

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem

Sistem yaitu suatu kesatuan yang memiliki beberapa komponen yang disatukan untuk melancarkan aliran informasi, materi atau energi guna mencapai tujuan (Effendy et al., 2023). Sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan untuk melakukan suatu aktivitas dengan sasaran tertentu. Sistem juga memiliki arti sebagai setiap kesatuan secara konseptual atau fisik yang tersusun dari bagian-bagian yang saling berkaitan satu sama lain (Duha & Juliani, 2020).

Prehanto et al. (2020:5-6) menjelaskan, sistem memiliki karakteristik tertentu, antara lain:

1) Komponen (*component*)

Sistem memiliki beberapa komponen, diantaranya melaksanakan interaksi yang membentuk suatu kesatuan serta saling bekerja sama yang tersusun dari berbagai cabang sistem.

2) Lingkungan luar sistem (*environment*)

Operasi sistem dipengaruhi oleh lingkungan luar pada sistem, yaitu lingkungan luar sistem.

3) Batasan sistem (*boundary*)

Ruang lingkup membatasi lingkungan luar sistem sesuai daerahnya.

4) Penghubung sistem (*interface*)

Satusub sistem akan dihubungkan dengan subsistem lain menggunakan alat bantu yang disebut dengan penghubung sistem.

5) Masukan sistem (*input*)

Sumber daya yang berupa perawatan (*maintenance input*), dan masukkan sinyal (*signal input*) yang dimasukkan ke dalam sistem, dikenal dengan masukan sistem.

6) Keluaran sistem (*output*)

Pemrosesan inputan akan menghasilkan energi keluaran yang dibuang ataupun dibutuhkan yang disebut dengan keluaran sistem.

7) Pengelola sistem

Pengelolaan sistem adalah bagian dari proses yang berfungsi merubah *input* ke *output*.

8) Sasaran sistem

Sasaran sistem adalah tujuan (*goal*) atau sasaran (*objective*).

Kesimpulan yang dapat diambil dari penjelasan diatas yaitu sistem merupakan kesatuan yang tersusun dari beberapa komponen atau elemen yang dikaitkan bersama untuk melancarkan aliran informasi, materi, atau energi guna mencapai tujuan. Sistem memiliki beberapa karakteristik, seperti komponen (*component*), lingkungan luar sistem (*environment*), batasan sistem (*boundary*), penghubung sistem (*interface*), masukan

sistem (*input*), keluaran sistem (*output*), pengelola sistem, dan sasaran sistem. Sistem dapat diartikan sebagai setiap kesatuan secara konseptual atau fisik yang tersusun dari bagian-bagian dalam keadaan saling berkaitan.

2. Pakar

Menurut Andriani (2016:14) sistem pakar merupakan dasar pengetahuan yang direpresentasi dari pengetahuan seorang pakar yang terdiri atas fakta dan kaidah. Fakta adalah informasi mengenai objek, peristiwa, dan situasi. Sedangkan, kaidah adalah langkah untuk menemukan fakta baru sesuai fakta sebelumnya. Dasar pengetahuan didapat oleh seorang pakar ataupun data histori yang berasal dari pengetahuan seorang pakar. Pakar dapat diartikan sebagai seseorang yang mempunyai pengetahuan atau keahlian spesial yang kebanyakan orang belum tentu memilikinya. Seorang pakar akan mudah memecahkan masalah yang sebelumnya sulit atau tidak bisa dipecahkan oleh orang awam. Sistem pakar dapat memuat pengetahuan yang berasal dari pikiran seorang pakar atau seorang ahlinya. Selain itu, pakar merupakan orang yang memiliki pengetahuan serta keahlian khusus yang untuk menangani sebuah masalah tertentu.

Kesimpulan yang didapat diambil dari pengertian diatas mengenai pakar dalam sistem pakar dapat diberikan kesimpulan bahwa pakar yaitu seorang pakar diartikan sebagai seorang yang berpengetahuan atau berkeahlian khusus yang belum tentu semua orang memilikinya. Seorang

pakar mampu memecahkan masalah yang sebelumnya sulit atau tidak bisa dipecahkan. Pengetahuan dalam sistem pakar berasal dari dasar pengetahuan seorang pakar. Dasar pengetahuan ini bisa diperoleh dari yang berasal dari pengetahuan seorang pakar.

3. Sistem Pakar

Menurut Andreswari et al. (2022) sistem pakar merupakan sistem berbasis pengetahuan yang memanfaatkan keahlian pakar untuk memberikan solusi dan saran kepada pengguna, memungkinkan mereka untuk menyelesaikan masalah secara mandiri, sehingga para pengguna dapat mengambil keputusan meskipun dirinya bukan seorang pakar. Definisi sistem pakar (*expert system*) yaitu sistem yang berasal pengetahuan manusia, di mana pengetahuan tersebut dituangkan ke dalam sebuah komputer, lalu digunakan guna menyelesaikan masalah yang membutuhkan keahlian manusia (Sholikhah et al., 2021).

Pengertian sistem pakar yaitu sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik pemikiran dalam menyelesaikan masalah, dan hanya mampu diselesaikan oleh seorang pakar yang menguasai suatu bidang spesifik (Bere et al., 2021). Pada dasarnya, sistem pakar dibuat untuk membantu dalam proses pemecahan masalah, seperti pembuatan keputusan (*Decision Making*), pemandu pengetahuan (*Knowledge*), pembuatan desain (*Designing*), perencanaan (*Planning*), prakiraan (*Forecasting*), peraturan (*Regulating*), pengendalian (*Controlling*), penjelasan (*Explaining*), diagnosis (*Diagnosing*),

perumusan (*Prescribing*), pemberi nasehat (*Advising*), dan pelatihan (*Tutoring*). Lain daripada itu, sistem pakar difungsikan juga sebagai asisten yang pandai dari seorang pakar (Firdaus & Yanti, 2022).

Berdasarkan pemaparan mengenai sistem pakar di atas, diperoleh kesimpulan yaitu sistem pakar merupakan suatu sistem yang menggunakan pengetahuan manusia untuk menyelesaikan masalah. Sistem pakar berfungsi guna mendukung kegiatan pemecahan masalah, termasuk pembuatan keputusan, pemandu pengetahuan, pembuatan desain, perencanaan, prakiraan, peraturan, pengendalian, penjelasan, diagnosis, perumusan, pemberi nasehat, dan pelatihan. Sistem pakar dapat digunakan oleh orang lain dalam pengambilan keputusan meskipun dirinya bukan seorang pakar.

4. Diagnosis

Diagnosis adalah identifikasi terkait suatu permasalahan. Diagnosis tersebut diperlukan dalam medis, ilmu pengetahuan, teknik, bisnis, dan lain-lain. Diagnosis utama adalah kondisi yang setelah pemeriksaan itu ternyata penyebab utama admission pasien ke rumah sakit untuk dirawat. Jadi, pengertian diagnosis secara terminologi merupakan penetapan keadaan yang menyimpang melalui dasar pemikiran juga pertimbangan ilmu pengetahuan (Fuad et al., 2022).

Diagnosis yaitu penentuan jenis masalah dengan meneliti latar belakang penyebab atau menganalisis gejala-gejala yang terlihat (Arifin, 2020). Menurut Kusumah et al. (2022) Kata "diagnosis" berasal dari

bahasa Yunani, dimana "gnosis" berarti "pengetahuan", menunjukkan hubungan etimologis antara kedua kata tersebut. Sedangkan secara terminologi, pengertian diagnosis yaitu kegiatan medis yang bertujuan untuk mengidentifikasi dan mendefinisikan suatu penyakit dengan menggunakan pengetahuan dan penalaran yang tepat.

Dari beberapa pengertian diagnosis dapat disimpulkan bahwa diagnosis diartikan identifikasi mengenai permasalahan yang digunakan dalam berbagai bidang, termasuk medis, ilmu pengetahuan, teknik, bisnis, dan lain-lain. Pengertian diagnosis secara terminologi adalah penetapan suatu keadaan yang menyimpang atau keadaan normal melalui dasar pemikiran dan pertimbangan ilmu pengetahuan. Diagnosis digunakan untuk menentukan jenis masalah dengan Menggali lebih dalam untuk menemukan sumber utama dari suatu masalah.

5. Penyakit

Menurut Yanuardi (2019) terdapat beberapa definisi penyakit berdasarkan pendapat para ahli:

- a) Menurut Dr. Beate Jacob, penyakit yaitu gangguan kesehatan pada tubuh dapat menyebabkan ketidakseimbangan dalam jiwa.
- b) Menurut Thoma Timmreck, penyakit diartikan sebagai keadaan di mana terdapat gangguan terhadap bentuk maupun fungsi bagian tubuh yang menimbulkan tubuh menjadi tidak dapat berfungsi dengan normal.

- c) Menurut Elizabeth J. Crown, penyakit adalah datangnya sekumpulan tanda-tanda tubuh mengindikasikan tidak normal, yang tidak dapat ditoleransi oleh manusia.

6. Sistem Pakar Diagnosis Penyakit

Kesimpulan yang dapat diambil dari sistem pakar diagnosis penyakit adalah salah satu teknologi yang didukung atau menggantikan sementara peran dokter dalam mendiagnosis penyakit dengan menggunakan basis pengetahuan yang didapatkan dari pakar tenaga kesehatan. Cara kerja sistem ini yaitu menerima input berupa tanda-tanda yang dialami pasien dan kemudian diproses melalui sistem pakar yang memanfaatkan basis pengetahuan. Sistem pakar diagnosis penyakit menghasilkan diagnosa sementara berdasarkan informasi yang dimasukkan, seperti gejala atau data medis pasien. Diagnosa ini memberikan panduan kepada pengguna termasuk orang awam yang bukan ahli medis untuk memahami kemungkinan penyakit yang mereka alami. Manfaat dari sistem ini adalah kemampuannya untuk menyediakan akses terus-menerus memungkinkan layanan diagnosis yang cepat tanpa menunggu waktu lama untuk mengetahui hasil akhirnya.

7. Definisi Website

a) Pengertian *Website*

Website adalah kumpulan dari halaman yang akan digunakan untuk menyajikan jenis informasi, termasuk gambar tak bergerak atau bergerak, teks, suara, animasi, serta kombinasi. Halaman-halaman ini

bisa bersifat statis atau dinamis, dan saling terhubung membentuk satu kesatuan yang saling terkait. Setiap halaman dari *website* dihubungkan oleh sebuah jaringan halaman yang memungkinkan pengguna untuk berpindah antara halaman-halaman tersebut (Dewi Utami et al., 2023).

Menurut Asmara jimi (2019) *website* merupakan semua halaman *website* yang ada sama satu domain dan memuat berbagai informasi. *Website* yang telah dipublikasikan dan memiliki domain dikenal dengan situs *website*. *Website* terdiri dari dinamis dan statis. *Website statis* adalah tipe *website* yang kontennya bersifat permanen dan tidak mudah diubah secara cepat, sedangkan *website dinamis* merupakan tipe *website* yang kontennya dapat diubah dengan mudah dan cepat, memungkinkan pembaruan secara real-time (Astuti & Agustina, 2022).

b) Jenis-Jenis *Website* Berdasarkan Sifat

Menurut Sukma et al. (2020) *website* telah mengalami perkembangan secara signifikan dengan seiring perkembangan teknologi informasi yang pesat. Macam *website* secara sifat adalah sebagai berikut:

1) *Website* Dinamis

Website dinamis yaitu suatu *website* yang menyajikan isi yang terus berubah secara berkala seperti pada www.kompas.com.

2) *Website Statis*

Website statis yaitu *website* yang isinya tidak sering diubah secara rutin dan berisikan saluran informasi. Contohnya pada *web profile* organisasi, dan lain-lain.

c) **Jenis-Jenis *Website* Dari Bahasa Pemrogramannya**

Menurut Saputra & Fahrizal (2021) jenis *website* dari sisi bahasa pemrograman terbagi atas:

1) *Server Side*

Server Side yaitu *website* yang bergantung pada ketersediaan *server* di mana *website* ini dibangun menggunakan bahasa pemrograman yaitu PHP, ASP, .NET, dan sejenisnya. Dalam konteks ini, ketersediaan *server* sangat penting karena *website* yang dikembangkan dengan bahasa pemrograman tersebut tidak dapat berfungsi dengan baik tanpa adanya *server* yang memproses dan menyajikan konten *website* kepada pengguna.

2) *Client Side*

Client side tidak memerlukan *server* dalam mengoperasikannya, cukup dengan mengakses melewati *browser*. Sebagai contoh yaitu HTML.

8. **Basis Data**

a) **Pengertian Basis Data**

Menurut Fathulloh & Adauwiyah (2021) basis data yaitu kumpulan data yang terstruktur dan terorganisir dalam komputer ini

disebut basis data, di mana data dapat dimanipulasi dan dianalisis untuk menghasilkan informasi yang bermanfaat bagi penggunanya. Adanya basis data yang penting sebagai wadah data untuk diolah lebih lanjut. Basis data dapat dikelompokkan dan untuk menghasilkan informasi yang berguna.

b) Perencanaan Basis Data

Menurut Sudarso (2022) perencanaan basis data yaitu bagian dari aktivitas manajemen yang melibatkan pengumpulan data, penentuan format data, pembuatan dokumentasi, serta desain dan implementasi basis data. Perencanaan basis data ini diintegrasikan menggunakan strategi sistem informasi organisasi secara keseluruhan. Ada dua bagian perencanaan basis data yang berhubungan dengan taktik sistem informasi yaitu:

- 1) Menetapkan misi utama dalam pembuatan basis data dengan mendefinisikan tujuan inti (*mission statement*).
- 2) Mengidentifikasi tugas-tugas khusus yang perlu didukung oleh basis data (*mission objectives*).

c) Sistem Basis Data

Basis data memiliki tujuan untuk menggambarkan batasan dan cakupan aplikasi basis data dengan menggunakan sudut pandang pengguna (*user view*). Dengan memperhatikan *user view*, memastikan bahwa semua pengguna diperhitungkan dan membantu dalam pengembangan basis data yang kompleks dengan memecahnya

menjadi sub-sub yang lebih sederhana (Sudarso, 2022). Menurut Mumtahana (2021:3) didapati komponen-komponen sistem basis data, di antaranya:

- 1) Perangkat Keras
- 2) Sistem Operasi
- 3) Basis Data
- 4) Aplikasi DBMS
- 5) Jaringan

d) Desain Basis Data

Dalam membuat basis data terdapat macam model yang bisa digunakan, antara lain yaitu model data datar, hirarki, jaringan, dan relasional. Di antara model-model tersebut, model basis data relasional atau RDBMS (Sistem Manajemen Basis Data Relasional) merupakan model yang paling sering digunakan. Pada pertengahan tahun 2000, muncul model database NoSQL (Mumtahana, 2021:4).

Menurut Sudarso (2022) desain basis data yaitu suatu proses desain yang menyokong operasional serta mencapai tujuan perusahaan untuk:

- 1) Memvisualisasikan sebuah hubungan antara informasi yang diperlukan dan pandangan pengguna (*user view*).
- 2) Memfasilitasi struktur data yang menunjang semua transaksi.
- 3) Menjelaskan secara detail mengenai desain dengan struktur sesuai kebutuhan.

9. *MySQL*

Menurut Silalahi (2022:1) MySQL berkembang pada pertengahan 1990-an, dan mencapai lebih dari 10 juta instalasi. MySQL termasuk sistem manajemen database paling populer untuk server web. Saat ini, MySQL merupakan teknologi matang yang menggerakkan banyak tujuan internet. PHP yang gratis digunakan menjadi salah satu alasan berhasilnya MySQL. Selain itu, MySQL dapat diartikan sebagai *Relational Database Management System* (RDBMS). Sistem perangkat lunak MySQL terdiri dari server MySQL sebagai komponen utama.

Menurut Marathe (2022:512) MySQL dipaparkan sebagai salah satu *frontend* DBMS di mana Taurus mendukung di dalamnya, serta Pada konfigurasi tersebut, MySQL memanfaatkan pengoptimal kueri dan mesin eksekusinya untuk menyusun dan menjalankan pertanyaan yang diberikan. Menurut Fathulloh & Adauwiyah (2021) MySQL dapat menjadi basis data yang populer seperti sekarang karena mempunyai kehebatan terhadap fitur-fiturnya, Salah satu fitur yang paling terkenal adalah kecepatan. E-Week membandingkan beberapa database, termasuk MySQL., MS SQL, Oracle, Sybase ASE, serta IBM DB2. Dibuktikan dengan MySQL dan Oracle yang mempunyai kemampuan terbaik perihal *performance* dan *scalability*. MySQL bisa dengan lancar dan cepat menyelesaikan tabel yang berjumlah puluhan ribu serta *record* data dengan jumlah milyaran.

Berdasarkan penjelasan para ahli terkait MySQL maka dapat diambil kesimpulan yaitu MySQL merupakan platform perangkat lunak

populer untuk mengelola dan mengatur data terstruktur dalam basis data relasional dikenal sebagai sistem manajemen basis data relasional, terutama dalam konteks server web. MySQL plugin yang banyak dikunjungi dan digunakan karena MySQL gratis serta berbagai fitur yang ditawarkannya. salah satu yang paling dikenal yaitu terkait kecepatan.

10. Laragon

Laragon merupakan perangkat lunak bebas yang berfungsi sebagai server mandiri atau *localhost*. *Laragon* memiliki banyak *services*, *tools*, dan fitur seperti apache, MySQL, PHP Server, Memcached, Redis, Composer, Xdebug, PhpMyAdmin, Cmdr serta Laravel (Harianto et al., 2019:14). Dengan demikian, laragon sangat berguna untuk membangun dan mengelola aplikasi web, serta memudahkan pengembang untuk mengelola berbagai proyek dengan lebih mudah.

Menurut Irianto et al. (2020) *laragon* diartikan sebagai *Universal development environment* untuk PHP, Node.js, Python, Java, Go, Ruby, yang portable dan terisolasi, laju, ringan, serta mudah digunakan. Laragon memiliki beberapa fungsi, di antaranya:

1) *Pretty URL*

Dengan laragon, kita dapat mengakses *project* melalui *app test* daripada menggunakan *localhost* atau *app*.

2) *Portable*

Folder *laragon* dapat dipindahkan tanpa merusak sistem.

3) *Isolated*

Sistem Laragon terisolasi dari sistem operasi yang utama dimiliki sehingga apapun yang dilakukan terhadap laragon tidak mempengaruhi komputer lokal.

4) *Easy Operation*

Laragon sudah mengotomatiskan tidak sedikit konfigurasi kompleks sehingga pengguna tinggal menggunakan saja. Pengguna dapat menambah versi baru PHP, Python, Ruby, MongoDB maupun yang lainnya tanpa kesukaran sehingga tidak akan mengganggu sistem yang sudah ada.

5) *Modern dan Powerful*

Laragon mempunyai arsitektur terbaru yang cocok untuk digunakan dalam pengembangan web modern. Sebagai contoh, pengguna dapat menggunakan Apache dan Nginx secara bersamaan.

Kesimpulan yang diperoleh dari penjelasan di atas adalah *laragon* merupakan perangkat lunak *open-source* yang mampu berjalan di berbagai sistem operasi dan berperan sebagai web server. *Laragon* menawarkan berbagai layanan dan fitur yang sangat memudahkan pengembang dalam membangun dan mengelola aplikasi berbasis web. *Laragon* menjadi pilihan yang disarankan bagi pengembang yang membutuhkan solusi pengembangan yang fleksibel, ringan, dan cepat.

11. Bahasa Pemrograman

a) *HTML (HyperText Markup Language)*

Menurut Adam Saputra (2019:2) pada tahun 1989, Tim Berners Lee memunculkan *Hyper Text Markup Language* (HTML) yang kemudian dikembangkan oleh *World Wide Web Condortium* (W3C), serta pada tahun 2004 dikembangkan *Web Hypertext Application Tecnology Working Group* (WHATG) hingga saat ini adanya perkembangan bahasa HTML. Menurut Annafri et al. (2023) *Hypertext Markup Language* (HTML) merupakan pemrograman dasar dan esensial untuk membangun jendela *website* yang dapat diakses dan menampilkan berbagai informasi. HTML berfungsi sebagai penyusun data dan informasi, memungkinkan dokumen terstruktur dan ditampilkan melalui internet dengan bantuan layanan web.

b) *CSS (Cascading Style Sheet)*

CSS (Cascading Style Sheets) merupakan pemrograman sederhana dan paling mudah digunakan untuk menambahkan gaya pada sebuah halaman web saat dibangun. CSS juga dikenal sebagai "*style sheets*" dan berbasis gaya yang dapat meningkatkan keindahan visual situs web (Kumar, 2022). Selain itu CSS berguna untuk menetapkan cara elemen HTML ditampilkan, layaknya menentukan posisi, mengubah warna teks dan lain sebagainya fungsi dari CSS (*Cascading Style Sheets*).

Cascading Style Sheets (CSS) yaitu bahasa pemrograman yang digunakan untuk mendesain sajian pada *website*. CSS membantu untuk memisahkan antara struktur HTML dengan tampilan visual yang diterapkan pada *website*. Menggunakan CSS dalam pembuatan *website* dapat mengatur warna, jenis huruf, ukuran *font*, posisi, dan lain sebagainya. CSS memberikan *fleksibilitas* yang lebih besar dalam mendesain tampilan *website* (Wibowo, 2023:16).

c) ***Javascript***

Javascript yaitu bahasa pemrograman yang dalam pemrosesannya dibuat di sisi *browser* dan dapat memanipulasi HTML dan CSS secara dinamis (Andriati, 2023). Menurut Rizki et al. (2024:5) *javascript* dapat diartikan sebagai bahasa pemrograman yang digunakan oleh para pengembang untuk menciptakan halaman web interaktif. *Javascript* menjadi salah satu teknologi inti yang membangun *World Wide Web (WWW)*. Contohnya seperti, menjelajahi internet, corosel gambar, menu *dropdown* yang dapat diklik untuk tampil, dan elemen yang berubah warna secara dinamis dihalaman web.

d) ***PHP (Hypertext Preprocessor)***

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman server-side yang bersinergi dengan HTML untuk membangun aplikasi web dinamis, memungkinkan pemrosesan dan manipulasi data secara interaktif. Semua *subrtax* yang diberikan akan

dijalankan pada server, sedangkan hasilnya akan dikirimkan ke *browser*. PHP merupakan bahasa berbentuk *script* yang diletakkan *browser*. PHP disebut juga dengan *scripting*, yaitu menggabungkan dengan tag-tag HTML, dieksekusi di server, dan dipakai guna membuat halaman web yang dinamis layaknya *Active Server Pages* (ASP) atau *Java Server Pages* (JSP), PHP termasuk sebuah *software Open Source* (Hermiati et al., 2021)

12. *Framework*

Menurut Yudhanto & Prasetyo (2019:10) *framework* merupakan sebuah kelompok kode (terutama kelas dan fungsi) yang bisa membantu pengembang dalam mengatasi berbagai permasalahan pemrograman, layaknya koneksi ke basis data, pengelolaan variabel, pengelolaan file, dan sebagainya, sehingga pengembangan aplikasi dapat dilakukan dengan lebih fokus dan cepat. *Framework* memiliki komponen pemrograman yang dapat dipakai ulang kapanpun sehingga pengembang tidak perlu membuat *script* dan tugas yang sama. Secara sederhana, dapat diterangkan bahwa *framework* merupakan kumpulan fungsi (*libraries*) yang pengembang tidak perlu membuatnya dari awal, yang biasanya disebut dengan *library*.

13. *Laravel*

a) **Pengertian *Laravel***

Menurut Asrozy et al. (2022) *Laravel* merupakan sebuah *framework* yang dimanfaatkan dalam pengembangan aplikasi web kontemporer. *Laravel* yaitu *framework open source* yang bisa

dijalankan secara bebas oleh banyak orang dalam membangun aplikasi web. Taylor Otwell merupakan pencipta dan pengembang utama *laravel*, dimulai pada tahun 2011. Menurut Purbo (2021) *laravel* dirilis pada tanggal 5 Juni 2011, *laravel* diperkenalkan dengan lisensi MIT License dan menggunakan GitHub untuk situs membagikan program.

Laravel didasarkan pada arsitektur *Model-View-Controller* (MVC). *Model-View-Controller* (MVC) merupakan pola desain berisi tiga bagian utama: model, tampilan dan pengontrol. Komponen-komponen tersebut dikonfigurasi sedemikian rupa Menangani aspek pengembangan spesifik aplikasi (Subecz, 2021). Model yang mewakili struktur data berisi berbagai fungsi yang menolong orang yang menangani basis data, seperti memasukkan dan memverifikasi data, dan lainnya. *View* adalah bagian yang mengatur tampilan ke pengguna biasanya disebut dengan halaman web. Bagian yang menghubungkan model dan tampilan disebut *controller*. *Controller* memiliki perintah-perintah yang berguna memproses suatu data dan mengirimkan ke halaman web (Yudhanto & Prasetyo, 2018:6).

b) Keunggulan *Laravel*

Menurut Tahir et al. (2019) *framework laravel* mempunyai kelebihan yakni, kinerja yang lebih cepat, reload data yang lebih stabil, keamanan data yang terjamin, penggunaan fitur canggih seperti *Blade* dengan konsep HMVC (*Hierarchical Model View Controller*),

dan tersedianya library yang siap pakai serta terdapat fitur pengelolaan *migrations* guna pembentukan skema *table* pada database. *Migrations* dapat membuat skema table pada *database* dengan cara lebih mudah sehingga *website* dapat berjalan dengan baik tidak adanya *error*.

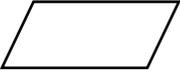
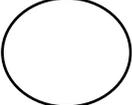
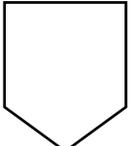
14. *Flowchart*

Flowchart adalah gambaran yang menunjukkan alur atau urutan langkah yang berbentuk bagan untuk menyelesaikan suatu masalah (Audita et al., 2022). *Flowchart* (diagram alir) merupakan bagan (*Chart*) yang mengarahkan alir (*flow*) termasuk program sistem secara logika. *Flowchart* diartikan sebagai cara untuk memaparkan tahapan penyelesaian masalah dengan merepresentasikan tanda-tanda tertentu yang mudah dipahami, digunakan dan standar (Syamsiah, 2019).

Flowchart berfungsi dalam memutuskan langkah dari proyek pembuatan program yang melibatkan banyak orang. Bagian alur akan membuat program lebih mudah dipahami dan lebih ringkas, sehingga mengurangi kemungkinan salah penafsiran. Penggunaan *flowchart* dalam dunia pemrograman merupakan cara yang baik guna mengaitkan antara kebutuhan teknis dan non-teknis (Septima et al., 2024).

Menurut Rahman (2019) berikut ini merupakan simbol-simbol dari *flowchart* sebagai berikut:

Tabel 2 1 Simbol *Flowchart*

No	Gambar	Simbol	Keterangan
1		Terminator	berfungsi guna menunjukkan titik awal maupun akhir sebuah program
2		Garis Alir	digunakan dalam menggambarkan alur atau aliran program
3		Proses	Simbol yang digunakan untuk memproses perintah pengolahan data
4		Input Output Data	digunakan guna menggambarkan proses memasukan serta mengeluarkan data
5		Decision	Simbol yang berfungsi untuk memberikan pilihan antara ya atau tidak
6		On Page Connector	Simbol yang menghubungkan bagian flowchart dalam page sama yang berbeda-beda.
7		Off Page Connector	Simbol yang menghubungkan part flowchart dalam page yang berbeda.

Sumber: Septima et al. (2024)

Berdasarkan penjelasan beberapa ahli di atas bisa disimpulkan bahwa *flowchart*, layaknya skema alur kerja, memvisualisasikan langkah-langkah dan urutan logis dalam suatu proses atau kegiatan, membantu pemahaman dan analisis. Diagram ini menggunakan simbol-simbol

tertentu yang memiliki arti masing-masing untuk menjelaskan tahap-tahap pemecahan masalah dan memudahkan pemahaman. *Flowchart* memiliki peran utama dalam menentukan langkah pembuatan suatu program atau proyek yang melibatkan banyak orang, serta memberikan gambaran yang jelas dan ringkas dari proses yang kompleks. *Flowchart* membuat proses dari sebuah program menjadi lebih mudah dipahami.

15. *Certainty Factor* (CF)

Menurut pendapat Sukiakhy et al. (2022) metode *Certainty Factor* (CF) digunakan untuk mengendalikan ketidakpastian sistem yang berbasis aturan. Metode CF dikembangkan oleh Shortliffe dan Buchanan sekitar tahun 1970 sebagai pendekatan pengelola ketidakpastian untuk MYCIN, sebuah sistem pakar yang berfungsi untuk mendiagnosa sekaligus mengobati infeksi darah dan meningitis. Sejak saat itu metode CF menjadi pendekatan manajemen ketidakpastian standar untuk suatu sistem yang berbasis aturan.

Menurut Sembiring et al. (2019) metode *Certainty Factor* (CF) mengenalkan konsep kepastian dan ketidakpastian. Rumus dasar *Certainty Factor* (CF) yaitu:

$$CF(H, E) = MB(H, E) - MD(H, E) \quad (1)$$

$$MB(H, E) = \frac{Max[P(H|E), P(H)] - P(H)}{Max[1,0] - P(H)} \dots P(H) = 1 \quad (2)$$

$$MD(H, E) = \frac{Min[P(H|E), P(H)] - P(H)}{Min[1,0] - P(H)} \dots P(H) = 0 \quad (3)$$

Keterangan:

- CF(H,E) : *Certainty Factor* (CF) mempengaruhi hipotesis H yang dipengaruhi oleh fakta E.
- MB(H,E) : *Measure of unbelief* (ukuran tingkat kepercayaan) terhadap hipotesis H, jika diberikan fakta E (dalam rentang antara 0 dan 1)
- MD(H,E) : *Measure of unbelief* (ukuran ketidakpercayaan) terhadap hipotesis H, jika diberikan bukti E (dalam rentang antara 0 dan 1)
- P(H) : Probability hipotesis H
- P(H|E) : Persamaan adalah bahwa H benar karena fakta E

16. UML

a) Definisi UML

Menurut Aurin et al. (2021) *Unified Modeling Language* (UML) yaitu sebuah bahasa yang dipergunakan guna mendefinisikan, membangun, memvisualisasikan, serta mengabadikan artefak-artefak dalam pengembangan sistem perangkat lunak. UML dapat digunakan dalam pemodelan bisnis dan sistem non-perangkat lunak lainnya. UML juga dapat digunakan sebagai bahasa pemodelan yang berbasis pada konsep pemrograman berorientasi objek. Menurut Prasetya et al. (2022) UML sangat mudah diaplikasikan dalam berbagai domain, termasuk sistem informasi perusahaan, sistem perbankan dan perekonomian, telekomunikasi, transportasi, penerbangan,

perdagangan, pelayanan elektronik, pengetahuan, dan pelayanan berbasis web terdistribusi.

b) Diagram Pemodelan UML

1) *Use Case*

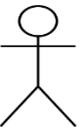
Menurut Stiawansyah et al. (2022) *use case* merupakan suatu pendekatan awal dalam pemodelan perilaku sistem informasi yang bermanfaat untuk menggambarkan interaksi umum antara pengguna suatu sistem dengan sistem itu sendiri. *Use case* menggambarkan interaksi tersebut melalui deskripsi penggunaan sebuah sistem. *Use case* diartikan sebagai rancangan awal sistem informasi dimana terdapat sistem dan aktor yang bersifat fungsional dan memiliki hubungan langsung terhadap sistem tersebut. Dengan demikian, *use case* dapat disimpulkan sebagai gambaran awal dari suatu sistem informasi dan interaksi *user* dengan sistem.

2) *Use case Diagram*

Menurut Hamas & Imaduddin (2019) *use case diagram* digunakan untuk menjelaskan hubungan antara sistem dengan dunia luar, atau menggambarkan dan merepresentasikan aktor, kasus penggunaan (*use case*), dan ketergantungan. Pengertian lain menurut Y. W. S. Putra et al. (2022) *use case diagram* digunakan untuk menjelaskan visualisasi konsep, struktur, atau hubungan antara elemen-elemen dalam suatu sistem atau proses.

berbagai macam aktivitas yang dapat dilakukan aktor dalam sistem yang berjalan sekaligus tujuan yang perlu dipenuhi oleh sistem menurut pendapat aktor. Simbol-simbol *use case* diagram dapat digambarkan sebagai berikut:

Tabel 2 2 Simbol *Use Case* Diagram

No	Gambar	Simbol	Keterangan
1.	 Actor	<i>Actor</i>	Elemen sistem yang menjelaskan hubungan peran atau entitas pengguna yang berinteraksi dengan sistem informasi yang dibuat..
2.		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) mewarisi struktur data dari objek induk(<i>ancestor</i>) yang berda di atasnya.
3.		<i>Include</i>	Ketentuan yang secara eksplisit menunjukkan bahwa suatu <i>use case</i> menggabungkan <i>use case</i> sumber ke dalamnya.
4.		<i>Extends</i>	Merupakan ketentuan relasi yang menyatakan bahwa <i>use case</i> memperluas perilaku <i>use case</i> sumber pada titik tertentu.

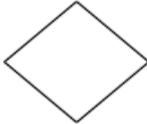
5.	<i>Association</i>	Koneksi penghubung antara satu objek dengan objek lainnya.
----	--------------------	--

Sumber: Hutabri & Putri (2019)

3) *Activity* Diagram

Menurut Musthofa & Adiguna (2022) *activity* diagram merupakan sebuah diagram yang digunakan untuk memodelkan berbagai proses atau aktivitas yang terjadi dalam suatu sistem. Diagram ini menggambarkan runtutan alur kerja atau proses secara visual dan terstruktur, seringkali dengan orientasi vertikal. Diagram aktivitas adalah bagian dari kumpulan diagram *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan *use case* dalam pengembangan sistem. Pengertian lain menurut Norrahim & Zainal (2022) *activity* diagram menggambarkan operasi bisnis dan perangkat lunak sebagai serangkaian tindakan yang dilakukan oleh manusia, komponen perangkat lunak, dan komputer. *activity* diagram ini digunakan untuk mendokumentasikan implementasi proses sistem serta untuk mendefinisikan proses bisnis dan kasus penggunaan. Simbol-simbol pada *activity* diagram dapat dilihat sebagai berikut:

Tabel 2 3 Simbol *Activity Diagram*

Simbol	Deskripsi
 Status awal/ akhir	Status awal dan akhir yang menunjukkan tempat eksekusi akan dimulai.
 Aktivitas	Kegiatan yang dilakukan sistem maupun merepresentasikan aktivitas yang terlibat dalam proses sistem.
 Decision (percabangan)	Hubungan guna memutuskan aktivitas yang harus dibuat dalam alur eksekusi dengan kondisi yang diberikan.
 Join (Penggabungan)	Merupakan titik alur eksekusi terbagi bertemu kembali menjadi satu eksekusi tunggal
 Swimlane	Merupakan pemisah organisasi dalam kelompok yang berbeda berdasarkan entitas yang terlibat.

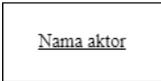
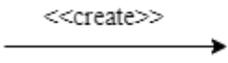
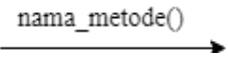
Sumber: Hutabri & Putri (2019)

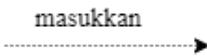
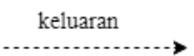
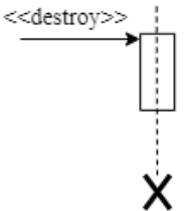
4) *Sequence Diagram*

Menurut Musthofa & Adiguna (2022) *sequence diagram* digunakan untuk menggambarkan interaksi antar objek dalam sebuah sistem atau proses, biasanya menunjukkan urutan pesan yang dikirim antar objek selama eksekusi waktu tertentu use case, menunjukkan urutan tindakan, pesan, dan respon. Menurut Hutabri & Putri (2019) Diagram Sekuen (*Sequence Diagram*)

bekerja dengan menjabarkan alur waktu hidup sebuah objek sekaligus pesan yang dikirim dan diterima objek tersebut. Simbol-simbol *sequence* diagram dapat dilihat sebagai berikut ini:

Tabel 2 4 Simbol *Sequence* Diagram

Simbol	Deskripsi
<p>Aktor nama aktor Atau</p>  <p>Tanpa waktu aktif</p> 	<p>Merupakan orang atau sistem atau proses di luar sistem yang dibuat dan masih memiliki hubungan dengan sistem tersebut.</p>
 <p><i>Lifeline</i> (garis hidup)</p>	<p>Garis aktif yang menunjukkan keaktifan objek.</p>
 <p>Objek</p>	<p>Objek yang menjalankan representasi pada entitas yang terlibat dalam hubungan antar entitas.</p>
 <p>Waktu aktif</p>	<p>Merupakan suatu objek yang sedang dalam keadaan aktif atau berinteraksi, semua yang terlibat terhadap waktu aktif adalah dalam keadaan berjalan.</p>
 <p>Pesan tipe <i>create</i></p>	<p>Menggambarkan pesan saat objek yang baru dibuat dan dikaitkan dengan objek yang sesuai.</p>
 <p>Pesan tipe <i>call</i></p>	<p>Merupakan gambaran yang menunjukkan pesan berupa pemanggilan operasi antara objek.</p>

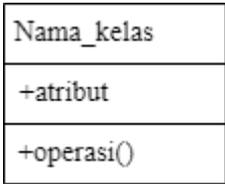
	Merupakan gambaran pengiriman pesan antar objek yang menunjukkan komunikasi.
Pesan tipe <i>send</i>	
	Merupakan objek dapat melakukan atau menjalankan metode yang pada operasi memberikan hasil pengembalian atau keluaran ke objek tertentu.
Pesan tipe <i>return</i>	
	Pernyataan bahwa satu objek mengakhiri hidup dari objek lain, jika ada <i>create</i> lebih baik ada <i>destroy</i> .
Pesan tipe <i>destroy</i>	

Sumber: Hutabri & Putri (2019)

5) Class Diagram

Class diagram merupakan diagram pada UML yang berfungsi untuk memberikan pola mengenai kelas-kelas yang ada pada suatu sistem yang menjelaskan mengenai sistem serta relasi yang dimiliki sistem (Nurwulan & Choldun, 2020). Menurut Arianti et al. (2022) mengartikan bahwa *class* diagram merupakan spesifikasi merupakan inti dari pengembangan atau desain yang berorientasi objek, dan dapat menghasilkan sebuah objek jika diinstansiasi. *Class* diagram digunakan untuk menjelaskan keadaan (atribut atau properti) dalam suatu sistem serta memaparkan layanan yang dapat mengubah keadaan tersebut (metode atau fungsi). Simbol-simbol pada *class* diagram dapat dilihat pada tabel berikut ini:

Tabel 2.5 Simbol *Class Diagram*

Simbol	Deskripsi
 <p><i>Class</i> (kelas)</p>	Memiliki tiga bagian horizontal, bagian nama kelas menjelaskan identitas kelas yang direpresentasikan, bagian tengah berisi atribut mewakili variabel data dari kelas tersebut, bagian metode kelas berisikan operasi atau fungsi yang dilakukan kelas.
 <p><i>Interface</i> (antarmuka)</p>	Merupakan kemiripan kelas yang digunakan untuk menggambarkan kontrak metode abstrak yang diimplementasikan tetapi dalam pendeklarasian tanpa isi dan tanpa atribut kelas.
 <p><i>Association</i></p>	Merupakan relasi antar kelas yang dilengkapi dengan <i>multiplicity</i> dan menunjukkan ketertarikan antar kelas tersebut.
 <p><i>Directed association</i></p>	Merupakan relasi antara dua kelas yang memodelkan interaksi dan ketergantungan antara kelas dalam sistem.
 <p>Generalisasi</p>	Merupakan relasi kelas dengan bermakna generalisasi.
 <p><i>Dependency</i></p>	Pewarisan antar kelas yang menunjukkan hierarki.
 <p><i>Aggregation</i></p>	Relasi kelas bermakna semua bagian (<i>whole-part</i>)

Sumber: Hutabri & Putri (2019)

17. Metode *Rapid Application Development* (RAD)

Rapid Application Development (RAD) merupakan model pengembangan perangkat lunak yang menitikberatkan pada pembuatan prototipe secara cepat serta mendapatkan umpan balik dengan segera, sehingga menghindari siklus pengembangan dan pengujian yang panjang dan terus menerus. Dengan metode ini, pengembang dapat melakukan beberapa iterasi dan pembaruan pada perangkat lunak dengan cepat tanpa harus memulai ulang setiap kali. RAD pertama kali diperkenalkan pada tahun 1980-an. (Aprilian Lusita Violita & Saputra Harry, 2020:63). Menurut Wijayanto et al. (2024:43-44) ada 4 tahapan dalam metode RAD sebagai berikut:

a. Perancangan Kebutuhan

Tahap ini merupakan langkah awal dalam pengembangan sistem. Pada tahap ini, masalah diidentifikasi dan data dikumpulkan dari para stakeholder atau pengguna untuk menentukan tujuan akhir dan kebutuhan informasi yang diinginkan dari sistem. Partisipasi kedua belah pihak sangat penting untuk mengidentifikasi kebutuhan sistem secara benar.

b. Desain Sistem

Pada tahap desain sistem, partisipasi aktif dari pengguna sangat penting karena proses desain dan perbaikan dilakukan berulang kali untuk memastikan kesesuaian desain dengan kebutuhan pengguna yang telah diidentifikasi sebelumnya. Hasil dari langkah ini adalah

spesifikasi perangkat lunak yang mencakup struktur data, organisasi sistem secara keseluruhan, dan elemen lainnya

c. Perancangan Sistem

Pada tahap ini, desain sistem yang telah dirancang dan disetujui diubah menjadi aplikasi versi beta sebelum mencapai versi final. Selama tahap ini, programmer harus terus melakukan pengembangan dan mengintegrasikan aplikasi dengan komponen lain sambil memperhatikan umpan balik dari pengguna atau klien. Jika proses ini berjalan lancar, pengembangan dapat berlanjut ke tahap berikutnya. Namun, jika aplikasi belum memenuhi kebutuhan, programmer akan kembali ke tahap desain sistem.

d. Implementasi

Pada tahap ini, programmer mengimplementasikan desain sistem yang telah disetujui sebelumnya. Sebelum sistem dijalankan, pengujian dilakukan untuk menemukan kesalahan pada program. Di tahap ini, programmer biasanya memberikan *feedback* terhadap sistem yang telah dibuat dan mendapatkan persetujuan.

18. **Black box Testing**

Black box testing adalah metode yang berfungsi untuk menguji perangkat lunak mengenai fungsionalitas aplikasi yang berlawanan dengan struktur internal. Metode *black box* difungsikan untuk pengujian program bertujuan untuk memastikan semua fitur berjalan sesuai dengan harapan. Metode pengujian *black box* dapat membuat pengembang *software* untuk

merancang himpunan kondisi input yang digunakan untuk melatih syarat fungsional dari suatu program. Pengujian ini dilakukan dengan memilih beberapa modul yang memiliki tipe data berbeda-beda agar dapat digunakan untuk memastikan bahwa program tersebut hanya dapat menerima input data yang sesuai dengan tipenya (A. P. Putra et al., 2020).

Menurut Ramdani et al. (2023) *black box* testing memiliki kelebihan dan kelemahan dalam pengujian sistem. Berikut kelebihan dan kelemahan pengujian *black box*:

a) Kelebihan *Black box*

- 1) Tidak memerlukan pengetahuan tentang bahasa pemrograman tertentu.
- 2) Pengujian dilakukan dari sudut pandang pengguna,
- 3) Ketergantungan antara programmer dan tester.

b) Kelemahan *Black box*

- 1) Terdapat kemungkinan bahwa programmer telah melakukan tes yang berulang.
- 2) Beberapa bagian *back end* dilewatkan dan tidak diuji sama sekali.

Menurut Muhardin et al. (2020) Pendekatan *black box* adalah metode pengujian yang melengkapi pengujian *white-box* dengan kemampuan untuk mengungkap lebih banyak jenis kesalahan. Pengujian *black box* fokus pada menemukan kesalahan dalam beberapa kategori berikut:

- a) Fungsi yang salah atau tidak ada
- b) Kesalahan antarmuka
- c) Kesalahan struktur data atau akses dasar data eksternal
- d) Kesalahan kinerja dalam sistem
- e) Kesalahan inisialisasi dan penghentian

Berdasarkan penjelasan diatas dapat diambil kesimpulan bahwa *blackbox* testing merupakan metode pengujian sebuah perangkat lunak yang berfokus pada fungsionalitas aplikasi dan tidak mempertimbangkan struktur internal atau kerja. Metode ini digunakan untuk memastikan semua fitur berjalan sesuai dengan harapan dan meningkatkan kemungkinan Pengembang perangkat lunak merancang himpunan kondisi input untuk melatih syarat fungsional suatu program. Kelebihan pengujian *blackbox* adalah bahwa mereka tidak memerlukan pengetahuan bahasa pemrograman tertentu; pengujian ini dilakukan dari sudut pandang pengguna, dan mereka menciptakan ketergantungan antara pembuat dan tester. Namun, kelemahan pengujian *blackbox* termasuk kemungkinan bahwa programmer telah mengulangi tes yang sudah dilakukan, serta kemungkinan bahwa beberapa komponen back end tidak diuji sama sekali.

B. Kajian Empiris

Pada Penelitian ini, ada beberapa kajian empiris yang menjadi referensi dalam melakukan penelitian, Hal ini penulis mengambil 4 penelitian yang relevan yaitu:

Penelitian yang sesuai mengenai pembuatan sistem pakar diagnosa penyakit pada kucing dengan metode *Certainty Factor* (CF) berbasis *website*. Penelitian ini dilakukan oleh Kharisma & Hakim (2022) berjudul Sistem Pakar Diagnosa Penyakit Sapi Dengan Metode *Certainty Factor* berbasis web. Metode pengembangan sistem mengikuti tahapan berurutan: perencanaan, analisis, desain, pengembangan, pengujian, implementasi, operasi, dan pemeliharaan. Sedangkan metode yang digunakan untuk menghitung tingkat ketidakpastian dan mengukur tingkat keyakinan dalam diagnosa adalah metode *Certainty Factor* (CF). Penelitian ini memiliki hasil bahwa sistem memungkinkan peternak sapi untuk melakukan diagnosa awal secara mandiri berdasarkan gejala yang diinputkan. Namun, disarankan agar hasil diagnosa tetap dikonsultasikan dengan dokter hewan terutama jika tingkat kepercayaan hasil diagnosa di bawah 75%, guna menghindari kesalahan dalam penanganan. Penelitian ini menunjukkan bahwa sistem pakar berbasis web berfungsi dengan baik dan dapat memberikan manfaat signifikan dalam membantu peternak sapi menangani penyakit pada ternak mereka.

Penelitian serupa yang dilakukan Bere et al. (2021) berjudul “Sistem Pakar Diagnosa Penyakit Pada Ayam Pedaging Berbasis Web Menggunakan Metode *Certainty Factor*” Sistem ini bertujuan untuk membantu peternak di daerah Cigalontang yang memiliki keterbatasan pengetahuan dan akses terhadap pakar penyakit ayam. Sistem tersebut menggunakan metode *Certainty Factor* (CF) dalam menghitung tingkat keyakinan dalam proses diagnosa penyakit dengan mempertimbangkan hubungan antara gejala dan penyakit

yang mungkin terjadi. Pengembangan sistem ini mengikuti tahapan ESDLC (*Expert System Development Life Cycle*), yang mencakup pengetahuan, representasi pengetahuan, validasi pengetahuan, desain mesin inferensi, desain antarmuka pengguna, pengujian dan evaluasi, serta implementasi dan pemeliharaan. Hasil pengujian menunjukkan bahwa sistem tersebut dapat memberikan diagnosa dalam waktu singkat.

Penelitian serupa yang dilakukan Arini et al. (2020) dengan judul "Analisis Akurasi Kombinasi Proses *Text Mining* dan *Certainty Factor* dalam Diagnosis Kerusakan Printer" dalam pembuatan sistem ini telah dilakukan kombinasi proses *tekx mining* dan *certainty factor* dalam mendiagnosis kerusakan printer dari gejala-gejala yang terlihat. Sedangkan pengembangan sistem menggunakan metode RAD (*Rapid Application Development*) yang terdiri dari tiga fase utama yaitu, tahap perencanaan, tahap *workshop* desain, dan tahap implementasi pengkodean. Pengembangan untuk sistem ini yang dibutuhkan relatif lebih singkat dan cepat sesuai dengan kebutuhan pengguna. Penelitian ini menggunakan PHP dengan *framework laravel* untuk mengembangkan aplikasi, serta MySQL untuk basis data. Penelitian ini menghasilkan aplikasi sistem pakar berbasis website untuk mendiagnosis kerusakan pada printer yang menggabungkan proses *text mining* dan *certainty factor* dengan akurasi 58,33%.

Peneliti serupa mengenai sistem berbasis website dengan menggunakan *framework laravel* dalam mendiagnosis penyakit rhinitis pada tubuh manusia. Penelitian ini dilakukan oleh NurJumala et al. (2022) dengan judul "Sistem Pakar

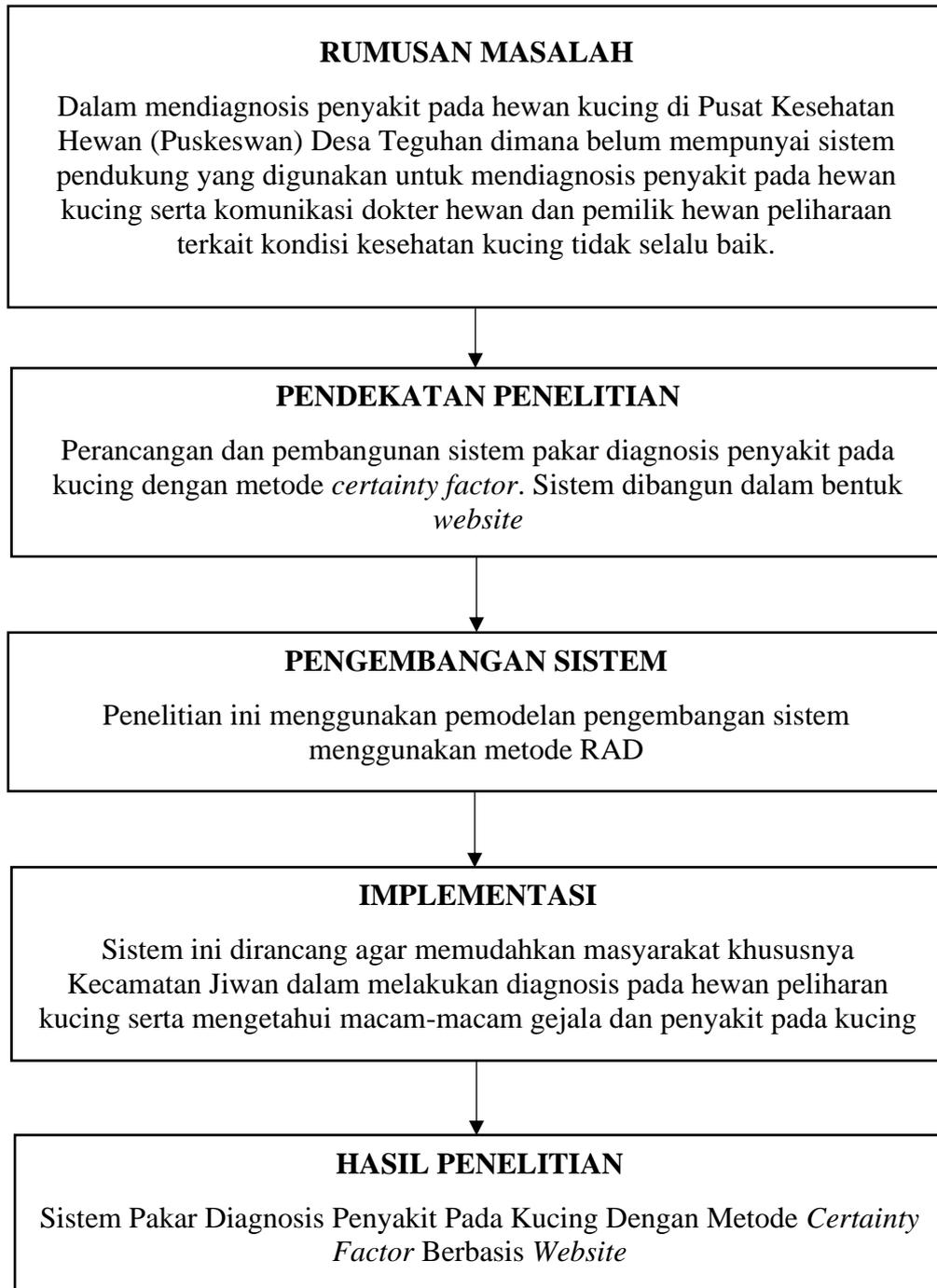
Diagnosis Penyakit Rhinitis Menggunakan Metode *Forward Chaining* Berbasis Web“. Sistem pakar tersebut dibuat menggunakan metode *forward chaining*, dengan menggunakan bahasa pemrograman PHP dan didukung oleh *framework laravel*. Hasil dari penelitian ini menyatakan bahwa sistem yang dikembangkan mampu mendiagnosa penyakit rhinitis dengan baik dan berhasil, dibuktikan dengan pengujian akurasi oleh pakar THT. Sistem pakar ini dapat memudahkan pengguna untuk melakukan diagnosis secara cepat tanpa harus menunggu lama untuk mengetahui hasilnya dan membantu meringankan kinerja seorang pakar.

Berdasarkan kajian empiris dari berbagai penelitian tentang sistem pakar berbasis *website* diatas dapat diambil kesimpulan, bahwa metode *Certainty Factor* (CF) tepat dalam menangani ketidakpastian dan menghitung tingkat keyakinan dalam proses diagnosa. Sistem-sistem yang dikembangkan berdasarkan penelitian terdahulu diatas berhasil seperti pendapat Kharisma & Hakim (2022) dan Bere et al. (2021) berhasil membantu peternak dalam melakukan diagnosa awal penyakit pada sapi dan ayam, menunjukkan bahwa sistem pakar ini sangat berguna terutama di daerah dengan akses terbatas. Selain itu, pengembangan sistem menggunakan pendekatan metode yang berbeda seperti Waterfall, ESDLC, dan RAD serta penerapan teknologi modern seperti PHP, *laravel*, dan MySQL sesuai yang diuraikan oleh Arina et al (2020) dan NurJumala et al. (2022). Penggunaan *framework laravel* dalam pengembangan aplikasi sistem pakar berbasis web untuk diagnosis penyakit menunjukkan bahwa hasil diagnosis berhasil, manfaat dari sistem pakar dalam

berbagai konteks, baik medis maupun teknis. Namun, meskipun sistem ini mampu memberikan diagnosa yang cepat tidak menunggu lama untuk mengetahui hasilnya, hasil diagnosa tetap harus dikonfirmasi dengan pakar, terutama jika tingkat keyakinan di bawah batas tertentu untuk memastikan penanganan yang tepat dan mencegah kesalahan. Secara keseluruhan, penelitian-penelitian ini memperlihatkan potensi besar sistem pakar berbasis web dalam memberikan solusi diagnosa yang jelas dan membantu dalam pengambilan keputusan yang lebih cepat dan memberikan informasi yang jelas.

C. Kerangka Berpikir

Berikut ini merupakan penjelasan tentang kerangka berpikir pada penelitian yang telah dilaksanakan.



Gambar 2. 1 Kerangka Berpikir