

BAB II

Tinjau Pustaka

II.1. Penelitian Terdahulu

Konsep Sistem Pendukung Keputusan pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System* (Sprague, 1982). Konsep pendukung keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tidak terstruktur. SPK (Sistem Pendukung Keputusan) adalah pengembangan dari sistem informasi manajemen terkomputerisasi, yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakaiannya. Sifat interaktif ini untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan, seperti prosedur, kebijakan, teknik analisis, serta pengalaman dan wawasan manajerial untuk membentuk suatu kerangka keputusan yang bersifat fleksibel. kemudian penelitian oleh (Eko Darmanto) dari Universitas Maria Kudus, yang menjelaskan bagaimana penerapan metode AHP (*Analytic Hierarchy*) untuk menentukan kualitas gula tumbuh dengan metode model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki.

II.2. Landasan Teori

II.2. 1 *Analytical Hierarchy Process (AHP)*

AHP dikembangkan pada tahun 1970an oleh Dr Thomas L. Satty untuk menyediakan pendekatan sistematis untuk menentukan prioritas dan pengambilan keputusan dalam suatu kompleks lingkungan. AHP dirancang untuk mencerminkan cara berpikir orang sebenarnya. Analytical Hierarchy Process (AHP) Adalah metode untuk memecahkan suatu situasi yang kompleks tidak terstruktur kedalam beberapa komponen dalam susunan yang hirarki, dengan memberi nilai subjektif tentang pentingnya setiap variabel secara relatif, dan menetapkan variabel mana yang memiliki prioritas paling tinggi guna mempengaruhi hasil pada situasi tersebut.

Proses pengambilan keputusan pada dasarnya adalah memilih suatu alternatif yang terbaik. Seperti melakukan penstrukturan persoalan, penentuan alternatif-alternatif, penentuan nilai kemungkinan untuk variabel aleatori, penetapan nilai, persyaratan preferensi terhadap waktu, dan spesifikasi atas resiko. (Priyo Sutopo dkk, 2019).

Untuk kegiatan perbandingan antar sepasang objek, metode AHP memberikan sebuah standar nilai perbandingan antar dua objek seperti dituangkan pada tabel II.1.

Tabel II.1. Nilai Perbandingan

Intensitas kepentingan	Keterangan	Penjelasan
1	Kedua elemen sama Pentingnya	Dua elemen mempunyai pengaruh yang sama besar terhadap tujuan
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya	Pengalaman, penilaian dan penilaian sedikit menyokong satu elemen dibandingkan elemen lainnya
5	Elemen yang satu lebih penting daripada elemen yang lainnya	Pengalaman dan penilaian sangat kuat menyokong satu elemen dibandingkan elemen lainnya
7	Elemen yang satu jelas lebih mutlak penting daripada yang lainnya	Satu elemen yang kuat disokong dan dominan terlihat dalam praktek
9	Satu elemen mutlak penting daripada elemen lainnya	Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat penegasan tertinggi yang mungkin menguatkan
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang Berdekatan	Nilai ini diberikan bila ada dua pendapat di antara dua pilihan
Kebalikannya	Jika aktifitas i mendapat satu angka dibandingkan dengan aktifitas j , maka j memiliki kebalikannya dibandingkan dengan i	

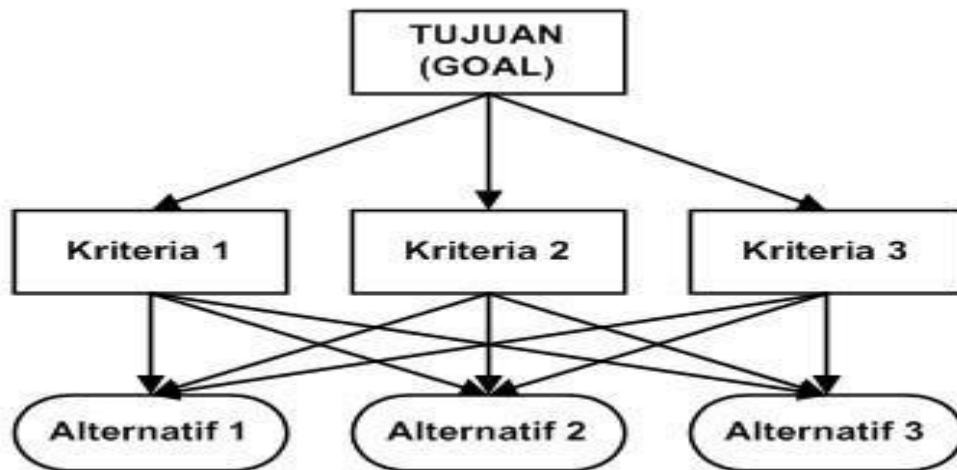
Sumber : (Suryadi & Ramdhani 2016)

II.2.2. Proses-proses dalam AHP

Adapun proses-proses yang harus dilakukan pada metode AHP adalah sebagai berikut (I Nyoman Yudha Astana, 2013) :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.

2. Membuat struktur hirarki yang diawali tujuan umum dilanjutkan dengan kriteria dan kemungkinan alternatif pada tingkatan kriteria paling bawah.



Gambar II.1. struktur hirarki
Sumber : Rosana Junita Sirait, et al., 2015)

Penjelasan:

- a. Tujuan (goal) yang akan di capai atau penyelesaian masalah yang dikaji.
 - b. Kriteria, kriteria apa saja yang harus dipenuhi oleh semua alternatif (penyelesaian) agar layak untuk menjadi pilihan ideal.
3. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap kriteria yang setingkat di atasnya.
 4. Melakukan perbandingan berpasangan sehingga diperoleh *judgment* (keputusan) sebanyak $n \times ((n-1)/2)$ kali, dengan n adalah banyaknya elemen yang dibandingkan.

5. Menghitung nilai *eigen* dan menguji konsistensinya jika tidak konsisten maka pengambilan data diulangi lagi.
6. Mengulangi langkah 3,4 dan 5 untuk setiap tingkatan hirarki.
7. Menghitung *vector eigen* dari setiap matrik perbandingan berpasangan.
8. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10 persen maka penilaian data *judgment* harus diperbaiki.

II.2.3 UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami dibandingkan dengan bahasa pemrograman. *Unified Modelling Language* (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan actor.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur static sebuah sistem dalam bentuk *class diagram*.

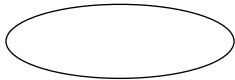
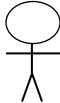
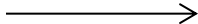
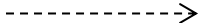
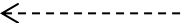
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams* UML.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionalty* dengan *stereo types*.

Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Agus Prayitno dkk, 2019).

II.2.3.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Agus Prayitno dkk, 2019). Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.2. Simbol *Use Case*

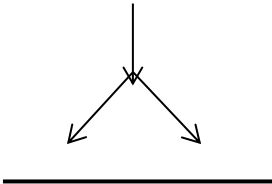
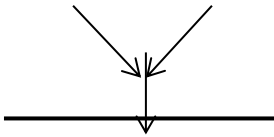
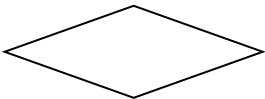
Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber: Agus Prayitno dkk, 2019)

II.2.3.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .

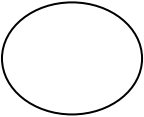
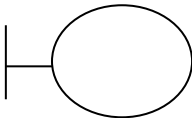
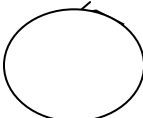
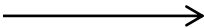
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
---	-----------------	--

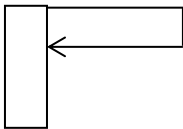
(Sumber: Agus Prayitno dkk, 2019)

II.2.3.3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.4. Simbol Sequence Diagram

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .

	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber: Agus Prayitno dkk, 2019)

II.2.3.4. Class Diagram

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

II.2.4. Database

Menurut (Yakub 2020), *database* diartikan sebagai gudang, tempat bersarang atau berkumpul. Prinsip utama *database* adalah pengaturan data dengan tujuan utama kecepatan dalam pengambilan data kembali. Adapun tujuan *database* adalah untuk menangani data dalam jumlah besar, meniadakan duplikasi dan dapat dicari secara cepat.(Yakub:2020)

Database yang dibuat pada sistem informasi ini menggunakan PHP *MyAdmin* semua pengeluaran biaya yang dilakukan di perusahaan akan dimasukkan di *database* kemudian di tinjau apakah biaya ini layak untuk dikeluarkan atau tidak (Evri:2019)

II.2.5 Xampp

Xampp adalah software pendukung ang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. (Juno:2020)

II.2.6. PHP MyAdmin

PHP Myadmin perangkat lunak bebas yang ditulis dalam bahasa pemrograman php yang digunakan untuk menangani administrasi MYSQL melalui jejaringan jagat jembar (World Wide Web). Phpmyadmin mendukung berbagai aplikasi mysql, diantaranya (mengelola basis data, table- table, bidang, relasi, indeks, pengguna, perizinan, dll) (Juno: 2020)

II.2.7. Internet

Internet atau *Interconnected Networking* merupakan dua komputer atau lebih yang saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer di dunia, yang saling berinteraksi dan bertukar informasi, terhadap masalah sosial misalnya dengan menggunakan alat-alat bantu online untuk mencapai bisnis elektronik, kepemilikan informasi dan interaksi dengan masyarakat. (Priyo Sutopo, 2019).

II.2.8. PHP (*Php Hypertext Preprocessor*)

PHP singkatan dari PHP: *Hypertext Preprocessor* yang digunakan sebagai bahasa *script serverside* dalam pengembangan web yang disisipkan dalam dokumen HTML. Penggunaan PHP memungkinkan web dapat dinamis sehingga maintenance situs web tersebut menjadi lebih mudah dan efisien. PHP merupakan software Open-source yang disebar dan dilisensikan secara gratis serta dapat di download secara bebas dari situs resminya <http://www.php.net>. (Aan Tri Wibowo, 2013 : 63).

II.2.9. MySql

My Structure Query Language adalah salah satu jenis Database server yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *Database*. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL sebuah *Database* mengandung satu atau sejumlah tabel. *Tabel* terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. (Priyo Sutopo dkk, 2019).

II.3.0. CSS (*Cascading Style Sheets*)

Menurut Juno (2020), *Coding* program yang di buat untuk mempercantik tampilan web secara langsung ataupun tidak langsung sehingga siapapun yang melihat merasa tertarik .(Juno:2020)