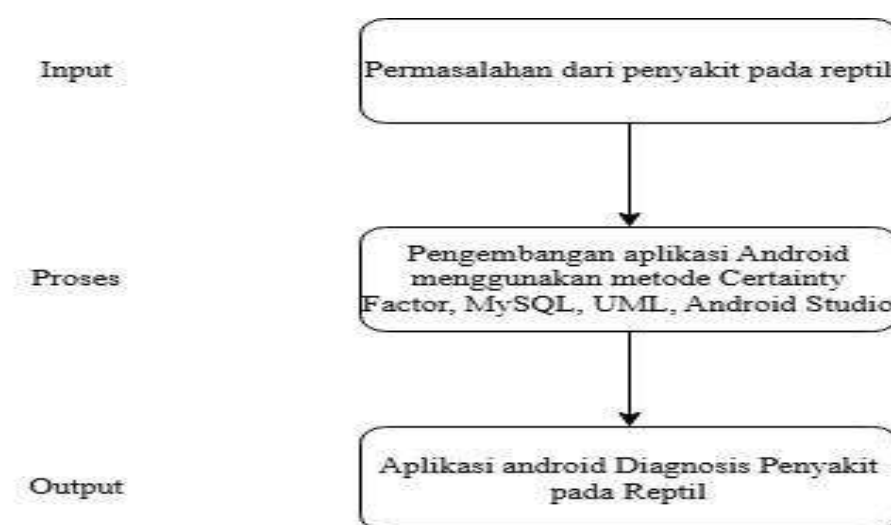
sehingga sistem ini dapat memberikan hasil diagnosis yang disertai dengan nilai kepastian. Sistem ini dirancang agar mudah digunakan diperangkat mobile.(Santoso et al., 2023)

2.5 Kerangka Pemikiran

Kerangka pemikiran adalah tujuan dan membimbing strategi untuk mencapai penelitian atau membimbing peneliti selama proses penelitian. (Silpiah et al., 2021)

Kerangka pemikiran pada penelitian ini menggambarkan alur proses identifikasi penyakit pada reptil menggunakan metode *certainty factor*.

Bagian ini menjelaskan alur berpikir peneliti dalam menyusun tahapan proses identifikasi penyakit dalam menyusun tahapan proses identifikasi penyakit reptil menggunakan metode *Certainty factor*. Kerangka pemikiran disusun untuk menggambarkan proses dari input gejala hingga menghasilkan output diagnosis berbagai sistem.



Gambar 2. 8 Kerangka Pemikiran

Sumber: Penelitian 2025

Kerangka di atas menunjukkan alur berpikir penelitian dalam tiga komponen utama:

1. Input

Permasalahan utama yang menjadi latar belakang penelitian adalah penyakit pada reptil dan kesulitan akses terhadap layanan medis hewan, khususnya di daerah terpencil. Hal ini mendorong kebutuhan akan solusi yang cepat dan mudah diakses.

2. Proses

Tahap ini mencakup perancangan dan pembuatan aplikasi sistem pakar diagnosis penyakit reptil. Proses ini melibatkan penerapan metode *Certainty Factor* untuk pengambilan keputusan diagnosis secara logis dan sistematis.

3. Output

Hasil dari proses ini adalah sebuah aplikasi Android yang mampu membantu pengguna (pemilik reptil) dalam mendiagnosis penyakit berdasarkan gejala-gejala yang dimasukkan.