

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) merupakan salah satu bidang dalam ilmu komputer yang dirancang untuk membantu proses pengambilan keputusan, terutama ketika dihadapkan pada persoalan yang bersifat kompleks, tidak pasti, semi terstruktur, maupun tidak terstruktur. Sistem ini berperan dalam mengubah data menjadi informasi yang relevan dan berguna, sehingga memudahkan pengambil keputusan dalam memilih alternatif terbaik di antara berbagai kemungkinan yang tersedia (Ginting, dkk., 2020:1).

Menurut (Agarina et al., 2023), Sistem Pendukung Keputusan (SPK) dapat diartikan sebagai sebuah perangkat atau alat bantu yang dirancang secara khusus untuk membantu individu dalam melakukan proses penilaian atau pengambilan keputusan pada situasi tertentu.

Secara umum, SPK memiliki beberapa karakteristik atau persyaratan utama yang menunjang fungsinya, antara lain:

a. Interaktif

Sistem pendukung keputusan menyediakan antarmuka pengguna yang memfasilitasi komunikasi, memungkinkan pengguna mendapatkan data secara efisien dan memperoleh pengetahuan yang diperlukan.

b. Fleksibel

Memberikan keleluasaan bagi pengguna dalam memilih dan mengkombinasikan berbagai faktor atau alternatif yang tersedia dalam proses pengambilan keputusan.

c. Data Kualitas

Mampu menerima data input bersifat subjektif dari pengguna yang kemudian dapat diolah secara kuantitatif, misalnya dengan memberikan nilai tertimbang untuk aspek-aspek yang dinilai secara kualitatif.

d. Prosedur Pakar

Mampu menerima data input bersifat subjektif dari pengguna yang kemudian dapat diolah secara kuantitatif, misalnya dengan memberikan nilai tertimbang untuk aspek-aspek yang dinilai secara kualitatif.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa Sistem Pendukung Keputusan (SPK) merupakan alat bantu yang penting dalam proses pengambilan keputusan, terutama dalam kondisi yang kompleks dan tidak terstruktur. Dengan karakteristik interaktif, fleksibel, mampu mengolah data subjektif secara kuantitatif, serta didukung oleh prosedur pakar, SPK mampu membantu pengambil keputusan untuk memperoleh hasil yang lebih akurat, efisien, dan dapat dipertanggungjawabkan.

2. *Profile Matching*

Metode *Profile Matching* merupakan salah satu pendekatan yang umum digunakan dalam sistem pendukung keputusan (*Decision Support System*), di mana proses penilaiannya dilakukan dengan membandingkan profil kompetensi individu terhadap profil standar yang telah ditentukan sebelumnya. Tujuan utama dari metode ini adalah untuk menghitung selisih atau *gap* antara nilai aktual dengan nilai ideal dari setiap kriteria penilaian. Semakin kecil nilai *gap* yang diperoleh, maka semakin tinggi kesesuaian kompetensi individu dengan standar yang diharapkan (Gede Surya Mahendra et al., 2023). Pendapat tersebut diperkuat oleh (Harahap et al., 2024), yang menyatakan bahwa metode *Profile Matching* memberikan bobot berbeda pada setiap kriteria berdasarkan tingkat kepentingannya. Hal ini bertujuan agar sistem dapat menyesuaikan penilaian dengan tipe kebutuhan yang spesifik. Salah satu keunggulan dari metode ini adalah pengelompokan indikator ke dalam dua kategori utama, yaitu *Core Factor* (CF) atau faktor inti, dan *Secondary Factor* (SF) atau faktor pendukung, yang masing-masing memiliki kontribusi berbeda dalam proses penilaian.

Dari uraian tersebut dapat disimpulkan bahwa *Profile Matching* merupakan metode komparatif dalam sistem pendukung keputusan yang menilai kompetensi berdasarkan selisih antara nilai aktual dan nilai standar, serta memberikan pembobotan berbeda pada setiap kriteria guna menyesuaikan dengan tingkat prioritasnya. Metode ini efektif digunakan

dalam proses seleksi atau evaluasi karena memperhatikan keseimbangan antara keakuratan perhitungan dan relevansi kebutuhan sistem.

Langkah-langkah dalam metode *profile matching* (Harahap et al., 2024) sebagai berikut :

- a. Langkah pertama yaitu menentukan aspek-aspek penilaian atau indikator penilaian pada *Core Factor* (CF) dan *Secondary Factor* (SF).
- b. Langkah kedua yaitu menghitung *GAP* atau pemetaan *GAP* dengan rumus : $GAP = \text{Nilai Kriteria} - \text{Nilai Minimal}$.
- c. Pembobotan setelah pemetaan *GAP* sesuai dengan tabel bobot nilai *GAP* seperti berikut :

Tabel 2. 1 Bobot Nilai Kriteria

NO	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih
2	1	4,5	Kelebihan 1 tingkat
3	-1	4	Kekurangan 1 tingkat
4	2	3,5	Kelebihan 2 tingkat
5	-2	3	Kekurangan 2 tingkat
6	3	2,5	Kelebihan 3 tingkat
7	-3	2	Kekurangan 3 tingkat
8	4	1,5	Kelebihan 4 tingkat
9	-4	1	Kekurangan 4 tingkat

Sumber : (Harahap et al., 2024)

d. Perhitungan dan pengelompokkan *Core Factor* (CF) dan *Secondary Factor* (SF) seperti persamaan berikut.

a) Rumus untuk mencari *Core Factor* (CF) :

$$NCF = \frac{\sum NC (aspek)}{\sum IC}$$

Keterangan :

NCF = Nilai rata-rata *Core Factor* (CF)

NC (aspek) = Jumlah nilai dari indikator dalam *Core Factor*

IC = Jumlah item dalam *Core Factor*

b) Rumus untuk mencari *Secondary Factor* (SF) :

$$NSF = \frac{\sum NS (aspek)}{\sum IS}$$

Keterangan :

NSF = Nilai rata-rata *Secondary Factor* (SF)

NS (aspek) = Jumlah nilai dari indikator dalam *Secondary Factor*

IS = Jumlah item dalam *Secondary Factor*

e. Perhitungan nilai total. Berikut rumus untuk mencari nilai total :

$$(x)\% NCF(aspek) + (x)\% NSF(aspek) = Ntotal(aspek)$$

Keterangan :

NCF (aspek) = Nilai rata-rata dari *Core Factor*

NSF (aspek) = Nilai rata-rata dari *Secondary Factor*

N (aspek) = Nilai total dari aspek yang dinilai

$(x)\%$ = Persentase bobot masing-masing faktor sesuai ketentuan sistem

- f. Langkah akhir adalah menentukan peringkat berdasarkan nilai total setiap aspek, menggunakan rumus:

$$Ranking = (x)\% Ns$$

Keterangan :

Ranking = Nilai akhir untuk menentukan urutan atau peringkat

Ns = Nilai dari setiap aspek

$(x)\%$ = Persentase bobot aspek terhadap nilai akhir

3. *Website*

Website merupakan sekumpulan halaman *web* yang saling terhubung dan dapat diakses melalui jaringan internet, yang dirancang untuk menyajikan berbagai bentuk informasi, media pembelajaran, konten *multimedia*, hingga menjalankan aplikasi secara daring dengan cara yang cepat, mudah, dan tepat (Puspita et al., 2021).

Dalam proses pembuatannya, *website* dibangun dengan memanfaatkan kombinasi *markup language* dan bahasa pemrograman. Menurut (Rahmatika et al., 2020), struktur dasar halaman *web* umumnya menggunakan HTML (*HyperText Markup Language*) untuk menyusun konten, CSS (*Cascading Style Sheets*) untuk mengatur tampilan visual, serta *JavaScript* dan *PHP* untuk mengatur interaktivitas serta logika aplikasi *web*. Penggunaan teknologi-teknologi tersebut memungkinkan pengembang menciptakan tampilan antarmuka yang menarik, responsif, dan fungsional

sesuai dengan kebutuhan pengguna. HTML berperan dalam membentuk kerangka dan isi halaman, CSS bertanggung jawab atas aspek visual dan estetika, sementara JavaScript dan PHP digunakan untuk mengelola logika program serta interaksi dinamis antara pengguna dan sistem. Selain itu, pengembangan website saat ini juga didukung oleh berbagai *framework* dan *Content Management System* (CMS), yang bertujuan untuk mempercepat proses pembangunan serta mempermudah pengelolaan konten situs *web* (Subariah et al., 2021).

Dapat disimpulkan bahwa *website* merupakan media digital yang sangat penting dalam penyampaian informasi dan layanan interaktif secara daring. Kombinasi teknologi *markup* dan pemrograman seperti HTML, CSS, JavaScript, dan PHP, serta dukungan *framework* atau CMS, memungkinkan *website* dikembangkan secara efisien dan fungsional untuk memenuhi berbagai kebutuhan pengguna.

4. *Framework*

Menurut (Sinlae et al., 2024), *framework* adalah komponen pemrograman yang dapat digunakan secara berulang oleh programmer dalam menyusun skrip untuk menyelesaikan tugas-tugas yang serupa. Sebagai contoh, ketika seorang programmer ingin membuat halaman *web* dengan fitur paginasi, programmer tidak perlu menulis kode dari awal, melainkan cukup menggunakan fungsi-fungsi yang telah disediakan oleh *framework* yang digunakan.

Di sisi lain, (Ikhsan et al., 2023) mengemukakan bahwa *framework* merupakan suatu kerangka kerja yang terdiri atas serangkaian fungsi, prosedur, serta kelas-kelas yang telah dirancang secara khusus untuk mendukung proses pengembangan perangkat lunak. Kehadiran *framework* bertujuan untuk menyederhanakan sekaligus mempercepat proses pengembangan, karena pengembang tidak perlu lagi membangun struktur dasar seperti fungsi atau kelas secara manual dari awal. Dengan menggunakan *framework*, efisiensi dan konsistensi dalam penulisan kode dapat lebih terjamin, sehingga pengembangan sistem menjadi lebih terstruktur dan terarah.

Berdasarkan pendapat-pendapat tersebut, dapat disimpulkan bahwa *framework* adalah sebuah kerangka kerja yang menyajikan komponen-komponen pemrograman yang dapat digunakan kembali oleh programmer. Dengan adanya kumpulan fungsi, prosedur, dan kelas yang telah terstruktur, *framework* memudahkan proses pengembangan sistem secara efisien dan konsisten.

5. *CodeIgniter 3*

CodeIgniter 3 adalah *framework* berbasis PHP yang populer digunakan untuk membangun aplikasi *web*. Keunggulan utama dari *framework* ini adalah sifatnya yang ringan, mudah dipelajari, dan memiliki performa yang cepat. *CodeIgniter 3* menawarkan pengelolaan file, keamanan, dan kecepatan pemrosesan yang baik, sehingga cocok digunakan dalam pengembangan sistem pendukung keputusan, seperti sistem pemilihan

karyawan terbaik dengan metode profile matching. Dengan dukungan arsitektur MVC (*Model-View-Controller*), *CodeIgniter 3* memungkinkan pengembangan untuk memisahkan logika aplikasi dari tampilan antarmuka, sehingga pengelolaan kode lebih terstruktur (Sabilillah et al., 2024).

Menurut (Sahi, 2020), *CodeIgniter* merupakan salah satu *framework* yang dirancang untuk memudahkan pengembang dalam membangun aplikasi web secara cepat dan efisien sejak tahap awal pengembangan. *Framework* ini tidak hanya memungkinkan pembuatan situs *web* yang lebih dinamis, tetapi juga mendukung pengembangan aplikasi yang ringan dan memiliki performa tinggi. Salah satu keunggulan utama dari *CodeIgniter* adalah tersedianya dokumentasi yang sangat lengkap, dilengkapi dengan contoh-contoh implementasi kode yang mempermudah proses pembelajaran dan penerapan dalam pengembangan sistem secara praktis.

Dapat disimpulkan bahwa *CodeIgniter 3* merupakan pilihan *framework* yang tepat untuk membangun sistem berbasis *web* yang membutuhkan performa tinggi, struktur kode yang rapi, serta efisiensi dalam proses pengembangan. Kemudahan penggunaan dan dokumentasi yang lengkap menjadikan *CodeIgniter 3* sangat mendukung pengembangan sistem pendukung keputusan seperti yang diangkat dalam penelitian ini.

6. *PHP*

PHP (*Hypertext Preprocessor*) merupakan sebuah bahasa pemrograman yang populer digunakan dalam sebuah *framework* yang bersifat *open source*. Menurut (Hermiati et al., 2021) menerangkan

bahwa *PHP* dikenal sebagai sebuah bahasa *scripting* yang menyatu dengan tag-tag *HTML*, dieksekusi di *server*, dan digunakan untuk membuat halaman *web* yang dinamis. (Sudaria et al., 2021) dalam penelitiannya menjelaskan bahwa *PHP* adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan kode program menjadi kode mesin yang dapat dimengerti dan dijalankan oleh komputer yang berbasis *server-side* yang juga dapat ditambahkan pada *HTML*. *PHP* juga bisa digunakan dalam berbagai sistem operasi seperti *linux*, *windows*, dan *unix* dengan salah satu fungsi utamanya adalah menangani kebutuhan standar pembuatan aplikasi *website* (Setiawan et al., 2022).

Dari pendapat diatas, maka dapat disimpulkan bahwa *PHP* merupakan bahasa pemrograman *open source* yang populer dan sering digunakan dalam pengembangan aplikasi *web*. Berfungsi untuk menerjemahkan kode program menjadi bahasa mesin yang dapat dijalankan oleh komputer.

7. Basis Data

Basis data merupakan elemen fundamental dalam pengembangan aplikasi berbasis *web*, karena berperan sebagai pusat penyimpanan dan pengelolaan data yang mendukung fungsionalitas sistem. Menurut (Jamaluddin et al., 2022:2), basis data didefinisikan sebagai sekumpulan informasi yang disimpan secara sistematis dalam komputer, sehingga memungkinkan pengguna untuk mengakses dan menelusuri data melalui

bantuan program komputer guna memperoleh informasi yang dibutuhkan secara efisien.

Sementara itu, (Tri Amri Wijaya et al., 2021) menjelaskan bahwa basis data adalah kumpulan data yang saling memiliki keterkaitan atau relasi, yang memungkinkan data tersebut untuk dimanipulasi, ditampilkan, serta dicari dengan cepat dan akurat. Hubungan antar data ini memungkinkan proses pengolahan informasi menjadi lebih terstruktur dan dapat diandalkan dalam mendukung proses bisnis atau operasional sistem.

Berdasarkan pandangan kedua sumber tersebut, dapat disimpulkan bahwa basis data merupakan kumpulan informasi yang tersimpan secara terorganisasi dalam sistem komputer, yang dirancang untuk mempermudah proses pencarian, pengambilan, dan pengelolaan data. Dalam konteks aplikasi *web*, basis data sering diibaratkan sebagai “otak” dari sistem karena memiliki peran sentral dalam mengatur aliran informasi serta menjaga konsistensi dan integritas data yang digunakan oleh seluruh komponen sistem.

8. *MySQL*

MySQL merupakan salah satu sistem manajemen basis data relasional yang dikembangkan berdasarkan konsep dasar *Structured Query Language* (SQL). Menurut (Joshua et al., 2021), SQL berperan sebagai bahasa utama dalam pengelolaan *database*, yang memungkinkan pengguna untuk melakukan operasi seperti pemilihan, pemasukan, dan pengolahan data secara otomatis dan efisien. MySQL, sebagai implementasi dari SQL,

menyediakan berbagai fungsi yang memudahkan dalam pengelolaan data berbasis relasional, menjadikannya salah satu sistem basis data yang banyak digunakan dalam pengembangan perangkat lunak modern.

Pendapat ini diperkuat oleh (Kalsum Siregar et al., 2024) yang menyatakan bahwa SQL merupakan bahasa pemrograman paling populer untuk mengakses dan mengelola basis data relasional. Dalam praktiknya, MySQL menjadi salah satu perangkat lunak pendukung SQL yang mampu menyederhanakan proses pengolahan data bagi para praktisi, baik dalam skala kecil maupun besar.

Berdasarkan uraian dari berbagai sumber tersebut, dapat disimpulkan bahwa SQL merupakan fondasi utama dalam pengoperasian basis data, yang memberikan kemudahan dalam proses manipulasi data secara otomatis. Sementara itu, MySQL berfungsi sebagai *platform* manajemen basis data relasional yang mendukung implementasi SQL, serta memberikan kemudahan dan efisiensi dalam pengelolaan data bagi pengguna dari berbagai skala kebutuhan.

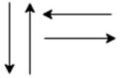
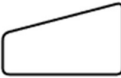

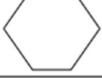



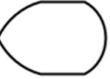








9. *Flowchart*

(Saputro, 2022) mendefinisikan diagram alur sebagai representasi grafis termasuk simbol-simbol tertentu yang menggambarkan urutan langkah demi langkah suatu proses dan interkoneksi antara berbagai proses di dalam suatu perangkat lunak.

Menurut (Fauzi, 2020), *diagram alur* atau *flowchart* merupakan representasi visual dari suatu algoritma yang disusun menggunakan simbol-

simbol grafis standar. Flowchart berfungsi untuk menggambarkan urutan langkah-langkah dalam suatu program, serta memperlihatkan hubungan antara proses yang berlangsung dan informasi yang terlibat dalam sistem tersebut. Setiap simbol dalam diagram ini merepresentasikan jenis aktivitas atau proses tertentu, sementara garis penghubung menunjukkan aliran atau urutan eksekusi dari satu proses ke proses lainnya.

Dengan memanfaatkan *flowchart*, pengembang sistem dapat memperoleh pemahaman yang lebih sistematis mengenai alur logika program, sekaligus memudahkan proses identifikasi terhadap bagian analisis yang mungkin terlewat atau belum optimal. Representasi grafis ini juga membantu dalam mengomunikasikan rancangan sistem secara lebih efektif kepada pihak-pihak terkait. Penjelasan lebih lanjut mengenai simbol-simbol yang digunakan dalam diagram alur dapat dilihat pada Gambar 2.1 berikut.

	Simbol Flow Direction Simbol penghubung antara simbol yang satu dengan simbol yang lain		Simbol Manual Input Simbol untuk input data secara manual on-line keyboard
	Simbol Terminator Simbol untuk permulaan (start) dan akhir (stop) untuk suatu kegiatan		Simbol Preparation Simbol persiapan penyimpanan di dalam storage
	Simbol Connector Simbol keluar-masuk proses dalam halaman yang sama		Simbol Predefine Proses Simbol pelaksanaan suatu sub-program/prosedur
	Simbol Connector Simbol keluar masuk proses dalam halaman yang berbeda		Simbol Display Simbol pernyataan output seperti layar, plotter, printer, dll
	Simbol Processing Simbol yang menunjukkan pengolahan oleh komputer		Simbol Disk and Online Storage Simbol pernyataan input dari disk/disimpan di disk
	Simbol Manual Operation Simbol yang menunjukkan pengolahan bukan dari komputer		Simbol Magnetik Tape Unit Simbol pernyataan input dari magnetik/disimpan di magnetik
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang baik		Simbol Punch Card Simbol pernyataan input dari punch card/disimpan di punch card
	Simbol Input Output Simbol pernyataan proses input output tanpa berdasarkan peralatan		Simbol Dokumen Simbol pernyataan input dari dokumen dalam bentuk kertas/output dicetak kertas

Gambar 2. 1 Simbol Flowchart

Sumber : (Fauzi, 2020)

10. Unified Modeling Language (UML)

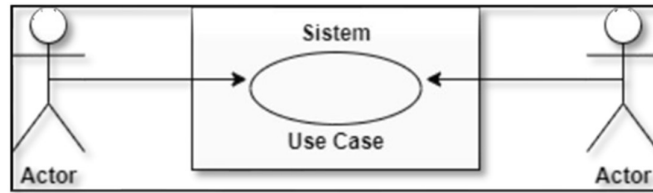
Unified Modeling Language (UML) berperan sebagai alat bantu penting dalam proses perancangan dan pendeskripsian sistem yang dibangun menggunakan pendekatan pemrograman berorientasi objek. UML memungkinkan pengembang untuk memvisualisasikan struktur dan perilaku sistem secara sistematis dan terstandar, sehingga mempermudah proses dokumentasi dan implementasi (Nistrina & Sahidah, 2022). Dalam penelitian yang dilakukan oleh (Suharni et al., 2023), dijelaskan bahwa UML merupakan bentuk pemodelan yang meniru kondisi nyata, di mana sistem direpresentasikan sebagai kumpulan objek yang saling berinteraksi.

Objek-objek tersebut divisualisasikan dengan simbol-simbol khusus yang dirancang untuk menggambarkan hubungan, aktivitas, serta fungsi sistem secara menyeluruh.

Sementara itu, (Siska Narulita et al., 2024) menambahkan bahwa UML memberikan standar dalam penyusunan *blueprint* sistem, yang mencakup berbagai komponen penting seperti pemodelan proses bisnis, perancangan kelas (*class*) yang dapat diimplementasikan dalam bahasa pemrograman tertentu, rancangan basis data, hingga komponen-komponen pendukung lain yang relevan dalam pengembangan perangkat lunak. UML terdiri dari beberapa jenis diagram yang berfungsi sebagai kerangka penyusun sistem, antara lain *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*. Masing-masing diagram ini memiliki fungsi spesifik dalam menggambarkan aspek struktural maupun dinamis dari sistem yang dirancang, sehingga memudahkan pemahaman dan pengelolaan sistem secara terintegrasi.

a. *Use case diagram*

Use case diagram merupakan suatu interaksi antara sistem dengan aktor yang memiliki alur dan kemudian diterapkan pada sistem yang nantinya dibuat, yang mana nantinya interaksi ini menjadi dasar perintah saat aktor melakukan tindakan maka sistem akan merespon tindakan tersebut dan menjalankan sesuai sebagaimana fungsinya (Sutrisno & Karnadi, 2020). Adapun gambar mengenai *use case diagram* dapat dilihat pada gambar 2.2 seperti berikut



Gambar 2. 2 Use Case Model

Sumber : (Saputra & Fahrizal, 2021)






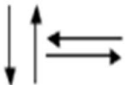
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Gambar 2. 3 Simbol Use Case Diagram

Sumber : (Rasiban et al., 2024)

b. *Activity diagram*

Activity Diagram merupakan diagram yang dapat memodelkan proses yang terjadi dalam sebuah sistem dengan memiliki alur berupa runtutan menu-menu proses bisnis yang terdapat dalam sistem. Diagram aktivitas tidak menjelaskan tentang kelakuan aktor, melainkan dipakai untuk menggambarkan alur kerja dan aktivitas sistem (Prasetya et al., 2022). Menurut Saputra, (2021) menjelaskan bahwa *activity diagram* merupakan teknik untuk mendeskripsikan logika prosedural, proses bisnis, serta aliran kerja dalam banyak kasus. Adapun simbol-simbol yang digunakan dalam *activity diagram* bisa dilihat pada gambar 2.4 berikut.

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya


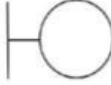



Gambar 2. 4 Simbol Activity Diagram

Sumber : (Saputra & Fahrizal, 2021)

c. *Sequence diagram*

Sequence diagram merupakan salah satu jenis diagram dalam pemodelan sistem yang digunakan untuk menggambarkan interaksi antar objek dalam sistem secara runtut berdasarkan urutan waktu. Diagram ini memvisualisasikan bagaimana suatu objek saling berkomunikasi melalui pengiriman pesan secara berurutan dalam rangka mengeksekusi suatu skenario atau kondisi tertentu dalam sistem. Dengan kata lain, *sequence diagram* berfungsi untuk menunjukkan dinamika komunikasi antar objek dari waktu ke waktu, serta memperlihatkan bagaimana alur proses berjalan dalam konteks tertentu.

Menurut (Acharya, 2024), *sequence diagram* adalah bentuk diagram interaksi yang menitik beratkan pada urutan kronologis pengiriman pesan antar objek. Secara grafis, diagram ini menyerupai tabel linier di mana setiap objek digambarkan secara vertikal, dan pesan-pesan yang dipertukarkan direpresentasikan dengan panah horizontal yang mengindikasikan arah dan urutan interaksi. Interaksi ini mencerminkan skenario implementasi sistem yang sedang dianalisis. Dengan demikian, diagram ini menjadi alat penting dalam mengidentifikasi tanggung jawab masing-masing objek dan memastikan bahwa alur komunikasi antar komponen sistem berlangsung secara logis dan terstruktur. Simbol dan ilustrasi lebih lanjut mengenai *sequence diagram* dapat dilihat pada Gambar 2.5 berikut.

Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

Gambar 2. 5 Simbol Sequence Diagram


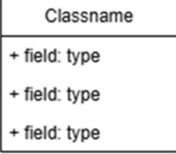




Sumber : (Rasiban et al., 2024)

d. *Class diagram*

Class diagram merupakan representasi visual dari struktur statis suatu sistem yang menggambarkan kumpulan kelas yang akan digunakan dalam proses pengembangan perangkat lunak berbasis objek. Diagram ini berfungsi untuk menunjukkan hubungan antar kelas, atribut yang dimiliki, serta metode atau operasi yang dapat dijalankan oleh masing-masing kelas. Dalam konteks rekayasa perangkat lunak, *class*

diagram sangat penting karena membantu pengembang memahami kerangka logis sistem sebelum implementasi dilakukan.

Menurut (Ahmad et al., 2022), *class diagram* dapat didefinisikan sebagai alat bantu visualisasi yang digunakan untuk menggambarkan struktur internal dari sebuah sistem atau program, terutama dalam konteks objek yang memiliki kesamaan atribut (properti), fungsi (operasi), dan hubungan (relasi). Diagram ini memberikan gambaran menyeluruh mengenai bagaimana objek-objek dalam sistem saling berinteraksi dan bagaimana struktur data diorganisasikan secara hierarkis maupun fungsional. Dengan demikian, *class diagram* tidak hanya membantu dalam tahap desain awal, tetapi juga mempermudah proses dokumentasi dan pengembangan berkelanjutan. Adapun simbol-simbol dan penjelasan terkait komponen *class diagram* dapat dilihat pada Gambar 2.6 berikut.

Simbol	Keterangan
Association 	Merupakan Class-class yang saling terhubung satu sama lain secara konseptual.
Class 	Merupakan kumpulan objek yang memiliki atribut dan operation yang sama. Atribut, merupakan <i>property</i> dari sebuah <i>class</i> . <i>Operation</i> merupakan sesuatu yang bisa dilakukan oleh sebuah <i>class</i> .
Agregation 	Merupakan suatu garis yang menangani objek-objek dimana salah satunya adalah bagian dari yang lain.
Composition 	Merupakan sebuah tipe agregasi yang kuat dimana bagian dari objek tergantung pada keseluruhan objek.
Dependency 	Merupakan sebuah garis yang berfungsi menunjukkan operasi pada suatu class yang menggunakan class lain.
Generalization 	Merupakan suatu garis yang berfungsi untuk mewariskan struktur data dan objek induk kepada objek anak yang dituju.

Gambar 2. 6 Simbol Class Diagram

Sumber : (Wandari, 2023)

11. *System Usability Scale (SUS)*

Metode *System Usability Scale (SUS)* telah digunakan dan diuji coba selama lebih dari tiga dekade, dan saat ini masih menjadi metode yang sangat diandalkan dalam mengevaluasi *usability* suatu sistem dengan merujuk pada standar industri (Dermawan Mulyodiputro et al., 2023). Metode ini melibatkan penggunaan kuesioner yang dirancang secara khusus untuk mengevaluasi sejauh mana pengguna puas dan mudah menggunakan suatu sistem (Kholifah, Siti Nur Heryana, Nono Nugraha, 2023). Kuesioner SUS terdiri dari sejumlah pernyataan yang menilai berbagai aspek *usability*,

termasuk tingkat kesulitan penggunaan, kejelasan antarmuka, dan kepuasan pengguna secara umum.

Pemilihan metode *System Usability Scale* (SUS) sebagai alat pengukuran kebergunaan pada penelitian ini didasarkan pada kehandalan, kesederhanaan, dan efisiensi yang dimilikinya. SUS telah terbukti menjadi standar industri yang efektif dalam menilai sejauh mana suatu sistem dapat dianggap berguna oleh penggunanya (Dermawan Mulyodiputro et al., 2023). Kelebihan utama SUS terletak pada pertanyaan yang sederhana dan dapat dijawab dengan cepat oleh responden, memungkinkan pengumpulan data yang efisien (Madawara et al., 2023).

B. Kajian Empiris

Berbagai penelitian mengenai penerapan *Decision Support System* (DSS) dengan metode *Profile Matching* telah banyak dilakukan dalam beragam konteks organisasi dan sektor industri. Tinjauan terhadap sejumlah studi terdahulu menunjukkan bahwa metode ini sering digunakan sebagai alat bantu dalam proses seleksi dan evaluasi, khususnya untuk mengidentifikasi individu yang paling sesuai berdasarkan sejumlah kriteria yang telah ditentukan. Meskipun begitu, hasil telaah juga mengungkapkan adanya ruang kontribusi yang masih dapat diisi melalui penelitian ini, terutama dalam hal kontekstualisasi sistem terhadap kebutuhan spesifik perusahaan.

Salah satu penelitian yang relevan dilakukan oleh (Danuarsa & Santoso, 2023) dalam karya mereka yang berjudul "*Sistem Pendukung Keputusan Penilaian Kinerja Karyawan Terbaik pada PT. Nusantara Bina Artha*

Menggunakan Metode Profile Matching". Penelitian tersebut bertujuan merancang sistem yang dapat membantu bagian HRD dalam melakukan penilaian kinerja karyawan yang sebelumnya bersifat subjektif. Dengan memanfaatkan metode *Profile Matching*, sistem yang dikembangkan mampu menghitung tingkat kesesuaian kompetensi karyawan terhadap kriteria yang telah ditentukan. Hasil dari studi ini menunjukkan bahwa sistem mampu mempercepat proses evaluasi dan meningkatkan objektivitas dalam pengambilan keputusan.

Penelitian serupa dilakukan oleh (Desty Nurmayanti et al., 2022) yang dipublikasikan dalam jurnal SATIN dengan judul "*Penerapan Metode Profile Matching untuk Sistem Penunjang Keputusan Pemilihan Karyawan Terbaik*". Studi ini menggunakan lima kriteria utama, yaitu produktivitas, kualitas kerja, kerjasama, inisiatif, dan kehadiran. Tujuan utamanya adalah menyediakan sistem berbasis desktop yang mampu membantu lembaga pemerintah dalam menentukan karyawan terbaik secara sistematis. Sistem menghasilkan *output* berupa perbandingan, yang mempermudah proses seleksi oleh pihak pengambil keputusan.

Adapun Prayogi Perdana (Nicolas et al., 2021) mengembangkan pendekatan yang lebih kompleks dalam penelitiannya "*Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik pada PT. XYZ dengan Metode Profile Matching dan Interpolasi*". Penelitian ini mengombinasikan metode *Profile Matching* dengan teknik interpolasi guna meningkatkan ketelitian dalam pengolahan nilai gap. Pendekatan ini memperhitungkan variasi antar nilai kriteria yang tidak

seragam, sehingga mampu menghasilkan perankingan yang lebih akurat dan presisi dalam menyeleksi karyawan unggulan.

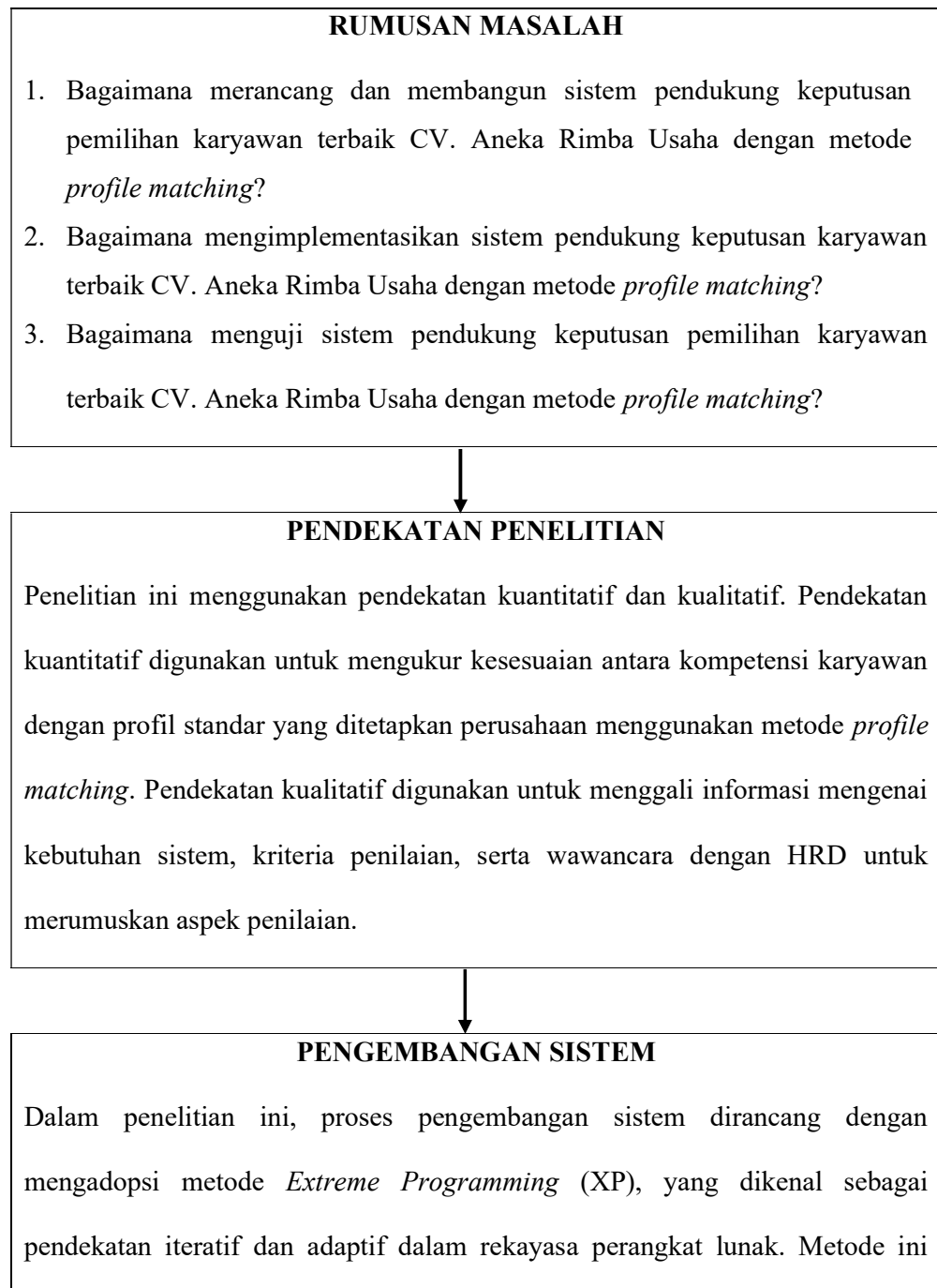
Dari perspektif global, (William et al., 2022) dalam jurnal internasional JAKAKOM memaparkan penelitian berjudul "*Perancangan Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik pada PT. Dipo Internasional Pahala Otomotif*". Penelitian ini mengimplementasikan metode *Profile Matching* pada sistem berbasis PHP dan MySQL. Hasilnya menunjukkan bahwa sistem tersebut mampu meningkatkan keadilan dalam pengambilan keputusan serta mengurangi dominasi intuisi subjektif dari pihak manajemen.

Terakhir, penelitian oleh (Akbar & Mufty, 2020) yang diterbitkan dalam Jurnal SENATIK dengan judul "*Seleksi Pemilihan Karyawan Terbaik Menggunakan Metode Profile Matching*", memberikan fokus pada pentingnya proses pembobotan dalam menilai kompetensi secara menyeluruh. Penelitian ini menekankan bahwa penerapan metode tersebut dapat mendorong efisiensi serta meningkatkan aspek keadilan dalam seleksi karyawan, meskipun pengembangannya masih terbatas pada sisi teknis dan belum sepenuhnya mempertimbangkan kebutuhan organisasi secara kontekstual.

Dari kelima penelitian tersebut, terdapat beberapa persamaan yang menonjol, yakni sama-sama menggunakan metode *profile matching*, bertujuan meningkatkan objektivitas penilaian, dan menghasilkan *output* perankingan karyawan. Namun, terdapat pula perbedaan, baik dari sisi pendekatan pengembangan sistem, penambahan metode interpolasi, maupun kriteria penilaian yang digunakan. Selain itu, beberapa penelitian belum

memperhatikan pengembangan sistem yang disesuaikan secara kontekstual dengan kebutuhan spesifik perusahaan.

C. Kerangka Berfikir



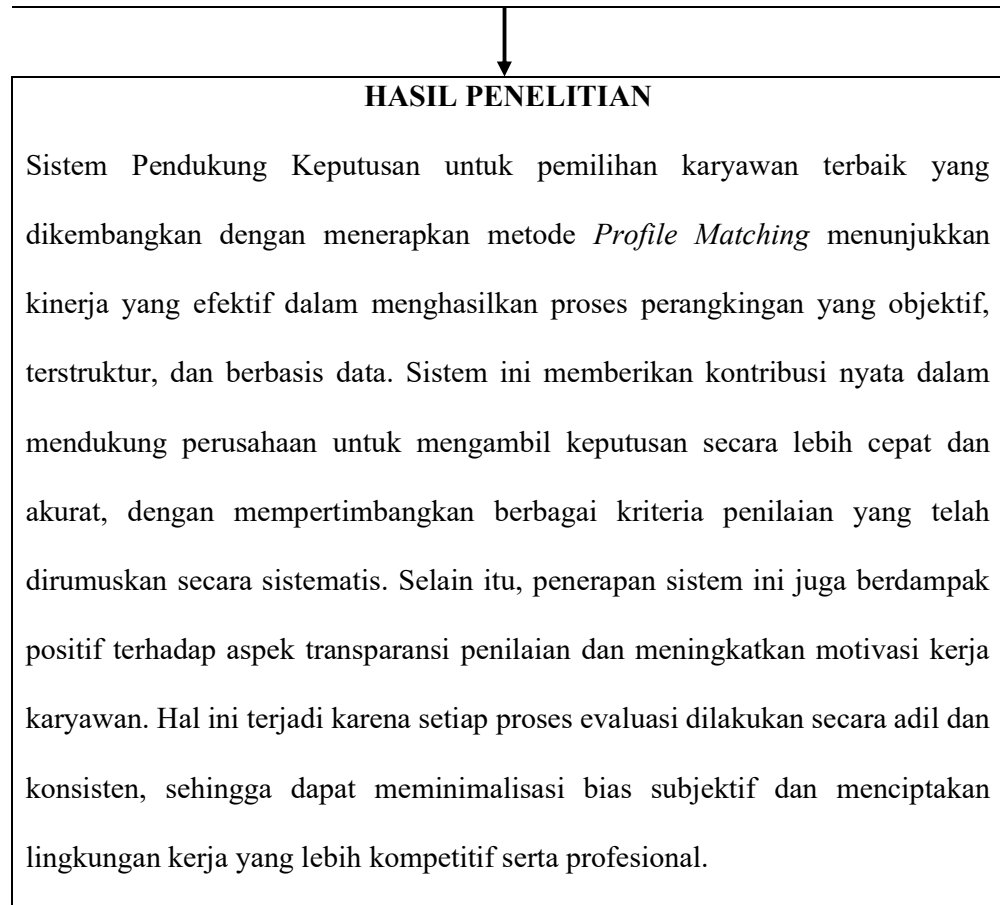
dipilih karena kemampuannya dalam menangani perubahan kebutuhan secara dinamis serta mendukung kolaborasi yang intens antara tim pengembang dan pengguna. Untuk implementasi teknis, sistem dibangun menggunakan framework *CodeIgniter* versi 3, yang menawarkan struktur pengembangan berbasis *Model-View-Controller* (MVC) guna menjaga keteraturan dan efisiensi dalam penulisan kode. Sementara itu, pengelolaan data dilakukan melalui sistem manajemen basis data *MySQL*, yang mendukung penyimpanan dan pengolahan data secara terstruktur dan andal. Kombinasi antara metode pengembangan dan perangkat teknologi tersebut diharapkan mampu menghasilkan sistem yang responsif, mudah dikembangkan, serta sesuai dengan kebutuhan pengguna di lingkungan organisasi.

EVALUASI SISTEM

Pengujian sistem ini menggunakan bantuan kuesioner dengan metode *system usability scale* (SUS) untuk mengetahui kebergunaan dan kepuasan pengguna

IMPLEMENTASI SISTEM

Sistem diimplementasikan dalam bentuk aplikasi berbasis *web* yang digunakan oleh pihak HRD untuk memasukkan data penilaian karyawan. Sistem secara otomatis akan menghitung nilai gap antara profil ideal dan kompetensi karyawan, melakukan pembobotan berdasarkan *core factor* dan *secondary factor*, lalu menghasilkan output berupa ranking karyawan yang dinilai paling sesuai untuk mendapatkan penghargaan.



Gambar 2. 7 Kerangka Berfikir