

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Pendukung Keputusan (SPK)

Menurut Prasetyo & Amin, (2022), Sistem pendukung keputusan merupakan sebuah sistem yang dirancang untuk membantu menghasilkan keputusan yang dapat menjadi solusi bagi pengguna. Dalam prosesnya, pengguna dapat terlebih dahulu menentukan faktor atau kriteria yang dianggap penting. Sistem ini telah banyak memberikan kontribusi bagi masyarakat, baik untuk kebutuhan jangka pendek maupun jangka panjang. SPK ini dapat diterapkan di berbagai bidang seperti Pendidikan, bisnis, hingga game, dan mendukung pengambilan keputusan baik secara individu maupun dalam skala besar melalui media berbasis komputer atau web.

Berdasarkan pendapat di atas, maka dapat disimpulkan bahwa Sistem Pendukung Keputusan (SPK) merupakan gabungan dari prosedur, komponen teknologi, dan sumber daya manusia yang berfungsi untuk mengolah data menjadi informasi yang bermakna. Informasi ini kemudian digunakan untuk mendukung proses pengambilan keputusan yang lebih tepat, serta meningkatkan efektivitas dan efisiensi dalam menyelesaikan suatu permasalahan atau kebutuhan tertentu.

2. Game *RPG (Role Playing Game)*

Game *RPG (Role Playing Game)* adalah salah satu genre permainan yang menawarkan pengalaman mendalam melalui alur cerita yang kompleks serta memungkinkan pemain untuk mengendalikan karakter dalam dunia imajinatif. Pemain berperan sebagai tokoh tertentu, menjalani misi, dan menghadapi berbagai tantangan yang berkaitan dengan perkembangan karakter. Game *RPG (Role Playing Game)* merupakan salah satu genre yang paling diminati dan terus mengalami perkembangan dalam dunia permainan digital. Genre ini ditandai dengan alur cerita yang kuat dan peran aktif pemain dalam mengendalikan karakter yang terlibat dalam dunia fiksi. *RPG* menyuguhkan pengalaman bermain yang mendalam, disertai dengan pengembangan karakter dan narasi yang kaya (Badruzzaman, 2024).

Wuthering Waves adalah game action *RPG (Role Playing Game)* yang dikembangkan oleh *Kuro Games* dan menampilkan dunia terbuka (*Open World*) Game ini dikembangkan dan dirilis oleh *Kuro Games* pada tanggal 22 Mei 2024 dengan sistem pertarungan *real-time* yang dinamis. Dalam game ini, pemain dapat memilih berbagai hero dengan elemen dan peran berbeda, salah satunya adalah hero bertipe *support* yang memiliki kemampuan untuk menyembuhkan, memberikan *buff*, atau melindungi rekan satu tim. Game ini mengalami lonjakan popularitas sejak perilisannya, dan menjadi booming di kalangan *Game RPG* karena kualitas grafisnya yang tinggi, mekanisme pertarungan

yang halus, serta sistem pengembangan hero yang kompleks. Pernyataan tersebut membuat para pemain tertarik untuk memperoleh karakter yang diinginkan. Hal yang menarik dalam hal ini adalah adanya sistem gacha yang digunakan untuk mendapatkan karakter atau item di dalam game (MAKKI, 2025).

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa *game RPG (Role Playing Game)* merupakan jenis permainan yang menekankan pada pemilihan karakter dan strategis permainan yang kompleks. Dalam game ini, pemain harus mempertimbangkan berbagai aspek seperti peran karakter, kemampuan khusus, serta kesesuaian dalam tim untuk mencapai hasil permainan yang optimal.

3. Metode *Analytical Hierarchy Process (AHP)*

AHP merupakan salah satu metode pengambilan keputusan multikriteria yang memungkinkan pengguna untuk menilai dan menentukan prioritas berbagai kriteria dan alternatif berdasarkan preferensi secara relatif. Dalam konteks skripsi ini, penerapan *AHP* digunakan untuk mendukung proses pemilihan hero support terbaik pada game *RPG Wuthering Waves*, sehingga keputusan yang diambil tidak hanya bersifat subjektif, melainkan melalui pendekatan yang lebih sistematis dan terukur. Dalam proses *AHP*, pengambil keputusan diminta membandingkan elemen secara berpasangan untuk menilai tingkat kepentingannya relatif satu sama lain. Hasil dari perbandingan ini kemudian dihitung untuk memperoleh bobot atau prioritas dari

masing-masing elemen. Bobot ini kemudian digunakan untuk menghitung skor total dari setiap alternatif. Metode *Analytic Hierarchy Process (AHP)* adalah suatu pendekatan dalam analisis pengambilan keputusan yang dikembangkan oleh *Thomas L. Saaty* pada tahun 1970. *AHP* telah dikenal luas sebagai metode yang efektif dalam menangani masalah pengambilan keputusan yang melibatkan banyak kriteria serta tingkat kompleksitas dan ketidakpastian yang tinggi. (Rozali et al., 2023).

a. Perhitungan AHP

Metode *Analytical Hierarchy Process (AHP)* adalah metode pengambilan keputusan yang digunakan untuk memecahkan masalah kompleks dengan menguraikannya ke dalam bentuk *hierarki*, yang terdiri dari tujuan, kriteria, sub-kriteria, dan alternatif. *AHP* memanfaatkan perbandingan berpasangan antar elemen untuk menentukan bobot prioritas berdasarkan Tingkat kepentingan relatif masing masing. Pendekatan ini bersifat sistematis, logis, dan kuantitatif karena menghasilkan nilai numerik yang merepresentasikan Tingkat kepentingan setiap elemen. Selain itu, *AHP* mampu mengintegrasikan pertimbangan kualitatif dan kuantitatif dengan dukungan dari penilaian para pengambil keputusan. Keunggulan *AHP* terletak pada kemampuannya dalam menangani ketidak konsistenan dalam proses penilaian melalui *rasio konsistensi (CR)*, sehingga dapat menjamin keakuratan dan validasi hasil pengambilan keputusan.

AHP juga memanfaatkan matriks perbandingan berpasangan, yang dihitung melalui Langkah-langkah berikut:

- 1) Membentuk matriks perbandingan berpasangan

Bandingkan kriteria atau alternatif dua per dua menggunakan skala 1-9 menurut tingkat kepentingan.

	K1	K2	K3
K1	1	0.50	3
K2	2	1	4
K3	0.33	0.25	1
Total	3.33	1.75	8

- 2) Menjumlahkan kolom matriks

$$\text{Jumlah kolom ke-}j = \sum_{i=1}^n a_{ij}$$

Dimana:

a_{ij} = Nilai Perbandingan antara kriteria ke-i dengan kriteria ke-j pada matriks AHP.

n = Jumlah Kriteria

$\sum a_{ij}$ = Penjumlahan total nilai pada kolom ke-j

Penjumlahan kolom ini dilakukan untuk setiap kolom dalam matriks perbandingan berpasangan sebagai Langkah awal sebelum melakukan normalisasi.

- 3) Melakukan normalisasi matriks

$$\tilde{a}_{ij} = \frac{a_{ij}}{\sum_{i=1}^n a_{ij}}$$

Dimana:

a_{ij} = Nilai hasil normalisasi antara kriteria ke-I dan ke-j

a_{ij} = Nilai asli perbandingan dari matriks AHP

$\sum a_{ij}$ = Jumlah total nilai pada kolom ke-j

Proses ini dilakukan dengan membagi setiap elemen pada kolom dengan total nilai kolom tersebut, untuk menghasilkan matriks yang ternormalisasi.

- 4) Menghitung bobot prioritas (*eigenvector*)

$$w_i = \frac{\sum_{j=1}^n \tilde{a}_{ij}}{n}$$

Dimana:

w_i = Bobot prioritas (*eigenvector*) untuk kriteria ke-i

a_{ij} = Nilai hasil normalisasi dari elemen pada baris ke-I dan kolom ke-j

n = Jumlah kriteria

Bobot dihitung dengan cara menjumlahkan seluruh nilai dalam baris ke-I pada matriks hasil normalisasi, lalu dibagi dengan jumlah kriteria. Bobot ini mencerminkan seberapa penting setiap kriteria secara relative dibandingkan yang lain.

- 5) Menghitung Konsistensi

- a) Hitung λ_{maks} (*eigen value*):

$$\lambda_{maks} = \sum (\text{Total kolom} \times \text{Bobot})$$

- b) Hitung *Konsistency Index (CI)*:

$$CI = \frac{\lambda_{\text{maks}} - n}{n - 1}$$

c) Hitung *Konsistency Ratio (CR)*:

$$CR = \frac{CI}{RI}$$

Keterangan:

RI = Random Index, tergantung jumlah kriteria.

Jika $CR < 0.1$, maka perbandingan dikatakan konsisten.

Penerapan metode Analytical Hierarchy Process (AHP) dilakukan melalui sejumlah langkah yang sistematis. Pertama, permasalahan keputusan dianalisis dan disusun ke dalam struktur hierarki yang mencakup tujuan utama, kriteria, dan alternatif. Setelah itu, dilakukan perbandingan berpasangan antara kriteria menggunakan matriks perbandingan berpasangan. Langkah selanjutnya adalah melakukan normalisasi dan menghitung nilai bobot dari setiap alternatif berdasarkan kriteria yang ditentukan. Terakhir, bobot kriteria dikalikan dengan bobot alternatif untuk mendapatkan skor akhir. Alternatif dengan skor tertinggi dianggap sebagai pilihan terbaik dalam pengambilan keputusan.

Dari pernyataan di atas, dapat disimpulkan bahwa metode AHP merupakan salah satu pendekatan dalam pengambilan keputusan multikriteria yang mengubah penilaian kualitatif menjadi nilai

kuantitatif melalui perbandingan berpasangan. Metode ini memfasilitasi proses pengambilan keputusan dengan menyusun permasalahan ke dalam struktur hierarki yang terdiri dari tujuan, kriteria, dan alternatif. Evaluasi dilakukan dengan membandingkan elemen-elemen dalam setiap tingkat *hierarki* secara berpasangan untuk menentukan tingkat kepentingannya. Nilai-nilai tersebut kemudian dinormalisasi dan dihitung bobot prioritasnya. Langkah utama dalam AHP meliputi: identifikasi tujuan dan kriteria, penyusunan struktur *hierarki*, perbandingan berpasangan, normalisasi matriks, perhitungan bobot (prioritas), serta pengujian konsistensi. Setelah itu, skor akhir untuk setiap alternatif diperoleh melalui agregasi bobot antar kriteria dan alternatif. Dengan demikian, AHP memberikan kerangka sistematis dan objektif dalam membandingkan alternatif, sekaligus mempertimbangkan konsistensi logis dalam penilaian pengambil keputusan.

4. *Website*

Website merupakan media informasi berbasis internet yang dirancang untuk menyajikan berbagai jenis konten, seperti teks, gambar, suara, dan animasi. Dengan dukungan teknologi jaringan komputer, *website* memungkinkan pengguna mengakses informasi secara global, kapan saja dan di mana saja. Selain sebagai sumber informasi, *website* juga dimanfaatkan untuk kebutuhan komunikasi dan interaksi sosial, termasuk dalam mendukung aktivitas bisnis. Seiring

perkembangan zaman, *website* menjadi salah satu alat utama dalam penyampaian informasi yang cepat, murah, dan efisien. Kemampuannya dalam menghubungkan berbagai halaman melalui *hyperlink* menjadikan penyajian informasi lebih terstruktur dan mudah diakses. *Website* sering dimanfaatkan sebagai sarana penyedia informasi. Selain itu, *website* juga berperan sebagai media sosial yang memungkinkan interaksi dan komunikasi antar pengguna menjadi lebih cepat dan efisien. (Firmansyah & Herman, 2023).

a. *Framework*

Framework adalah sekumpulan instruksi, kelas, dan fungsi yang sudah disusun secara sistematis untuk membantu pengembang dalam proses pembuatan aplikasi. Dengan adanya *framework*, *developer* tidak perlu menulis ulang kode program yang sama secara berulang, sehingga proses pengembangan aplikasi menjadi lebih cepat dan efisien. *Framework* merupakan sekumpulan instruksi yang disusun dalam bentuk *class* dan fungsi fungsi tertentu, yang masing masing memiliki peran khusus untuk memudahkan *developer* dalam penggunaannya tanpa perlu menulis ulang sintaks program yang sama secara berulang, sehingga dapat menghemat waktu dalam proses pengembangan (Sallaby & Kanedi, 2020).

Framework merupakan sebuah kerangka kerja yang dirancang secara sistematis untuk membantu pengembangan aplikasi, baik berbasis web, mobile, maupun desktop. Di dalam *framework* tersedia

beragam komponen, variabel, serta fungsi dasar yang siap digunakan untuk mempercepat proses pengembangan sistem. Dengan adanya struktur yang terorganisir dan komponen yang dapat digunakan ulang, framework memungkinkan pengembang bekerja secara lebih efisien, konsisten, dan fokus pada logika bisnis aplikasi tanpa harus membangun semuanya dari awal. *Framework* adalah suatu kerangka kerja yang dirancang untuk mempermudah dalam pengembangan sebuah *website*. Di dalamnya terdapat berbagai komponen dan variabel yang diperlukan oleh pengembang *web* guna memudahkan proses pembacaan kode, perencanaan, pengujian, serta pemeliharaan aplikasi (Prasena & Sama, 2020).

Dari pernyataan di atas dapat disimpulkan bahwa *framework* adalah suatu perangkat bantu berbasis perangkat lunak yang menyediakan kerangka kerja sistematis dan komponen-komponen fungsional yang dapat digunakan ulang dalam proses pengembangan aplikasi. Penggunaan *framework* tidak hanya mempercepat proses pembangunan sistem, tetapi juga membantu menciptakan kode program yang lebih terstruktur, bersih, dan mudah untuk dikembangkan maupun dipelihara. Dalam pengembangan berbagai jenis aplikasi, baik itu berbasis web, mobile, maupun desktop, framework memegang peran penting dalam meningkatkan produktivitas dan konsistensi hasil akhir. Dengan adanya pola dan aturan baku yang telah disediakan, pengembang tidak perlu membuat seluruh kode dari awal, sehingga proses pengembangan

menjadi lebih hemat waktu, efisien, dan mampu mengurangi potensi kesalahan dalam penulisan program.

b. *CodeIgniter*

Menurut Sulistiani et al., (2020), *CodeIgniter* merupakan salah satu *framework open source* berbasis *PHP* yang dirancang untuk memudahkan proses pengembangan aplikasi web. *Framework* ini mengadopsi pola arsitektur *Model-View-Controller (MVC)*, yang memungkinkan pemisahan antara logika bisnis, tampilan antarmuka, dan kontrol alur program, sehingga pengembangan aplikasi menjadi lebih rapi dan terstruktur. Keunggulan *CodeIgniter* terletak pada kemampuannya dalam mempercepat proses pembuatan aplikasi karena telah menyediakan berbagai resource dan library bawaan yang ringan, mudah digunakan, serta efisien dalam hal penggunaan sumber daya. Hal ini menjadikan *CodeIgniter* sangat cocok digunakan untuk membangun aplikasi web dinamis secara cepat, terutama dalam situasi yang membutuhkan performa dan fleksibilitas tinggi.

CodeIgniter merupakan *framework* atau kerangka kerja yang digunakan dalam pengembangan aplikasi *web* berbasis *PHP*. Dengan memanfaatkan *framework* ini, pengembang tidak perlu menulis seluruh kode dari awal karena telah tersedia berbagai library dan fungsi yang terstruktur untuk mempercepat proses pembuatan aplikasi. *CodeIgniter* menerapkan pola desain *Model, View, Controller (MVC)*, yang membantu memisahkan bagian logika, tampilan, dan kontrol aplikasi

secara jelas. Penggunaan *framework CodeIgniter* dipilih karena memungkinkan pengembangan program tanpa harus menulis kode dari awal, sehingga proses pengerjaannya menjadi lebih efisien dan cepat (Ridwan et al., 2022).

Dari pernyataan di atas dapat disimpulkan bahwa *CodeIgniter* merupakan sebuah kerangka kerja (*framework*) *PHP* sumber terbuka yang diciptakan untuk menyederhanakan dan mempercepat pengembangan aplikasi *web*. Dengan mengadopsi pola desain *Model-View-Controller (MVC)*, *CodeIgniter* memungkinkan pemisahan yang jelas antara logika bisnis aplikasi, antarmuka pengguna, dan mekanisme kontrol. Oleh karena itu, *CodeIgniter* merupakan solusi yang efektif untuk membangun aplikasi *web* yang dinamis dan tertata rapi.

c. *PHP*

Menurut Noviana, (2022), *PHP* merupakan bahasa pemrograman *scripting* sisi *server* yang dirancang khusus untuk pengembangan *website*. Namun, *PHP* juga dapat digunakan sebagai bahasa pemrograman umum. Bahasa ini pertama kali dibuat oleh *Rasmus Lerdorf* pada tahun 1994. Saat ini, *PHP* merupakan singkatan dari *PHP: Hypertext Preprocessor*, yaitu sebuah *akronim rekursif* di mana kepanjangannya mengandung nama itu sendiri. *PHP* bersifat gratis (*free*) dan *open source*, serta didistribusikan di bawah lisensi *PHP*

License, yang sedikit berbeda dari *lisensi GNU General Public License (GPL)* yang umum digunakan dalam proyek proyek *open source*.

Menurut Utomo & Hamdani, (2021), *PHP (Perl Hypertext Protocol)* merupakan bahasa pemrograman berbasis *skrip* yang dijalankan di sisi *server* dan diproses langsung oleh *server*. Selain itu, PHP juga termasuk salah satu dari banyak bahasa pemrograman yang *mendukung HTML (Hypertext Markup Language)*. PHP banyak digunakan dalam pembuatan aplikasi *web* dinamis yang mampu berjalan secara otomatis. Sebagai bahasa *skrip*, PHP memiliki fungsi utama dalam mengumpulkan serta mengevaluasi data seperti hasil survei atau *input* lainnya yang kemudian disimpan ke dalam server basis data (*database*). Selanjutnya, proses ini dapat memicu serangkaian aksi berantai, di mana *skrip* lain akan berinteraksi dengan *database* untuk mengelola, mengelompokkan, dan menyajikan informasi sesuai dengan permintaan pengguna (*user request*) melalui antarmuka situs *web*.

Sehingga dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman sumber terbuka (*open source*) yang serbaguna, dirancang khusus untuk mendukung pembuatan dan pengembangan situs *web*.

5. Basis Data

Menurut Rizki & Ferico, (2021), Basis data merupakan sekumpulan data atau informasi yang tersimpan secara terstruktur di dalam komputer, sehingga dapat diakses dan dimanfaatkan oleh suatu program

untuk memperoleh informasi dari data yang tersimpan tersebut. Data tersebut dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi atau perusahaan, baik dalam jangka pendek maupun panjang. Peran basis data sangat penting karena menjadi inti dari sistem informasi, memungkinkan banyak pengguna untuk mengakses, memproses, dan mengelola data dengan cepat dan efisien. Oleh karena itu, baik perusahaan kecil maupun besar sangat bergantung pada basis data untuk menjalankan aktivitas operasional secara terstruktur dan konsisten.

Menurut Putri, (2021), Sistem basis data merupakan perkembangan dari sistem file sebelumnya. Sistem file digunakan untuk menyimpan data dalam jangka waktu panjang serta mampu menampung data dalam jumlah besar. Informasi tersebut tersimpan dalam bentuk tabel tabel yang saling terhubung, memungkinkan pengelolaan data yang efisien, konsisten, dan mudah diakses. Melalui sistem manajemen basis data (DBMS), proses penyimpanan, pencarian, dan manipulasi data dapat dilakukan dengan cepat dan terorganisir, menjadikan basis data komponen penting dalam berbagai sistem informasi modern.

Sehingga dapat disimpulkan bahwa basis data adalah sekumpulan informasi yang terorganisir secara rapi dan tersimpan dalam sistem komputer, sehingga memudahkan proses penyimpanan, pengelolaan, dan pengambilan data secara cepat serta efisien. Basis data memiliki

peran yang sangat penting dalam mendukung kelancaran aktivitas operasional pada berbagai organisasi atau perusahaan, karena menyediakan akses informasi yang akurat dan dapat digunakan untuk jangka waktu yang panjang.

a. *MySQL*

MySQL merupakan salah satu jenis *server* database yang sangat populer. *MySQL* memanfaatkan bahasa *SQL* untuk mengakses basis datanya. *MySQL* tersedia dalam lisensi *FOSS License Exception*, namun juga memiliki versi berbayar untuk keperluan komersial. *MySQL* dikenal dengan slogannya “*The World's Most Popular OpenSource Database*”. (Ramadhan & Mukhaiyar, 2020).

6. *Flowchart*


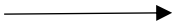

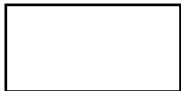

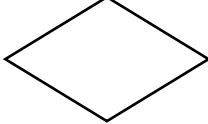
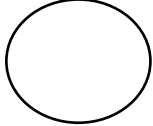
Flowchart adalah representasi *grafis* dari tahapan tahapan serta urutan proses dalam suatu program. *Flowchart* sistem menggambarkan runtutan proses dalam sistem dengan menunjukkan perangkat *input*, *output*, serta media yang digunakan untuk menyimpan data selama proses pengolahan berlangsung. Sementara itu, *flowchart* program merupakan *diagram* yang menggunakan simbol simbol tertentu untuk menunjukkan detail urutan proses, serta hubungan antara satu proses (instruksi) dengan proses lainnya dalam suatu program. Dalam penyusunannya, *flowchart* harus mengikuti pedoman tertentu agar mudah dipahami, yaitu digambar dari atas ke bawah atau dari kiri ke kanan, aktivitas yang ditampilkan harus jelas dan mudah dimengerti,

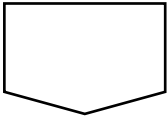

serta memiliki titik awal dan akhir yang tegas. Setiap langkah dalam aktivitas harus dijabarkan dengan kata kerja dan disusun secara sistematis sesuai urutan yang benar. Ruang lingkup aktivitas perlu diperhatikan dengan cermat, dan jika terdapat percabangan yang tidak berkaitan langsung, cukup digambarkan dengan simbol konektor dan bisa diletakkan di halaman terpisah. Penggunaan simbol-simbol standar juga sangat penting agar *flowchart* dapat dibaca dan dipahami oleh semua pihak yang terlibat dalam proses analisis dan pemrograman (Zalukhu Agustinus et al., 2023).

Flowchart adalah bentuk penulisan algoritma yang disusun menggunakan notasi *grafis*. *Flowchart* menampilkan *diagram* atau ilustrasi yang menunjukkan tahapan tahapan dalam suatu program beserta hubungan antar proses dan pernyataannya. Setiap tahapan dalam proses tersebut digambarkan menggunakan simbol tertentu, di mana masing masing simbol merepresentasikan jenis proses yang berbeda, sedangkan garis penghubung digunakan untuk menunjukkan alur antar proses. Penggunaan *flowchart* sangat membantu dalam memeriksa kemungkinan adanya bagian proses yang terlewat saat melakukan analisis permasalahan. *Flowchart* juga mempermudah analisis sistem maupun *programmer* dalam membagi permasalahan ke dalam bagian bagian kecil serta mengevaluasi berbagai alternatif solusi yang dapat diterapkan dalam sistem.

Secara umum, *flowchart* dapat didefinisikan sebagai rangkaian bagan yang memiliki aliran proses, menggambarkan langkah langkah penyelesaian masalah secara runtut. *Flowchart* juga menjadi representasi visual dari prosedur dalam sebuah program, yang disusun berdasarkan urutan eksekusi proses secara logis dan sistematis (Fauzi, 2020).

Tabel 2. 1 *Flowchart*

BAGAN	NAMA	FUNGSI
	<i>Terminator</i>	Awal atau akhir program
	<i>Flow Line</i>	Aliran program
	<i>Input / Output</i>	Input / Output data
	<i>Process</i>	Menyatakan suatu proses
	Sub Program	Permulaan sub program
	<i>Decision</i>	Menyatakan suatu kondisi Kemungkinan, ya/tidak.
	<i>On Page Connector</i>	Penghubung proses ke proses lain pada halaman yang sama.

BAGAN	NAMA	FUNGSI
	<i>Off page Connector</i>	Penghubung bagian berbeda halaman
	<i>Preparation</i>	Inisialisasi / pemberian nilai awal

Sumber : (Zalukhu Agustinus et al., 2023)

7. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan bahasa pemodelan berbasis grafis yang digunakan untuk merancang serta mendokumentasikan sistem dalam pengembangan perangkat lunak. UML banyak dimanfaatkan oleh *developer* atau pengembang perangkat lunak untuk mempermudah proses visualisasi, pencatatan, dan pemahaman terhadap struktur dan rancangan sistem yang akan dibuat. UML memungkinkan para pengembang perangkat lunak untuk memvisualisasikan struktur, perilaku, dan interaksi dalam sistem secara sistematis. Dengan menggunakan berbagai jenis diagram, *seperti use case, class, activity, sequence*, dan lainnya, UML membantu tim pengembang dalam memahami rancangan sistem yang akan dibangun, memfasilitasi komunikasi antar anggota tim, serta memastikan bahwa kebutuhan pengguna dapat diterjemahkan dengan baik ke dalam desain teknis (Gunawan et al., 2021).

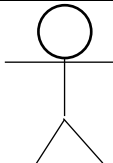


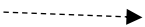
Unified Modeling Language (UML) adalah bahasa pemodelan perangkat lunak yang telah distandarisasi untuk digunakan sebagai panduan atau cetak biru (*blueprint*) dalam pengembangan perangkat lunak (menurut *Pressman*). UML dapat dimanfaatkan untuk memvisualisasikan, merinci, membangun, dan mendokumentasikan berbagai komponen dalam suatu sistem perangkat lunak. Dengan kata lain, sebagaimana seorang arsitek menyusun cetak biru untuk dijadikan acuan dalam proses pembangunan sebuah gedung oleh tim konstruksi, maka arsitek perangkat lunak membuat diagram UML untuk membantu *developer* atau *programmer* dalam mengembangkan sistem perangkat lunak. UML juga memudahkan komunikasi antar pemangku kepentingan (*stakeholders*), seperti programmer, analis sistem, pemilik proyek, dan pengguna akhir, karena mampu menyajikan desain sistem dalam bentuk diagram yang mudah dipahami (Sumiati et al., 2021).

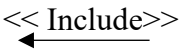

a) *Use case diagram*

Menurut A. R. Putri et al., (2021), *Use case diagram* berfungsi untuk memvisualisasikan interaksi antara pengguna (aktor) dengan aktivitas-aktivitas yang terdapat dalam sistem. Pada perancangan sistem ini, use case diagram mencakup tiga peran utama, yaitu pengguna sebagai pelanggan, dokter hewan, serta administrator. Melalui *use case*, dapat diidentifikasi berbagai fungsionalitas yang dimiliki sistem, bagaimana interaksi antara user dan sistem terjadi, serta keterhubungan antara aktor (pengguna) dengan fungsi-fungsi utama dalam sistem tersebut.

Menurut Helsalia et al., (2021), *Use case diagram* merupakan salah satu jenis diagram dalam *UML (Unified Modeling Language)* yang menunjukkan interaksi antara aktor dan sistem. *Use case* menggambarkan bentuk interaksi antara pengguna dan sistem yang digunakan. Diagram ini menjelaskan perilaku dari sistem atau sebagian sistem secara keseluruhan dan berfungsi sebagai representasi dari serangkaian tindakan yang diinginkan oleh pengguna sistem atau perangkat lunak yang sedang dikembangkan. *Use case* berperan dalam mendeskripsikan bagaimana sistem berperilaku, baik secara menyeluruh maupun pada bagian tertentu, melalui rangkaian tindakan atau tahapan yang diharapkan oleh pengguna terhadap sistem atau aplikasi yang tengah dirancang.

Tabel 2. 2 *Use case diagram*

Simbol	Elemen	Keterangan
	<i>Aktor</i>	Merepresentasikan peran pengguna, sistem yang berinteraksi dengan sistem melalui use case.
	<i>Use Case</i>	Menggambarkan interaksi atau fungsi yang terjadi antara sistem dengan aktor secara abstrak.
	<i>Association</i>	Merupakan hubungan antara aktor dan use case yang merepresentasikan komunikasi di antaranya
	<i>Generalisasi</i>	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari

Simbol	Elemen	Keterangan
		use case lainnya jika suatu kondisi terpenuhi.
	<i>Include</i>	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
	<i>Extend</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan dengan use case.


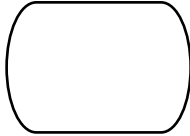


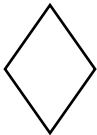
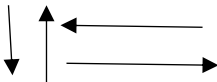
Sumber : (Gunawan et al., 2023)

b) *Activity diagram*

Menurut Saputra & Borman, (2020), *Activity diagram* dimanfaatkan untuk menggambarkan alur proses dari layanan perawatan salon, sesi konsultasi dengan dokter, serta proses pembelian. Dengan menyusun diagram aktivitas ini, proses proses yang terdapat dalam sistem dapat dipahami dengan lebih mudah. Diagram ini memodelkan urutan aktivitas dan keputusan yang terjadi dalam proses, sehingga memudahkan pemahaman terhadap logika alur kerja sistem. Seiring perkembangannya, *activity diagram* telah mengalami perubahan signifikan. Pada versi UML 1.X, penggunaannya lebih terbatas dan difokuskan untuk menggambarkan proses komputasional seperti login, dengan tampilan yang menyerupai *flowchart*. Namun, sejak UML versi 2.X, batasan tersebut telah dihapus sehingga *activity diagram* kini dapat dimanfaatkan secara lebih luas, tidak hanya untuk menggambarkan alur

sistem, tetapi juga untuk merepresentasikan proses bisnis yang kompleks dan dinamis.

Tabel 2. 3 *Activity diagram*

Simbol	Elemen	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Titik awal dari diagram aktivitas
	<i>Activity Final Node</i>	Titik akhir dari diagram aktivitas
	<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
	<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

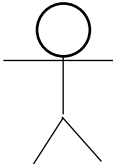
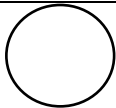
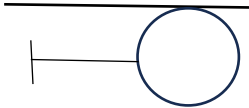
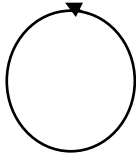
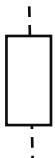
Sumber : (Gunawan et al., 2023)


c) *Sequence diagram*

Menurut Arianti et al., (2022), *Sequence diagram* merupakan sebuah jenis diagram yang menunjukkan interaksi antar objek yang bekerja sama dalam suatu sistem, dengan menampilkan urutan pesan yang

dikirimkan di antara elemen-elemen dari suatu class. Diagram ini menunjukkan bagaimana objek-objek yang terlibat dalam suatu *use case* saling berkomunikasi melalui pertukaran pesan, serta menyajikan urutan pengiriman pesan tersebut secara kronologis. Dengan demikian, *Sequence Diagram* memfasilitasi visualisasi alur komunikasi dan interaksi yang terjadi antara aktor (pengguna) dan sistem, maupun antar objek dalam sistem itu sendiri. Hal ini sangat membantu pengembang dalam memahami dinamika proses yang berlangsung di dalam sistem secara lebih terstruktur, runtut, dan logis, sehingga mempermudah proses analisis dan perancangan perangkat lunak.

Tabel 2. 4 *Sequence diagram*

Simbol	Elemen	Keterangan
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
	<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem.
	<i>Control Class</i>	Menggambarkan penghubung antar boundary dengan tabel
	<i>A Focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message.

	<i>Message</i>	Menggambarkan pengiriman pesan.
---	----------------	---------------------------------








Sumber : (Gunawan et al., 2023)

d) *Class diagram*

Menurut Ramdany et al., (2024), *Class diagram* adalah representasi hubungan antar kelas beserta penjabaran detail setiap kelas dalam model perancangan suatu sistem. Diagram ini juga menampilkan aturan serta tanggung jawab masing masing entitas yang membentuk perilaku dari sistem tersebut. Dalam tahap analisis, *class diagram* membantu mengidentifikasi entitas utama dan relasinya, sedangkan pada tahap desain, diagram ini digunakan untuk menangkap struktur dari semua kelas yang membentuk arsitektur sistem. *Class diagram* terdiri dari kumpulan kelas dan relasi-relasinya, di mana masing-masing kelas direpresentasikan dalam bentuk persegi yang terdiri atas tiga bagian: nama kelas, atribut, dan metode. Selain itu, *class diagram* juga berfungsi sebagai acuan untuk memahami alur struktur database dan keterkaitan antarkomponen dalam sistem, sehingga menjadi salah satu jenis diagram *Unified Modeling Language* (UML) yang paling umum digunakan dalam pengembangan dan desain perangkat lunak berorientasi objek.

Tabel 2. 5 Tabel Class diagram

Simbol	Elemen	Keterangan
--------	--------	------------

	Garis lurus (<i>Generalization</i>)	Menunjukkan hubungan objek anak (descendent) dan induk (ancestor) dalam hal berbagai perilaku dan struktur datanya.
	<i>Nary Association</i>	Suatu upaya untuk menghindari asosiasi yang melebihi 2 objek.
	<i>Class</i>	Suatu himpunan dari objek-objek dalam sistem
	<i>Collaboration</i>	Urutan aksi-aksi dalam Sistem.
	<i>Realization</i>	Operasi yang dilakukan oleh objek dalam sistem.
	<i>Dependency</i>	Suatu hubungan ketergantungan antar kelas
	<i>Association</i>	Menghubungkan objek yang satu dengan yang lainnya.

Sumber : (Gunawan et al., 2023)

B. Kajian Empiris

Penelitian ini disusun dengan menggunakan beberapa referensi jurnal yang berhubungan dengan topik yang diangkat yaitu sistem pendukung keputusan (SPK) pada pemilihan hero *support* terbaik dalam game *RPG Wuthering Waves*, yang digunakan sebagai acuan dan disusun secara sistematis dengan harapan dapat membantu dalam merancang dan membangun sistem ini. Studi literatur dari penelitian terdahulu digunakan sebagai dasar untuk meperkuat dan mematangkan desain sistem, terutama dalam hal penerapan metode *Analytical Hierarchy Process (AHP)*.

Penelitian yang dilakukan oleh Abidin et al., (2024), membahas pengembangan sistem pendukung keputusan untuk membantu pemain dalam memilih hero terbaik di dalam game *Mobile Legends* pada tier *Epic*. Penelitian ini menggunakan metode gabungan *AHP* dan *TOPSIS* dalam proses perhitungan bobot dan perangkingan. Kriteria yang digunakan meliputi kemampuan *crowd control*, serangan area, daya tahan, potensi late game, dan kemampuannya menghindar. Proses dimulai dengan pembobotan kriteria menggunakan *ahp*, dilanjutkan dengan perhitungan preferensi antar alternatif hero menggunakan *TOPSIS*. Sistem dibangun berbasis web dan diuji melalui metode *black box* serta *alpha test* untuk mengevaluasi kinerja sistem. Hasilnya menunjukkan bahwa sistem dapat memberikan rekomendasi hero secara tepat dan efisien sesuai peran dan kriteria yang dipilih oleh pengguna. Temuan dari penelitian ini menunjukkan bahwa penerapan metode *AHP* efektif dalam menghasilkan bobot kriteria yang valid dan konsisten, serta dapat diadaptasikan dalam pengembangan sistem rekomendasi hero pada game lain, seperti *Wuthering Waves*.

Penelitian yang dilakukan Wirahaji et al., (2022), Yang mengembangkan sistem pendukung keputusan dalam pemilihan smartphone gaming terbaik menggunakan metode *Analytical Hierarchy Process (AHP)*. Penelitian ini bertujuan untuk membantu pengguna dalam memilih smartphone yang sesuai berdasarkan kriteria seperti *RAM*, *prosesor*, baterai, dan harga. Proses *AHP* digunakan untuk

menentukan bobot setiap kriteria melalui matriks perbandingan berpasangan, dilanjutkan dengan penghitungan nilai *eigen vektor* dan konsistensi. Berdasarkan hasil pembobotan dan perhitungan akhir, sistem berhasil merekomendasikan smartphone dengan peringkat tertinggi sebagai pilihan terbaik. Hasil dari penelitian ini menunjukkan bahwa metode *AHP* efektif dalam menyusun prioritas kriteria secara objektif, serta dapat digunakan sebagai pendekatan sistematis dalam pengambilan keputusan. Hal ini menjadi dasar referensi penting untuk mengembangkan sistem serupa dalam konteks yang berbeda, seperti pemilihan hero *support* dalam game *RPG Wuthering Waves*, dengan menyesuaikan kriteria berdasarkan kebutuhan *gameplay*.

Penelitian yang dilakukan Herdiansah, (2020), mengembangkan sebuah sistem pendukung keputusan (SPK) berbasis metode *Analytical Hierarchy Process (AHP)* untuk membantu siswa kelas XII IPA dalam menentukan jurusan studi lanjut di fakultas teknik. Permasalahan utama dalam penelitian ini adalah minimnya pemahaman siswa terhadap jurusan yang sesuai dengan kemampuan akademik mereka, serta belum tersedianya sistem yang mampu memberikan rekomendasi secara objektif. Kriteria yang digunakan meliputi lima mata pelajaran utama yaitu Bahasa Inggris, Matematika, Fisika, Biologi, dan Kimia, dengan bobot yang diperoleh dari lima ketua program studi teknik sebagai narasumber ahli. Proses *AHP* digunakan untuk membentuk struktur *hierarki*, menghitung bobot kriteria melalui matriks perbandingan

berpasangan, dan melakukan konsistensi pengujian. Hasil akhir menunjukkan sistem mampu memberikan peringkat jurusan berdasarkan kecocokan nilai siswa, dengan jurusan Teknik Mesin menjadi pilihan teratas pada contoh kasus.

Penelitian yang dilakukan oleh Siregar et al., (2022), membangun sistem pendukung keputusan untuk menentukan guru terbaik di SMK Sirajul Falah Parung dengan menggunakan metode gabungan AHP dan TOPSIS. Latar belakang dari penelitian ini adalah adanya sistem pemilihan guru yang sebelumnya masih bersifat manual dan subjektif, serta belum terintegrasi secara digital. Kriteria yang digunakan dalam proses pengambilan keputusan meliputi moralitas, tanggung jawab, kedisiplinan, dan penguasaan materi. Metode AHP digunakan untuk menentukan bobot prioritas tiap kriteria, sementara TOPSIS digunakan untuk menghitung nilai preferensi dan menentukan perangkingan alternatif. Hasil akhir menunjukkan bahwa guru dengan nama Abdul Choir, S.Kom menjadi alternatif terbaik dengan nilai preferensi 1.000, menjadikannya kandidat paling layak sebagai guru terbaik.

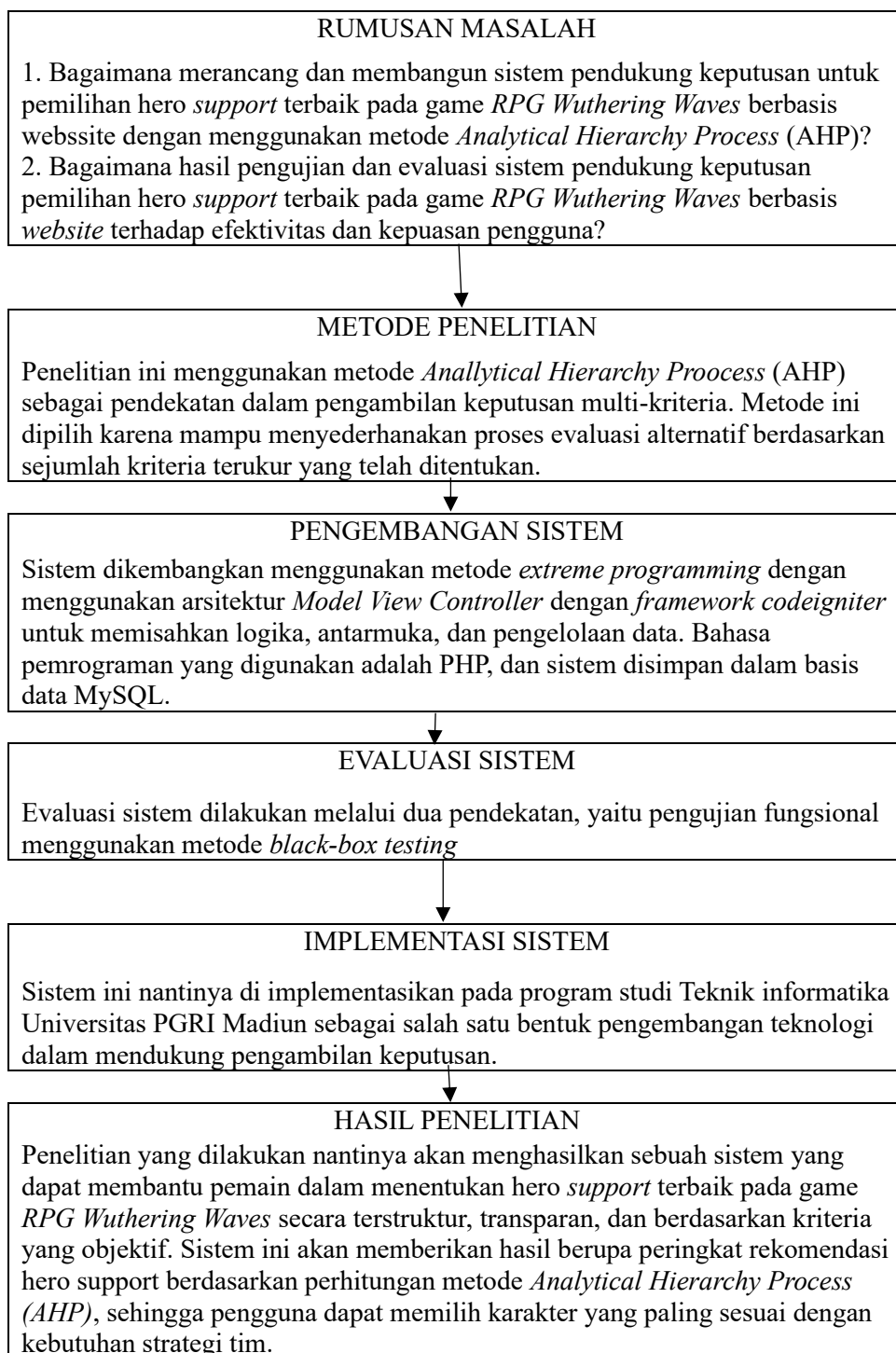
Penelitian yang dilakukan oleh Prawira & Amin, (2022), Penelitian ini mengimplementasikan sebuah sistem pendukung keputusan menggunakan metode *Analytical Hierarchy Process (AHP)* untuk pemilihan karyawan terbaik di PT. Citra Prima Batara. Metode *AHP* dipilih karena kemampuannya dalam membobot kriteria dan menentukan prioritas, serta menggunakan konsep *eigen vector* untuk

perangkaian prioritas berdasarkan matriks perbandingan berpasangan, yang cocok untuk masalah pengambilan keputusan multi-objektif dan multi-kriteria. Kriteria yang digunakan dalam penilaian kinerja karyawan meliputi perilaku kerja, disiplin kerja, kualitas kerja, tanggung jawab kerja, kehadiran, dan kejujuran. Proses pengambilan keputusan dilakukan dengan menilai alternatif berdasarkan kriteria yang telah ditetapkan. Hasil perhitungan *AHP* menunjukkan bahwa urutan kriteria paling penting adalah kejujuran, diikuti oleh tanggung jawab kerja, kehadiran, kualitas kerja, perilaku kerja, dan disiplin kerja. Berdasarkan perhitungan matriks alternatif, Bapak Rega memperoleh nilai tertinggi sebesar 35,3%, menjadikannya rekomendasi utama sebagai karyawan terbaik. Konsistensi preferensi responden juga terbukti melalui nilai *Consistency Ratio (CR)* yang konsisten di bawah 0,100 untuk semua kriteria dan alternatif. Secara keseluruhan, sistem ini berhasil membantu perusahaan dalam meningkatkan proses perhitungan dan memberikan rekomendasi karyawan terbaik berdasarkan peringkat.

Berdasarkan kajian literatur yang telah dijelaskan sebelumnya, pemilihan metode yang tepat dalam proses pengembangan sistem menjadi hal penting agar sistem yang dibangun dapat berjalan secara optimal dan sesuai dengan tujuan. Dalam penelitian ini, peneliti menggunakan metode *Extreme Programming (EP)* sebagai pendekatan pengembangan perangkat lunak karena fleksibel dan mendukung

proses iteratif yang responsif terhadap perubahan kebutuhan. Untuk mendukung struktur sistem, digunakan pola *Model View Controller (MVC)* dengan *framework CodeIgniter*. Pemilihan *framework* ini mempermudah pemisahan antara logika bisnis, tampilan antarmuka, dan pengelolaan data sehingga sistem lebih terorganisir dan mudah dikembangkan. *Metode Analytical Hierarchy Process (AHP)* dipilih sebagai metode utama dalam sistem pendukung keputusan, karena mampu memberikan pembobotan kriteria secara objektif dan logis dalam konteks pemilihan hero support terbaik dalam game *Wuthering Waves*. Evaluasi sistem dilakukan melalui pengujian fungsional menggunakan metode *blackbox testing* untuk memastikan bahwa semua fitur berjalan sesuai dengan fungsinya. Dengan pendekatan ini, sistem yang dibangun diharapkan mampu menghasilkan proses rekomendasi yang terstruktur, transparan, dan sesuai dengan kebutuhan pengguna.

C. Kerangka Berfikir



Gambar 2. 1 Kerangka Berfikir