

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Pendukung Keputusan

Sudah tidak asing bagi kalangan manajerial mengenai *Decision Support System* (DSS) atau Sistem Pendukung Keputusan (SPK) sudah menjadi *tool* bagi penunjang keputusan yang dipakai oleh para manajer, organisasi atau pun perusahaan dalam pengambilan keputusan. DSS merupakan “bagian dari sistem informasi berbasis komputer yang digunakan untuk memberikan penunjang dalam dunia bisnis maupun pengambilan keputusan dalam suatu perusahaan atau organisasi” (Andoyo et al., 2021:36).

Sistem pendukung keputusan adalah sistem komputer yang dapat memberikan saran atau rekomendasi dalam proses pengambilan keputusan (Hafiez et al., 2022:375). Menurut Bonczek (dalam Lubis et al., 2022:2), “mendefinisikan sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi yaitu sistem bahasa, sistem pengetahuan dan sistem pemrosesan masalah”. Dapat disimpulkan bahwa sistem pendukung keputusan dapat digunakan seseorang dalam mendukung pengambilan keputusan yang dapat memberikan penjelasan permasalahan yang ada dengan lebih cepat.

Sistem pendukung keputusan diciptakan dengan tingkat fleksibilitas dan adaptabilitas yang tinggi, sehingga mudah menyesuaikan diri dengan perubahan kebutuhan pengguna dan kondisi lingkungan (Indra & Anggara, 2023:20). Pemanfaatan teknologi dalam proses pengambilan keputusan sangat diperlukan agar keputusan dapat diambil lebih cepat dan kesalahan yang paling sedikit (Pibriana & Aswadi, 2021:114). Petunjuk pengambilan keputusan yang optimal akan sangat penting untuk mencapai kepuasan organisasi (Schelling et al., 2021:5). Kegiatan pengambilan keputusan meliputi identifikasi masalah, mencari alternatif pemecahan masalah, mengevaluasi alternatif pemecahan tersebut, dan memilih alternatif yang baik (Novirda et al., 2022:304). Adapun ciri-ciri dari sistem pendukung keputusan menurut Kusri (dalam Andoyo et al., 2021:36) sebagai berikut :

- a. Sistem pendukung keputusan bertujuan sebagai pembantu pengambilan keputusan yang dikatakan kurang terstruktur, terutama oleh manajer di tingkat tinggi.
- b. Sistem pendukung keputusan menerapkan kombinasi antara model kualitatif dan data.
- c. Sistem pendukung keputusan mempunyai fitur interaktif yang memudahkan interaksi manusia dengan komputer.
- d. Sistem pendukung keputusan mempunyai sifat fleksibel dan mudah menyesuaikan deformasi-deformasi yang berlangsung.

2. *Simple Additive Weighting (SAW)*

Konsep awal dari metode *Simple Additive Weighting (SAW)* adalah “mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut” (F. Susanto, 2021:8). Teknik *Simple Additive Weighting (SAW)* sering difungsikan saat menghadapi situasi *Multiple Attribute Decision Making (MADM)*.

Metode *Simple Additive Weighting (SAW)* atau sering dikenal dengan metode penjumlahan tertimbang merupakan bagian dari *Multi Criteria Decision Making (MCDM)* yang menghitung penjumlahan bobot kinerja setiap alternatif terhadap seluruh kriteria yang dimiliki (Widarma et al., 2021:69). Teknik *Simple Additive Weighting (SAW)* memerlukan normalisasi matriks keputusan (X) dengan skala yang sebanding untuk semua alternatif yang tersedia (Nugraha, 2021:18). Teknik SAW dianggap sebagai salah satu cara yang dapat menghasilkan pemeringkatan yang tergolong lebih akurat melalui berbagai kriteria evaluasi, dan hasil akhirnya ditentukan dengan melakukan teknik *Simple Additive Weighting (SAW)* sehingga memaparkan hasil yang lebih akurat (Fajri, 2021:163). Dalam teknik *Simple Additive Weighting (SAW)*, terdapat kelebihan dan kekurangan dalam penerapannya. Berikut adalah kelebihan dan kekurangan teknik *Simple Additive Weighting (SAW)* menurut (Susanto, 2021:9) :

a. Kelebihan

- 1) Sederhana : Perhitungan cara *Simple Additive Weighting* (SAW) hanya empat langkah, diawali dengan menetapkan kriteria, menetapkan peringkat kesesuaian, menciptakan matriks keputusan, dan menghitung peringkat berdasarkan penjumlahan perkalian yang dinormalisasikan.
- 2) Penentuan bobot : Metode ini memungkinkan penentuan bobot untuk setiap atribut. Selanjutnya, opsi terbaik dapat dipilih berdasarkan peringkat yang dihasilkan.
- 3) Akurasi : Evaluasi menggunakan metode SAW lebih akurat sebab didasarkan pada prioritas bobot yang telah ditetapkan.
- 4) Normalisasi : Metode SAW juga melibatkan normalisasi matriks berdasarkan nilai atribut (termasuk *benefit* dan *cost*).

b. Kelemahan

- 1) Perhitungan : Metode SAW dapat menggunakan bilangan *crisp* dan *fuzzy*.
- 2) Normalisasi : Terdapat perbedaan dalam perhitungan normalisasi matriks, tergantung pada nilai atribut (baik nilai *benefit* dan *cost*).

Berdasarkan uraian kelebihan dan kekurangan teknik *Simple Additive Weighting* (SAW) di atas mengandung istilah-istilah yang perlu dipahami lebih lanjut.

- a. Kriteria : ukuran yang digunakan untuk penilaian. Kriteria dapat dikategorikan menjadi *benefit* dan *cost*. Semakin tinggi benefit, semakin baik, sedangkan untuk *cost* harus semakin kecil.
- b. Alternatif : solusi atau pilihan yang akan dievaluasi.
- c. Atribut : faktor-faktor yang digunakan untuk mengukur kinerja alternatif.
- d. Data *crisp* : data yang bebas dari dimensi dan digunakan untuk normalisasi nilai atribut.

3. *Website*

Secara terminologi, sebuah *website* merupakan sekumpulan halaman yang tergabung dalam satu domain atau subdomain di *World Wide Web* (WWW) di internet. Setiap slide halaman *website* berbentuk dokumen yang dituangkan dalam format *Hyper Text Markup Language* (HTML) kemudian dapat diakses melalui protocol HTTP. Protokol ini memungkinkan informasi dari *server website* ditunjukkan kepada pengguna melalui *web browser*.

Menurut Rohi Abdullah (dalam Sa'ad, 2020:3), *website* atau *web* adalah kumpulan halaman berisi informasi yang dijelaskan dalam bentuk data digital meliputi teks, gambar, video, file audio dan animasi. Informasi ini dapat tersampaikan melalui bantuan koneksi internet. *Website* menurut

Yuhefizar (dalam Sa'ad, 2020:3) *website* mencakup seluruh halaman *website* yang tersedia dalam suatu *domain* dan memuat berbagai berita.

Menurut Novendri (dalam Haslindah et al., 2022:198) *website* adalah kumpulan halaman yang menggunakan bahasa pemrograman terkait dan berfungsi menayangkan informasi, gambar bergerak, gambar diam, suara, atau kombinasi seluruhnya. *Website* dapat bersifat statis atau dinamis.

Dengan demikian, disimpulkan bahwa *website* adalah sebuah laman atau wadah yang menyediakan informasi di internet. Internet diibaratkan sebagai sebuah pusat informasi terbesar di dunia sehingga seseorang dapat mengakses *website* tanpa terpaut waktu dan tempat asalkan terhubung dengan internet.

4. *Database*

Database dapat dianggap sebagai 'jiwa' dari sebuah aplikasi. Hal ini karena melalui penggunaan *database*, semua fitur, alat, menu, dan fasilitas lainnya dalam aplikasi dapat terhubung. *Database* bukan hanya tempat penyimpanan data semata. *Database* merupakan komponen terpenting ketika membangun suatu sistem informasi karena di sinilah semua data dalam sistem disimpan dan diatur serta tempat *programmer* dapat mencari dan mengedit informasi dalam berbagai format (Ricki & Devitra, 2019:256).

Database adalah kombinasi data yang tersimpan secara sistematis pada komputer pribadi. Data ini diproses atau dimanipulasi oleh aplikasi


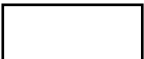

untuk menghasilkan informasi (Kurniawan et al., 2022:304). *Database* digunakan untuk memfasilitasi pengguna yang memerlukan pemrosesan data, baik untuk analisis maupun evaluasi.

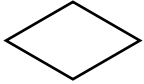



5. *Flowchart*

Flowchart adalah diagram yang mengilustrasikan proses kerja atau sistem terkait. *Flowchart* digunakan sebagai gambaran dalam menyelesaikan suatu masalah (Safira & Purtingrum, 2023:19). Pengertian *flowchart* menurut Putri et al., (2022:19) “*Flowchart* merupakan diagram alir dengan tampilan grafis yang digambarkan secara sistematis dan digunakan untuk menggambarkan suatu proses logika program yang disusun secara berurutan sehingga bisa dipelajari untuk mendapatkan pemecahan masalah dalam pengembangan aplikasi atau program”.

Oleh sebab itu, dapat disimpulkan bahwa *flowchart* adalah teknik representasi visual dari algoritma menggunakan simbol yang telah ditetapkan. Simbol-simbol yang sering digunakan dalam *flowchart* dijelaskan lebih lanjut pada tabel 2.1.

Tabel 2.1 Simbol *Flowchart*

No	Nama	Keterangan	Simbol
1.	Terminator	Simbol dalam diagram alur mewakili titik terminal, seperti titik awal, titik akhir, berhenti, tunda, atau interupsi.	
2.	<i>Process</i>	Simbol yang mewakili segala jenis fungsi pemrosesan.	
3.	<i>Document</i>	Simbol yang mewakili fungsi <i>output</i> dan <i>input</i> dimana mediana berupa dokumen	

4.	<i>Decision</i>	Simbol keputusan mewakili operasi tipe <i>switching</i> yang menentukan jalur alternatif mana yang harus diambil.	
5.	Data	Simbol masukan/keluaran merepresentasikan fungsi <i>input/output</i> , yaitu menyediakan informasi untuk diproses (<i>input</i>) atau merekam informasi yang telah diproses (<i>output</i>).	
6.	<i>Connector</i>	Simbol ini mewakili jalan keluar ke atau <i>entry</i> dari bagian lain diagram alur.	
7.	<i>Flowline</i>	Simbol yang menghubungkan antara simbol-simbol menunjukkan urutan informasi dan operasi yang tersedia dan dapat dieksekusi.	

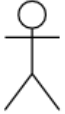
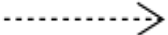

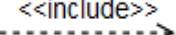
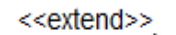




Sumber : Putri et al., (2022)


Penggunaan *flowchart* memiliki beberapa manfaat, antara lain memudahkan komunikasi saat menyelesaikan masalah dalam aplikasi, meningkatkan efektivitas dalam analisis, dan mempercepat penulisan kode karena pemetaan masalah dan instruksi program yang jelas.

6. Use Case Diagram

Diagram *use case* mengilustrasikan aktivitas yang dikerjakan oleh sistem informasi dari sudut pandang eksternal (Aulia et al., 2022:281). Use case diagram mengilustrasikan subjek dan objek untuk kejelasan mengoperasikan sistem (Aseprianto, 2021:760). *Use case* adalah fitur sistem yang memungkinkan pengguna atau aplikasi memahami aplikasi yang mereka bangun (Harmayani & Armadi, 2020:143). Ini memungkinkan pengguna dan perancang sistem untuk memahami alur sistem yang akan dibuat. Simbol-simbol *use case* tercantum dalam tabel 2.2.

Tabel 2.2 Simbol *Use Case* Diagram.

No	Nama	Keterangan	Simbol
1.	<i>Actor</i>	Mengilustrasikan tugas seseorang saat berinteraksi dengan <i>use case</i> .	
2.	<i>Dependency</i>	Relasi dependensi ini terjadi ketika perubahan pada komponen tunggal mempengaruhi komponen yang bergantung.	
3.	<i>Generalization</i>	Relasi pewarisan ini terjadi ketika objek anak (<i>descendant</i>) mewarisi watak dan struktur data dari objek yang berada di objek induk (<i>ancestor</i>).	
4.	<i>Include</i>	Spesifikasi <i>use case</i> utama menjelaskan secara eksplisit mengenai <i>use case</i> utama.	
5.	<i>Extend</i>	Spesifikasi <i>use case</i> target menyatakan bagaimana <i>use case</i> target memperluas perilaku dari <i>use case</i> utama pada titik tertentu.	
6.	<i>Association</i>	Mengilustrasikan koneksi antara objek satu dengan objek lainnya.	
7.	<i>System</i>	Sistem menyebutkan paket yang membatasi tampilan.	
8.	<i>Use Case</i>	Deskripsi urutan aksi menjelaskan urutan langkah-langkah yang menimbulkan hasil terukur bagi seorang aktor.	
9.	<i>Collaboration</i>	Sinergi komponen mengilustrasikan interaksi aturan dan komponen lain yang terkoneksi untuk memberikan perilaku	


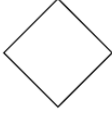
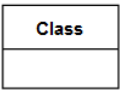

		yang lebih besar daripada jumlah komponennya.	
10.	<i>Note</i>	Komponen fisik merujuk pada komponen fisik yang tersedia ketika aplikasi dijalankan dan memperlihatkan sumber daya komputasi.	

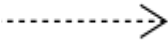

Sumber : Sa'ad (2020)

7. *Class Diagram*

Diagram kelas atau *class diagram* adalah variasi diagram struktural dalam *Unified Modeling Language* (UML) yang dengan jelas mengilustrasikan struktur dan deskripsi kelas, atribut, metode, serta hubungan antar objek (Sumirat et al., 2023:88). *Class diagram* menjelaskan kelas-kelas yang termuat dalam pengembangan rekayasa perangkat lunak (Nasri & Kania, 2019:126). Berikut penjelasan singkat simbol-simbol *class diagram* tercantum pada tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

No	Nama	Keterangan	Simbol
1.	<i>Generalization</i>	Relasi objek anak (<i>descendant</i>) yang mewarisi watak dan struktur data awal objek yang berada di objek induk/ <i>ancestor</i> .	
2.	<i>N-ary Association</i>	Untuk menghindari hubungan dengan lebih dari dua objek.	
3.	<i>Class</i>	Komponen dari objek yang berbagi atribut dan operasi yang sama.	
4.	<i>Collaboration</i>	Ilustrasi langkah-langkah yang ditunjukkan oleh sistem dan menimbulkan hasil yang terukur bagi suatu aktor.	

5.	<i>Dependency</i>	Relasi perubahan pada komponen tunggal memengaruhi komponen yang bergantung.	
6.	<i>Association</i>	Menyambungkan antara satu objek dengan objek lainnya.	




Sumber : Sa'ad (2020)



8. *Activity Diagram*

Activity diagram atau diagram aktivitas mengilustrasikan seluruh aliran kerja sistem yang diciptakan, termasuk kerja setiap aliran diawali, keputusan yang diambil, dan kerja sistem berakhir (Kurniawan, 2020:199). Setiap *use case* biasanya memiliki setidaknya satu diagram aktivitas. Selain itu, diagram ini menerangkan teknik bisnis dan urutan aktivitas dalam suatu proses. Namun, perlu dicatat bahwa *activity* diagram tidak menerangkan teknik dan jalur aktivitas dari level atas secara umum. Sebaliknya, fokusnya adalah mengilustrasikan teknik yang sedang berlangsung, sementara itu *use case* memaparkan bagaimana aktor menjalankan sistem untuk melakukan aktivitas.

Terdapat beberapa simbol-simbol penting yang harus diketahui sebelum membuat *activity* diagram sesuai pada tabel 2.4.

Tabel 2.4 Simbol *Activity* Diagram

No	Nama	Keterangan	Simbol
1.	<i>Activity</i>	Menunjukkan cara kelas UI terhubung satu sama lain.	
2.	<i>Action</i>	Menunjukkan eksekusi dari suatu objek dalam sistem.	
3.	<i>Initial Node</i>	Menjelaskan cara objek dibentuk atau diinisialisasi.	

4.	<i>Activity Final Node</i>	Mengilustrasikan cara objek dihancurkan atau dihapus.	
5.	<i>Fork Node</i>	Menunjukkan alur yang pada tahap tertentu dapat berubah menjadi beberapa alur.	



Sumber : Sa'ad (2020)

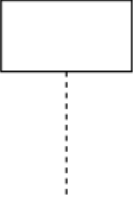


9. Sequence Diagram

Pengertian *sequence* diagram menurut Muhammad (dalam Sa'ad, 2020:52) *sequence* diagram atau diagram urutan mengilustrasikan interaksi antar objek dan komunikasi antar objek-objek tersebut. Diagram ini secara rinci menjabarkan interaksi antar objek dalam suatu sistem (Sumirat et al., 2023:86). Selain itu, *sequence* diagram juga menampilkan urutan pesan yang dikirim oleh objek untuk melakukan tugas atau tindakan tertentu.

Seluruh objek disusun mulai kiri ke kanan, dengan aktor yang memulai interaksi dan komunikasi ditempatkan di bagian paling kiri dalam diagram. Berikut komponen simbol utama yang sering digunakan dalam *sequence* diagram sesuai pada tabel 2.5.

Tabel 2.5 Simbol *Sequence* Diagram

No	Nama	Keterangan	Simbol
1.	Objek Interaksi	Menunjukkan objek-objek yang berinteraksi, pesan, atau <i>instance</i> dari sebuah kelas dengan nama objek di bagian dalamnya.	
2.	Aktor	Mendeskripsikan seseorang, cara atau sistem lain yang komunikasi dengan sistem informasi yang akan diciptakan. Perlu dicatat bahwa meskipun aktor	

		digambarkan sebagai visual orang, belum tentu itu selalu mengacu pada individu manusia.	
3.	<i>Lifeline</i>	Menyatakan keberadaan suatu objek atau menunjukkan kehidupan objek. <i>Lifeline</i> biasanya berupa titik-titik garis vertikal yang ditarik dari objek.	
4.	<i>Activation/Objek Aktif</i>	Digambarkan sebagai segi empat yang disimpan di <i>lifeline</i> , menunjukkan objek kondisi aktif dan berkomunikasi melalui pesan.	
5.	<i>Message/Pesan</i>	Digambarkan sebagai anak panah, menyatakan komunikasi antar objek-objek.	

Sumber : Sa'ad (2020)

Berdasarkan gambar 2.5 *sequence* diagram berfungsi ketika mengilustrasikan serangkaian langkah sebagai respon terhadap suatu kejadian dengan tujuan menghasilkan output tertentu. Diagram ini membantu memvisualisasikan interaksi antar objek dan pesan yang dikirim di dalam sistem.

10. *Hypertext Preprocessor (PHP)*

Pengertian PHP menurut Enterprise (2019:184) “PHP merupakan bahasa pemrograman yang digunakan untuk membuat *website* dinamis dan interaktif. Dinamis artinya, *website* tersebut dapat berubah-ubah tampilan kontennya sesuai kondisi tertentu dan interaktif artinya dapat memberi *feedback* bagi *user* (misalnya menampilkan hasil pencarian produk)”.

Singkatnya bahasa pemrograman PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang banyak digunakan untuk menciptakan dan mengembangkan situs *web*. PHP disisipkan langsung ke dalam kode HTML, dan tag HTML digunakan untuk mengelilingi kode PHP yang tersedia dalam file HTML (Nadhif & Fiati, 2022:2). Ada beberapa tipe data yang dikenal dalam lingkungan PHP. Tipe data tersebut antara lain data *integer*, *string*, *array*, dan *boolean*.

Pengolahan data dalam PHP dilakukan menggunakan berbagai operator sesuai pada tabel 2.6.

Tabel 2.6 Operator PHP

No	Operator	Kegunaan	Keterangan
1.	Aritmatika	Untuk memulai operasi matematika	+ (Penjumlahan), - (Pengurangan), * (Perkalian), / (Pembagian)
2.	Penugasan	Untuk mengisi nilai ke variabel	= (Penugasan), += (Penugasan dengan Penjumlahan), -= (Penugasan dengan Pengurangan), *= (Penugasan dengan Perkalian), /= (Penugasan dengan Pembagian)
3.	Perbandingan	Untuk membandingkan dua nilai	== (Sama Dengan), != (Tidak Sama Dengan), < (Kurang Dari), > (Lebih Dari), <= (Kurang Dari atau Sama Dengan), >= (Lebih Dari atau Sama Dengan)
4.	Logika	Untuk menggabungkan kondisi	&& (AND), (OR), ! (NOT)

5.	<i>Increment dan Decrement</i>	Untuk menambah atau mengurangi nilai variabel	++ (<i>increment</i>), -- (<i>decrement</i>)
6.	<i>Concatenation</i>	Untuk menggabungkan string	. (titik)

Sumber : Enterprise (2019)

Berdasarkan tabel 2.6 menunjukkan bahwa bahasa pemrograman PHP memiliki banyak operator yang dapat digunakan oleh programmer dalam membangun suatu *website*. PHP bertumbuh dan bekerja pada perencanaan *website* yang kuat, bukan untuk membuat halaman *web* sederhana saja, melainkan untuk situs *web* yang dikenal dan digunakan oleh jutaan orang (Margaretha & Nababan, 2020:25).

11. MySQL

MySQL merupakan *database* pertama yang mendukung bahasa pemrograman *scripting* (PHP dan Perl) untuk internet (Lestari & Murti, 2020:5). MySQL adalah sistem manajemen basis data open source berdasarkan *Structured Query Language* (SQL) (Gopi et al., 2023:4). Bahasa pemrograman *scripting* PHP biasa digunakan dalam pengembangan aplikasi. Menurut Nugroho (dalam Sa'ad, 2020:35), pengertian MySQL adalah program pembuatan *database open source* yang dapat digunakan siapa saja.

Pengertiannya lainnya menurut Arief (dalam Sa'ad, 2020:36) menjelaskan MySQL merupakan salah satu golongan server *database* yang familiar dan mayoritas menggunakannya sebagai alternatif menciptakan aplikasi *web* yang melibatkan *database* sebagai pondasi

pengelolaan data. MySQL juga dapat digunakan sebagai *Relational Database Management System* (RDBMS) karena dengan membuat *database* di MySQL maka dapat memilih tabel dua dimensi yang berbeda (Supriady et al., 2019:2).

Dengan demikian, MySQL umumnya digunakan untuk membangun aplikasi *website*. MySQL dan PHP sering dianggap sebagai kombinasi perangkat lunak yang ideal untuk mengembangkan situs *web*.

B. Kajian Empiris

Penelitian yang membahas tentang sistem pendukung keputusan dalam memilih produk perawatan kulit telah banyak diterapkan sehingga dijadikan bahan referensi. Perancangan dan pengembangan sistem semacam ini memerlukan penelitian yang tepat dari berbagai sumber penelitian yang relevan.

Penelitian sebelumnya oleh Rahayu et al. (2022) membahas tentang sistem rekomendasi untuk memilih produk perawatan kulit dasar. Hasil penelitian menunjukkan bahwa menerapkan cara *Simple Additive Weighting* (SAW) dalam sistem rekomendasi pemilihan produk *skincare* berdasarkan jenis kulit dapat menjadi solusi pengguna dalam menemukan produk yang cocok dengan kebutuhan mereka. Sistem ini melibatkan tahap persiapan data, analisis data, dan perankingan untuk menentukan produk *skincare* dengan rating tertinggi berdasarkan kriteria yang dipilih oleh pengguna. Pengujian sistem menerangkan bahwa cara *Simple Additive Weighting* (SAW) efektif dalam memberikan rekomendasi *skincare*. Saran untuk pengembangan sistem

meliputi penyediaan informasi lebih lanjut mengenai masalah kulit wajah dan eksplorasi penggunaan algoritma selain *Simple Additive Weighting* (SAW).

Penelitian selanjutnya dilaksanakan oleh Maarif et al. (2019) mengenai penggunaan teknik logika *fuzzy* untuk sistem pendukung keputusan pemilihan *skincare*. Hasil penelitian ini yaitu pengembangan sistem pendukung keputusan pemilihan *skincare* berbasis *android* menggunakan logika *fuzzy* bertujuan untuk membantu pengguna dalam memilih jenis produk *facial foam* yang cocok untuk jenis kulit pengguna. Aplikasi ini diharapkan dapat memberikan hasil yang akurat berdasarkan jawaban pengguna, namun tidak memberikan solusi penyembuhan untuk permasalahan kulit.

Penelitian oleh Imania et al. (2023) tentang sistem pendukung keputusan pemilihan *skincare* yang dikhususkan untuk tipe kulit sensitif di klinik kecantikan La Beautee Care menerapkan teknik *Simple Additive Weighting* (SAW) dan *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS). Hasil penelitian menunjukkan bahwa teknik SAW lebih relevan daripada teknik TOPSIS dalam pemilihan *skincare* untuk wajah sensitif di klinik tersebut. Hasil penelitian ini dapat membantu dalam memilih produk *skincare* terbaik untuk tipe kulit wajah sensitif, dengan hasil perhitungan menunjukkan bahwa produk Ponds adalah paling cocok berdasarkan nilai V terbesar.

Penelitian yang dilaksanakan oleh Purnomo et al. (2021) penelitian tersebut menerapkan teknik *Simple Additive Weighting* (SAW) untuk memilih produk *skincare*. Hasil penelitian menunjukkan bahwa *brand* Wardah dengan

produk Wardah Renew You menjadi pilihan terbaik dengan preferensi tertinggi yaitu 3,7 untuk remaja milenial. Kemudian *brand* Garnier dengan produk Garnier Sakura White dengan preferensi 3,6 menjadi pilihan kedua untuk remaja milenial.

Penelitian yang dijalankan oleh Susanto et al. (2023) mengenai penggunaan cara *Simple Additive Weighting* (SAW) yang diterapkan pada sistem pendukung keputusan menentukan rekomendasi pemilihan produk *skincare* di Klinik Almahira Skincare. Dari hasil penilaian, alternatif A1 (*Radiance White Sunscreen & Radiance White Night*) memiliki nilai tertinggi sebesar 1, sehingga dapat menjadi rekomendasi sebagai produk *skincare* yang cocok dengan kriteria yang telah ditetapkan.

Berdasarkan penelitian-penelitian di atas, dapat disimpulkan bahwa sistem pendukung keputusan memiliki peran krusial dalam membantu pengambilan keputusan di berbagai organisasi. Sistem pendukung keputusan tersebut dapat bekerja secara digital dan lebih akurat dengan menerapkan kriteria-kriteria pendukung keputusan dalam sistem.

C. Kerangka Berpikir

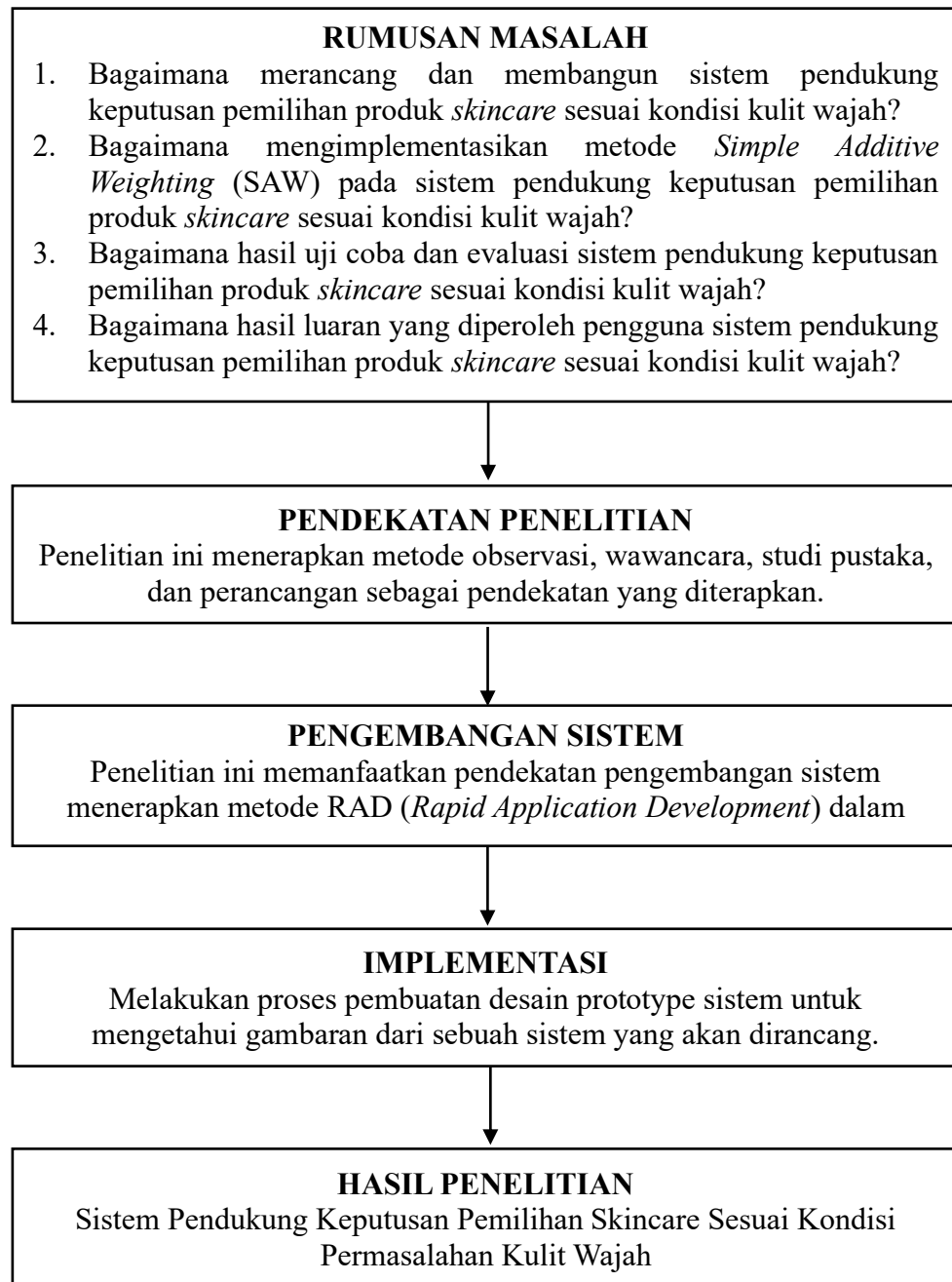
Perancangan dan pembuatan sistem pendukung keputusan untuk memilih *skincare* sesuai kondisi permasalahan kulit wajah ini diharapkan dapat mempermudah pengguna *skincare* dalam memilih produk *skincare* yang tepat. Di era maraknya *brand-brand skincare* yang saling menonjolkan keunggulan produknya. Sehingga menimbulkan dampak bagi pengguna *skincare* yang merasa kebingungan memilih produk yang sesuai dan tepat. Jika salah dalam

memilih produk *skincare* maka dapat menimbulkan dampak negatif pula bagi pengguna *skincare*.

Solusi dari permasalahan penelitian ini adalah dengan merancang dan mengembangkan sistem pendukung keputusan untuk memilih produk *skincare* dengan metode *Simple Additive Weighting* (SAW). SAW dipilih karena didasari oleh perhitungan yang relatif akurat dan mudah dalam penerapannya.

Pendekatan pengembangan sistem pendukung keputusan ini menerapkan teknik RAD (*Rapid Application Development*) karena berfokus pada pengembangan aplikasi secara terstruktur melalui pengulangan dan *feedback* berulang.

Berikut bagan kerangka berpikir pada penelitian ini.



Gambar 2.1 Kerangka Berpikir