BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Pendukung Keputusan

Sistem pendukung keputusan/*DSS* (*Decision Support System*) adalah bahan atau model dalam ilmu komputer yang dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang berdasarkan situasi yang semistruktur dan situasi yang tidak struktur (Kusrini, 2021).

Sistem Pendukung Keputusan (SPK) adalah sistem komputer yang dirancang untuk membantu pengambilan keputusan dalam situasi yang kompleks dan tidak pasti. Dengan menggunakan SPK, pengambil keputusan dapat mengakses informasi yang terkini dan akurat, menganalisis berbagai skenario dan alternatif,serta mengevaluasi konsekuensi dari setiap keputusan yang diambil. Selain itu, SPK juga dapat membantu mengidentifikasi pola dan tren yang mungkin tidak terlihat secara manual, sehingga memungkinkan perusahaan atau organisasi untuk merespons perubahan pasar atau lingkungan dengan lebih cepat dan efektif (Yusfrizal, 2020).

Dapat disimpulkan bahwa SPK atau Sistem Pendukung Keputusan atau *DSS* (*Decision Support System*) merupakan ilmu komputer yang digunakan untuk membantu menyelesaikan berbagai masalah seperti situasi semistruktur dan tidak struktur, situasi

kompleks dan tidak pasti. Sehingga dengan menggunakan SPK/*DSS* pengambilan keputusan dapat mengakses informasi yang terkini dan akurat.

2. *Profile Matching*

Metode *profile matching* adalah sebuah metode dari sistem pendukung keputusan yang melakukan proses penilaian dengan membandingkan antara satu profil nilai dengan beberapa profil nilai lainnya dengan tujuan untuk mengetahui selisihnya atau nilai *gap*. Selisih dari kompetensi disebut dengan gap, dimana *gap* yang semakin kecil memiliki nilai yang semakin tinggi (Mahendra et al., 2023).

Metode *profile matching* merupakan metode yang memberikan bobot berbeda pada setiap kriteria, tujuannya adalah agar kriteria sesuai dengan tipe atau standar kepentingan. Metode ini sering dimanfaatkan untuk merancang sistem pendukung keputusan karena memiliki keunggulan dengan adanya (CF) *core factor*/faktor utama dan (SF) *secondary factor*/faktor pendukung (Harahap et al., 2024).

Berdasarkan penjelasan di atas dapat di ambil kesimpulan bahwa metode *profile matching* merupakan salah satu metode yang digunakan dalam penerapan sistem pendukung keputusan yang memiliki proses penilaian perbandingan selisih nilai berdasarkan indikator penilaian. Metode ini juga memberikan bobot pada setiap kriteria dengan tujuan untuk menyesuaikan dengan standar kepentingan. Selain itu, metode ini juga memiliki keunggulan yaitu

berupa adanya *core factor* dan *secondary factor* pada tahapan proses penilaian untuk mencari nilai *gap*/selisih.

(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* = Nilai Kriteria – Nilai Minimal.
c. Pembobotan setelah pemetaan *GAP* sesuai dengan tabel bobot nilai *GAP* seperti tabel 2.1 berikut (Harahap et al., 2024).

Tabel 2.1. Nilai Pembobotan

NO	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih (Kompetensi sesuai kebutuhan)
2	1	4,5	Kompetensi individu kelebihan 1 tingkat
3	-1	4	Kompetensi individu kekurangan 1 tingkat
4	2	3,5	Kompetensi individu kelebihan 2 tingkat

5	-2	3	Kompetensi individu kekurangan 2 tingkat
6	3	2,5	Kompetensi individu kelebihan 3 tingkat
7	-3	2	Kompetensi individu kekurangan 3 tingkat
8	4	1,5	Kompetensi individu kelebihan 4 tingkat
9	-4	1	Kompetensi individu kekurangan 4 tingkat

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

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

$$NCF = \frac{\sum\ NC\ (aspek)}{\sum\ IC} \quad (2.1)$$

Keterangan :

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

NS (aspek) = Jumlah nilai *Core Factor*

IC = Jumlah item *Core Factor*

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

$$NSF = \frac{\sum\ NS\ (aspek)}{\sum\ IS} \quad (2.2)$$

Keterangan :

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

NS (aspek) = Jumlah nilai *Secondary Factor*

IS = Jumlah item *Secondary Factor*

e. Perhitungan nilai total dengan rumus seperti persamaan berikut.

$$(x)\%NCF(aspek) + (x)\%NSF(aspek) = N(aspek) \quad (2.3)$$

Keterangan :

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

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

N(aspek) = Nilai total dari aspek (x)% = Nilai persen yang diinputkan

f. Perhitungan nilai rangking untuk menetukan perangkingan dengan rumus seperti persamaan berikut.

$$Rangking = (x)\%Ns \quad (2.4)$$

Keterangan :

Ns = Nilai aspek

(x)% = Nilai persen yang diinputkan

3. *Website*

Website adalah sebuah media yang bersifat *hyperlink*, yaitu memiliki halaman yang saling terhubung dengan halaman lainnya yang dimana *website* memuat informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya. *Website* memiliki karakteristik utama yaitu halaman-halaman yang saling terhubung dan dilengkapi dengan *domain* sebagai alamat (*url*) atau *world wide web* (*www*) dan juga *hosting* sebagai media penyimpanan data yang banyak (Elgamar, 2020).

Situs web / website merupakan kumpulan informasi digital yang ditampilkan pada halaman web yang sering dihubungkan melalui

tautan yang disajikan dalam bentuk teks atau gambar. Situs web adalah konten online apa pun yang terkandung dalam domain dan menyediakan informasi. Untuk membuat website yang baik, hal yang harus dipertimbangkan yaitu kriteria fungsionalitas, desain, konten, orisinalitas, profesionalisme, dan efektivitas (Purnomo et al., 2023)

Dapat disimpulkan bahwa *website* merupakan sebuah halaman hyperlink atau mempunyai karakteristik utama saling terhubung yang dilengkapi dengan *domain* dan *hosting* sebagai media penyimpanan data serta memiliki kegunaan sebagai gerbang utama menuju informasi dan interaksi dengan tujuan untuk menampilkan informasi, menawarkan produk, atau membangun sebuah komunitas.

4. *Framework*

Framework adalah sebuah komponen pemrograman yang siap digunakan secara berulang kali oleh seorang *programer* dalam membuat skrip untuk tugas yang serupa. Sebagai contoh, seorang *programer* ingin membuat halaman web yang menampilkan data dengan paginasi, maka seorang *programer* hanya perlu menggunakan fungsi yang telah ditetapkan dalam *framework* yang ingin digunakan (Sinlae et al., 2024).

Framework merupakan kerangka kerja yang terdiri dari sekumpulan fungsi/prosedur dan kelas-kelas yang memiliki tujuan tertentu dan siap digunakan untuk mempermudah dan mempercepat

pekerjaan seorang *programer* tanpa harus membuat fungsi atau kelas (Ikhsan et al., 2023).

Dapat disimpulkan bahwa *framework* merupakan kerangka kerja yang memuat tentang komponen-komponen pemrograman yang bisa digunakan secara berulang kali olehn *programer* untuk membuat skrip dan memiliki sekumpulan fungsi maupun prosedur serta kelas-kelas tertentu untuk mempermudah juga mempercepat pekerjaan *programer*.

5. *Laravel*

Laravel ialah salah satu *framework PHP* terbaik yang dicipta oleh Taylor Otwell pada April 2011. *Laravel* dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*model view controller*). *Laravel* adalah kerangka kerja pengembangan situs web berbasis MVP yang ditulis dalam *PHP*. Tujuan utamanya adalah untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan. Selain itu, *Laravel* juga bertujuan untuk meningkatkan pengalaman pengguna dalam menggunakan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu (Prasetyo, 2019).

Laravel merupakan *framework PH*P yang dirancang untuk mengembangkan aplikasi *website* dengan cara terstruktur dan lebih rapi. *Laravel* dirancang untuk lebih menekankan pada kesederhanaan dan fleksibilitas pada desainnya serta dilengkapi dengan *command*

line tool yang bernama *"artisan"* yang dapat digunakan untuk *packaging bund*le dan instalasi *bundle* (Muthia Kansha et al., 2023).

Dapat disimpulkan bahwa *laravel* merupakan salah satu contoh *framework PHP* yang dirancang dan dibangun dengan konsep MVC (*model view controller*) dengan tujuan untuk mempermudah dan mempercepat seorang *programer* dalam pekerjaannya dalam mengembangkan aplikasi *website* dengan cara terstruktur dan lebih rapi.

6. *PHP*

PHP : Hypertext Preprocessor merupakan bahasa pemrograman berjenis *server-sides*, yaitu bahasa pemrograman yang diproses oleh server yang hasil olahannya akan dikirim kembali ke *browser*. Dengan demikian, salah satu *tool* yang harus tersedia sebelum memulai *PHP* adalah server, yang dimana server yang sering digunakan dalam memulai *PHP* yaitu *XAMPP* (Yoga, 2019)

PHP adalah bahasa pelengkap *HTML* yang memungkinkan pembuatan aplikasi dinamis dengan kemampuan pengolahan dan pemrosesan data. Semua sintaks yang diberikan akan dijalankan sepenuhnya di server, sementara yang dikirimkan ke *browser* hanya hasilnya saja. *PHP* adalah bahasa *scripting* yang terintegrasi dengan tag-tag *HTML*, dieksekusi di server, dan digunakan untuk membuat halaman web dinamis seperti *Active Server Pages* (*ASP*) atau *Java Server Pages* (*JSP*) (Hermiati et al., 2021).

Dapat disimpulkan bahwa *PHP* (*Hypertext Preprocessor*) merupakan pemrograman *server-side* yang diproses oleh server dan mengirim hasilnya kembali ke *browser*. *PHP* adalah bahasa pemrograman tambahan untuk *HTML* yang memungkinkan pengembangan aplikasi dinamis dengan kemampuan pengolahan dan pemrosesan data. *PHP* adalah bahasa *scripting* yang terintegrasi dengan *tag-tag HTML*.

7. *MySQL*

MySQL atau *structure query language* adalah *database* yang sering digunakan oleh *programer* untuk media penyimpanan data dibandingkan *database* yang lainnya dengan alasan dinilai lebih stabil dan sangat kuat. *MySQL* tergolong *software RDBMS* yang mempunyai peran sebagai *server database* yang juga mempunyai lisensi di bawah GNU *General Public License* (GPL) dan juga mempunyai sifat *open source* dengan fitur *multiplatform*, andal, cepat, dan mudah digunakan (Meidyan Permata Putri et al., 2023).

MySQL merupakan sistem manajemen *database* yang berguna untuk menyimpan data dalam tabel terpisah dan mempunyai fungsi menempatkan semua data dalam satu ruang yang besar dengan memiliki keunggulan yaitu bersifat *scalable* atau bisa di naik atau di turunkan skalanya sesuai dengan kebutuhan (Rahmatiyah et al., 2023).

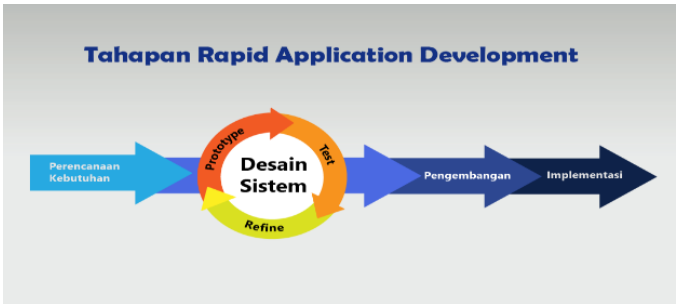
Dari kedua definisi di atas dapat disimpulkan bahwa *MySQL* tergolong *software RDBMS* yang mempunyai peran sebagai *server*

database yang juga mempunyai lisensi di bawah GNU *General Public License* dan juga mempunyai sifat *open source* dengan fitur *multiplatform*, andal, cepat, dan mudah digunakan. *MySQL* merupakan sistem manajemen *database* yang berguna untuk menyimpan data dalam tabel terpisah dan mempunyai fungsi menempatkan semua data dalam satu ruang yang besar dengan memiliki keunggulan yaitu bersifat *scalable* atau bisa di naik atau di turunkan skalanya sesuai dengan kebutuhan.

8. *RAD (Rapid Application Development)*

Rapid Application Development (RAD) merupakan salah satu metode pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak dengan proses perkembangan bersifat *software* sekuensial linier, yaitu menekankan siklus perkembangan yang sangat pendek. Dengan kata lain, model *RAD* ini merupakan model pengembangan sistem dengan adaptasi kecepatan tinggi (Habibi & Aprilian, 2020).

Metode *RAD* merupakan metode pengembangan sistem yang memanfaatkan konsep inkremental iteratif dengan menekankan tenggat waktu dan efisien biaya sesuai dengan kebutuhan (Utami & Zein, 2023). Adapun tahapan-tahapan dalam proses pengembangan sistem menggunakan metode *rapid application development* bisa dilihat dalam gambar 2.1 berikut.



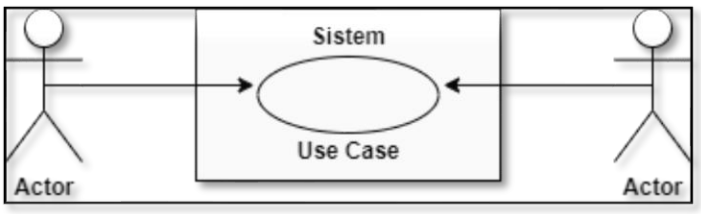
Gambar 2.1. Tahapan *RAD*

Sumber : (Utami & Zein, 2023)

9. *UML (Unified Modelling Language)*

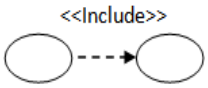
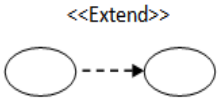
a. Use Case Diagram

Use Case Diagram merupakan Salah satu jenis diagram UML yang digunakan untuk menggambarkan aspek dinamis sistem. *Use case* berguna untuk memvisualisasikan dan mendeskripsikan sistem pada tahap analisis dan perancangan (Dias et al., 2023). Adapun gambar mengenai *use case* diagram dapat dilihat pada gambar 2.2 seperti berikut.



Gambar 2.2. *Use Case* Model

Sumber : (Saputra & Fahrizal, 2021)

Simbol	Keterangan
	UseCase: menggambarkan bagaimana seseorang akan menggunakan/memanfaatkan sistem.
	Aktor: seseorang/sesuatu yang berinteraksi dengan sisitem yang sedang kita kembangkan.
	Relasi: sebagai penghubung antara aktor-*usecase*, *usecase-usecase* dll.
	Relasi Asosiasi: relasi terjadi antara aktor dengan *usecase* biasanya berupa garis lurus dengan kepala panah disalah satu ujungnya.
	*Include Relationship* (relasi cakupan): memungkinkan suatu *usecase* untuk menggunakan fungsionalitas yang disediakan oleh *usecase* yang lainnya.
	*Extend Relationship*: memungkinkan *usecase* memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh *usecase* yang lainnya.

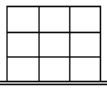
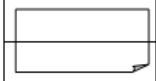
Gambar 2.3. Simbol *Use Case* Diagram

Sumber : (Saputra & Fahrizal, 2021)

b. *Activity* Diagram

Activity Diagram merupakan diagram yang dapat memodelkan prose 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 dilhat pada gambar 2.4 berikut.

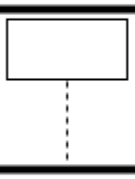
Simbol	Keterangan
	Lingkaran penuh. Memulai proses dalam suatu diagram aktivitas.
	Mata banteng. Akhir dari proses.
	Segi empat panjang, kejadian, aktivitas, atau pemicu.
	Berlian. Sebuah cabang.
	**Tabel. Suatu *file* komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis.**
	Dokumen. Menunjukan dokumen sumber atau laporan.
	Garis putus-putus. Alur informasi antar kejadian.
	Garis tidak terputus. Urutan dari satu kejadian atau aktivitas ke yang berikutnya.
	Catatan. Memberikan acuan bagi pembaca pada diagram atau dokumen lain untuk memperincinya

Gambar 2.4. Simbol *Activity* Diagram

Sumber : (Saputra & Fahrizal, 2021)

c. *Sequence* Diagram

Menurut (Acharya, 2024) *Sequence* Diagram merupakan diagram interaksi yang menekankan pada pengurutan waktu pesan yang secara grafis *sequence* diagram merupakan tabel dengan objek yang disusun memanjang. Diagram interaksi menggambarkan urutan pesan yang dipertukarkan antar peran tersebut mengimplementasikan suatu sistem. Adapun gambar mengenai simbol sequence dapat dilihat pada gambar berikut.

Simbol	Keterangan
	Aktor : Seseorang / sesuatu yang akan berinteraksi dengan sistem yang sedang kita kembangkan
	Objek : Menambah objek baru pada diagram
	Aktivasi : Menggambarkan langkah-langkah dalam aliran kerja
	Pesan : Menggambarkan pesan antara dua objek

Gambar 2.5. Simbol Sequence Diagram

Sumber : (Saputra & Fahrizal, 2021)

d. Class Diagram

Menurut Ahmad, (2022) *Class* Diagram merupakan sebuah alat untuk membantu memvisualisasikan maupun mengambarkan struktur dari sebuah kelas yang terdapat pada suatu sistem atau program dengan menggambarkan kelompok suatu objek bersama properti, operasi dan relasi yang sama. *Class* diagram juga merupakan sebuah penggambaran dari struktur sistem dengan mendefinisikan kelas yang hendak dibangun sistem/aplikasi. Adapun simbol dan penjelasan mengenai *Class* Diagram dapat dipelajari pada Gambar 2.6 berikut.

Simbol	Keterangan
Association	Merupakan Class-class yang saling terhubung satu sama lain secara konseptual.
Class: Classname + field: type + field: type + field: type	Merupakan kumpulan objek yang memiliki atribut dan operation yang sama. Atribut, merupakan *property* dari sebuah *class*. *Operation* merupakan sesuatu yang bisa dilakukan oleh sebuah *class*.
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 ----------Use--------->	Merupakan sebuah garis yang berfungsi menunjukkan operasi pada suatu class yang menggunakan class lain.
Generalization ----------Extends----------▷	Merupakan suatu garis yang berfungsi untuk mewariskan struktur data dan objek induk kepada objek anak yang dituju.

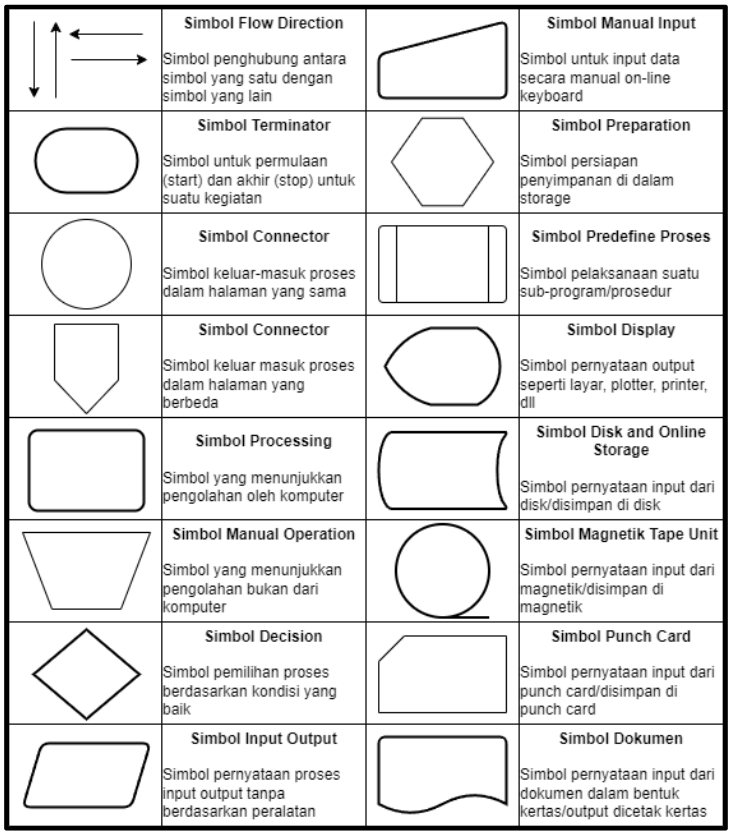
Gambar 2.6. Simbol *Class* Diagram

Sumber : (Wandari, 2023)

e. Flowchart

Flowchart atau diagram alir merupakan jenis diagram yang mewakili suatu algoritma instruksional dalam suatu sistem. Seorang analis sistem menggunakan diagram alur sebagai bukti dokumen untuk menjelaskan gambaran logis dari sistem yang dibuat kepada pemrogram. Dengan cara ini, diagram alur dapat memberikan solusi atas masalah yang timbul selama pembangunan sistem (Rosaly & Prasetyo, 2019).

Menurut Fauzi, (2020) diagram alur adalah cara menulis algoritma menggunakan simbol grafis. *Flowchart* adalah gambaran atau diagram yang menunjukkan urutan atau langkah-langgah suatu program dan hubungan antara proses dan informasinya. Deskripsi ini ditunjukkan dengan simbol. Setiap simbol mewakili suatu proses. Pada titik ini, proses dan garis yang terhubung ditampilkan. Dengan menggunakan diagram alur, akan lebih mudah untuk memeriksa bagian-bagian analisis masalah yang terlewatkan. Adapun simbol dan penjelasan mengenai diagram alur akan dijelaskan pada gambar 2.7 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 2.7 Simbol *Flowchart*

Sumber : (Fauzi, 2020)

10. *System Usability Scale (SUS)*

Menurut Beny, (2019) *system usability scale* merupakan metode pengujian yang sering digunakan untuk menilai kegunaan (*usability*) suatu sistem dengan memiliki beberapa kelebihan seperti proses kalkulasi yang sederhana, memiliki hasil luaran berupa skala skor dari 0-100 sehingga lebih mudah untuk dipahami, tidak memerlukan biaya dalam penggunaannya, dan memiliki ukuran sampel yang relatif kecil tetap terbukti *valid* dan *reliable*.

System Usability Scale (*SUS*) adalah metode pengujian sistem secara sederhana menggunakan sepuluh skala yang memberikan pandangan secara menyeluruh dari evaluasi tujuan kebergunaan. *SUS* berupa skala *likert* yang sederhana dengan responden yang diharuskan untuk menjawab kesetujuan dan ketidaksetujuan dalam skala 5 atau 7 poin (Salamah, 2019). Adapun proses untuk pengujian menggunakan *SUS* akan berlaku persamaan seperti berikut.

Skor *System Usability Scale* =

$$((\boldsymbol{R1} - \mathbf{1}) + (\mathbf{5} - \boldsymbol{R2}) + (\boldsymbol{R3} - \mathbf{1}) + (\mathbf{5} - \boldsymbol{R4}) + (\boldsymbol{R5} - \mathbf{1}) + (\mathbf{5} - \boldsymbol{R6}) + (\boldsymbol{R7} - \mathbf{1}) + (\mathbf{5} - \boldsymbol{R8}) + (\boldsymbol{R9} - \mathbf{1}) + (\mathbf{5} - \boldsymbol{R10})) * (\mathbf{2}, \mathbf{5}) \quad (2.5)$$

$$\boldsymbol{Skor\ akhir} = (\boldsymbol{Total\ Skor} \times \boldsymbol{Faktor\ Koreksi}) \quad (2.6)$$

$$\boldsymbol{Rata\ rata\ Skor} = \frac{Skor\ Akhir}{Jumlah\ Responden} \quad (2.7)$$

B. Kajian Empiris

Berdasarkan penelitian yang dilakukan oleh Nabila, (2022) untuk menentukan pemilihan duta Generasi Berencana (Genre) dengan menggunakan metode *profile matching*, terdapat 2 aspek kriteria yaitu aspek pengetahuan dan aspek kepribadian dengan 5 sub kriteria yaitu *public speaking*, berbahasa inggris, wawasan, *atitude*, dan inisiatif. Penelitian tersebut mendapatkan hasil akhir dengan terpilihnya duta Genre laki-laki dan juga duta Genre Perempuan dengan menggunakan perhitungan metode *profile matching*.

Penelitian selanjutnya yaitu dilakukan oleh (Harahap et al., 2024) sebagai pendukung keputusan dalam pemilihan alat kontrasepsi agar tidak terjadi kehamilan dengan tujuan mengurangi pertumbuhan penduduk juga menggunakan *profile matching* sebagai metode sistem pendukung keputusan. Penelitian tersebut bertujuan untuk pemilihan alat kontrasepsi untuk keluarga berencana dengan mempertimbangkan aspek-aspek penilaian yang dilakukan. Sistem dibangun untuk menentukan alat kontrasepsi yang dapat memberikan kemudahan kepada peserta program keluarga berencana, dapat merumuskan permasalahan yang dihadapi oleh pserta program keluarga berencana dalam menerapkan sistem pendukung keputusan dalam menentukan alat kontrasepsi, sistem yang dirancang oleh penulis upaya untuk menghasilkan *output* yang lebih berguna dan bermanfaat bagi peserta program keluarga berencana.

Berdasarkan penelitian yang dilakukan oleh (Badrul, 2021) yaitu tentang seleksi pegawai, menyimpulkan bahwa metode *profile matching* dapat memberikan alternatif keputusan terbaik dalam pengambilan keputusan untuk menentukan kandidat calon ketua OSIS untuk disaring menjadi 3 calon kandidat terbaik. Hasil dari perhitungan menggunakan *profile matching* untuk penelitian tersebut merupakan perangkingan tertinggi ke rendah dan nilai tertinggi merupakan hasil yang dibutuhkan sebagai bahan pertimbangan untuk mentukan kandidat ketua OSIS.

C. Kerangka Berfikir

Pemilihan Putra Putri Unipma merupakan kegiatan yang penting untuk memilih kandidat yang mencerminkan nilai-nilai kampus. Namun, proses seleksi seringkali bersifat subjektif dan kurang sistematis. Hal tersebut menjadi sebuah ide dari penulis untuk mengatasi masalah tersebut yaitu dengan membuat sebuah sistem yang dapat membantu teman-teman panitia dalam membuat keputusan yang objektif dan transparan. Dengan demikian, penulis mempunyai maksud untuk membuat sistem pendukung keputusan yang menerapkan *profile matching* sebagai metode untuk pengambilan keputusan.

Dalam tahapan maupun proses perancangannya, penulis akan mencakup beberapa langkah diantaranya, analisis, desain, pengembangan, pelaksanaan dan juga evaluasi. Langkah pertama yang dilakukan yaitu dengan mengidentifikasi masalah. Peneliti sudah melakukan analisis mengenai permasalahan tersebut dan msudah mendapatkan hasil yang

dimana hasil analisis itu akan diterapkan ke dalam implementasi sistem ini. Selanjutnya peneliti melakukan analisis mengenai kebutuhan sistem. Hal ini mencakup hal-hal yang berkaitan dengan data yang kemudian akan dilakukan pengumpulan dan pengelompokkan.

Dalam penerapan *profile matching* ini, peneliti sudah memiliki kriteria yang didapatkan dari hasil pengamatan dan juga wawancara sebelum akhirnya akan digunakan dalam proses perhitungan untuk penerapan metode ini. Pada tahapan perhitungan akan melalui beberapa tahapan lagi yaitu melalui perumusan-perumusan yang sudah ditetapkan pada metode *profile matching*. Selanjutnya pada tahapan implementasi akan di aplikasikan dalam bentuk *website* yang akan dirancang menggunakan *PHP Laravel* dan akan melalui tahapan evaluasi sistem atau pengujian sistem dengan tujuan untuk mendapatkan hasil yang sesuai dengan kebutuhan.

Gambar 2.8. Kerangka Berpikir