

## **BAB II**

### **KAJIAN PUSTAKA**

#### **A. Kajian Teoritis**

##### **1. Pengertian Sistem Pendukung Keputusan (SPK)**

Menurut Hutahaean et al. (2023), Sistem Pendukung Keputusan (SPK) adalah salah satu elemen dari sistem informasi berbasis komputer yang berfungsi membantu pihak yang berkepentingan dalam mengambil keputusan dengan menggunakan data dan model analisis tertentu. Sistem ini memanfaatkan data yang tersedia, model perhitungan, dan teknik analisis tertentu untuk menghasilkan rekomendasi atau pilihan yang dapat dipertimbangkan oleh pengguna. Dengan adanya SPK, proses pengambilan keputusan menjadi lebih efisien dan akurat karena informasi yang diberikan lebih terarah dan relevan.

Dari uraian tersebut, dapat disimpulkan bahwa Sistem Pendukung Keputusan (SPK) merupakan sistem komputer interaktif yang dirancang untuk membantu pengambil keputusan dalam menangani permasalahan yang bersifat kompleks, dengan cara mengolah data dan menggunakan model analisis agar proses pengambilan keputusan menjadi lebih efektif.

## 2. Beasiswa

Beasiswa adalah bantuan finansial yang diberikan kepada seseorang sebagai bentuk dukungan untuk melanjutkan jenjang pendidikan. Bantuan ini dapat diperoleh dari berbagai sumber, termasuk instansi pemerintah, organisasi swasta, maupun lembaga non-profit seperti yayasan., maupun institusi pendidikan. Selain untuk mendukung pendidikan formal, beasiswa juga bisa diberikan sebagai bentuk penghargaan atas prestasi atau dalam rangka peningkatan kualitas sumber daya manusia, misalnya oleh tempat kerja penerima beasiswa (Alamsyah et al., 2021).

Beasiswa merupakan upaya yang bertujuan untuk mengurangi beban keuangan mahasiswa selama menjalani proses pendidikan. Beasiswa dapat diartikan sebagai bentuk apresiasi atau bantuan yang diberikan kepada seseorang untuk mendukung kelanjutan pendidikannya ke tingkat yang lebih tinggi. Bantuan ini bisa berupa kemudahan akses ke lembaga pendidikan maupun dukungan finansial secara langsung. Secara umum, bantuan keuangan dalam program beasiswa mencakup biaya pendidikan seperti uang kuliah, biaya buku, hingga biaya hidup, sehingga meringankan tanggungan finansial mahasiswa atau keluarganya. Program ini tidak hanya membantu secara ekonomi, tetapi juga mendorong motivasi dan prestasi belajar penerimanya (Susanti & Nawangsit, 2023).

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa beasiswa merupakan bentuk dukungan finansial untuk pendidikan yang bisa diberikan atas dasar kebutuhan ekonomi maupun prestasi akademik.

Agar manfaatnya optimal, pemberian beasiswa harus mempertimbangkan kriteria penerima seperti kompetensi, kualitas, dan kelayakan, sehingga dapat menunjang pengembangan pendidikan dan peningkatan sumber daya manusia secara merata.

### **3. Kartu Indonesia Pintar Kuliah**

Melalui Program Indonesia Pintar, pemerintah meluncurkan Kartu Indonesia Pintar Kuliah (KIP Kuliah) pada tahun 2020 sebagai bentuk dukungan pendidikan bagi lulusan SMA, SMK, atau yang setara dari keluarga kurang mampu. Inisiatif ini bertujuan untuk memberikan kesempatan kepada mereka agar dapat melanjutkan pendidikan ke perguruan tinggi atau akademi. Pemanfaatan KIP Kuliah diharapkan dapat menjangkau seluruh wilayah Indonesia, sehingga para penerima tidak hanya mampu meningkatkan taraf hidup keluarganya, tetapi juga turut berkontribusi dalam pembangunan bangsa. Program ini memberikan jaminan keberlanjutan studi di perguruan tinggi melalui pembebasan biaya pendidikan serta bantuan biaya hidup bulanan bagi mahasiswa yang memenuhi kriteria (Kementerian Pendidikan dan Kebudayaan, 2021).

Penerima bantuan Kartu Indonesia Pintar Kuliah (KIP Kuliah) harus berasal dari keluarga miskin atau rentan miskin yang telah melalui proses verifikasi oleh perguruan tinggi. Mahasiswa yang menjadi prioritas penerima KIP Kuliah mencakup mereka yang telah memiliki KIP pada jenjang pendidikan menengah, tercantum dalam Data Terpadu Kesejahteraan Sosial (DTKS), atau menerima bantuan sosial dari

kementerian terkait. Selain itu, program ini juga menyasar mahasiswa dari kelompok masyarakat dalam desil 1 hingga 3 berdasarkan Data Percepatan Penghapusan Kemiskinan Ekstrem (PPKE), serta mereka yang berasal dari panti asuhan atau panti sosial, dengan syarat telah lulus seleksi masuk perguruan tinggi melalui jalur SNBP, SNBT, atau seleksi mandiri di perguruan tinggi negeri maupun swasta (Kemdiktisaintek, 2025).

Dari pemaparan di atas dapat disimpulkan bahwa KIP Kuliah merupakan wujud nyata komitmen pemerintah dalam memastikan kesempatan mengenyam pendidikan tinggi bagi lulusan SMA, SMK, atau setara yang berasal dari keluarga kurang mampu. Selain menanggung biaya pendidikan di perguruan tinggi, program ini juga menyediakan bantuan biaya hidup setiap bulan bagi mahasiswa yang memenuhi persyaratan baik secara ekonomi maupun akademik. KIP Kuliah diprioritaskan bagi mereka yang telah terverifikasi sebagai penerima bantuan sosial dari pemerintah, termasuk pemilik KIP pendidikan menengah, peserta DTKS, data PPKE desil 1–3, serta anak panti asuhan, sehingga diharapkan dapat meningkatkan taraf hidup dan kontribusi mereka terhadap pembangunan bangsa.

#### **4. *Metode Multi Attribute Utility Theory (MAUT)***

*Metode Multi Attribute Utility Theory (MAUT)* merupakan metode yang sering digunakan untuk membandingkan berbagai pilihan berdasarkan kombinasi antara biaya, risiko, dan manfaat. Setiap kriteria dalam metode ini memiliki beberapa alternatif solusi yang memungkinkan.

Dalam menentukan alternatif yang paling sesuai dengan preferensi pengguna, dilakukan proses perhitungan dengan cara mengalikan setiap nilai alternatif dengan skala prioritas yang telah ditentukan. Alternatif yang memperoleh skor tertinggi dianggap sebagai pilihan paling baik. MAUT juga dikenal sebagai metode pengambilan keputusan multikriteria yang mengintegrasikan preferensi subjektif dari pengambil keputusan dengan data objektif dari masing-masing alternatif, di mana setiap kriteria diberikan bobot sesuai tingkat kepentingannya (Hutahaean et al, 2023).

a. Perhitungan MAUT

Metode MAUT adalah pendekatan yang digunakan untuk mengevaluasi suatu alternatif (x) dengan cara menghitung nilai akhir  $v(x)$ , yang diperoleh dari penjumlahan antara bobot dan nilai relevan dari setiap dimensi atau kriteria. Nilai akhir ini biasa disebut sebagai nilai utilitas. MAUT berfungsi untuk mengubah berbagai aspek kepentingan menjadi bentuk nilai numerik dalam rentang 0 hingga 1, di mana angka 0 menunjukkan alternatif dengan performa paling rendah, dan angka 1 menunjukkan alternatif terbaik. Dengan konversi ini, berbagai kriteria yang memiliki satuan berbeda dapat dibandingkan secara langsung. Perhitungan dalam metode ini dilakukan menggunakan sejumlah rumus matematis yang telah ditentukan (A. Saputra et al., 2024):

1) Membentuk matriks keputusan

$$X_{ij} = \begin{pmatrix} X_{11} & X_j & X_{1n} \\ X_{ij} & X_{ij} & X_{in} \\ X_{m1} & X_{mj} & X_m \end{pmatrix}$$

Dimana:

m adalah jumlah alternatif kandidat, n adalah jumlah kriteria evaluasi dan  $X_{ij}$  kinerja alternatif sehubungan dengan kriteria j.

- 2) Melakukan normalisasi dengan rumus.

$$U(x) = \frac{X - X_{i-}}{X_{i+} - X_{i-}}$$

Dimana:

$U(x)$  = Nilai normalisasi bobot alternatif x.

X = Nilai alternatif

$x_{i-}$  = Nilai terendah dari kriteria alternatif ke - x

$x_{i+}$  = Nilai tertinggi dari kriteria alternatif ke - x

- 3) Perhitungan Nilai Preferensi

$$v(x) = \sum_{j=1}^n w_j \cdot X_{ij}$$

Keterangan :

$V(x)$  = Evaluasi total dari alternatif ke- x.

$W_j$  = Bobot relatif kriteria

n = Jumlah kriteria.

$X_{ij}$  = Hasil evaluasi atribut (kriteria) untuk alternatif ke- x.

Penerapan metode MAUT dilakukan melalui beberapa tahapan sistematis. Pertama, masalah keputusan dibagi ke dalam beberapa kriteria atau dimensi yang relevan. Selanjutnya, ditentukan bobot relatif untuk setiap kriteria berdasarkan tingkat

kepentingannya. Setelah itu, seluruh alternatif yang akan dievaluasi didaftarkan. Masing-masing alternatif kemudian diberikan nilai utilitas sesuai atributnya terhadap setiap kriteria. Langkah terakhir adalah mengalikan nilai utilitas dengan bobot kriteria untuk memperoleh skor akhir dari setiap alternatif, sehingga dapat diketahui alternatif mana yang paling sesuai untuk dijadikan keputusan (Botutihe & Biki, 2023).

Dari pernyataan di atas dapat disimpulkan bahwa metode MAUT merupakan salah satu pendekatan dalam pengambilan keputusan multikriteria yang mengubah berbagai kepentingan atau atribut menjadi nilai numerik dalam skala 0 hingga 1, di mana 0 menunjukkan pilihan terburuk dan 1 terbaik. Proses evaluasi dilakukan dengan menghitung nilai akhir dari setiap alternatif berdasarkan bobot dan nilai utilitas dari masing-masing kriteria. Langkah-langkah utama dalam MAUT meliputi: identifikasi kriteria, penentuan bobot, pencatatan alternatif, pemberian nilai utilitas, serta perhitungan akhir dengan mengalikan nilai utilitas dan bobot. Dengan cara ini, MAUT memudahkan pembuat keputusan untuk membandingkan berbagai alternatif secara objektif dan sistematis.

## 5. *Website*

*Website* merupakan produk kemajuan teknologi informasi yang digunakan untuk menyajikan berbagai jenis konten, seperti teks, gambar,

audio, dokumen, dan video. Keunggulan utamanya terletak pada kemampuan menghubungkan antar halaman melalui *hyperlink*, sehingga mempermudah pengguna dalam mengakses informasi secara terstruktur dan interaktif. Di era digital saat ini, *website* menjadi sarana vital dalam penyebaran informasi secara luas, cepat, dan *real-time* (Rahmi et al., 2023).

a. *Framework*

Selama proses pengembangan *web*, alat bantu yang berbentuk kerangka kerja dikenal sebagai *framework*. Fokus utama penggunaan *framework* adalah untuk membantu pengembang *web* (*web developer*) menulis kode program dengan lebih mudah dan cepat. Jika *framework* memiliki struktur dan aturan yang jelas, pengembang tidak perlu membangun aplikasi dari awal secara manual. Dengan menawarkan komponen dasar yang dapat digunakan kembali, *framework* membuat proses pembuatan *web* lebih efisien, terorganisir, dan konsisten (Nasution et al., 2023).

*Framework* merupakan perangkat lunak yang memiliki struktur kerangka kerja terorganisir dan dirancang untuk membantu pengembang dalam membangun aplikasi, baik berbasis *web*, *mobile*, maupun *desktop*. Di dalamnya terdapat berbagai perintah dan fungsi dasar yang bersifat umum, yang dapat digunakan kembali dalam proses pengembangan aplikasi. Dengan memanfaatkan *framework*, proses pembuatan aplikasi menjadi lebih cepat, efisien, serta

menghasilkan struktur kode yang rapi dan teratur (Kansha et al., 2023).

Dari pernyataan di atas dapat disimpulkan bahwa *framework* merupakan alat bantu berupa perangkat lunak yang menyediakan struktur kerja terorganisir dan komponen-komponen dasar yang dapat digunakan kembali dalam pengembangan aplikasi. *Framework* tidak hanya mempercepat proses pembuatan aplikasi, tetapi juga memastikan bahwa kode yang ditulis oleh pengembang menjadi lebih rapi, terstruktur, dan mudah dipelihara. Baik dalam pengembangan aplikasi *web*, *mobile*, maupun *desktop*, *framework* berperan penting dalam meningkatkan efisiensi kerja dan konsistensi sistem yang dibangun. Dengan adanya aturan dan pola yang sudah ditetapkan di dalam *framework*, pengembang tidak perlu memulai dari nol, sehingga waktu dan tenaga dapat lebih dihemat serta risiko kesalahan dalam pengkodean dapat diminimalkan.

b. *CodeIgniter*

Menurut Martani et al. (2022), *CodeIgniter* adalah salah satu *framework open source* untuk pengembangan aplikasi web yang menggunakan bahasa pemrograman PHP. *Framework* ini mengadopsi pola arsitektur MVC (*Model, View, Controller*) yang memungkinkan pengembangan aplikasi web dinamis menjadi lebih cepat dan terstruktur. Keunggulan utama dari *CodeIgniter* terletak

pada performanya yang ringan, cepat, dan didukung dokumentasi yang sangat lengkap, termasuk contoh penerapan kode yang jelas.

*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. Tujuan utama dari penggunaan *CodeIgniter* adalah menyediakan berbagai alat bantu seperti *helper* dan *library* yang mendukung pelaksanaan tugas-tugas umum dalam pengembangan, sehingga proyek dapat diselesaikan lebih efisien dan terorganisir (Krisna et al., 2022).

Dari pernyataan di atas dapat disimpulkan bahwa *CodeIgniter* merupakan *framework open source* berbasis PHP yang dirancang untuk mempermudah dan mempercepat proses pengembangan aplikasi web. Dengan pola arsitektur *Model-View-Controller* (MVC), *CodeIgniter* membantu memisahkan logika aplikasi, tampilan, dan kontrol secara terstruktur. Dengan begitu, *CodeIgniter* menjadi pilihan yang efisien untuk membangun aplikasi web yang dinamis dan terorganisir.

c. PHP

Menurut (Noviana, 2022), PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman skrip yang dijalankan di sisi server dan dirancang khusus untuk membangun aplikasi berbasis web. Meskipun utamanya digunakan dalam pengembangan web, PHP juga dapat berfungsi sebagai bahasa pemrograman umum. Bahasa ini pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994. Pada awal kemunculannya, PHP merupakan singkatan dari *Personal Home Page*, namun seiring perkembangannya, singkatan tersebut diubah menjadi *PHP: Hypertext Preprocessor*, yang merupakan singkatan rekursif yaitu bentuk kepanjangan yang menyertakan akronim itu sendiri. Sebagai perangkat lunak yang bersifat open source, PHP dapat digunakan secara gratis dan kode sumbernya dapat dimodifikasi oleh siapa saja. PHP dirilis di bawah lisensi PHP License, yang memiliki perbedaan tertentu dengan lisensi GNU *General Public License* (GPL) yang umum digunakan dalam proyek-proyek perangkat lunak sumber terbuka lainnya.

Menurut Sabarudin et al. (2022), PHP merupakan bahasa pemrograman yang bersifat *open source* dan bersifat universal, dirancang khusus untuk menangani proses pembuatan serta pengembangan situs web. 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 . 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 merupakan bahasa pemrograman yang bersifat open source dan bersifat universal, dirancang khusus untuk menangani proses pembuatan serta pengembangan situs web.

## 6. Basis Data

Menurut Putri (2021), basis data adalah kumpulan informasi yang disimpan dalam jangka waktu lama, bahkan bisa mencapai bertahun-tahun, dan berperan penting dalam operasional berbagai jenis usaha. Baik perusahaan besar maupun kecil sangat bergantung pada basis data untuk mengelola informasi penting secara terstruktur dan sistematis. Basis data bekerja di balik sistem perusahaan untuk memastikan akses data yang cepat dan efisien ketika dibutuhkan.

Menurut Connolly & Kroenke dalam Gunawan et al. (2023), basis data merupakan sekumpulan data yang disusun secara sistematis dan terstruktur dalam suatu sistem komputer. Data tersebut tersimpan dalam beberapa tabel yang saling berelasi, sehingga memungkinkan pengelolaan dan pengambilan informasi secara efisien dan terorganisir.

Sehingga dapat disimpulkan bahwa basis data adalah kumpulan data atau informasi yang disusun secara sistematis dan terstruktur dalam sistem komputer, yang memungkinkan penyimpanan, pengelolaan, serta pengambilan data secara efisien dan terorganisir. Basis data berperan penting dalam mendukung operasional berbagai instansi atau perusahaan, karena menyediakan akses cepat terhadap informasi yang dibutuhkan dalam jangka panjang.

a. MySQL

MySQL merupakan sistem manajemen basis data (DBMS) bersifat *opensource* yang mendukung dua model lisensi, yakni lisensi bebas (gratis) dan versi komersial (berbayar). Dengan lisensi GNU *General Public License* (GPL), pengguna dapat memanfaatkan MySQL secara gratis, baik untuk keperluan individu maupun bisnis. Kemudahan akses tersebut, ditambah dengan komunitas pengguna yang luas serta dokumentasi teknis yang lengkap, menjadikan MySQL sebagai salah satu platform favorit dalam pengembangan sistem aplikasi. (Siregar et al., 2024).

## 7. *Flowchart*




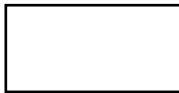

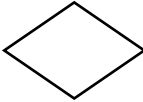
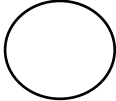
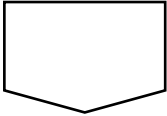

*Flowchart* merupakan representasi grafis yang menggambarkan urutan langkah atau prosedur dalam suatu program atau sistem secara terstruktur. Flowchart dibagi menjadi dua jenis, yaitu *flowchart* sistem yang menjelaskan alur proses suatu sistem lengkap dengan *media input*, *output*, serta media penyimpanan yang digunakan, dan *flowchart* program

yang menunjukkan urutan proses secara rinci dalam bentuk simbol-simbol standar yang saling berhubungan antar instruksi 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 et al., 2023).

Flowchart merupakan representasi grafis yang menggambarkan urutan proses atau tahapan secara sistematis dalam menjalankan sebuah program. Diagram ini membantu dalam proses analisis, perancangan, dan pengkodean untuk menyelesaikan suatu permasalahan secara lebih rinci dalam aktivitas operasional tertentu. Flowchart biasanya dimanfaatkan untuk mempermudah identifikasi solusi ketika dilakukan evaluasi lebih lanjut terhadap suatu sistem. Dalam konteks lain, flowchart dapat dijelaskan sebagai diagram yang memakai simbol-simbol grafis untuk menyatakan aliran suatu proses. Simbol-simbol tersebut mewakili berbagai tahapan atau tindakan yang harus dilakukan. Secara umum,

flowchart adalah representasi visual dari langkah-langkah atau urutan prosedural dalam suatu program atau kegiatan yang dirancang untuk mencapai tujuan tertentu (Listyoningrum et al., 2023).

Tabel 2. 1 *Flowchart*

<b>SIMBOL</b>	<b>NAMA</b>	<b>FUNGSI</b>
	<i>Terminator</i>	Menunjukkan titik awal atau akhir dari suatu alur program.
	<i>Flow Line</i>	Menandakan arah aliran proses atau urutan langkah dalam program.
	<i>Input / Output</i>	Menggambarkan Input / Output data
	<i>Process</i>	Merepresentasikan aktivitas atau tindakan tertentu dalam program.
	<i>Sub Program</i>	Menandai dimulainya suatu proses terpisah atau prosedur tambahan dalam sistem.
	<i>Decision</i>	Menyatakan suatu kondisi Kemungkinan, ya/tidak.
	<i>On Page Connector</i>	Penghubung proses ke proses lain pada halaman yang sama.
	<i>Off page Connector</i>	Menghubungkan alur proses yang terletak pada halaman yang berbeda.
	<i>Preparation</i>	Menyatakan langkah inialisasi atau pemberian nilai awal sebelum proses utama dimulai.



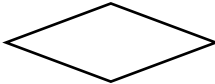

Sumber : (Zalukhu et al., 2023)

## 8. *Entity Relationship Diagram (ERD).*

ERD merupakan representasi konseptual dari model data yang digunakan untuk menunjukkan relasi antar entitas dalam sebuah sistem informasi. Tujuannya adalah untuk membantu pengembang perangkat lunak dalam proses pemahaman, perancangan, dan penerapan struktur basis data secara efisien dan terorganisir. Fungsi utama ERD antara lain mempermudah analisis basis data dengan cepat dan efisien, menyajikan relasi antar data yang memiliki keterkaitan melalui objek dan hubungan, mendokumentasikan data secara sistematis dengan mengidentifikasi setiap entitas dan relasinya, serta memfasilitasi proses pengujian terhadap model yang telah dirancang (Gunawan et al., 2023).

Entitas Relationship Diagram (ERD) dirancang untuk mempermudah proses analisis sistem serta mendukung tahapan perancangan berikutnya. Penyusunan ERD dilakukan dengan memvisualisasikan seluruh entitas beserta hubungan antarentitas yang ada, termasuk relasi antar dua atau lebih entitas. Dengan demikian, ERD memberikan gambaran menyeluruh mengenai struktur dan keterkaitan data dalam sistem yang akan dikembangkan (Sayuti et al., 2023).

Tabel 2. 2 *Entity Relationship Diagram*

Simbol	Arti Penggunaan
	Menunjukkan entitas, melambangkan objek nyata atau abstrak yang datanya perlu disimpan dalam sistem.
	Garis, menghubungkan antara relasi dan entitas, serta antara entitas dan atribut dalam diagram.
	Relasi, menggambarkan keterkaitan atau hubungan logis antara dua entitas yang berbeda.
	Atribut merupakan karakteristik atau informasi penting yang dimiliki oleh suatu entitas.

Sumber : (Gunawan et al., 2023).

## 9. *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan bahasa pemodelan grafis yang digunakan secara luas dalam pengembangan perangkat lunak untuk merancang dan mendokumentasikan sistem. 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., 2023).

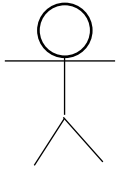

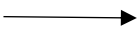
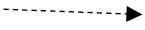
UML adalah bahasa pemodelan grafis yang digunakan dalam pengembangan perangkat lunak berorientasi objek untuk membantu memvisualisasikan, merancang, membangun, serta mendokumentasikan sistem. UML memiliki peran penting dalam mendeskripsikan struktur serta perilaku sistem secara terstruktur melalui beragam jenis diagram, sehingga memudahkan pemahaman, perancangan, serta komunikasi antar tim pengembang dalam membangun sistem yang kompleks dan terstruktur (D. Saputra et al., 2023).

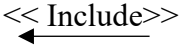

*a) Use case diagram*

Menurut Setiyani (2021), *Use case* adalah representasi fungsional dari sebuah sistem yang disusun berdasarkan perspektif pengguna. Di dalamnya dijelaskan proses-proses yang akan dijalankan oleh sistem beserta elemen-elemen yang terlibat dalam pelaksanaannya. *Use case* beroperasi berdasarkan skenario, yakni berupa penjabaran langkah-langkah yang menggambarkan proses interaksi antara sistem dan pengguna, baik dari sisi pengguna ke sistem maupun sebaliknya. 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 (Gunawan et al., 2023).

Menurut Helsalia et al. (2021), *Use Case Diagram* adalah salah satu bentuk diagram dalam UML (*Unified Modeling Language*) yang berfungsi untuk memvisualisasikan relasi dan interaksi antara sistem dengan aktor yang terlibat. Diagram ini menunjukkan berbagai kemungkinan interaksi yang dapat terjadi antara pengguna dan sistem. *Use case* berfungsi untuk menjelaskan perilaku sistem, baik secara keseluruhan maupun sebagian, melalui serangkaian aksi atau langkah-langkah yang diharapkan oleh calon pengguna terhadap perangkat lunak atau sistem yang sedang dikembangkan.

Tabel 2. 3 *Use case diagram*

Simbol	Elemen	Keterangan
	<i>Aktor</i>	Pengguna atau sistem eksternal yang melakukan interaksi dengan sistem melalui <i>use case</i> .
	<i>Use Case</i>	Menjelaskan fungsi atau aktivitas utama yang dilakukan antara sistem dan aktor dalam bentuk abstraksi.
	<i>Association</i>	Menyatakan koneksi atau hubungan komunikasi antara aktor dan use case yang saling terhubung.
	<i>Generalisasi</i>	Menunjukkan bahwa suatu use case adalah bagian dari fungsi tambahan yang muncul jika kondisi tertentu terpenuhi.


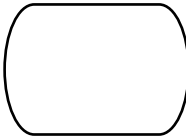


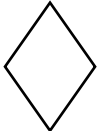
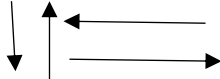
Simbol	Elemen	Keterangan
	<i>Include</i>	Mengindikasikan bahwa seluruh proses dalam use case merupakan bagian dari use case lainnya yang lebih besar.
	<i>Extend</i>	Menggambarkan ekstensi atau perluasan fungsionalitas yang terjadi dalam kondisi tertentu pada <i>use case</i> lain.

Sumber : (Gunawan et al., 2023)

#### b) *Activity diagram*

Menurut Sabarudin et al. (2022), *activity diagram* berfungsi untuk menggambarkan *workflow* (alur kerja) atau aktivitas dari suatu sistem maupun proses bisnis secara visual. 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. 4 Activity diagram

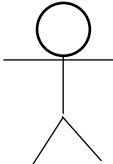
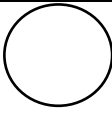

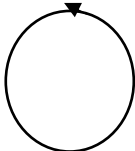
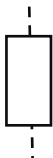

Simbol	Elemen	Keterangan
	<i>Activity</i>	Menunjukkan interaksi antar kelas atau komponen antarmuka dalam suatu proses sistem.
	<i>Action</i>	Merepresentasikan kondisi sistem yang menggambarkan pelaksanaan suatu tindakan tertentu.
	<i>Initial Node</i>	Simbol yang menunjukkan permulaan dari alur aktivitas dalam diagram.
	<i>Activity Final Node</i>	Simbol yang menandakan akhir dari jalannya proses dalam diagram aktivitas.
	<i>Decision</i>	Digunakan untuk menyatakan pilihan tindakan berdasarkan kondisi tertentu.
	<i>Line Connector</i>	Menggambarkan koneksi antara satu simbol dengan simbol lainnya dalam alur aktivitas.

Sumber : (Gunawan et al., 2023)

c) *Sequence diagram*

Menurut Mahardika et al. (2023), *Sequence Diagram* berfungsi untuk memvisualisasikan interaksi antar objek dalam sebuah sistem berdasarkan alur waktu. Diagram ini menggambarkan urutan komunikasi antar 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. 5 *Sequence diagram*

Simbol	Elemen	Keterangan
	Aktor	Merepresentasikan pengguna atau pihak eksternal yang berinteraksi langsung dengan sistem.
	Kelas Entitas	Menunjukkan relasi atau keterkaitan yang dilakukan oleh entitas dalam sistem.
	Kelas Batas	Bertugas sebagai perantara komunikasi antara sistem dan lingkungan luar.
	Kelas Kontrol	Berperan sebagai pengatur alur proses antara kelas batas dan komponen internal seperti database.
	Fokus Kontrol & Garis Hidup	Menyimbolkan waktu dimulainya hingga berakhirnya interaksi atau pengiriman pesan pada objek tertentu.
	Pesan	Menjelaskan proses pertukaran informasi atau instruksi antar objek dalam sistem.



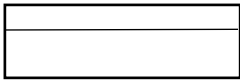

Sumber : (Gunawan et al., 2023)

#### d) *Class diagram*

Menurut (Ramdany, 2024), class diagram merupakan representasi visual dari hubungan antar kelas beserta penjelasan detail setiap kelas dalam model desain suatu sistem. Diagram ini memperlihatkan struktur logis dari sistem perangkat lunak yang akan dibangun, serta menggambarkan aturan-aturan dan tanggung jawab

dari masing-masing entitas yang memengaruhi perilaku sistem. 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. 6 Tabel *Class diagram*

Simbol	Elemen	Keterangan
	Garis lurus ( <i>Generalization</i> )	Menunjukkan relasi antara kelas turunan (anak) dengan kelas induk, mencakup perilaku dan struktur data yang diwariskan.
	<i>Nary Association</i>	Digunakan untuk menggambarkan hubungan yang melibatkan lebih dari dua objek dalam sistem.
	<i>Class</i>	Representasi kumpulan objek dalam sistem yang memiliki atribut dan metode tertentu.
	<i>Collaboration</i>	Urutan aksi-aksi dalam Sistem.

Simbol	Elemen	Keterangan
←-----	<i>Realization</i>	Menjelaskan fungsi atau aksi yang dijalankan oleh suatu objek dalam sistem.
-----→	<i>Dependency</i>	Menggambarkan relasi saling bergantung antara satu kelas dengan kelas lainnya.
—————	<i>Association</i>	Menghubungkan objek yang satu dengan yang lainnya.

Sumber : (Gunawan et al., 2023)

## 10. Black Box Testing

Black Box Testing, yaitu salah satu pendekatan pengujian fungsional yang berfokus pada evaluasi keluaran sistem berdasarkan berbagai masukan yang diberikan, tanpa memerhatikan struktur internal kode program atau logika pemrograman yang digunakan. Pada tahap ini, seluruh fitur dan fungsi sistem yang sebelumnya telah dideskripsikan dalam use case diagram diuji secara menyeluruh oleh tim peneliti dengan pendekatan black-box. Setiap fungsi diberikan masukan yang sesuai, dan hasilnya diamati serta dibandingkan dengan keluaran yang diharapkan (Kusuma et al., 2021).

### B. Kajian Empiris

Penelitian ini disusun dengan mengacu pada sejumlah referensi jurnal yang relevan dengan topik mengenai sistem pendukung keputusan seleksi penerima beasiswa berbasis metode *Multi Attribute Utility Theory (MAUT)*. Referensi tersebut dijadikan sebagai dasar acuan yang disusun secara sistematis guna mendukung proses pengembangan sistem. Studi literatur dari penelitian sebelumnya dimanfaatkan sebagai landasan untuk

memperkuat serta menyempurnakan rancangan sistem yang dibangun, melalui proses analisis perbandingan dan evaluasi terhadap sistem-sistem serupa yang telah dikembangkan pada penelitian terdahulu.

Penelitian yang dilakukan oleh Lumbanbatu et al. (2022), membahas penerapan metode *Multi Attribute Utility Theory* (MAUT) dengan pembobotan *Rank Order Centroid* (ROC) dalam penentuan calon mahasiswa penerima Kartu Indonesia Pintar (KIP). Penelitian ini bertujuan mengatasi kendala proses seleksi manual yang kurang efisien dan rawan subjektivitas akibat banyaknya mahasiswa dengan kriteria yang serupa. Dengan menggunakan empat kriteria utama, yaitu prestasi akademik, jumlah tanggungan orang tua, pekerjaan orang tua, dan penghasilan orang tua. Hasil akhir menunjukkan bahwa mahasiswa bernama Johannes Pardede (A1) memperoleh skor utilitas tertinggi sebesar 0,7737 dan direkomendasikan sebagai penerima beasiswa.

Penelitian yang dilakukan Indriyani et al. (2024). melakukan penelitian yang berfokus pada pemodelan prediksi penerima Beasiswa KIP-Kuliah dengan menggunakan metode *Weight Product*. Dalam studi ini, variabel-variabel yang digunakan mencakup faktor-faktor seperti pekerjaan dan penghasilan orang tua, kepemilikan kartu tertentu, status kendaraan dan rumah, luas tempat tinggal, serta fasilitas MCK. Seluruh data dianalisis melalui pendekatan statistik dan teknik pembobotan untuk membentuk model prediksi yang akurat. Proses pembobotan dilakukan dengan menetapkan bobot relatif pada masing-masing kriteria sesuai tingkat

pengaruhnya terhadap keputusan penerimaan. Berdasarkan hasil analisis, sebanyak 74 calon mahasiswa direkomendasikan sebagai penerima beasiswa KIP dan 12 tidak direkomendasikan dikarenakan kurang sesuai dengan kriteria untuk calon penerima beasiswa KIP.

Penelitian yang dilakukan Hasanah et al. (2024), mengkaji penerapan metode SMART (*Simple Multi Attribute Rating Technique*) dalam sistem pendukung keputusan (SPK) untuk menentukan penerima beasiswa KIP-Kuliah di ITS Paluta. Latar belakang penelitian ini adalah proses seleksi beasiswa yang masih bersifat manual dan kurang efisien, serta belum adanya sistem berbasis digital yang mendukung pengambilan keputusan secara objektif. Dalam studi ini, digunakan lima kriteria utama, yaitu jumlah tanggungan orang tua, penghasilan orang tua, prestasi akademik, kelengkapan berkas, dan prestasi non-akademik. Hasil akhir menunjukkan bahwa mahasiswa dengan skor tertinggi berdasarkan nilai utility total, seperti Irma Hasanah dengan nilai 0,90, menjadi prioritas utama untuk menerima beasiswa.

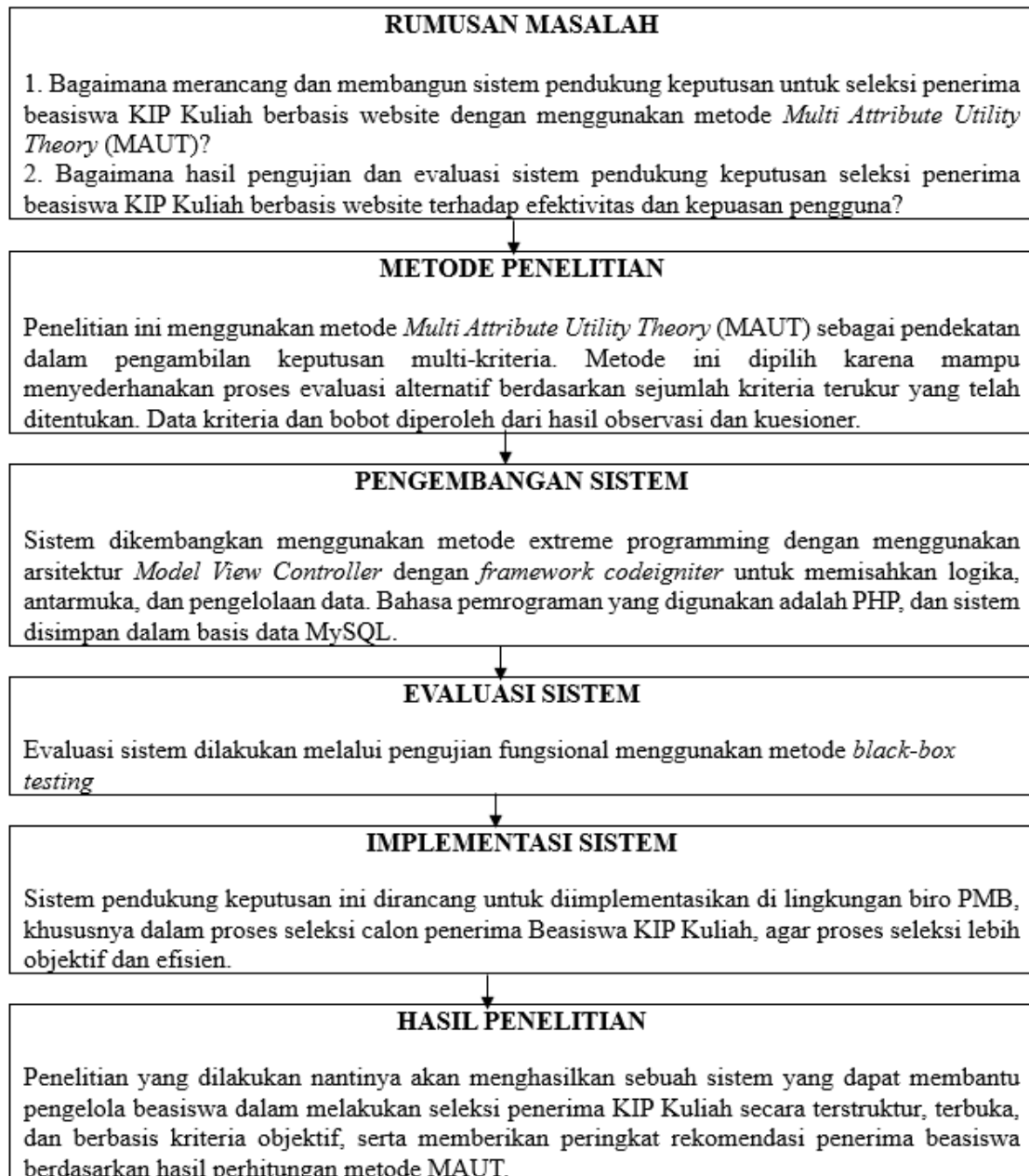
Penelitian yang dilakukan oleh Botutihe & Biki (2023), bertujuan membangun sistem pendukung keputusan (SPK) berbasis metode *Multi Attribute Utility Theory* (MAUT) untuk membantu sekolah dalam melakukan seleksi awal penerima beasiswa Program Indonesia Pintar (PIP) secara cepat dan objektif. Hasil penelitian menunjukkan bahwa metode MAUT mampu menghasilkan peringkat alternatif yang objektif, di mana Abdul Malik Kadili memperoleh nilai tertinggi sebesar 0,75, menunjukkan

bahwa ia paling layak sebagai penerima beasiswa. Pengujian sistem menggunakan *Black Box* menghasilkan sistem yang berfungsi sesuai dengan harapan.

Penelitian yang dilakukan oleh Kusuma et al. (2021), mengungkapkan bahwa penerapan metode *extreme programming* dalam pengembangan sistem pendukung keputusan mampu meningkatkan efisiensi dan fleksibilitas proses pembangunan sistem. Selain itu, hasil dari pengujian *black box testing* menunjukkan bahwa sistem yang dikembangkan bebas dari kesalahan, dengan semua fungsi dalam aplikasi berjalan dengan baik saat diuji, serta mendapatkan respons positif dari pengguna.

Dari beberapa sumber literatur diatas, maka penggunaan metode yang tepat dalam proses pembangunan sistem sangat diperlukan. Pada proses pembangunan ini, peneliti memilih menggunakan metode *extreme programming* dengan menggunakan arsitektur *Model View Controller* dengan *framework codeigniter*. Selain itu, metode *Multi Attribute Utility Theory* (MAUT) dipilih karena dianggap sesuai untuk kebutuhan sistem pendukung keputusan (SPK), kemudian dalam pengujiannya dilakukan melalui pendekatan pengujian fungsional menggunakan metode *black-box testing*, sehingga menghasilkan sistem yang lebih terstruktur, dan sesuai dengan kebutuhan pengguna.

### C. Kerangka Berfikir



Gambar 2. 1 Kerangka Berfikir