

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem

Menurut Ahmad & Hasti (2018), sistem merupakan serangkaian prosedur yang saling terkait dan bekerja sama dalam melakukan suatu aktivitas atau mencapai tujuan tertentu.

Menurut Sidh (2013), sebuah sistem terdiri dari tiga komponen pokok: *input*, proses, dan *output*. *Input* adalah bagian yang memberikan dorongan atau tenaga untuk menjalankan sistem, sedangkan *output* adalah hasil akhir dari proses tersebut. Secara sederhana, *output* adalah tujuan atau hasil yang diinginkan dari operasi sistem, sementara proses merujuk pada kegiatan yang mengubah *input* menjadi *output*.

Kesimpulan dari kedua pendapat tersebut adalah, sistem merupakan suatu entitas yang terdiri dari prosedur-prosedur yang saling terkait, memanfaatkan *input* untuk melakukan proses tertentu, dan menghasilkan *output* untuk mencapai tujuan yang ditetapkan.

2. Sistem Pendukung Keputusan (SPK)

Turban dkk. (2005) mendefinisikan sistem pendukung keputusan sebagai sistem yang diciptakan untuk membantu para pengambil keputusan dalam menghadapi situasi di mana keputusan yang harus diambil tidak sepenuhnya terstruktur. Sistem ini dirancang sebagai alat untuk meningkatkan kemampuan pengambil keputusan tanpa

menggantikan proses evaluasi yang telah mereka lakukan.

Manurung (2018) berpendapat bahwa sistem pendukung keputusan bertujuan untuk meningkatkan kemampuan para pengambil keputusan dengan memberikan bantuan, tetapi tidak menggantikan evaluasi yang mereka lakukan secara langsung.

Dari kedua pendapat tersebut dapat disimpulkan bahwa Sistem Pendukung Keputusan (SPK) adalah sebuah sistem yang dirancang untuk membantu para pengambil keputusan dalam menghadapi situasi di mana keputusan tidak sepenuhnya terstruktur. SPK bertujuan untuk meningkatkan kemampuan pengambil keputusan dengan memberikan bantuan, namun tidak bertujuan menggantikan proses penilaian yang dilakukan oleh mereka secara langsung.

3. Karyawan

Menurut Wiliandari (2019), karyawan merupakan aset utama bagi setiap organisasi atau perusahaan, yang memiliki peran sebagai perencana, pelaksana, dan pengendali dalam mencapai tujuan perusahaan. Karyawan merupakan makhluk sosial yang berperan aktif dalam mewujudkan visi perusahaan.

Septiana & Widjaja (2020) berpendapat bahwa karyawan adalah elemen kunci dalam sebuah organisasi yang tidak hanya merencanakan tetapi juga secara aktif terlibat dalam semua kegiatan organisasional. Mereka membawa beragam pikiran, perasaan, keinginan, dan latar

belakang yang beragam dalam hal pendidikan, usia, dan jenis kelamin. Hal tersebut membuat mereka berbeda dengan aset seperti mesin, uang, dan material yang bersifat pasif dan dapat sepenuhnya dikendalikan untuk mendukung pencapaian tujuan organisasi.

Dari kedua pandangan tersebut, dapat disimpulkan bahwa karyawan merupakan aset yang sangat berharga dan elemen kunci dalam sebuah organisasi atau perusahaan. Mereka memainkan peran krusial sebagai perencana, pelaksana, dan pengendali dalam mencapai tujuan perusahaan. Selain itu, karyawan dianggap sebagai individu sosial yang berperan aktif dalam mewujudkan visi perusahaan. Mereka membawa berbagai pemikiran, perasaan, keinginan, dan latar belakang yang berbeda, menjadikan mereka berbeda dengan aset lainnya seperti mesin, uang, dan bahan baku.

4. Penerimaan Karyawan

Menurut Taufiq dkk. (2018), penerimaan karyawan merupakan suatu proses seleksi dalam perusahaan untuk menemukan individu yang cocok untuk suatu posisi tertentu. Hal tersebut bertujuan agar karyawan yang masuk di perusahaan mampu bekerja secara optimal dan mempertahankan keanggotaannya dalam perusahaan dalam jangka waktu yang panjang.

Menurut Widowati & Agustin (2021), penerimaan karyawan adalah langkah-langkah untuk mencari, menemukan, mengundang, dan

menempatkan sejumlah individu dari dalam atau luar perusahaan sebagai calon tenaga kerja, dengan memperhatikan karakteristik yang telah ditetapkan dalam perencanaan sumber daya manusia.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa penerimaan karyawan merupakan suatu proses seleksi yang bertujuan untuk menemukan individu yang cocok untuk suatu posisi tertentu dalam perusahaan. Tujuan utamanya adalah agar karyawan yang diterima mampu bekerja secara optimal dan mempertahankan keanggotaannya dalam perusahaan dalam jangka waktu yang panjang. Langkah-langkah dalam proses penerimaan karyawan mencakup pencarian, penemuan, undangan, dan penempatan sejumlah individu dari dalam atau luar perusahaan, dengan memperhatikan karakteristik yang telah ditetapkan dalam perencanaan sumber daya manusia.

5. *Simple Additive Weighting (SAW)*

Afshari dkk. (2010) menggambarkan *Simple Additive Weighting (SAW)*, yang juga dikenal sebagai metode kombinasi linier berbobot atau metode *scoring*, sebagai teknik sederhana dalam pengambilan keputusan multi-atribut yang sering digunakan. Dalam metode ini, setiap alternatif dinilai dengan mengalikan nilai atributnya dengan bobot kepentingan relatif yang ditetapkan oleh pengambil keputusan, kemudian hasilnya dijumlahkan untuk semua kriteria. Keuntungan utama dari metode ini adalah bahwa metode ini secara proporsional mentransformasikan data

asli, sehingga urutan relatif skor yang dinormalisasi tetap konsisten.

Menurut Meri (2020), *Simple Additive Weighting* (SAW) adalah algoritma yang dikenal sebagai metode penjumlahan berbobot. Algoritma ini melibatkan normalisasi matriks keputusan (x) untuk menghasilkan skala yang memungkinkan perbandingan antara semua nilai alternatif yang dinilai.

Berdasarkan kedua pendapat tersebut, maka dapat disimpulkan bahwa *Simple Additive Weighting* (SAW) adalah sebuah teknik pengambilan keputusan multi-atribut yang sederhana dan umum digunakan. Metode ini melibatkan penilaian alternatif dengan mengalikan nilai atribut dengan bobot kepentingan relatif, kemudian menjumlahkan hasilnya untuk semua kriteria. Salah satu keuntungan utama SAW adalah transformasi linier proporsional dari data asli, yang menjaga urutan relatif skor yang dinormalisasi tetap sama. Metode ini juga memerlukan tahap normalisasi matriks keputusan agar dapat membandingkan semua penilaian alternatif yang ada.

Misalkan terdapat m alternatif dan n kriteria dalam pengambilan keputusan multi-atribut. Setiap alternatif i dinilai menggunakan vektor keputusan $V_i = (x_{i1}, x_{i2}, \dots, x_{in})$, di mana x_{ij} adalah nilai dari kriteria j untuk alternatif i . Langkah-langkah untuk menggunakan metode SAW adalah sebagai berikut:

a. Normalisasi Matriks Keputusan

Normalisasi pada rumus (1) dilakukan untuk mengubah setiap nilai x_{ij} ke dalam skala yang dapat dibandingkan dengan kriteria lainnya. Misalkan r_{ij} adalah nilai yang sudah dinormalisasi dari x_{ij} .

$$r_{ij} = \frac{x_{ij} - \min(x_{1j}, x_{2j}, \dots, x_{mj})}{\max(x_{1j}, x_{2j}, \dots, x_{mj}) - \min(x_{1j}, x_{2j}, \dots, x_{mj})} \quad (1)$$

Di sini, $\min(x_{1j}, x_{2j}, \dots, x_{mj})$ adalah nilai terkecil dari kriteria j , dan $\max(x_{1j}, x_{2j}, \dots, x_{mj})$ adalah nilai terbesar dari kriteria j .

b. Penetapan Bobot

Setelah matriks keputusan dinormalisasi, setiap kriteria diberikan bobot relatif w_j yang mencerminkan pentingnya kriteria tersebut dalam pengambilan keputusan.

c. Perhitungan Skor Alternatif

Skor S_i untuk setiap alternatif i dihitung dengan menjumlahkan hasil perkalian nilai yang sudah dinormalisasi dengan bobot relatif pada rumus (2):

$$S_i = \sum_{j=1}^n w_j \times r_{ij} \quad (2)$$

di mana w_j adalah bobot kepentingan untuk kriteria j , dan x_{ij} adalah nilai ter-normalisasi dari alternatif i terhadap kriteria j .

d. Pemilihan Alternatif Terbaik

Alternatif yang memiliki skor S_i tertinggi dianggap sebagai

alternatif terbaik sesuai dengan kriteria yang ditetapkan.

6. *Website*

Prasetyo (2015) menjelaskan bahwa sebuah *website* atau situs web adalah hasil dari implementasi bahasa pemrograman web, yang terdiri dari serangkaian halaman yang mengandung berbagai informasi seperti teks, gambar diam atau bergerak, animasi, suara, video, atau kombinasi dari semua elemen tersebut. Konten-konten dalam *website* dapat berupa statis atau dinamis, dan saling terhubung membentuk sebuah struktur yang terdiri dari halaman-halaman yang terkoneksi menggunakan *hyperlink*.

Rochmawati (2019) menyatakan bahwa situs web merupakan kumpulan informasi yang terdiri dari halaman-halaman web yang saling terkoneksi, yang disediakan oleh individu, kelompok, atau organisasi. Keunggulan sebuah situs web yang baik dapat dilihat dari aspek tampilan visual yang menarik dan fungsional sesuai dengan kebutuhan pengguna.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa situs web adalah hasil dari implementasi bahasa pemrograman web yang terdiri dari berbagai halaman yang mengandung teks, gambar, animasi, suara, dan video. Situs web dapat memiliki karakteristik statis atau dinamis dan terhubung melalui *hyperlink*. Tujuan utama dari situs web adalah untuk menyediakan informasi kepada pengguna. Kualitas yang baik dari sebuah situs web mencakup tampilan visual yang menarik dan berfungsi dengan baik sesuai dengan kebutuhan pengguna.

7. *HyperText Markup Language (HTML)*

Sari dkk. (2019:9) berpendapat bahwa HTML adalah sebuah bahasa pemrograman yang digunakan untuk menampilkan halaman web. Sebagai bahasa pemrograman gratis, HTML tidak dimiliki oleh individu tertentu, melainkan dikembangkan secara kolaboratif oleh banyak orang dari berbagai negara. Hal ini mencerminkan kerjasama global dalam pengembangan bahasa tersebut.

Mukhlis dkk. (2023:17) berpendapat bahwa HTML, yang disingkat dari *HyperText Markup Language*, adalah bahasa *markup* yang menjadi standar dalam pembuatan dan desain halaman web. Tim Berners-Lee mengembangkannya pada tahun 1991, dan sejak itu, HTML telah menjadi dasar dari struktur konten di web. Seiring berjalannya waktu, HTML telah mengalami perkembangan dan berbagai versi, dengan versi terkini adalah HTML5. Fungsi utama HTML adalah untuk menentukan kerangka dasar dari sebuah halaman web, serta menentukan cara elemen-elemen dalam halaman tersebut ditampilkan dan berinteraksi satu sama lain. HTML bekerja dengan menggunakan *tag* dan atribut, di mana *tag* merupakan elemen-elemen dasar yang membentuk struktur halaman web.

Kesimpulan dari kedua pendapat tersebut adalah bahwa HTML merupakan sebuah bahasa pemrograman yang digunakan untuk menampilkan halaman web. Sebagai bahasa pemrograman gratis, HTML dikembangkan secara kolaboratif oleh banyak orang dari berbagai negara,

mencerminkan kerjasama global dalam pengembangan bahasa tersebut. HTML, singkatan dari *HyperText Markup Language*, adalah bahasa *markup* standar dalam pembuatan dan desain halaman web. Dikembangkan oleh Tim Berners-Lee pada tahun 1991, HTML telah menjadi dasar dari struktur konten di web. Perkembangan HTML telah menghasilkan berbagai versi, dengan versi terkini adalah HTML5. Fungsi utama HTML adalah menentukan kerangka dasar sebuah halaman web dan cara elemen-elemen dalam halaman tersebut ditampilkan dan berinteraksi satu sama lain menggunakan *tag* dan atribut sebagai elemen dasarnya.

8. *HyperText Preprocessor* (PHP)

Menurut Santoso (2022:27), PHP atau singkatan dari *HyperText Preprocessor* adalah bahasa pemrograman web yang pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994.

Menurut Siswanto (2021:1), PHP adalah bahasa pemrograman server-side yang digunakan untuk mengembangkan berbagai jenis situs web, termasuk situs web statis, dinamis, dan aplikasi web. PHP merupakan singkatan dari *Hypertext Pre-processor*, sebelumnya dikenal sebagai *Personal Home Pages*. Sebagai bahasa pemrograman *server-side*, *script* atau program PHP dieksekusi dan diproses oleh *server*, berbeda dengan JavaScript yang berjalan di sisi klien.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa PHP adalah bahasa pemrograman web yang awalnya dikembangkan oleh

Rasmus Lerdorf pada tahun 1994. PHP merupakan singkatan dari *Hypertext Pre-processor* dan merupakan bahasa pemrograman *server-side* yang digunakan untuk mengembangkan berbagai jenis situs web, termasuk yang statis, dinamis, dan aplikasi web. *Script* atau program PHP dieksekusi dan diproses oleh *server*, berbeda dengan JavaScript yang berjalan di sisi klien.

9. Basis Data

Menurut Fikry (2019:2), basis data adalah kumpulan data yang saling terhubung dan diatur sesuai dengan skema atau struktur tertentu. Data-data tersebut disimpan di dalam perangkat keras komputer dan dimanipulasi menggunakan perangkat lunak untuk melakukan berbagai operasi seperti pembaruan, pencarian, pengolahan dengan perhitungan khusus, dan penghapusan, semuanya dilakukan dengan tujuan tertentu.

Prayoga dkk. (2023:1) berpendapat bahwa basis data adalah rangkaian data yang tersusun secara terstruktur dan disimpan secara elektronik. Sistem ini berfungsi untuk menyimpan, mengelola, dan mengambil informasi dengan efisien. Basis data umumnya digunakan dalam berbagai situasi, mulai dari lingkungan bisnis dan organisasi hingga aplikasi web dan perangkat lunak.

Berdasarkan kedua pandangan tersebut, dapat disimpulkan bahwa basis data adalah koleksi data yang terkait satu sama lain dan tersusun sesuai dengan struktur atau skema tertentu. Data ini disimpan di

perangkat keras komputer dan dikelola menggunakan perangkat lunak untuk melakukan berbagai manipulasi seperti pembaruan, pencarian, pengolahan dengan perhitungan khusus, dan penghapusan, semuanya dilakukan untuk tujuan tertentu. Basis data juga dapat diartikan sebagai sekumpulan data yang terstruktur dan disimpan secara elektronik, berfungsi untuk menyimpan, mengelola, dan mengambil informasi dengan efisien. Penggunaan basis data meluas dalam berbagai konteks, dari lingkungan bisnis dan organisasi hingga aplikasi web dan perangkat lunak.

10. MySQL

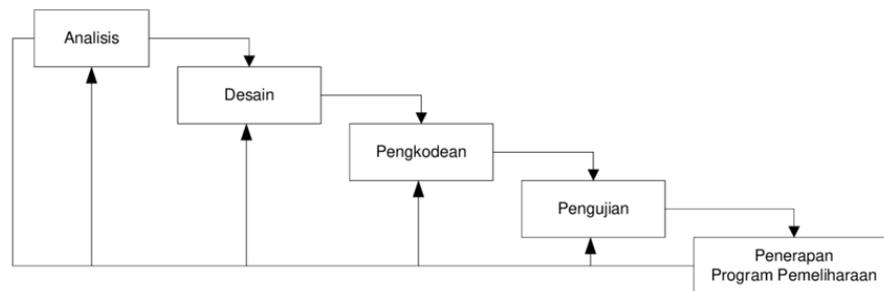
Menurut Risawandi (2019:49), MySQL adalah sebuah platform manajemen basis data *open-source* yang sedang populer saat ini. Data dalam MySQL disimpan dalam objek basis data yang disebut tabel. Tabel terdiri dari berbagai entri yang terhubung satu sama lain, ditampilkan dalam bentuk kolom dan baris.

Menurut Rawat dkk. (2021), MySQL adalah salah satu sistem manajemen basis data yang gratis dan bersifat *open-source*. MySQL mendukung berbagai platform seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, dan lainnya. MySQL memungkinkan pengguna untuk menjalankannya secara simultan tanpa mengalami masalah atau konflik.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa MySQL adalah platform manajemen basis data yang populer, *open-source*,

dan tersedia secara gratis. Data dalam MySQL disimpan dalam tabel yang terhubung melalui kolom dan baris. MySQL dapat dijalankan pada berbagai platform seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, dan Amiga tanpa mengalami masalah atau konflik saat digunakan secara bersamaan.

11. Metode Pengembangan *Waterfall*



Gambar 2. 1 Metode Pengembangan *Waterfall*

Sumber: Pressman (dalam Wahyudi & Aristantia, 2017)

Badrul & Ardy (2021) menjelaskan bahwa metode pengembangan *waterfall* adalah pendekatan dalam pengembangan perangkat lunak yang memungkinkan proses pembuatan sistem dilakukan secara terstruktur dan sistematis, mengikuti urutan yang telah ditentukan sesuai dengan siklus pengembangan yang telah ditetapkan.

Menurut Firzatullah (2021), metode pengembangan *waterfall* adalah suatu metode dalam pengembangan perangkat lunak yang dijalankan secara terstruktur dan terencana, memastikan proses pembangunan sistem informasi mengikuti alur yang jelas. Analoginya,

proses pengembangan ini seperti air terjun, di mana setiap tahapannya dilaksanakan berurutan dari atas ke bawah.

Berdasarkan pendapat di atas, dapat disimpulkan bahwa metode pengembangan *waterfall* adalah pendekatan sistematis dalam pengembangan perangkat lunak. Proses pembuatan sistem mengikuti urutan yang telah ditentukan sesuai dengan siklus pengembangan yang ada. Analoginya, metode ini mirip dengan aliran air terjun, di mana setiap tahapannya dilaksanakan secara berurutan dari atas ke bawah, memastikan jalur yang jelas dalam proses pembangunan sistem informasi.

Berikut merupakan tahapan dari metode *waterfall*:

a. Analisis Kebutuhan

Tahap analisis kebutuhan adalah awal dari proses rancang bangun pada perangkat lunak. Pada tahap ini, tim pengembang berinteraksi dengan pelanggan atau pemangku kepentingan untuk memahami dan mendefinisikan kebutuhan sistem yang akan dikembangkan. Tujuan utama dari tahap ini adalah untuk mengidentifikasi apa yang harus dicapai oleh sistem yang akan dibangun. Hasil dari tahap ini adalah dokumen spesifikasi kebutuhan yang akan digunakan sebagai panduan selama sisa siklus pengembangan.

b. Perancangan Sistem

Setelah kebutuhan sistem teridentifikasi, tahap berikutnya

adalah merancang struktur dan arsitektur sistem secara keseluruhan. Perancangan sistem mencakup perencanaan solusi untuk memenuhi kebutuhan yang telah ditentukan pada tahap analisis. Hal ini mencakup pemilihan platform, pemodelan data, arsitektur perangkat lunak, dan rancangan antarmuka pengguna. Hasil dari tahap ini adalah dokumen rancangan sistem yang mendetail.

c. Pembuatan Sistem / Pengkodean

Tahap ini merupakan tahap di mana pengembangan perangkat lunak sebenarnya dilakukan. Berdasarkan rancangan sistem yang telah disetujui, kode program dan komponen-komponen sistem disusun. Tim pengembang menulis kode, mengintegrasikan komponen-komponen, dan mengimplementasikan sistem secara keseluruhan sesuai dengan spesifikasi yang telah ditentukan.

d. Pengujian

Setelah sistem selesai dibuat, tahap pengujian dilakukan untuk memastikan bahwa sistem bekerja sesuai dengan yang diharapkan dan memenuhi semua persyaratan yang telah ditentukan pada tahap analisis. Pengujian meliputi pengujian fungsional untuk memeriksa apakah sistem berperilaku seperti yang diharapkan, serta pengujian perhitungan *Simple Additive Weighting* agar memastikan bahwa perhitungan berjalan dengan akurat.

e. Implementasi / Penerapan

Tahap terakhir dalam metode *waterfall* adalah implementasi atau penerapan sistem. Pada tahap ini, sistem yang telah melewati pengujian dan disetujui diimplementasikan ke lingkungan produksi atau dioperasikan secara nyata. Implementasi mencakup instalasi perangkat lunak, konfigurasi sistem, dan peluncuran operasional agar sistem siap digunakan oleh pengguna akhir.

12. *Unified Modeling Language (UML)*

Hasbid dkk. (2021) berpendapat bahwa UML adalah notasi yang digunakan untuk merancang desain yang berorientasi objek. Beberapa contoh diagram UML termasuk diagram *use case*, *class diagram*, *activity diagram* dan *sequence diagram*.

Menurut Voutama (2022), UML digunakan untuk mempermudah perancangan sistem dan mengurangi kesalahan dalam pengembangan program. Dengan menerapkan UML, *engineer* dapat menggambarkan siapa yang terlibat dalam sistem, apa yang dilakukan oleh setiap pihak, serta proses dan mekanisme yang ada dalam sistem yang sedang dikembangkan.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa UML adalah notasi yang digunakan untuk merancang desain yang berorientasi objek, dan beberapa contoh diagram UML meliputi diagram *use case*, *class diagram*, *activity diagram*, dan *sequence diagram*. UML juga digunakan untuk mempermudah perancangan sistem dan mengurangi

kesalahan dalam pengembangan program. Dengan menerapkan UML, pengembang dapat menggambarkan aktor yang terlibat dalam sistem, aktivitas yang dilakukan oleh setiap aktor, serta proses dan mekanisme yang ada dalam sistem yang sedang dikembangkan. Kesimpulannya, UML merupakan alat yang berguna dalam perancangan sistem dan pengembangan program karena dapat menyediakan notasi untuk merancang desain berorientasi objek serta memfasilitasi pemahaman tentang siapa yang terlibat dalam sistem dan bagaimana prosesnya berjalan.

13. *Flowchart*

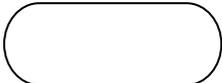
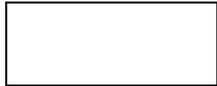
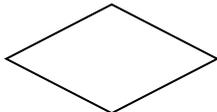
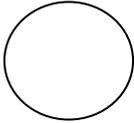
Menurut Jogiyanto (dalam Irviani & Oktaviana, 2017), *flowchart* merupakan gambaran visual yang menggambarkan urutan logis dari operasi dalam sebuah program atau prosedur sistem. Fungsinya adalah untuk memfasilitasi komunikasi dan dokumentasi dalam pengembangan sistem.

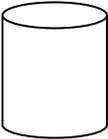
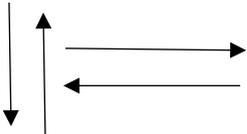
Abdurrahman (2017) berpendapat bahwa *flowchart* digunakan untuk mengilustrasikan langkah-langkah kegiatan di dalam suatu organisasi. *Flowchart* merupakan diagram yang menggambarkan secara visual seluruh sistem, termasuk aktivitas manual serta jalur aliran dokumen yang digunakan dalam sistem tersebut.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa *flowchart* adalah gambaran visual yang digunakan untuk menggambarkan

urutan logis dari operasi dalam sebuah program atau prosedur sistem. Fungsinya adalah untuk memfasilitasi komunikasi, dokumentasi, serta mengilustrasikan langkah-langkah kegiatan dalam suatu organisasi. *Flowchart* mencakup baik aktivitas manual maupun aliran dokumen yang terjadi dalam sistem tersebut.

Tabel 2. 1 Simbol *Flowchart*

No.	Simbol	Nama	Deskripsi
1		<i>Terminator</i>	Awal atau akhir dari alur proses.
2		<i>Process</i>	Aktivitas atau proses yang dilakukan dalam <i>flowchart</i> .
3		<i>Input / Output Data</i>	<i>Input</i> atau <i>output</i> data yang digunakan atau dihasilkan dalam proses.
4		<i>Decision</i>	Simbol yang menunjukkan titik dalam <i>flowchart</i> di mana keputusan harus dibuat, memiliki dua jalur berbeda tergantung pada kondisi yang ditentukan.
5		<i>On Page Connector</i>	Simbol yang digunakan untuk menghubungkan alur di dalam satu halaman <i>flowchart</i> yang besar atau kompleks.

6		<i>Database</i>	Simbol yang menunjukkan tempat penyimpanan atau basis data dalam <i>flowchart</i> .
7		Dokumen	Simbol yang mewakili dokumen atau informasi tertulis dalam <i>flowchart</i> .
8		<i>Flow Line</i>	Garis yang menghubungkan simbol-simbol dalam <i>flowchart</i> , menunjukkan alur atau urutan dari satu simbol ke simbol lainnya.

14. Use Case Diagram

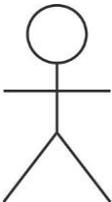
Djaelangkara dkk. (2015) menjelaskan bahwa *use case diagram* adalah kumpulan deskripsi atau skenario yang terhubung membentuk sebuah sistem terstruktur, yang dilakukan atau diawasi oleh pengguna. Diagram *use case* digunakan untuk mengilustrasikan perilaku objek atau entitas dalam suatu model, dan implementasinya melibatkan kerja sama antara elemen-elemen tersebut. Umumnya, *use case* digambarkan dalam bentuk elips dengan garis solid dan nama yang spesifik.

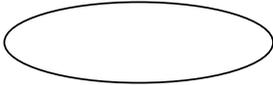
Artina (2006) mengungkapkan bahwa *use case* adalah pendekatan untuk mengidentifikasi kebutuhan fungsional dari sistem baru atau yang mengalami perubahan. Setiap *use case* terdiri dari serangkaian skenario yang menggambarkan interaksi antara sistem dengan pengguna atau sistem lainnya untuk mencapai tujuan bisnis tertentu. Pendekatan ini

tidak memperinci cara kerja internal sistem atau implementasinya, melainkan berfokus pada langkah-langkah yang dilakukan pengguna dalam menggunakan perangkat lunak.

Berdasarkan sumber-sumber yang disebutkan, dapat disimpulkan bahwa *use case diagram* merupakan representasi visual dari serangkaian deskripsi atau skenario yang saling terkait dalam suatu sistem. Diagram ini menggambarkan interaksi antara pengguna dengan elemen sistem lainnya, menunjukkan perilaku objek atau entitas dalam model tanpa memperhatikan detail implementasi internal sistem. Tujuan utama dari *use case* adalah untuk mengidentifikasi kebutuhan fungsional sistem serta menjelaskan interaksi antara pengguna dengan sistem tersebut untuk mencapai tujuan bisnis tertentu. *Use case diagram* umumnya mengilustrasikan *use case* sebagai elips dengan garis solid dan nama yang spesifik, dengan fokus pada langkah-langkah yang dilakukan pengguna dalam penggunaan perangkat lunak.

Tabel 2. 2 Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Merupakan peran orang, sistem yang lain, atau alat ketika berhubungan dengan <i>use case</i> .

Simbol	Nama	Keterangan
	<i>Use Case</i>	Mewakili fungsi atau aktivitas yang dapat dilakukan oleh satu atau lebih aktor dalam sistem.
	Asosiasi	Hubungan antara aktor dan kasus penggunaan yang menunjukkan interaksi antara pengguna (aktor) dengan fungsionalitas sistem.
	Generalisasi	Hubungan hierarkis di antara <i>use case</i> yang menunjukkan bahwa <i>use case</i> yang lebih spesifik mewarisi atau memperluas fungsionalitas dari <i>use case</i> yang lebih umum.
	<i>Include</i>	Hubungan di mana <i>use case</i> mencakup fungsionalitas dari <i>use case</i> lainnya dalam semua situasi.
	<i>Extend</i>	Hubungan di mana suatu <i>use case</i> dapat menambahkan fungsionalitas tambahan ke dalam <i>use case</i> utama hanya pada situasi tertentu atau opsional.

15. Class Diagram

Menurut Arribe dkk. (2023), *class diagram* adalah suatu representasi visual dari struktur suatu sistem, yang mencakup banyak kelas yang saling terhubung dalam struktur tersebut.

Menurut Wira dkk. (2019), *class diagram* adalah representasi visual dari struktur sistem yang menampilkan definisi kelas-kelas yang digunakan dalam pembangunan sistem. Diagram ini mencakup atribut dan operasi dari setiap kelas, yang memungkinkan pengembang perangkat

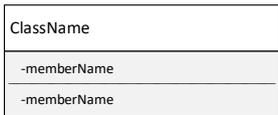
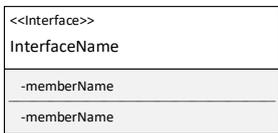
lunak untuk menetapkan hubungan antara berbagai elemen dalam dokumentasi desain dan implementasi perangkat lunak.

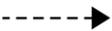
Dari kedua pendapat tersebut dapat disimpulkan bahwa *class diagram* memberikan representasi visual yang jelas tentang struktur suatu sistem dengan kelas-kelas yang saling terhubung, serta mendefinisikan atribut dan operasi yang memungkinkan pengembang program untuk merancang dan mengimplementasikan perangkat lunak dengan tepat.

Sebuah *class* memiliki tiga komponen utama, antara lain:

- a. Nama, identitas unik yang dimiliki oleh *class* tersebut.
- b. Atribut, karakteristik atau properti yang dimiliki oleh objek dari *class* tersebut, menentukan batasan nilai yang dapat dimiliki.
- c. Operasi, tindakan atau fungsi yang dapat dilakukan oleh *class* itu sendiri atau oleh kelas lain terhadap kelas tersebut.

Tabel 2. 3 Simbol *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	Representasi dari entitas yang memiliki atribut dan metode yang mendefinisikan perilaku entitas tersebut dalam sistem.
	<i>Interface</i>	Kumpulan dari metode tanpa implementasi yang dapat diimplementasikan oleh kelas lain.
	<i>Association</i>	Hubungan antara dua atau lebih kelas yang menunjukkan bahwa objek dari kelas satu dapat berhubungan dengan objek

Simbol	Nama	Keterangan
		dari kelas lain.
	<i>Indirect Association</i>	Hubungan antara dua kelas melalui kelas ketiga. Ketika dua kelas tidak berhubungan langsung tetapi memiliki ketergantungan melalui kelas lain, ini disebut sebagai hubungan asosiasi tidak langsung. Misalnya, kelas A memiliki hubungan asosiasi dengan kelas C, dan kelas C memiliki hubungan asosiasi dengan kelas B.
	<i>Generalization</i>	Hubungan hierarkis antara kelas-kelas yang menunjukkan bahwa kelas anak (<i>subclass</i>) mewarisi atribut dan metode dari kelas induk (<i>superclass</i>).
	<i>Aggregation</i>	Hubungan di mana kelas berisi objek dari kelas lain sebagai bagian dari dirinya sendiri.
	<i>Dependency</i>	<i>Dependency</i> menunjukkan bahwa satu kelas menggunakan layanan atau metode dari kelas lain tanpa memegang referensi ke objek tersebut.

16. Activity Diagram

Artina (2006) berpendapat bahwa *activity diagram* merupakan suatu diagram yang digunakan untuk merepresentasikan proses dan alur kerja yang dijelaskan dalam *use case diagram*. Biasanya, dalam suatu sistem, terdapat beberapa *activity diagram* yang menggambarkan berbagai

proses yang terjadi.

Arianti dkk. (2022) berpendapat bahwa *activity diagram* adalah gambaran visual dari aliran data atau kontrol, serta aksi-aksi terstruktur yang dirancang secara cermat dalam suatu sistem.

Dari kedua perspektif ini, dapat disimpulkan bahwa *activity diagram* digunakan untuk menggambarkan proses dan alur kerja dalam suatu sistem secara visual, dengan menyoroti aliran data, kontrol, dan aksi-aksi yang terstruktur.

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status awal	Titik awal dari aktivitas dalam <i>activity diagram</i> , menunjukkan dimulainya aliran aktivitas yang akan dilakukan oleh sistem atau objek yang diwakili.
	Aktivitas	Representasi dari tindakan atau aktivitas yang dilakukan oleh sistem atau objek dalam <i>activity diagram</i> , menunjukkan unit kerja yang spesifik atau langkah dalam proses yang sedang dilakukan.
	Percabangan / <i>Decision</i>	Titik dalam <i>activity diagram</i> di mana pilihan harus dibuat oleh sistem atau objek berdasarkan kondisi tertentu, menciptakan jalur alternatif dalam aliran aktivitas.
	Penggabungan / <i>Join</i>	Titik dalam <i>activity diagram</i> di mana jalur yang berbeda dari percabangan atau jalur paralel bergabung kembali menjadi satu aliran aktivitas tunggal.

Simbol	Nama	Keterangan
	Status Akhir	Titik akhir dari aliran aktivitas dalam <i>activity diagram</i> , menunjukkan bahwa proses atau aktivitas telah selesai atau mencapai tujuan yang diinginkan.
	<i>Swimlane</i>	Kompartemen horizontal atau vertikal dalam diagram aktivitas yang menunjukkan aktor atau entitas yang bertanggung jawab atas aktivitas tertentu, membantu dalam memodelkan interaksi dan tanggung jawab antar aktor atau unit organisasi dalam proses bisnis atau sistem.

17. Sequence Diagram

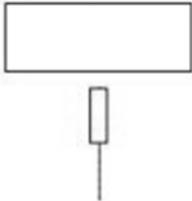
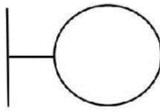
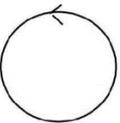
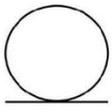
Arianti dkk. (2022) berpendapat bahwa *sequence diagram* adalah representasi visual yang menunjukkan bagaimana objek-objek saling berkomunikasi dan berinteraksi satu sama lain dalam konteks suatu kelas atau sistem.

Putri dkk. (2021) berpendapat bahwa *sequence diagram* adalah visualisasi dari hubungan antara objek-objek yang menunjukkan bagaimana komunikasi atau pesan ditukar antara objek tersebut.

Berdasarkan kedua pendapat tersebut, maka dapat disimpulkan bahwa *sequence diagram* adalah representasi visual yang menunjukkan bagaimana objek-objek dalam suatu kelas atau sistem berkomunikasi dan berinteraksi satu sama lain. *Sequence diagram* juga merupakan visualisasi dari hubungan antara objek-objek yang menunjukkan bagaimana

komunikasi atau pesan ditukar antara objek tersebut.

Tabel 2. 5 Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	Entitas di luar sistem yang berinteraksi dengan sistem dan berperan sebagai pengguna langsung dari sistem atau objek yang direpresentasikan dalam <i>sequence diagram</i> .
	<i>Lifeline</i>	Garis vertikal pada diagram urutan yang mewakili eksistensi objek atau entitas selama urutan waktu tertentu, menunjukkan kapan objek aktif dalam sistem.
	<i>General</i>	Kelas abstrak yang tidak dapat diinstansiasi langsung, tetapi dapat digunakan sebagai induk untuk kelas-kelas turunannya.
	<i>Boundary</i>	Kelas yang bertanggung jawab untuk menangani interaksi antara sistem dengan aktor eksternal atau elemen lain di luar sistem.
	<i>Control</i>	Kelas yang mengendalikan aliran dari suatu sistem, menangani logika aplikasi dan mengkoordinasikan antara kelas <i>Boundary</i> dan <i>Entity</i> .
	Entitas	Representasi dari objek atau data yang memiliki perilaku dan informasi yang relevan untuk sistem yang sedang dimodelkan.

Simbol	Nama	Keterangan
	<i>Activation</i>	Periode waktu di mana suatu objek atau entitas sedang aktif atau sedang melakukan operasi dalam urutan diagram.
	<i>Message Entry</i>	Pesan yang dikirim dari aktor atau objek luar ke dalam sistem, memulai urutan operasi atau interaksi.
	<i>Message to Self</i>	Pesan yang dikirim dari objek ke dirinya sendiri, menunjukkan bahwa objek tersebut memicu suatu operasi atau respons terhadap kejadian tertentu.
	<i>Message Return</i>	Pesan yang dikirim dari objek atau entitas kembali ke pengirim pesan awal setelah menyelesaikan operasi atau mengirim respons.

18. Pengujian *Black Box*

Supriyono (2020) berpendapat bahwa pengujian *black box* adalah strategi pengujian perangkat lunak yang mengoperasikan berdasarkan spesifikasi fungsionalnya. Salah satu teknik dalam pengujian *black box* adalah analisis nilai batas / *Boundary Value Analysis* (BVA), yang terutama memperhatikan rentang *input* di mana setiap nilai yang diuji berada di antara batas minimum dan maksimum yang ditentukan.

Menurut Kumar dkk. (2015), pengujian *black box* adalah metode pengujian yang tidak memperhatikan struktur internal komponen atau sistem. Dalam metode ini, *tester* tidak perlu memiliki pemahaman

mendalam tentang pemrograman karena hanya menguji fungsi dasar dari sistem tanpa memperhatikan detail teknisnya.

Dari kedua pendapat tersebut dapat disimpulkan bahwa pengujian *black box* adalah strategi pengujian perangkat lunak yang tidak memperhatikan struktur internal komponen atau sistem. Fokus dari pengujian tersebut adalah pengujian berdasarkan spesifikasi fungsional dan fungsi dasar dari sistem, tanpa memerlukan pemahaman mendalam tentang pemrograman dari penguji. Teknik seperti *Boundary Value Analysis* (BVA) sering digunakan dalam pengujian *black box*.

B. Kajian Empiris

Penelitian yang dilakukan oleh Somya & Wahyudi (2020) bertujuan untuk membantu PT Visionet Data Internasional dalam memperbaiki proses perekrutan karyawan melalui pengembangan sistem pendukung keputusan menggunakan metode TOPSIS. Dengan sistem ini, Admin HRD dapat memproses data calon karyawan baru berdasarkan kriteria yang telah ditetapkan perusahaan dan menentukan alternatif terbaik secara efisien. Hasil penelitian menunjukkan bahwa sistem tersebut berhasil memberikan solusi dalam menyediakan alternatif pengambilan keputusan saat perekrutan karyawan baru, namun disarankan untuk melakukan peningkatan pada tampilan, keamanan, dan fungsionalitas sistem dalam rangka peningkatan optimalisasi kerja.

Penelitian yang dilakukan oleh Nugroho & Purwanto (2015)

bertujuan untuk meningkatkan proses seleksi dan penerimaan karyawan di PT. Bank Rakyat Indonesia dengan memanfaatkan Sistem Pendukung Keputusan menggunakan metode *Profil Matching*. Metode ini mengevaluasi calon karyawan berdasarkan tiga aspek utama: Kapasitas Intelektual, Sikap Kerja, dan Perilaku. Hasilnya menunjukkan bahwa metode ini efektif dalam memberikan bobot nilai kepada setiap calon karyawan, memungkinkan manajemen untuk membuat keputusan penerimaan yang lebih obyektif dengan waktu yang lebih efisien. Selain itu, disarankan agar sistem ini jika ingin dikembangkan, maka dikembangkan dalam bentuk berbasis web untuk kemudahan penggunaan yang meningkatkan efektivitas waktu dalam proses penggunaannya.

Penelitian yang dilakukan oleh Sundari & Taufik (2014) bertujuan untuk meningkatkan profesionalisme dalam proses penerimaan pegawai baru di perusahaan dengan memanfaatkan Sistem Pendukung Keputusan berbasis metode *Simple Additive Weighting* (SAW). Metode SAW dipilih karena memungkinkan penilaian yang sistematis dengan menentukan bobot untuk setiap kriteria dan kemudian melakukan pengurutan terhadap calon pegawai baru. Hasil pengujian menunjukkan bahwa sistem yang dikembangkan dapat memudahkan dan mempercepat proses penyeleksian serta membantu manajer divisi Sumber Daya Manusia (SDM) dalam pengambilan keputusan yang lebih tepat dalam penerimaan pegawai baru.

Penelitian yang dilakukan oleh Satria (2023) bertujuan untuk

mengembangkan sebuah Sistem Pendukung Keputusan untuk proses penerimaan staf administrasi menggunakan metode VIKOR. Dalam penelitian ini, dilakukan analisis terhadap kriteria-kriteria yang digunakan dalam penilaian calon karyawan, serta pembuatan aplikasi penerimaan karyawan menggunakan *framework* Laravel 8 untuk memudahkan proses seleksi. Metode VIKOR digunakan untuk melakukan pengurutan terhadap calon karyawan, dengan hasil terdapat calon yang direkomendasikan sebagai kandidat utama untuk mengisi posisi staf administrasi. Selain itu, pengolahan data tanggapan responden memberikan hasil yang positif terhadap kepercayaan, risiko penggunaan, persepsi kegunaan, dan persepsi kemudahan penggunaan, dengan keseluruhan kriteria untuk penerimaan teknologi dinilai baik.

Penelitian yang dilakukan oleh Handayani & Muzakir (2018) bertujuan untuk mengembangkan sebuah Sistem Pendukung Keputusan untuk proses penerimaan karyawan di PT. Virtus Venturama menggunakan metode *Analytical Hierarchy Process* (AHP). Latar belakang penelitian ini adalah kesulitan dalam menentukan pelamar yang benar-benar berkompeten dan layak menjadi karyawan, serta kurangnya efisiensi dalam penempatan karyawan pada departemen yang tersedia. Dalam penelitian ini, AHP digunakan sebagai model untuk sistem pendukung keputusan, dengan hasil akhir menunjukkan prioritas penerimaan karyawan, di mana Karyawan E mendapat peringkat tertinggi dengan persentase 27,6%. Kesimpulan dari

penelitian ini menyatakan bahwa berdasarkan hasil analisis data, pelamar yang disebutkan memiliki peringkat tertentu sesuai dengan prioritas dalam penerimaan karyawan di PT. Virtus Venturama.

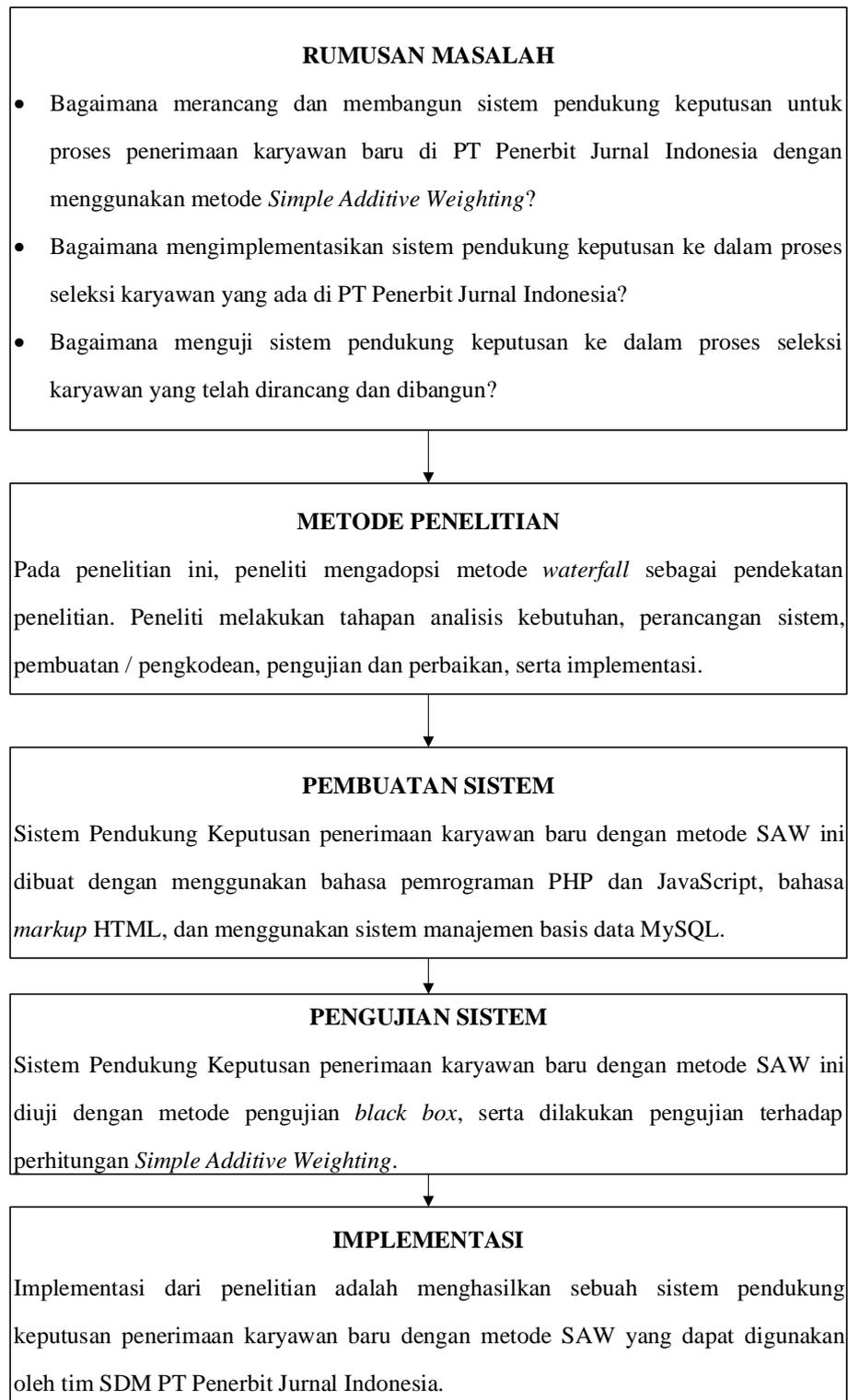
C. Kerangka Berpikir Penelitian

PT Penerbit Jurnal Indonesia, sebuah perusahaan penerbitan jurnal ilmiah di Indonesia, secara rutin membuka peluang bagi calon karyawan baru untuk berbagai posisi di berbagai departemen melalui tim Sumber Daya Manusia (SDM). Namun, proses penerimaan karyawan masih dilakukan secara manual. Pendekatan manual ini sering kali menimbulkan tantangan, seperti risiko kesalahan dan potensi kehilangan berkas calon karyawan.

Dalam penelitian ini, akan dikembangkan sebuah sistem pendukung keputusan menggunakan algoritma *Simple Additive Weighting* (SAW) dengan metode pengembangan *waterfall*. Prosesnya mencakup analisis kebutuhan, perancangan, implementasi, pengujian, integrasi, dan operasi. Sistem ini akan dibangun dengan menggunakan PHP dan HTML sebagai bahasa pemrograman, serta MySQL sebagai sistem manajemen basis data.

Hasil dari penelitian ini diharapkan dapat menghasilkan sistem pendukung keputusan untuk proses penerimaan karyawan baru dengan algoritma SAW. Sistem ini bertujuan untuk meningkatkan efisiensi dan objektivitas dalam proses seleksi karyawan, serta mengurangi risiko kesalahan dan potensi kehilangan dokumen pelamar.

Berdasarkan penjelasan yang telah dijabarkan, maka dapat dibuat kerangka berpikir sebagai berikut:



Gambar 2. 2 Kerangka Berpikir Penelitian