

BAB II

KAJIAN PUSTAKA

A. Kajian Teoris

1. Capaian Pembelajaran Lulusan (CPL)

Menurut panduan yang diterbitkan STIE Kasih Bangsa (2021:1), Capaian Pembelajaran Lulusan (CPL) merupakan seperangkat kompetensi yang mencakup aspek keterampilan, pengetahuan, maupun sikap yang wajib dimiliki oleh mahasiswa sebagai buah dari proses pembelajaran yang telah ditempuh. Desain pembelajaran yang berorientasi pada perkembangan mahasiswa ini memberikan kemudahan bagi perguruan tinggi dalam menyusun jalur belajar yang lebih personal dan adaptif sesuai kebutuhan berbagai kelompok peserta didik. Dalam rumusan CPL terkandung pernyataan-pernyataan yang bersifat konkret dan terukur mengenai dimensi pengetahuan, kecakapan, serta sikap yang menjadi tolok ukur ketercapaian tujuan pendidikan. Keberadaan CPL memberikan manfaat ganda, yakni membantu mahasiswa dalam memahami secara jelas arah dan substansi pembelajaran mereka, sekaligus mempermudah dosen dalam merancang proses pembelajaran, melaksanakan penilaian, serta menjadi acuan bagi lembaga akreditasi dalam memverifikasi kesesuaian mata kuliah dengan standar yang berlaku.

CPL dapat dipahami sebagai suatu pernyataan yang menggambarkan tujuan serta kapabilitas yang diharapkan tercapai melalui

proses pendidikan, yang meliputi dimensi pengetahuan, pemahaman mendalam, maupun keterampilan praktis yang seharusnya dikuasai oleh mahasiswa sesuai menjalani satu periode studi tertentu (Adilah et al., 2022:85). Lebih lanjut, CPL menjalankan fungsi strategis sebagai landasan dalam menganalisis kebutuhan pembelajaran, mengidentifikasi sumber daya yang diperlukan, serta menjadi pijakan dalam pembentukan dan pengorganisasian mata kuliah. Di samping itu, manfaat CPL turut dirasakan oleh berbagai pemangku kepentingan, mulai dari mahasiswa yang mendapatkan kejelasan mengenai kompetensi yang akan diraih, dosen yang terbantu dalam mengefisienkan perencanaan pembelajaran dan evaluasi, hingga lembaga akreditasi yang menggunakannya sebagai instrumen penilaian ketercapaian persyaratan dan tujuan program studi (Wijaya et al., 2023:138).

Berdasarkan pengertian Capaian Pembelajaran Lulusan (CPL) di atas, dapat disimpulkan bahwa Capaian Pembelajaran Lulusan merupakan rangkuman kompetensi yang bersifat menyeluruh, meliputi dimensi pengetahuan, keterampilan, dan sikap, yang wajib diraih oleh mahasiswa sebagai capaian akhir dari proses pendidikan yang ditempuh. CPL tidak hanya berfungsi sebagai acuan pedagogis bagi dosen dan mahasiswa, tetapi juga berperan sebagai instrumen penjaminan mutu yang menghubungkan desain kurikulum dengan standar akreditasi secara sistematis dan terstruktur.

2. Sistem Pakar

Menurut Tajrin et al. (2025:2), sistem pakar merupakan suatu perangkat berbasis komputer yang bersifat andal dan interaktif dalam mendukung proses pengambilan keputusan, yang bekerja dengan mengaplikasikan heuristik dan fakta untuk menyelesaikan permasalahan, serta dianggap sebagai representasi dari puncak keahlian dan kecakapan manusia dalam bidang tertentu. Sistem pakar dirancang untuk menyelesaikan permasalahan yang paling sulit dalam bidang tertentu dengan memanfaatkan kemampuan komputasi secara optimal. Lebih lanjut, sistem ini memungkinkan komputer untuk berperan sebagai pengganti seorang pakar dalam memberikan rekomendasi maupun solusi atas permasalahan yang kompleks dan membutuhkan keahlian khusus (Wahyuni & Nirsal, 2024:3).

Sistem pakar tergolong sebagai bagian dari kecerdasan buatan (AI) yang didesain untuk menirukan proses pengambilan keputusan manusia dalam suatu domain yang spesifik. Dengan mengintegrasikan basis pengetahuan dan mekanisme inferensi, sistem ini mampu memberikan solusi atas permasalahan yang pada umumnya hanya dapat diselesaikan oleh seorang ahli di bidangnya. Perkembangan sistem pakar mencerminkan evolusi teknologi yang semakin canggih, di mana komponen utamanya berperan penting dalam menentukan efisiensi dan keandalan sistem, meskipun tantangan seperti akurasi dan kompleksitas perancangan tetap menjadi perhatian utama (Sumpala et al., 2025:15).

Berdasarkan pengertian sistem pakar di atas, dapat disimpulkan bahwa sistem pakar merupakan salah satu wujud penerapan kecerdasan buatan yang dirancang guna mengadaptasi pola berpikir dan mekanisme pengambilan keputusan seorang ahli manusia dalam suatu bidang yang spesifik. Sistem ini dioperasikan melalui perangkat komputer dengan memanfaatkan basis pengetahuan, data, dan teknik penalaran, sehingga mampu memecahkan masalah kompleks yang biasanya memerlukan keahlian khusus. Dengan adanya mekanisme inferensi dan aturan pengambilan keputusan, sistem pakar dapat memberikan solusi secara efisien tanpa menghadirkan seorang pakar secara langsung.

3. Metode *Rule-based*

Menurut Sumpala et al. (2025:48), metode berbasis aturan merupakan salah satu pendekatan utama dalam sistem pakar yang bekerja berdasarkan aturan-aturan yang telah ditetapkan sebelumnya, di mana sistem ini meniru cara kerja seorang pakar dalam mengambil keputusan melalui seperangkat aturan *IF-THEN* guna menganalisis suatu permasalahan dan menghasilkan kesimpulan. Sejalan dengan hal tersebut, Hafizhan & Fauzi (2025:51) menambahkan bahwa metode *rule-based reasoning* bekerja dengan mencocokkan gejala yang diinput oleh pengguna terhadap basis aturan yang telah ditentukan oleh pakar, sehingga dapat menghasilkan diagnosis secara otomatis dan cepat. Metode ini memanfaatkan aturan-aturan logika sebagai dasar pengambilan keputusan,

di mana setiap gejala yang dimasukkan akan diproses dan dibandingkan dengan pengetahuan yang tersimpan dalam sistem.

Metode *rule-based* adalah pendekatan yang menggunakan seperangkat aturan berbentuk *if-then* untuk melakukan penalaran dalam sistem pakar. Aturan-aturan tersebut merupakan representasi pengetahuan yang diperoleh dari pakar dan digunakan untuk menghubungkan kondisi tertentu dengan kesimpulan yang dihasilkan. Dengan mekanisme ini, sistem dapat mengevaluasi fakta yang diberikan dan menentukan solusi atau keputusan secara sistematis sesuai dengan aturan yang berlaku (Abubakar et al., 2021:21). Selain itu, metode ini juga termasuk dalam jenis sistem pakar yang mengandalkan aturan logis untuk menghasilkan keputusan, sehingga proses penalarannya bersifat transparan dan mudah dipahami (Mustafa et al., 2023:2).

Berdasarkan pengertian metode *rule-based* di atas, dapat disimpulkan bahwa metode *rule-based* merupakan metode dalam sistem pakar yang menggunakan kumpulan aturan logika berbentuk *if-then* untuk mengolah data atau gejala menjadi suatu keputusan. Metode ini meniru cara berpikir seorang pakar dengan memanfaatkan basis pengetahuan serta mekanisme inferensi untuk menghasilkan solusi secara otomatis, sistematis, dan menyerupai cara pengambilan keputusan manusia.

4. Metode Pembobotan

Menurut Mahendra (2025:47), *Rank Order Centroid* (ROC) merupakan salah satu metode pembobotan dalam ranah pengambilan

keputusan yang berfungsi untuk menetapkan nilai bobot pada setiap kriteria yang dipertimbangkan, dengan dasar penetapan yang mengacu pada urutan prioritas maupun tingkat kepentingan masing-masing kriteria tersebut. Metode ini pertama kali digagas oleh F. Hutton Barron pada tahun 1992 dan kerap diimplementasikan dalam analisis multiatributif guna memilih alternatif terbaik di antara berbagai pilihan yang melibatkan sejumlah kriteria dengan bobot kepentingan yang tidak seragam.

Pembobotan pada dasarnya merupakan suatu pendekatan yang diterapkan ketika masing-masing karakteristik atau parameter memiliki peran dan kontribusi yang berbeda-beda dalam suatu proses pengambilan keputusan, di mana kriteria-kriteria atau variabel-variabel yang relevan dijadikan sebagai acuan perbandingan untuk menghasilkan keputusan yang tepat (Cahayati & Muzdalifah, 2024:24). Lebih lanjut, metode ROC secara khusus difungsikan untuk menghasilkan nilai bobot pada tiap-tiap kriteria yang terlibat dalam proses evaluasi, sehingga setiap kriteria mendapatkan representasi bobot yang proporsional sesuai tingkat kepentingannya (Purba et al., 2023:213). Dalam implementasinya, proses ini mencakup kegiatan pemberian bobot terhadap setiap kriteria, yang selanjutnya digunakan sebagai landasan perhitungan skor bagi setiap alternatif berdasarkan performa atau kondisi aktualnya (Yudhistira & Wahyudi, 2024:83).

Berdasarkan pengertian metode pembobotan di atas, dapat disimpulkan bahwa metode pembobotan merupakan suatu prosedur

sistematis yang dipergunakan untuk menentukan derajat kepentingan relatif dari setiap kriteria dalam proses pengambilan keputusan, dengan tujuan akhir menghasilkan penilaian yang lebih objektif, terukur, dan dapat dipertanggungjawabkan. Melalui proses pemberian bobot yang proporsional pada setiap kriteria, sistem yang mengintegrasikan metode ini mampu mengolah berbagai parameter secara terstruktur sehingga menghasilkan keluaran berupa rekomendasi atau keputusan yang dapat dipertanggungjawabkan secara ilmiah dan akurat.

5. *Website*

Menurut Dermawan et al. (2021:1), *website* merupakan salah satu medium penyampaian informasi yang berciri khas modern, di mana akses terhadapnya dilakukan melalui jaringan internet dengan memanfaatkan perangkat lunak penelusur atau *browser*. *Website* memiliki sifat unik karena setiap halamannya dapat terhubung satu sama lain melalui *hyperlink text*, dan seluruh informasi yang terdapat di dalamnya disimpan dalam *server*. Seiring dengan perjalanan waktu, fungsi *website* tidak lagi terbatas pada penyebaran informasi semata, melainkan telah berkembang menjadi medium serba guna yang mendukung beragam aktivitas seperti streaming musik, pengolahan foto, bahkan pengeditan video. *Website* pertama kali diperkenalkan oleh seorang ilmuwan asal Inggris bernama Tim Berners-Lee pada tahun 1989, kala itu ia masih bertugas di lembaga riset CERN, dengan misi awal memungkinkan pertukaran informasi di antara para peneliti dan institusi ilmiah di berbagai penjuru dunia.

World Wide Web, yang kerap disingkat menjadi *Web*, adalah suatu sistem pengaksesan informasi berbasis internet yang menggunakan protokol *HTTP* sebagai sarana transmisi data antar perangkat. *Web* mengandalkan perangkat penelusur seperti Internet Explorer maupun Firefox untuk membuka dokumen-dokumen *web* yang dikenal sebagai halaman *web*, di mana setiap halaman tersebut terkoneksi satu sama lain melalui *hyperlink* dan dapat memuat konten dalam berbagai bentuk, mulai dari grafis, audio, teks, hingga video (Sastradipraja & Munawar, 2022:3). Sejalan dengan hal tersebut, *website* juga dapat dipahami sebagai satu atau kumpulan halaman *web* yang saling berkaitan dan dapat dijangkau dari manapun di seluruh dunia selama terhubung dengan jaringan internet, di mana setiap halaman memiliki identitas alamat yang unik berupa URL (*Uniform Resource Locator*), serta mampu menyajikan beragam jenis informasi dan fitur interaktif seperti formulir kontak, kolom komentar, maupun layanan obrolan daring (Sukri et al., 2023:1).

Berdasarkan pengertian *website* di atas, dapat disimpulkan bahwa *website* merupakan suatu sistem yang berjalan di atas infrastruktur internet, tersusun dari satu atau beberapa halaman web yang saling terkoneksi melalui tautan *hyperlink*, dan dapat diakses secara bebas oleh siapa saja di seluruh penjuru dunia menggunakan perangkat *browser*. *Website* berfungsi sebagai wahana penyimpanan sekaligus penyampaian informasi dalam aneka bentuk, mulai dari teks, citra visual, konten audio, hingga video. Seiring dengan perkembangannya yang pesat, *website* tidak

lagi sekadar berperan sebagai sumber informasi, melainkan juga telah bertransformasi menjadi platform interaktif yang mengakomodasi berbagai kebutuhan pengguna dalam ekosistem digital.

6. *Framework*

Menurut Kurniawan et al. (2023:22), *framework* merupakan paduan antara bahasa pemrograman dan kerangka kerja yang dimanfaatkan untuk membangun aplikasi *web* secara lebih efektif dan efisien, di mana kerangka tersebut menyediakan struktur dasar, seperangkat kaidah, serta komponen-komponen yang membantu para pengembang dalam menata dan mempercepat alur pengembangan aplikasi berbasis *web*. Senada dengan hal tersebut, Abdulloh (2022:7) menjelaskan bahwa *framework* dapat dimaknai sebagai kumpulan kode program yang dilengkapi dengan aturan-aturan baku yang wajib dipatuhi oleh pengembang, mencakup ketentuan mengenai susunan folder, pengaturan berkas, maupun struktur penulisan kode, sehingga berbagai tugas pemrograman yang kompleks dapat disederhanakan dan diselesaikan dengan lebih mudah.

Framework pengembangan *web* menyediakan kerangka kerja standar untuk membangun dan mengelola aplikasi *web*, sehingga dapat mengurangi jumlah pekerjaan berulang yang harus dilakukan pengembang sekaligus membantu terciptanya aplikasi yang lebih terorganisir dan aman (Purwidianoro, 2024:33). Lebih jauh, Muthohir (2025:5) menegaskan bahwa *framework* merupakan infrastruktur perangkat lunak yang

menyediakan fondasi dalam pengembangan aplikasi, meliputi perpustakaan kode, perkakas pendukung, serta praktik-praktik terbaik yang telah terbukti keandalannya, sehingga para pengembang dapat menciptakan aplikasi dalam waktu yang lebih singkat, dengan efisiensi yang lebih tinggi, dan menghasilkan kualitas yang lebih baik..

Berdasarkan pengertian *framework* di atas, dapat disimpulkan bahwa *framework* adalah suatu kerangka kerja berbasis perangkat lunak yang menyediakan struktur, kaidah, dan komponen yang siap digunakan, dengan tujuan menyederhanakan sekaligus mengakselerasi proses pengembangan aplikasi *web* secara signifikan. Keberadaan *framework* memungkinkan pengembang untuk tidak memulai segalanya dari titik nol, sehingga proses kerja menjadi lebih efisien dan terstruktur, serta menghasilkan aplikasi dengan kualitas yang lebih tinggi, tingkat keamanan yang lebih baik, dan kemudahan dalam proses pemeliharaan jangka panjang.

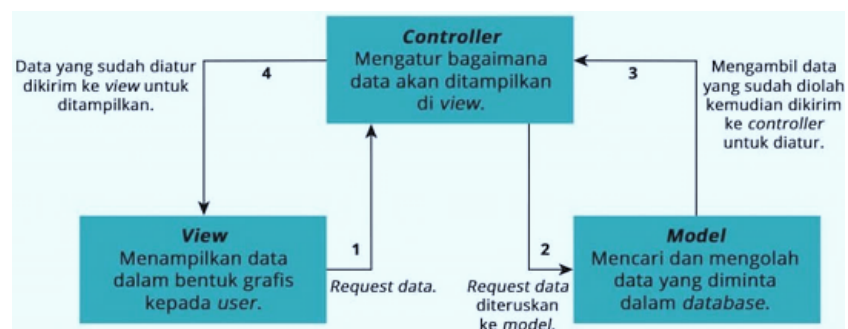
7. *Laravel*

Menurut Abdulloh (2022:10), *Laravel* merupakan salah satu *framework PHP* yang bersifat bebas biaya, lahir dari tangan seorang pengembang berkebangsaan Amerika bernama Taylor Otwell pada tahun 2011. Pasca dirilis secara publik, *Laravel* secara bertahap berhasil menarik perhatian para pengembang di berbagai belahan dunia dan berkembang menjadi salah satu *framework* yang paling diminati, bahkan melampaui popularitas berbagai *framework* yang telah hadir lebih dahulu. Menurut

Setiawansyah et al. (2023:18), *Laravel* adalah *framework PHP* yang bersifat *open-source* dan dilengkapi dengan berbagai modul fundamental yang dirancang untuk memaksimalkan performa *PHP*, sehingga proses pengembangan *web* dapat berlangsung lebih cepat, lebih aman, dan lebih sederhana, *framework* ini pertama kali diluncurkan pada 9 Juni 2011 sebagai alternatif dari *CodeIgniter*.

Menurut Sabaruddin et al. (2024:6), *Laravel* adalah *framework web* berbasis *PHP* yang bersifat gratis dan *open-source*, dirancang khusus untuk pengembangan aplikasi *web* dengan mengadopsi pola arsitektur MVC (*Model View Controller*), di mana komponen *Model* bertanggung jawab atas pengelolaan basis data, *View* bertugas menampilkan antarmuka kepada pengguna, dan *Controller* berfungsi sebagai jembatan penghubung antara *Model* dan *View* sekaligus bertindak sebagai pengendali alur sistem secara keseluruhan. Berkat kemudahan dalam penggunaannya, kelengkapan dokumentasi, serta ekosistem *library* yang kaya, *Laravel* kini telah menjelma menjadi salah satu *framework PHP* yang paling banyak diadopsi oleh para pengembang di seluruh dunia. Hal ini sejalan dengan hasil penelitian Sinlae et al. (2024:128) yang menyatakan bahwa pemanfaatan fitur-fitur yang tersedia pada *Laravel* terbukti mampu meningkatkan produktivitas pengembang, mempersingkat waktu pengembangan aplikasi, serta menghasilkan aplikasi dengan performa dan skalabilitas yang tinggi, sehingga *Laravel* dinilai layak dijadikan standar dalam pengembangan aplikasi *web* yang efisien dan berkelanjutan.

Berdasarkan pengertian *Laravel* di atas, dapat disimpulkan bahwa *Laravel* merupakan *framework PHP* yang bersifat gratis dan *open-source*, dibangun oleh Taylor Otwell sejak tahun 2011. *Laravel* didesain untuk menyederhanakan proses pengembangan aplikasi berbasis *web* dengan menerapkan pola arsitektur MVC dan didukung oleh ekosistem pengembangan yang lengkap, sehingga menempatkannya sebagai salah satu *framework PHP* paling populer yang dipercaya oleh jutaan pengembang di seluruh penjuru dunia.



Gambar 2.1 Arsitektur MVC *Framwork Laravel*

Sumber : Setiawansyah et al. (2023:11)

8. Basis Data (*Database*)

Menurut Royadi (2023:7), istilah basis data tersusun dari dua kata, yakni "basis" dan "data". Kata "basis" dapat dimaknai sebagai pusat, wadah penyimpanan, atau lokasi penghimpunan data, sementara "data" merupakan representasi fakta dari realita yang merepresentasikan suatu entitas, seperti individu (contohnya pegawai, peserta didik, pelanggan, atau konsumen), produk, kejadian, konsep, kondisi, serta berbagai unsur lainnya. Data tersebut dapat disajikan dalam berbagai bentuk, mulai dari

angka, huruf, simbol, teks, gambar, suara, hingga kombinasi dari elemen-elemen tersebut.

Basis data berfungsi sebagai tempat penyimpanan yang dapat mengelola serta menyusun data ke dalam format yang terstruktur, sehingga memungkinkan akses terhadap informasi secara cepat dan efisien (Rafianto & Voutama, 2025:896). Senada dengan hal tersebut, Gunawan et al. (2023:2) mendefinisikan basis data sebagai kumpulan data yang tersusun secara rapi dan terstruktur dalam lingkungan sistem komputer. Basis data terbentuk dari sejumlah tabel yang saling berinterelasi melalui hubungan-hubungan tertentu, serta difungsikan sebagai sarana penyimpanan, pengelolaan, dan pengaksesan data secara efisien. Lebih lanjut, Rao et al. (2022:8) memperjelas bahwa basis data merupakan himpunan data yang disimpan secara elektronik dan dapat diakses kapan saja melalui perangkat *Database Management System* (DBMS), yaitu sistem yang memfasilitasi proses penyimpanan, pengambilan, manipulasi, dan pengelolaan data sesuai dengan kebutuhan dan kepentingan pengguna.

Berdasarkan pengertian basis data di atas, dapat disimpulkan bahwa basis data merupakan himpunan data yang disusun secara terstruktur dan terorganisir di dalam suatu sistem komputer, yang terdiri atas tabel-tabel yang saling terhubung dalam jaringan relasi. Basis data dirancang untuk mempermudah dan mengoptimalkan proses penyimpanan, pengelolaan, serta pengambilan informasi secara cepat dan

efisien, sehingga menjadikannya komponen yang krusial dalam pembangunan sistem informasi maupun sistem pakar.

9. *MYSQL*

MySQL menempati posisi sebagai salah satu sistem manajemen basis data yang paling luas penggunaannya dalam lanskap teknologi informasi global. Menurut Widiyanto et al. (2026:2-3), *MySQL* merupakan salah satu *Relational Database Management System* (RDBMS) yang mendominasi ekosistem teknologi informasi secara global, di mana sistem ini mengimplementasikan model data relasional dengan menyimpan informasi ke dalam tabel-tabel yang saling berelasi melalui mekanisme kunci primer dan kunci asing, serta menggunakan *Structured Query Language* (SQL) sebagai bahasa standar yang digunakan untuk mendefinisikan, memanipulasi, dan mengendalikan data. Senada dengan hal tersebut, Silalahi (2022:1) menjelaskan bahwa *MySQL* merupakan sistem manajemen basis data yang dikembangkan agar berjalan secara cepat dan efisien, khususnya untuk mendukung kebutuhan situs *web* dinamis yang memerlukan pengelolaan data secara berkelanjutan, serta telah berkembang menjadi teknologi yang matang dengan lebih dari 10 juta instalasi di seluruh dunia.

Dari sisi asal-usul dan karakteristiknya, *MySQL* memiliki latar belakang pengembangan yang cukup panjang. Dalimunthe (2022:4) memaparkan bahwa *MySQL* dikembangkan oleh *MySQL* dikembangkan oleh *MySQL AB* sekitar tahun 1994 di Swedia, dengan tujuan awal untuk

menopang pengembangan aplikasi berbasis *web*, di mana *MySQL* merupakan perangkat lunak manajemen basis data yang memiliki karakteristik *multithread* dan *multi-user*, didistribusikan secara cuma-cuma di bawah naungan lisensi GPL (*GNU General Public License*) sehingga dapat digunakan secara bebas dengan ketentuan tidak diperkenankan untuk dijadikan produk turunan yang bersifat komersial.. Selain itu, dari aspek performa, Kurniadi et al. (2025:158) menegaskan bahwa *MySQL* menawarkan kemampuan pemrosesan yang memadai dengan konsumsi memori yang rendah, sehingga sangat ideal untuk diimplementasikan pada aplikasi yang beroperasi dengan sumber daya terbatas.

Berdasarkan pengertian *MySQL* di atas, dapat disimpulkan bahwa *MySQL* adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan *SQL* sebagai bahasa baku dalam pengelolaan data, bersifat *open source*, ringan, dan hemat sumber daya komputasi. *MySQL* telah terbukti menjadi pilihan utama dalam pengembangan sistem informasi berbasis *web* maupun berbagai aplikasi digital lainnya, berkat kemampuannya dalam mengelola data secara terstruktur, andal, dan konsisten di berbagai skala kebutuhan.

10. *Flowchart*

Menurut Khairunisa et al. (2023:28), *flowchart* merupakan diagram berbentuk visual yang berfungsi untuk melukiskan rangkaian langkah atau proses yang berlangsung dalam suatu sistem maupun

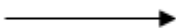
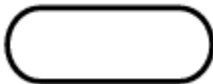
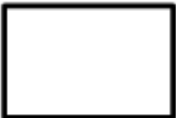
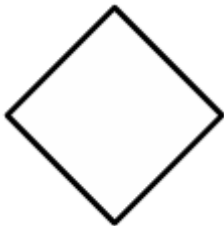
program, di mana alur kerja beserta pergerakan informasi dari satu tahap menuju tahap berikutnya direpresentasikan melalui simbol-simbol grafis yang saling dihubungkan oleh tanda panah penunjuk arah. Senada dengan hal tersebut, Khairunnisa et al. (2023:87) menjelaskan bahwa *flowchart* merupakan pendekatan diagramatik dalam merepresentasikan algoritma guna menghasilkan solusi atas suatu permasalahan, di mana setiap simbol dengan bentuk berbeda mewakili jenis proses yang berbeda pula, sementara panah-panah penghubung menunjukkan jalur yang harus ditempuh dalam menelusuri alur tersebut.

Lebih lanjut, *flowchart* tidak sekadar berfungsi sebagai alat perancangan teknis, melainkan juga memiliki peran komunikatif yang signifikan. Burhanuddin & Sukirman (2024:480) menegaskan bahwa diagram alir berperan sebagai alat bantu komunikasi antar-*programmer* dalam sebuah tim proyek, sehingga rangkaian logika yang panjang dan rumit dapat disajikan secara lebih mudah dipahami. Pernyataan ini diperkuat oleh Zimmermann et al. (2024:2) yang menyatakan bahwa *flowchart* dirancang menggunakan simbol-simbol, teks baku, dan/atau elemen grafis yang saling terhubung melalui panah yang merepresentasikan arah informasi maupun proses.

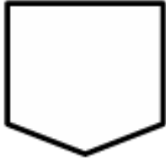
Berdasarkan pengertian *flowchart* di atas, dapat disimpulkan bahwa *flowchart* adalah representasi visual berupa diagram yang menggunakan simbol-simbol grafis dan panah untuk menggambarkan urutan langkah, aliran informasi, serta logika suatu proses atau algoritma

secara sistematis. Selain berfungsi sebagai alat bantu perancangan *program*, *flowchart* juga berperan penting dalam memfasilitasi komunikasi dan pemahaman bersama terhadap proses yang kompleks, baik dalam konteks pemrograman maupun penyelesaian masalah secara umum. Simbol-simbol *flowchart* disajikan pada Tabel 2.1 berikut.

Tabel 2.1 Simbol - Simbol *Flowchart*

Simbol	Nama	Fungsi
	Aliran Data / <i>Flowline</i>	Simbol berbentuk anak panah yang digunakan untuk menunjukkan arah perpindahan data atau informasi dari satu tahapan ke tahapan berikutnya dalam alur proses.
	Terminal / <i>Terminator</i>	Simbol berbentuk oval atau persegi panjang bersudut membulat yang berfungsi sebagai penanda titik awal maupun titik akhir dari suatu alur proses atau program.
	Proses / <i>Process</i>	Simbol berbentuk persegi panjang yang digunakan untuk merepresentasikan suatu tindakan atau serangkaian instruksi yang dijalankan dalam proses atau program.
	Keputusan / <i>Decision</i>	Simbol berbentuk belah ketupat yang digunakan untuk menandai percabangan dalam alur proses berdasarkan suatu kondisi tertentu. Setiap panah yang keluar dari simbol ini mewakili kemungkinan hasil dari keputusan yang diambil.

	Konektor	Simbol berbentuk lingkaran kecil yang berfungsi sebagai penghubung antara bagian-bagian alur proses yang secara fisik terpisah namun memiliki keterkaitan secara logis dalam satu rangkaian proses.
	<i>Input/Output</i>	Simbol berbentuk jajaran genjang yang digunakan untuk merepresentasikan proses pemasukan data ke dalam sistem (<i>input</i>) maupun keluaran hasil yang dihasilkan dari proses (<i>output</i>).
	<i>Predefined Process</i>	Simbol berbentuk persegi panjang dengan garis vertikal di kedua sisinya, digunakan untuk mengacu pada suatu sub-proses atau rangkaian langkah yang telah didefinisikan dan dirancang sebelumnya secara terpisah.
	Dokumen	Simbol berbentuk persegi panjang dengan bagian bawah bergelombang yang digunakan untuk merepresentasikan dokumen atau berkas informasi yang terlibat dan diproses dalam suatu alur kerja.
	Penggabungan (<i>Merge</i>) dan Pemisahan (<i>Split</i>)	Simbol yang digunakan untuk merepresentasikan proses penggabungan beberapa aliran proses menjadi satu (<i>merge</i>) atau pemecahan satu aliran menjadi beberapa jalur proses yang berbeda (<i>split</i>).

	<i>Off-page Connector</i>	Simbol berbentuk segi lima menyerupai rumah yang berisi huruf atau angka pengenal, digunakan untuk menghubungkan langkah-langkah atau bagian-bagian dari <i>flowchart</i> yang berlanjut ke halaman lain yang berbeda.
---	---------------------------	--

Sumber: Khairunisa et al. (2023: 33-36)

11. *Unified Modeling Language (UML)*

Menurut Gunawan et al. (2023:51) *Unified Modeling Language* (UML) adalah sebuah bahasa pemodelan berbasis grafis yang berperan dalam proses perancangan dan dokumentasi sistem perangkat lunak. Melalui bahasa ini, para pengembang dapat memvisualisasikan serta memahami struktur dan rancangan sistem yang hendak dibangun. Senada dengan hal tersebut, Sumirat et al. (2023:73) UML adalah bahasa berbasis grafis yang tidak hanya dimanfaatkan untuk memvisualisasikan dan menspesifikasikan suatu sistem, tetapi juga untuk membangun dan mendokumentasikan sistem perangkat lunak yang berorientasi objek. Lebih dari itu, UML memiliki kemampuan untuk diintegrasikan dengan berbagai bahasa pemrograman seperti Java, C++, dan Visual Basic, serta basis data berorientasi objek, sehingga cakupannya tidak terbatas hanya sebagai alat pemrograman visual semata.

Berdasarkan pengertian *Unified Modeling Language* (UML) di atas, dapat disimpulkan bahwa *Unified Modeling Language* (UML) adalah bahasa pemodelan grafis yang mencakup fungsi perancangan,

visualisasi, spesifikasi, pembangunan, sekaligus dokumentasi sistem perangkat lunak berbasis objek. Keberadaan UML sangat membantu pengembang dalam memahami dan menggambarkan sistem secara lebih terstruktur dan komprehensif. Selain itu, kemampuannya untuk berintegrasi dengan berbagai bahasa pemrograman maupun basis data berorientasi objek menjadikan UML sebagai salah satu instrumen penting dalam pengembangan perangkat lunak. Adapun jenis-jenis UML yang digunakan oleh penulis dalam penelitian ini meliputi:

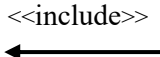
a. *Use Case Diagram*

Menurut Gunawan et al. (2023:55) *Use Case Diagram* adalah salah satu jenis diagram dalam UML yang digunakan untuk memodelkan bagaimana pengguna berinteraksi dengan sistem, dengan tujuan menampilkan kebutuhan fungsional utama yang perlu disediakan oleh sistem berdasarkan sudut pandang pengguna. Sementara itu, Sumirat et al. (2023:80-81) menambahkan bahwa *Use Case Diagram* merupakan representasi visual yang menggambarkan hubungan antara aktor eksternal dan sistem guna merepresentasikan fungsi serta alur proses bisnis. Diagram ini juga memiliki peran penting sebagai alat verifikasi kebutuhan, identifikasi aktor, spesifikasi antarmuka, serta sarana komunikasi resmi antara analis sistem dan para pemangku kepentingan.

Berdasarkan pengertian *Use Case Diagram* di atas, dapat disimpulkan bahwa *Use Case Diagram* merupakan salah satu diagram

UML yang berfungsi untuk memodelkan interaksi antara pengguna dan sistem secara visual. Diagram ini tidak hanya menggambarkan kebutuhan fungsional utama yang harus dipenuhi oleh sistem, tetapi juga merepresentasikan alur proses bisnis yang berjalan. Di samping itu, *Use Case Diagram* turut berperan dalam proses verifikasi kebutuhan, identifikasi aktor yang terlibat, spesifikasi antarmuka sistem, serta menjadi media komunikasi yang efektif antar entitas yang berkepentingan dalam pengembangan sistem. Simbol-simbol *Use Case Diagram* disajikan pada Tabel 2.2 berikut.

Tabel 2.2 Simbol *Use Case Diagram*

Simbol	Elemen	Keterangan
	Aktor	Merepresentasikan peran yang dimainkan oleh seseorang, sistem eksternal, atau perangkat tertentu ketika berinteraksi dan berkomunikasi dengan <i>use case</i> yang ada dalam sistem.
	<i>Use Case</i>	Menggambarkan abstraksi dari suatu fungsi atau layanan yang disediakan oleh sistem sebagai hasil dari interaksi yang terjadi antara sistem dengan aktor.
	<i>Association</i>	Merepresentasikan hubungan atau jalur komunikasi yang terbentuk antara aktor dengan <i>use case</i> , menunjukkan bahwa aktor tersebut terlibat dalam pelaksanaan <i>use case</i> yang bersangkutan.
	<i>Generalisasi</i>	Menggambarkan hubungan pewarisan antara aktor atau <i>use case</i> , di mana elemen yang lebih khusus (anak) mewarisi karakteristik dan kemampuan dari elemen yang lebih umum (induk).
	<i>Include</i>	Menandakan bahwa suatu <i>use case</i> secara wajib menyertakan dan menjalankan fungsionalitas dari <i>use case</i> lain sebagai bagian yang tidak terpisahkan dari prosesnya.

	<i>Extend</i>	Menandakan bahwa suatu <i>use case</i> dapat memperluas atau menambahkan fungsionalitas tambahan pada <i>use case</i> lain, namun hanya akan dijalankan apabila kondisi atau syarat tertentu terpenuhi.
---	---------------	---

Sumber: Gunawan et al. (2023:56)

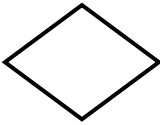
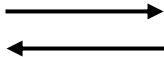
b. *Activity* Diagram

Menurut Hariyanto dalam Khairunisa et al. (2023:175-176) *activity* diagram merupakan salah satu jenis diagram yang dimanfaatkan dalam pemodelan sistem atau proses bisnis guna mendeskripsikan rangkaian tindakan yang berlangsung secara terstruktur. Sejalan dengan hal tersebut, Gunawan et al. (2023:64) mengemukakan bahwa *activity* diagram merupakan bagian dari diagram UML yang menggambarkan alur kerja atau serangkaian aktivitas dalam suatu sistem maupun proses bisnis, dengan tujuan untuk memperlihatkan jalannya suatu proses sekaligus mengidentifikasi bagian-bagian yang berpotensi untuk dikembangkan demi tercapainya efisiensi yang lebih optimal.

Berdasarkan pengertian *Activity* Diagram di atas, dapat disimpulkan bahwa *activity* diagram adalah diagram berbasis UML yang difungsikan untuk memodelkan rangkaian aktivitas atau alur kerja dalam sebuah sistem maupun proses bisnis. Diagram ini berperan penting dalam memperlihatkan jalannya suatu proses secara visual, menguraikan tahapan-tahapan yang berlangsung secara berurutan, serta membantu dalam menemukan bagian yang perlu

diperbaiki agar sistem dapat beroperasi dengan lebih efisien. Simbol-simbol *Activity Diagram* disajikan pada Tabel 2.3 berikut.

Tabel 2.3 Simbol *Activity Diagram*

Simbol	Elemen	Keterangan
	<i>Activity</i>	Menggambarkan sebuah aktivitas atau pekerjaan yang dilakukan dalam alur proses, memperlihatkan bagaimana setiap elemen antarmuka saling berinteraksi dan berkoordinasi satu dengan yang lainnya.
	<i>Action</i>	Menrepresentasikan kondisi atau keadaan sistem pada saat tertentu yang mencerminkan pelaksanaan sebuah aksi atau tindakan spesifik yang tidak dapat dibagi lebih lanjut.
	<i>Initial Node</i>	Berfungsi sebagai penanda titik permulaan dari suatu diagram aktivitas, menunjukkan dari mana alur proses mulai dijalankan.
	<i>Activity Final Node</i>	Berfungsi sebagai penanda titik berakhirnya seluruh alur dalam diagram aktivitas, menandakan bahwa keseluruhan proses telah selesai dijalankan.
	<i>Decision</i>	Digunakan untuk merepresentasikan suatu titik percabangan dalam alur proses, di mana terdapat dua atau lebih kemungkinan jalur yang akan ditempuh bergantung pada kondisi atau kriteria tertentu yang dievaluasi.
	<i>Line Connector</i>	Berupa garis penghubung yang digunakan untuk mengaitkan satu simbol dengan simbol lainnya, menunjukkan arah aliran atau urutan dari suatu aktivitas ke aktivitas berikutnya.

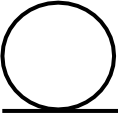
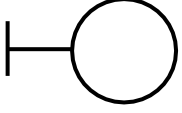
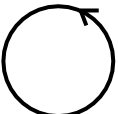
Sumber: Gunawan et al. (2023:65)

c. *Sequence* Diagram

Menurut Gunawan et al. (2023:60), *sequence* diagram merupakan salah satu jenis diagram UML yang menggambarkan interaksi antar objek dalam suatu sistem secara terurut guna memperlihatkan alur logika dari sebuah fitur atau fungsi tertentu. Diagram ini menyajikan urutan pesan maupun pemanggilan antar objek melalui tanda panah yang disusun berdasarkan dimensi waktu, sehingga dinamika interaksi antar objek dalam sistem dapat terlihat dengan jelas. Senada dengan hal tersebut, Sumirat et al. (2023:86) menjelaskan bahwa *sequence* diagram adalah diagram yang merepresentasikan interaksi secara rinci antar objek dalam sistem, mencakup pesan atau instruksi yang dikirimkan beserta waktu pelaksanaannya, di mana objek-objek yang terlibat pada umumnya disusun dari kiri ke kanan mengikuti jalannya proses.

Berdasarkan pengertian *Sequence* Diagram di atas, dapat disimpulkan bahwa *sequence* diagram merupakan pemodelan interaksi secara rinci antar objek dalam suatu sistem yang disusun secara berurutan mengacu pada aliran waktu. Diagram ini menampilkan rangkaian pesan atau pemanggilan antar objek melalui garis panah, sehingga alur logika suatu fitur atau fungsi dapat tergambarkan dengan jelas, sekaligus memperlihatkan waktu eksekusi setiap pesan yang terjadi di dalam sistem. Simbol-simbol *Sequence* Diagram disajikan pada Tabel 2.4 berikut.

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	Elemen	Keterangan
	Aktor	Merepresentasikan pengguna atau pihak eksternal yang memulai dan terlibat langsung dalam interaksi dengan sistem yang sedang dimodelkan.
	<i>Entity Class</i>	Merepresentasikan objek atau entitas dalam sistem yang menyimpan data dan memiliki keterlibatan dalam hubungan atau relasi yang terbentuk selama jalannya suatu skenario interaksi.
	<i>Boundary Class</i>	Merepresentasikan antarmuka yang berfungsi sebagai perantara dalam mengelola komunikasi dan pertukaran informasi antara aktor eksternal dengan lingkungan internal sistem.
	<i>Control Class</i>	Merepresentasikan objek pengendali yang berfungsi sebagai jembatan koordinasi antara kelas <i>boundary</i> yang berinteraksi dengan pengguna dan kelas <i>entity</i> yang mengelola data dalam sistem.
	<i>A Focus of Control & A Life Line</i>	<i>Lifeline</i> merepresentasikan keberadaan suatu objek sepanjang dimensi waktu dalam diagram, sedangkan <i>focus of control</i> menggambarkan periode aktif di mana objek tersebut sedang menjalankan atau mengendalikan suatu proses hingga selesai.
	<i>A Message</i>	Merepresentasikan komunikasi atau instruksi yang dikirimkan dari satu objek kepada objek lainnya dalam rangkaian interaksi, digambarkan melalui garis panah yang menunjukkan arah pengiriman pesan tersebut.

Sumber: Gunawan et al., (2023:60-61)

d. Class Diagram

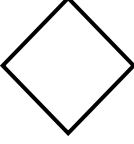
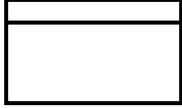
Menurut Sumirat et al. (2023:88-89), *class diagram* merupakan salah satu jenis diagram struktural dalam UML yang

secara eksplisit menggambarkan kelas, atribut, metode, serta relasi antar objek. Diagram ini memiliki sifat statis karena lebih menitikberatkan pada representasi hubungan antar kelas daripada alur interaksi yang terjadi di dalamnya. Sejalan dengan hal tersebut, Gunawan et al. (2023:57) mengemukakan bahwa *class* diagram adalah diagram dalam UML yang digunakan untuk merepresentasikan struktur kelas beserta keterkaitannya dalam suatu sistem, dengan tujuan memberikan gambaran menyeluruh mengenai objek-objek yang ada beserta relasi di antara objek-objek tersebut.

Berdasarkan pengertian *Class* Diagram di atas, dapat disimpulkan bahwa *class* diagram adalah diagram UML yang merepresentasikan struktur kelas dalam suatu sistem, meliputi atribut, metode, serta hubungan antar kelas atau objek. Diagram ini bersifat statis karena menekankan pada struktur dan keterkaitan antar kelas, bukan alur interaksi, sehingga memberikan gambaran menyeluruh mengenai komponen sistem dan bagaimana objek-objek tersebut saling berhubungan. Simbol-simbol *Class* Diagram disajikan pada Tabel 2.5 berikut.

Tabel 2.5 Simbol *Class* Diagram

Simbol	Elemen	Keterangan
	Garis lurus (<i>Generalization</i>)	Merepresentasikan hubungan pewarisan antara kelas turunan (<i>subclass</i>) dan kelas induk (<i>superclass</i>), di mana kelas turunan mengadopsi dan mewarisi perilaku serta struktur data yang dimiliki oleh kelas induknya.

	<i>Nary Association</i>	Merepresentasikan hubungan asosiasi yang melibatkan lebih dari dua kelas sekaligus dalam satu relasi, digunakan ketika sebuah asosiasi tidak dapat cukup digambarkan hanya dengan menghubungkan dua objek saja.
	<i>Class</i>	Merepresentasikan cetak biru atau <i>template</i> dari sekumpulan objek dalam sistem yang memiliki kesamaan dalam hal <i>atribut</i> (sifat) dan operasi (perilaku) yang dimilikinya.
	<i>Collaboration</i>	Menggambarkan kerja sama antar objek atau kelas melalui serangkaian aksi yang terkoordinasi dalam sistem, dengan tujuan menghasilkan suatu keluaran atau hasil yang dapat diukur dan diverifikasi.
	<i>Realization</i>	Merepresentasikan hubungan antara sebuah antarmuka (<i>interface</i>) dengan kelas yang mengimplementasikannya, menunjukkan bahwa kelas tersebut benar-benar mewujudkan dan menjalankan operasi-operasi yang telah ditetapkan oleh antarmuka.
	<i>Dependency</i>	Menggambarkan hubungan ketergantungan antar elemen, di mana perubahan yang terjadi pada elemen <i>independen</i> (yang tidak bergantung) berpotensi memberikan dampak atau pengaruh terhadap elemen lain yang bersifat <i>dependen</i> (bergantung) padanya.
	<i>Association</i>	Merepresentasikan hubungan struktural yang menghubungkan satu kelas dengan kelas lainnya, menandakan bahwa objek dari kelas yang satu memiliki keterkaitan atau dapat berkomunikasi dengan objek dari kelas yang lain.

Sumber: Gunawan et al. (2023:57-58)

B. Kajian Empiris

Penelitian ini membahas pengembangan sistem pakar untuk menganalisis Capaian Pembelajaran Lulusan (CPL) menggunakan metode pembobotan dan *rule-based*. Dalam pembuatan sistem ini diperlukan referensi dari sumber-sumber terdahulu yang relevan. Kajian empiris yang sesuai juga dibutuhkan untuk memberi landasan teori serta mendukung proses pengembangan sistem pakar yang sedang dikembangkan.

Pada penelitian yang dilakukan oleh Ishaq et al. (2023:187), yang berjudul Aplikasi Probe Untuk Penilaian Capaian Pembelajaran Mahasiswa Pada Kurikulum OBE, penelitian ini bertujuan mengembangkan aplikasi PROBE berbasis web untuk menyajikan penilaian CPL mahasiswa dalam kerangka kurikulum OBE menggunakan metode *waterfall*. Sistem ini dirancang untuk memudahkan pengelolaan kurikulum sekaligus menyajikan data capaian secara terukur melalui analisis kebutuhan fungsional dengan pendekatan terstruktur. Hasilnya, aplikasi mampu menampilkan CPL secara terukur per tahun ajaran maupun keseluruhan, serta mendukung pengambilan kebijakan terkait evaluasi dan revisi kurikulum.

Pada penelitian yang dilakukan oleh Adilah et al. (2022:84), yang berjudul Pengembangan Sistem Informasi Rencana Pembelajaran Semester dan Evaluasi Capaian Pembelajaran Lulusan Berbasis *Progressive Web App*, penelitian ini bertujuan mengatasi ketidakefisienan penggunaan spreadsheet dalam penyusunan RPS dan evaluasi CPL. Sistem dibangun menggunakan metode *prototype* melalui tahapan iteratif mulai dari pengumpulan kebutuhan

hingga umpan balik pengguna. Hasilnya, sistem memudahkan dosen menyusun RPS, membantu program studi memonitor ketercapaian CPL, serta memberikan kemudahan bagi mahasiswa untuk mengakses rencana pembelajaran dan hasil evaluasi secara terintegrasi.

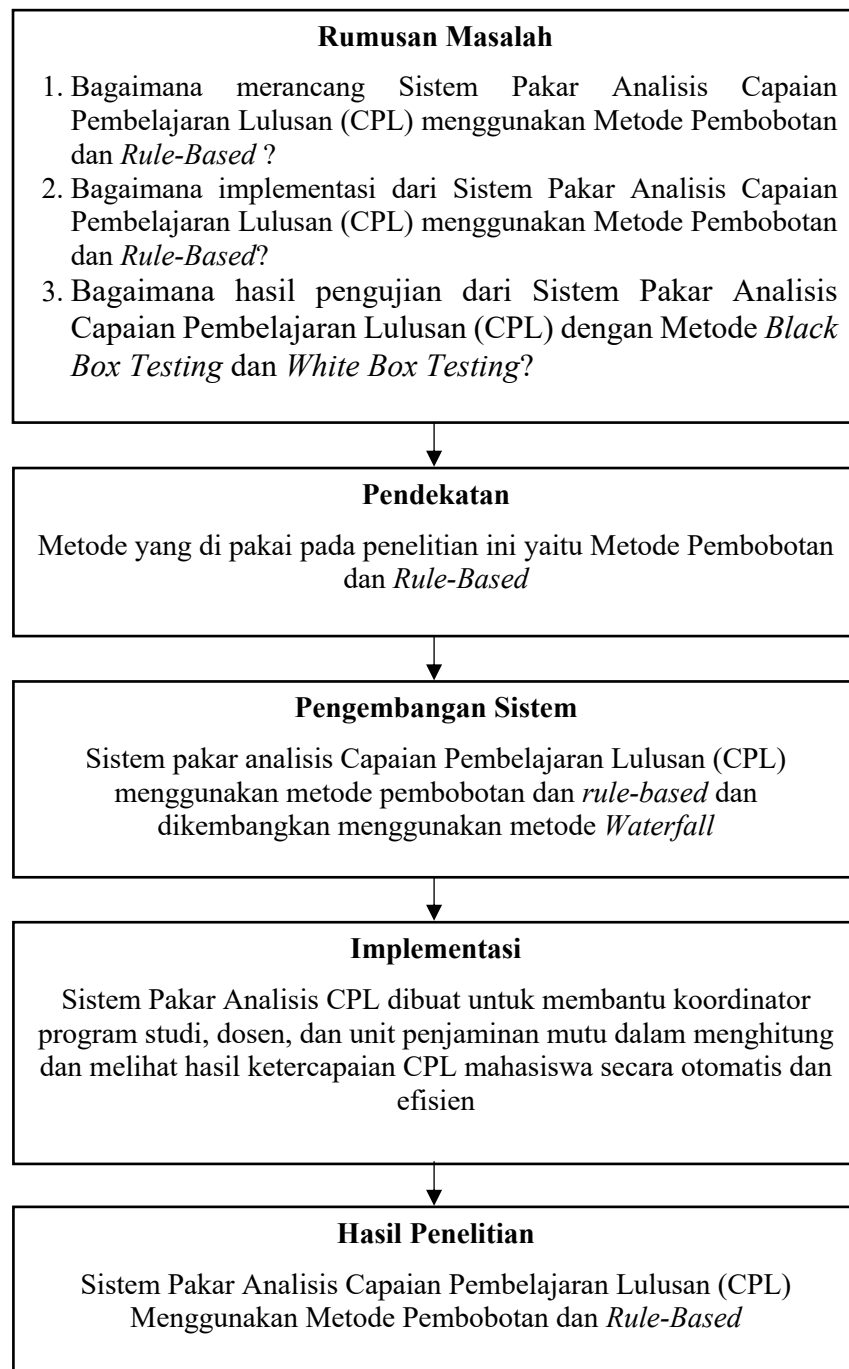
Pada penelitian yang dilakukan oleh Wijaya et al. (2023:137), yang berjudul *Analisa dan Perancangan Aplikasi Evaluasi Capaian Pembelajaran Lulusan*, penelitian ini bertujuan merancang aplikasi evaluasi CPL berbasis *web* menggunakan metode *waterfall* dengan pendekatan penilaian kumulatif berbasis CPMK pada program studi Teknik Informatika Universitas Satya Negara Indonesia. Pengujian dilakukan melalui *test case* dan *User Acceptance Testing* (UAT) untuk memastikan kesesuaian sistem dengan kebutuhan pengguna. Hasilnya, aplikasi mampu mengelola dan merekap nilai mahasiswa secara terkomputerisasi, serta memudahkan dosen dan mahasiswa dalam memantau perkembangan capaian pembelajaran.

Ketiga penelitian yang dilakukan oleh Ishaq et al., Adilah et al., serta Wijaya et al. telah menghasilkan sistem evaluasi CPL yang terkomputerisasi dan memberikan kontribusi nyata dalam pengelolaan pembelajaran berbasis OBE. Namun, ketiga sistem tersebut masih berfokus pada rekap dan monitoring nilai tanpa mengintegrasikan kemampuan analisis otomatis berbasis inferensi. Oleh karena itu, penelitian ini mengembangkan sistem pakar yang mampu menganalisis ketercapaian CPL secara otomatis menggunakan kombinasi metode pembobotan dan *rule-based*.

C. Kerangka Berpikir

Permasalahan utama dalam penelitian ini adalah proses analisis ketercapaian Capaian Pembelajaran Lulusan (CPL) di Program Studi Teknik Informatika UNIPMA yang masih dilakukan secara manual menggunakan spreadsheet tidak terintegrasi, sehingga rentan kesalahan input pada saat pemetaan nilai mata kuliah terhadap indikator CPL, membutuhkan waktu lama karena perhitungan bobot dan status ketercapaian dikerjakan ulang secara terpisah, dan hasilnya sering terlambat dijadikan dasar evaluasi kurikulum. Kondisi ini diperparah dengan belum adanya standar baku dalam pembobotan kontribusi mata kuliah terhadap CPL, serta belum tersedianya mekanisme yang mampu memberikan rekomendasi status ketercapaian CPL secara otomatis, baik pada level mahasiswa maupun program studi.

Oleh karena itu, dikembangkan Sistem Pakar Analisis CPL Menggunakan Metode Pembobotan dan *Rule-Based*, di mana metode pembobotan menentukan kontribusi setiap mata kuliah terhadap indikator CPL, sementara metode *rule-based* berupa aturan *IF-THEN* menyimpulkan status ketercapaian (tercapai, cukup tercapai, atau belum tercapai) berdasarkan hasil pembobotan tersebut. Sistem dirancang menggunakan UML, diimplementasikan dengan *framework* Laravel dan basis data MySQL, lalu diuji melalui *Black Box Testing* dan *White Box Testing* agar mampu menganalisis ketercapaian CPL secara otomatis, objektif, dan efisien guna mendukung pengambilan keputusan akademik yang lebih tepat. Kerangka berpikir penelitian ini ditunjukkan pada Gambar 2.2.



Gambar 2.2 Kerangka Berpikir