

BAB II

LANDASAN TEORI

2.1. Sistem Pendukung Keputusan

Konsep SPK (Sistem Pendukung Keputusan) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision System. Istilah SPK mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan, berikut ini adalah pendapat para ahli tentang pengertian SPK, diantaranya oleh Man dan Watson yaitu SPK (Sistem Pendukung Keputusan) adalah suatu sistem yang dapat membantu mengambil keputusan melalui penggunaan data dan model keputusan untuk memecahkan masalah yang sifatnya semi terstruktur maupun yang tidak terstruktur (Huang, 2011).

Maksud dan tujuan dari SPK, yaitu untuk mendukung pengambil keputusan memilih alternatif keputusan yang merupakan hasil pengolahan informasi-informasi yang diperoleh /tersedia dengan menggunakan model-model pengambil keputusan serta untuk menyelesaikan masalah-masalah bersifat terstruktur, semi terstruktur dan tidak terstruktur (Afrisawati, 2017).

Dari uraian di atas dapat disimpulkan bahwa, sistem pendukung keputusan adalah suatu sistem berbasis komputer yang dapat menghasilkan alternatif terbaik yang telah ditentukan berdasarkan kriteria-kriteria tertentu untuk membantu para pengambil keputusan dalam menentukan keputusan secara objektif.

2.2. *Elimination Et Choix Traduisant la Realite* (ELECTRE)

Elimination Et Choix Traduisant la Realite (ELECTRE) adalah salah satu metode pengambilan keputusan multi kriteria berdasarkan pada konsep Outranking

dengan menggunakan perbandingan berpasangan dari alternatif alternatif berdasarkan setiap kriteria yang sesuai. Metode *Elimination Et Choix Traduisant La Realite* (ELECTRE) digunakan pada kondisi dimana alternatif yang kurang sesuai dengan kriteria dieliminasi, dan alternatif yang sesuai dapat dihasilkan. Dengan kata lain, *Elimination Et Choix Traduisant La Realite* (ELECTRE) digunakan untuk kasus-kasus dengan banyak alternatif namun hanya sedikit kriteria yang dilibatkan. Suatu alternatif dikatakan mendominasi alternatif yang lainnya jika satu atau lebih kriterianya melebihi (dibandingkan dengan kriteria dari alternatif yang lain) dan sama dengan kriteria lain yang tersisa. (Marlinda, 2016).

Langkah-langkah yang dilakukan dalam penyelesaian masalah menggunakan metode *Elimination Et Choix Traduisant La Realite* (ELECTRE) adalah sebagai berikut:

1. Normalisasi matriks keputusan

Dalam prosedur ini, setiap atribut diubah menjadi nilai comparable.

Setiap normalisasi dari nilai x_{ij} dapat dilakukan dengan rumus:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, \text{ untuk } i = 1, 2, 3 \dots, m \text{ dan } j = 1, 2, 3 \dots, n$$

Sehingga didapat matriks R hasil normalisasi,

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{1n} \\ r_{21} & r_{22} & r_{2n} \\ r_{m1} & r_{m2} & r_{mn} \end{bmatrix}$$

R adalah matriks yang telah di normalisasi, dimana m menyatakan alternatif, n menyatakan kriteria dan r_{ij} adalah normalisasi pengukuran pilihan dari alternatif ke-i dalam hubungannya dengan kriteria ke-j

2. Pembobotan pada matriks yang telah dinormalisasi

Setelah dinormalisasi, setiap kolom dari matriks R dikalikan dengan bobot-bobot (W_j) yang ditentukan oleh pembuat keputusan. Sehingga $V=RW$ yang ditulis sebagai:

$$V = R.W$$

$$\begin{bmatrix} v_{11} & v_{12} & v_{1n} \\ v_{21} & v_{22} & v_{2n} \\ v_{m1} & v_{m2} & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_1 r_{12} & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & w_n r_{2n} \\ w_1 r_{m1} & w_2 r_{m2} & w_n r_{mn} \end{bmatrix}$$

Dimana W adalah

$$W = \begin{bmatrix} w_1 & 0 & 0 \\ v_{21} & w_2 & 0 \\ 0 & 0 & w_n \end{bmatrix}$$

3. Menentukan concordance dan discordance.

Untuk setiap pasang dari alternatif k dan l ($k, l = 1, 2, 3, \dots, m$ dan $k \neq l$) kumpulan kriteria J dibagi menjadi dua subsets, yaitu concordance dan discordance. Bilamana sebuah kriteria dalam suatu alternatif termasuk concordance adalah:

$$C_{kl} = \{j, v_{kj} \geq v_{lj}\}, \text{ untuk } j = 1, 2, 3 \dots n$$

Sebaliknya, komplementer dari himpunan concordance adalah himpunan discordance, yaitu bila:

$$D_{kl} = \{j, v_{kj} < v_{lj}\}, \text{ untuk } j = 1, 2, 3 \dots n$$

4. Hitung matriks concordance dan discordance

a. Menghitung matriks concordance Untuk menentukan nilai dari elemen-elemen pada matriks concordance adalah dengan menjumlahkan bobot-bobot yang termasuk pada himpunan concordance, secara matematis:

$$C_{kl} = \sum_{j \in C_{kl}} w_j$$

- b. Menghitung matriks discordance Untuk menentukan nilai dari elemen-elemen pada matriks discordance adalah dengan membagi maksimum selisih kriteria yang termasuk ke dalam himpunan bagian discordance dengan maksimum selisih nilai seluruh yang ada, secara matematis:

$$d_{kl} = \frac{\max \{ |v_{kj} - v_{ij}| \} j \in D_{kl}}{\max \{ |v_{kj} - v_{ij}| \} \forall j}$$

5. Menentukan matriks dominan concordance dan discordance

- a. Menentukan matriks dominan concordance

Matriks F sebagai matriks dominan concordance dapat dibangun dengan bantuan nilai threshold, yaitu membandingkan setiap nilai elemen matriks concordance dengan nilai threshold.

$$C_{kl} \geq c$$

Dengan nilai threshold (c) adalah:

$$c = \frac{\sum_{k=1}^m \sum_{i=1}^m C_{kl}}{m(m-1)}$$

Sehingga elemen matriks F ditentukan sebagai berikut:

$$f_{kl} = \begin{cases} 1, & \text{jika } C_{kl} \geq c \\ 0, & \text{jika } C_{kl} < c \end{cases}$$

- b. Menentukan matriks dominan discordance.

Untuk membangun matriks dominan discordance juga menggunakan bantuan nilai threshold, yaitu:

$D_{kl} \geq d$ Sehingga elemen matriks G ditentukan sebagai berikut:

$$g_{kl} = \begin{cases} 1, & \text{jika } d_{kl} < d \\ 0, & \text{jika } d_{kl} \geq d \end{cases}$$

- c. Menentukan aggregate dominance matriks Langkah selanjutnya adalah menentukan aggregate dominance matrix sebagai matriks E, yang setiap elemennya merupakan perkalian antara elemen matriks F dengan elemen matriks G, sebagai berikut:

$$E_{kl} = f_{kl} \times g_{kl}$$

Eliminasi alternatif yang less favourable Matriks E memberikan urutan pilihan dari setiap alternatif, yaitu bila $e_{kl} = 1$ maka alternatif A_k merupakan pilihan yang lebih baik dari pada A_l . Sehingga baris dalam matriks E yang memiliki jumlah $e_{kl} = 1$ paling sedikit dapat dieliminasi. Dengan demikian alternatif terbaik adalah yang mendominasi alternatif lainnya. (Marlinda, 2016).

2.3. Analytical Hierarchy Process (AHP)

Analytical Hierarchy Process (AHP) dikembangkan oleh Thomas Saaty pada tahun 1970-an dalam menghadapi masalah perencanaan militer Amerika Serikat untuk menghadapi berbagai pilihan (*contingency planning*) (Hermawati, 2013). AHP merupakan sistem pembuat keputusan dengan menggunakan model matematis (Daihani, 2015). Pada umumnya AHP digunakan dengan tujuan untuk menyusun prioritas dari berbagai alternatif pilihan yang ada dan pilihan-pilihan tersebut bersifat kompleks atau multikriteria (Kusumadewi, Sri; Hartati, Sri; Harjoko, Agus; Wardoyo, 2013).

Beberapa kelebihan penggunaan metode AHP adalah sebagai berikut (Maghfiroh, 2010) :

1. Struktur yang berbentuk hirarki sebagai konsekuensi dari kriteria yang dipilih sampai pada subkriteria yang paling dalam.
2. Memperhatikan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan.
3. Memperhitungkan daya tahan atau ketahanan keluaran analisis sensitivitas pembuat keputusan.

Selain itu metode AHP mempunyai kemampuan untuk memecahkan masalah yang multi-objektif dan multikriteria yang berdasar pada perbandingan preferensi dari setiap elemen dalam hirarki. Jadi metode AHP merupakan suatu bentuk pemodelan pembuatan keputusan yang sangat komprehensif. Terdapat 4 aksioma-aksioma yang terkandung dalam model AHP:

1. *Reciprocal Comparison* artinya pengambilan keputusan harus dapat memuat perbandingan dan menyatakan preferensinya. Preferensi tersebut harus memenuhi syarat resiprokal yaitu apabila A lebih disukai daripada B dengan skala x , maka B lebih disukai daripada A dengan skala $1/x$.
2. *Homogeneity* artinya preferensi seseorang harus dapat dinyatakan dalam skala terbatas atau dengan kata lain elemen- elemennya dapat dibandingkan satu sama lainnya. Kalau aksioma ini tidak dipenuhi maka elemen- elemen yang dibandingkan tersebut tidak homogen dan harus dibentuk cluster (kelompok elemen) yang baru.
3. *Independence* artinya preferensi dinyatakan dengan mengasumsikan bahwa kriteria tidak dipengaruhi oleh alternatif-alternatif yang ada melainkan oleh objektif keseluruhan. Ini menunjukkan bahwa pola ketergantungan dalam AHP adalah searah, maksudnya perbandingan antara elemen-elemen dalam

satu tingkat dipengaruhi atau tergantung oleh elemen-elemen pada tingkat di atasnya.

4. *Expectation* artinya untuk tujuan pengambil keputusan. Struktur hirarki diasumsikan lengkap. Apabila asumsi ini tidak dipenuhi maka pengambil keputusan tidak memakai seluruh kriteria atau objectif yang tersedia atau diperlukan sehingga keputusan yang diambil dianggap tidak lengkap.

Pada dasarnya terdapat beberapa langkah yang perlu diperhatikan dalam menggunakan metode AHP, antara lain (Maghfiroh, 2010) :

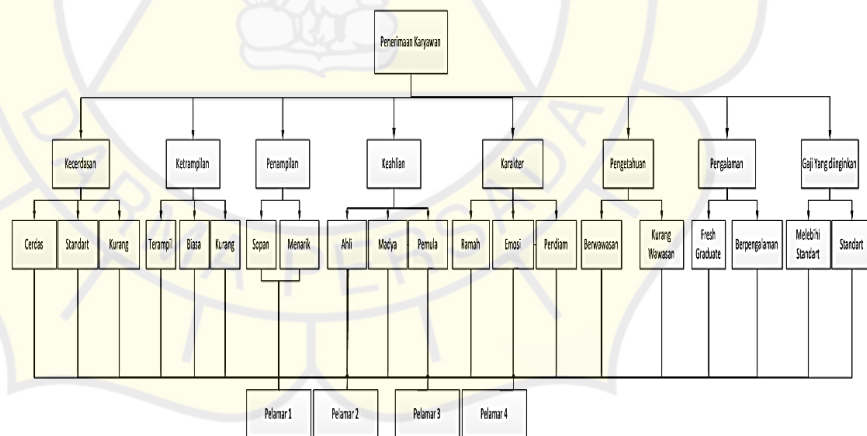
1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Membuat struktur hirarki yang diawali dengan tujuan umum dilanjutkan dengan subtujuan-subtujuan, kriteria dan kemungkinan alternatif-alternatif pada tingkatan kriteria yang paling bawah.
3. Membuat matriks perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap masing-masing tujuan atau kriteria yang setingkat di atasnya. Perbandingan dilakukan berdasarkan judgment dari pembuat keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya.
4. Melakukan perbandingan berpasangan sehingga diperoleh nilai judgment seluruhnya yaitu sebanyak $n \times [(n-1)/2]$ buah dengan n adalah banyaknya elemen yang dibandingkan.
5. Menghitung nilai lamda max dan menguji konsistensinya jika tidak konsisten maka pengambilan data diulangi.
6. Mengulangi langkah 3, 4 dan 5 untuk seluruh tingkat hirarki.

7. Menghitung vektor eigen dari setiap matriks perbandingan berpasangan. Nilai vektor eigen merupakan bobot setiap elemen. Langkah ini untuk mensintesis judgment dalam penentuan prioritas elemen-elemen pada tingkat hirarki terendah sampai pencapaian tujuan.
8. Memeriksa konsistensi hirarki. Jika nilai lebih dari 10% (persen) atau 0,1 maka penilaian data harus diperbaiki.

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah (Hermawati, 2013):

1. Membuat Hirarki

Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen - elemen pendukung, menyusun elemen secara hirarki, dan menggabungkannya atau mensistesisnya. Untuk lebih jelasnya dapat dilihat pada gambar 2.1.



Gambar 2. 1 Struktur Hirarki AHP

2. Penilaian kriteria dan alternative

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala

perbandingan Saaty dapat dilihat pada Tabel di bawah ini (Kosasi, S., & Kuway, 2012) :

Tabel 2. 1 Skala Penilaian Perbandingan Berpasangan (Kosasi, S., & Kuway, 2012)

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya.
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya.
5	Elemen yang satu lebih penting daripada elemen yang lainnya.
7	Satu elemen jelas lebih mutlak penting daripada elemen yang lainnya.
9	Satu elemen mutlak penting daripada elemen yang lainnya.
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan.
Kebalikan	Jika aktivitas i mendapat satu angka dibandingkan dengan aktivitas j, maka j memiliki nilai kebalikannya dibandingkan dengan i.

Pengisian nilai tabel perbandingan berpasangan dilakukan berdasarkan kebijakan pembuat keputusan dengan melihat tingkat kepentingan antar satu elemen dengan elemen yang lainnya. Proses perbandingan berpasangan, dimulai dari perbandingan kriteria misalnya A1, A2 dan A3. Maka susunan elemen-elemen yang dibandingkan tersebut akan tampak seperti pada tabel 2 di bawah ini:

Tabel 2. 2 Contoh Matriks Perbandingan Berpasangan

	A1	A2	A3
A1	1		
A2		1	
A3			1

Untuk menentukan nilai kepentingan relatif antar elemen digunakan skala bilangan dari 1 sampai 9 yang dapat dilihat pada Tabel 2.2. Apabila suatu elemen dibandingkan dengan dirinya sendiri maka diberi nilai 1. Jika elemen i dibandingkan dengan elemen j mendapatkan nilai tertentu, maka elemen j dibandingkan dengan elemen i merupakan kebalikannya.

Pengujian konsistensi dilakukan terhadap perbandingan antar elemen yang didapatkan pada tiap tingkat hirarki. Konsistensi perbandingan ditinjau dari per matriks perbandingan dan keseluruhan hirarki untuk memastikan bahwa urutan prioritas yang dihasilkan didapatkan dari suatu rangkaian perbandingan yang masih berada dalam batas-batas preferensi yang logis. Setelah melakukan perhitungan bobot elemen, langkah selanjutnya adalah melakukan pengujian konsistensi matriks. Untuk melakukan perhitungan ini diperlukan bantuan tabel *Random Index* (RI) yang nilainya untuk setiap ordo matriks dapat dilihat pada tabel berikut ini:

Tabel 2. 3 Tabel Nilai Random Index

Urutan Matriks	1	2	3	4	5	6	7	8	9	10
(RI)	0.00	0.01	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

3. *Synthesis of Priority* (Penentuan Prioritas)

Setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*Pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan judgement yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot atau prioritas dihitung dengan manipulasi matriks atau melalui penyelesaian persamaan matematika.

4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna, pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

Penggunaan metode AHP biasa digunakan untuk permasalahan pemilihan alternatif yang memiliki multikriteria. Setiap kriteria dan sub kriteria memiliki bobot, sebagai acuan prioritas perbandingan antar kriteria atau sub kriteria.

2.4. *Unified Modeling Language* (UML)

Menurut Windu, & Gata, (2013). UML (*Unified Modeling Language*) adalah Bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Ada beberapa diagram dalam UML (*Unified Modelling Language*) antara lain :





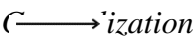
1) *Use Case Diagram*

Ini menggambarkan fungsionalitas yang diharapkan dari sebuah system.

Use case merepresentasikan sebuah interaksi antara actor dengan system.

Adapun symbol dari use case diagram antara lain :

Tabel 2. 4 Simbol *Use Case* Diagram



No	Simbol	Keterangan
1	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	Aktor / <i>Actor</i>  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	Asosiasi / <i>Association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Ekstensi / <i>Extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5	Generalisasi /  <i>ization</i>	Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi




		yang lebih umum dari lainnya.
6	<p>Menggunakan / <i>Include</i> / <i>uses</i></p> <p><<include>> -----></p>	Relasi use case tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan.

2) Activity Diagram

Menggambarkan berbagai alur aktifitas dalam system yang sedang dirancang, bagaimana masing-masing alur berawal dan decision yang mungkin terjadi dan bagaimana alur berakhir. Adapun symbol dari Activity Diagram antara lain:

Tabel 2. 5 Simbol Activity Diagram


No	Gambar	Keterangan
1.	 <i>Activity</i>	Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
2.	 <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

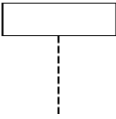

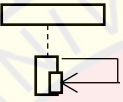
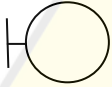
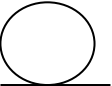
3.	 Status Awal	Status awal aktiviatas sistem, sebuah diagram aktivitas memiliki status awal.
4.	 Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5.	 <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

3) Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek didalam dan sekitar sistem (termasu pengguna dan display) berupa message yang digambarkan terhadap waktu, *Sequence* Diagram terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait). Setiap pesan yang dikirimkan bisa memberi respon (return) relative pada scenario yang dirancang di *Use Case* Diagram. Interaksi yang terjadi bisa bersifat instansiasi sebuah objek maupun static metod dari sebuah class. Berikut ini adalah symbol-simbol dari *sequence* diagram :

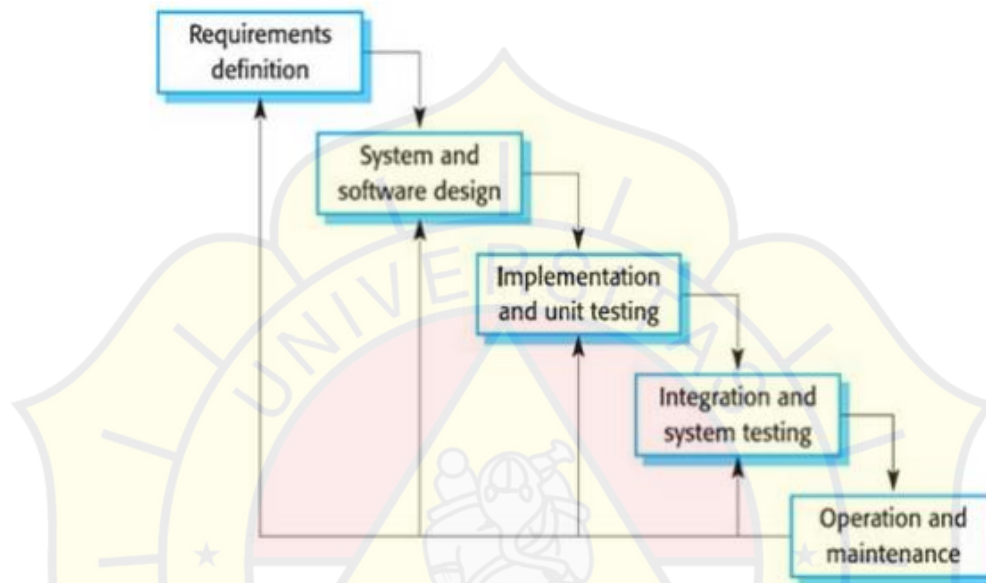
Tabel 2. 6 Simbol *Sequence*Diagram

No	Gambar	Keterangan
1.		Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary</i> sistem

	Aktor	
2.	 <i>Lifeline</i>	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)
3.	 <i>Message</i>	Menunjukkan aliran informasi atau kendali transaksi antar elemen
4.	 <i>Sel-Message</i>	Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i> . Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i>
5.	 <i>Boundary</i>	<i>Boundary</i> adalah objek stereotip yang memodelkan batasan sistem. Biasanya layar <i>user interface</i>
6.	 <i>Control</i>	<i>Control</i> adalah objek stereotip yang mengontrol atau mengatur entitas
7.	 <i>Entity</i>	Merupakan table pada <i>database</i> yang merupakan model penyimpanan data yang menangkap dan menyimpan informasi dalam sistem

2.5. Waterfall

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode waterfall (Sasmito, 2017). Metode waterfall merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Metode Waterfall memiliki tahapan-tahapan sebagai berikut :



Gambar 2. 2 Tahapan Metode Waterfall (Sasmito, 2017)

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and Software Design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak

melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer.

5. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. Maintenance melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

2.6. HTML

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. *HTML* merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)* (Kadir, 2014). *HTML* sebenarnya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

Mendesain *HTML* berarti melakukan suatu tindakan pemrograman. Namun *HTML* bukanlah sebuah bahasa pemrograman. Namun *HTML* hanyalah berisi

perintah-perintah yang telah terstruktur berupa tag-tag penyusun. Menuliskan tag-tag *HTML* tidaklah sebatas hanya memasukkan perintah-perintah tertentu agar *HTML* kita dapat di akses oleh browser. Mendesain *HTML* adalah sebuah seni tersendiri. *Homepage* yang merupakan implementasi dari *HTML* adalah refleksi dari orang yang membuatnya. Untuk itu kita perlu mendesainnya dengan baik agar para pengunjung homepage yang kita buat merasa senang dan bermanfaat. Mendesain *HTML* dapat dilakukan dengan cara menggunakan *HTML Editor*, seperti notepad++, adobe dreamweaver dan lain-lain.

2.7. PHP

Menurut Sidik, (2014) PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

Madcoms, (2016), *PHP* adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. *PHP* adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*.

Menurut Abdulloh, (2016) "*PHP* singkatan dari *hypertext preprocessor* yang merupakan *server-side* programing, yaitu bahasa pemrograman yang diproses disisi *server*.

Dari penjelasan di atas dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman yang digunakan untuk membuat dan mendesain pengembangan dari sebuah server web.

2.8. MySQL

Menurut Nugroho, (2015) , “*MYSQL* adalah *software* atau program *Database Server*”. Sedangkan *SQL* adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam *MYSQL* itu sendiri. *SQL* juga dipakai dalam *software database server* lain, seperti *SQL Server*, *Oracle*, *PostgreSQL* dan lainnya.

Menurut Fais, M. N., Susanto, A., & Listyorini, (2014) “*Mysql* merupakan sistem manajemen database yang bersifat *relational*. Artinya, data yang dikelola dalam database akan diletakan pada beberapa *table* yang terpisah sehingga manipulasi data akan menjadi lebih cepat”.

2.9. Blackbox Testing

Black box testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenai proses testing dibagian luar. Jenis testing ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan (Wicaksono, 2017).

Beberapa keuntungan yang diperoleh dari black box testing antara lain :

1. Anggota tim tester tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dari perangkat lunak ataupun bug seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
3. Hasil dari black box testing dapat memperjelas kontradiksi ataupun kerancuan yang mungkin timbul dari eksekusi sebuah perangkat lunak.

4. Proses testing dapat dilakukan lebih cepat dibandingkan white box testing.

2.10. Tinjauan Terhadap Penelitian Terkait

Berikut beberapa penelitian yang terkait dan menjadi referensi pada penelitian ini:

Penelitian (Prahesti, 2017) dengan judul “SISTEM REKOMENDASIPEMILIHAN SEKOLAH MENENGAH ATAS (SMA) SEDERAJAT KOTA MALANG MENGGUNAKAN METODE AHP-ELECTRE DAN SAW”. Pendidikan merupakan aspek yang penting bagi masyarakat. Salah satu jenjang pendidikan formal adalah pendidikan menengah yaitu Sekolah Menengah Atas (SMA). Sekolah adalah lembaga yang memberikan pengajaran untuk siswa dibawah pengawasan guru. Setiap tahun ajaran baru, siswa akan memilih sekolah terbaik yang sesuai dengan keinginannya. Terdapat banyak pilihan sekolah dengan berbagai tawaran yang diberikan. Dengan begitu, calon siswa akan mengalami kesulitan dalam menentukan sekolah yang sesuai dengan kriteria yang diinginkan. Penelitian ini bertujuan untuk memberikan rekomendasi sekolah berdasarkan kriteria yang diinginkan oleh calon siswa dengan menerapkan metode AHP-ELECTRE dan SAW ke dalam sistem. Metode AHP digunakan dalam pembobotan dari setiap kriteria, metode ELECTRE melakukan klasifikasi alternatif yang masuk ke dalam kelompok direkomendasikan, dan SAW melakukan perankingan alternatif. Untuk pengujian, hasil pengujian akurasi sistem adalah sebesar 82,98%. Hasil akurasi didapatkan dengan membandingkan hasil rekomendasi sistem dengan data yang telah didapatkan.

Penelitian (Saribu, 2020) dengan judul “PENERAPAN METODE TOPSIS PADA SISTEM PENDUKUNG KEPUTUSAN UNTUK PEMILIHAN TANAMAN HIAS BONSAI”. Banyaknya konsumen yang membutuhkan tanaman hias memberi prospek yang baik bagi masa depan bisnis tanaman hias khususnya bagi petani tanaman hias. Tanaman hias tidak terbatas hanya pada tanaman hias yang hidup di pot, tetapi juga meliputi bunga potong, kaktus, bonsai, tanaman hidroponik dan bunga tabor. Penelitian ini bertujuan untuk membuat Sistem Pendukung Keputusan (SPK) pemilihan tanaman hias bonsai. Kriteria pemilihan tanaman hias bonsai adalah memiliki bentuk dasar yang indah, berasal dari tanaman berumur panjang, batangan dahannya mudah dibentuk, permukaan kulit menarik dan berlekuk-lekuk, berdaun kecil dan cukup rimbun, kondisi tanaman sudah cukup umur dan tanaman kuat untuk dibentuk. Sistem dibangun dengan bahasa Pemrograman Visual Basic dan menerapkan metode TOPSIS untuk menentukan tanaman hias bonsai. Implementasi sistem pendukung keputusan menunjukkan dari 5 (lima) alternatif pilihan tanaman yang layak menjadi tanaman hias bonsai adalah Beringin, Asoka Cina dan Melati.