BAB II

LANDASAN TEORI

2.1. Sistem Pendukung Keputusan

Pengambilan keputusan pada dasarnya adalah suatu bentuk pemilihan berdasarkan alternatif – alternatif yang ada, dengan harapan pengambilan keputusan tersebut menghasilkan suatu keputusan yang terbaik dan tepat. Dalam pengambilan suatu keputusan selalu diupayakan secara objektif, cepat dan juga tepat. Penerapannya dikenal Sistem Pendukung Keputusan (SPK), SPK merupakan bagian dari sistem informasi yang berbasis komputer. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi sistem informasi untuk mengambil keputusan dari suatu masalah. (Abiyoga, 2014).

2.2. ELECTRE

Penelitian ini difokuskan dalam penerapan *Multi Attribute Decision Making (MADM)* pada Sistem Pendukung Keputusan (SPK) pemilihan paket *tour* Pulau Lombok dan atau Pulau Balimengunakan metode *ELimination Et Choix Traduisant la Realita* (ELECTRE) (Marlinda, 2016). Metode ELECTRE termasuk pada metode analisis pengambilan keputusan multikriteria yang berasal dari Eropa pada tahun 1960an. ELECTRE merupakan salah satu metode pengambilan keputusan multikriteria berdasarkan pada konsep outranking dengan menggunakan perbandingan berpasangan dari alternatif - alternatif berdasarkan setiap kriteria yang

sesuai. Metode ELECTRE digunakan pada kondisi dimana alternatif yang kurang sesuai dengan kriteria dieliminasi dan alternatif yang sesuai dapat dihasilkan. Dengan kata lain, ELECTRE digunakan untuk kasus-kasus dengan banyak alternatif namun hanya sedikit kriteria yang dilibatkan. (Salahudin, Astuti, & Kridalaksana, 2016).

Langkah – langkah yang dilakukan dalam penyelesaian masalah dengan menggunakan metode ELECTRE ialah sebagai berikut (Silalahi, 2016):

1. Normalisasi matriks keputusan. Setiap atribut diubah menjadi nilai yang *comparable* membentuk perbandingan berpasangan setiap alternatif pada setiap kriteria (xij). Nilai tersebut harus dinormalisasikan ke dalam suatu skala yang dapat diperbandingkan (rij).

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$ Untuk i = 1,2,3,…, m dan j = 1,2,3,…,n.

Rumus 2. 1. Rumus Normalisasi

Sehingga didapat matriks R hasil normalisasi,

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

Rumus 2. 2. Matriks R

Dimana:

R = matriks yang telah dinormalisasi

m = *alternative*

n = kriteria

r_{ij} = normalisasi pengukuran pilihan dari alternatif ke-i dalam hubungan dengan kriteria ke-j

x_{ij} = nilai X pada alternatif ke-i dan kriteria ke-j

2. Pembobotan pada matriks yang telah dinormalisasi. Setelah melakukan langkah normalisasi, pengambil keputusan memberikan bobot (faktor kepentingan) pada setiap kriteria (Wj), dengan cara setiap kolom dari matrik R dikalikan dengan bobot-bobot yang sebelunya telah ditentukan oleh pembuat keputusan. Sehingga matriks pembobot yang dinotasikan sebagai V=RW yang ditulis sebagai:

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

Rumus 2. 3. Mencari Matriks V

Dimana W adalah matriks pembobotan, R matriks yang telah dinormalisasi dan V matriks hasil perkalian antara matriks pembobotan dan matriks yang telah dinormalisasi.

$$W = \begin{bmatrix} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & W_n \end{bmatrix}$$

Rumus 2. 4. Matriks W

Keterangan:

V = matriks pembobotan

R = matriks normalisasi

W = bobot

3. Menentukan himpunan *concordance* dan *discordance* index. Untuk setiap pasang dari alternatif k dan l (k,l = 1,2,3,...,m dan k ≠ l) kumpulan kriteria J dibagi menjadi dua himpunan bagian, yaitu *concordance* dan *discordance*. Sebuah kriteria dalam suatu alternatif termasuk *concordance* jika :

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

Rumus 2. 5. Penentuan *Concordance*

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

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

Rumus 2. 6 Penentuan Disordance

4. Hitung matriks *concordance* dan *discordance*.

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

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

Rumus 2. 7. Menghitung *Concordance*

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

$$d_{kl} = \frac{\max\{|v_{kl} - v_{ij}|\} j \in v_{kl}}{\max\{|v_{kl} - v_{ij}|\} \forall_j}$$

Rumus 2. 8. Menghitung *Disordance*

5. Menentukan matriks dominan *concordance* dan *discordance*.

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

$$C_{kl} \geq \underline{c}$$

Rumus 2. 9. Menentukan Dominan Concordance

dengan nilai *threshold* (c̲) adalah :

$$\underline{c} = \frac{\sum_{k=1}^{m} \sum_{l=1}^{m} c_{kl}}{m(m-1)}$$

Rumus 2. 10. Menghitung Nilai Threshold c̲

Sehingga elemen matriks F ditentukan sebagai berikut :

$$f_{kl} = \begin{Bmatrix} 1, jika\ C_{kl} \geq \underline{c} \\ 0, jika\ C_{kl} < \underline{c} \end{Bmatrix}$$

Rumus 2. 11. Matriks F

b. *Discordance* matriks G sebagai dominan *discordance* dapat dibangun dengan menggunakan bantuan nilai threshold, yaitu :

$$\underline{d} = \frac{\sum_{k=1}^{m} \sum_{l=1}^{m} d_{kl}}{m(m-1)}$$

Rumus 2. 12. Menghitung *Disordance*

Elemen matriks G dapat ditentukan sebagai berikut :

$$g_{kl} = \begin{Bmatrix} 1, jika\ d_{kl} \geq \underline{d} \\ 0, jika\ d_{kl} < \underline{d} \end{Bmatrix}$$

Rumus 2. 13. Matriks G

6. Menentukan *aggregate dominance matrix*. Matriks E sebagai *aggregate dominance* matriks adalah matriks yang setiap elemennya merupakan perkalian antara elemen matriks F dengan elemen matriks G dirumuskan sebagai berikut:

$$e_{kl} = f_{kl}\ x\ g_{kl}$$

Rumus 2. 14. Matriks E

7. Eliminasi alternatif yang *less favourable*. Matriks E memberikan urutan pilihan dari setiap alternatif, yaitu bila $e_{kl} = 1$maka alternatif merupakan A_k pilihan yang lebih baik daripada. Sehingga A_1baris dalam matriks E yang memiliki jumlah paling sedikit $e_{kl} = 1$dapat dieliminasi. Dengan demikian alternatif terbaik adalah yang mendominasi alternatif lainnya.

2.3. *Analytical Hierarchy Process* (AHP)

Metode Analytical Hierarchy Process (AHP) pertama kali diperkenalkan oleh Saaty (1980). Proses AHP melakukan penyelesaian masalah dengan menjadikan struktur hierarki. Model AHP umumnya melibatkan 6 langkah penyelesaian, yang pertama adalah mendefinisikan tujuan dan masalah yang tidak terstruktur, setelah itu langkah yang kedua menentukan detail dari kriteria dan alternatif, langkah yang ketiga menerapkan perbandingan berpasangan untuk menyiapkan matriks perbandingan, selanjutnya langkah yang keempat adalah menggunakan teknik *eigenvalue*untuk menentukan bobot *relative* dari faktor keputusan, langkah yang kelima menghitung konsistensi index dari matriks dan yang terakhir memperoleh peringkat keseluruhan untuk alternatif. (Razandi, Pourghasemi, Neisani, & Rahmati, 2015)

2.4. Android Studio

Android studio adalah IDE *(integrated development environment)*resmi untuk aplikasidan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event* Google I/O *Conference* untuk tahun 2013.(Andi Juansyah, 2015)

2.5 PHP

PHP merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, maksudnya dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. PHP/FI merupakan nama awal dari PHP. PHP adalah *Personal Home Page*, FI adalah *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program yang dikhususkan untuk menerima input melalui *form* yang ditampilkan dalam *browser web*. *Software* ini disebarkan dan dilisensikan sebagai perangkat lunak *Open Source*.PHP secara resmi merupakan kependekan dari PHP *Hypertext Preprocessor*, merupakan bahasa *scriptserver-side* yang disisipkan pada HTML.(Rini Sovia dan Jimmy Febio, 2011)

2.6 JSON

JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di *website*. JSON dirancang untuk memudahkan pertukaran data

pada situs dan merupakan perluasan dari fungsi-fungsi *javascript*. (Destian Wijaya, E.M.A, & Fiade, 2015)

2.7. PhpMyAdmin

PhpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi database MySQL melalui jaringan lokal maupun internet. *PhpMyAdmin* mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain).(Restu, 2017)

2.8. Yii2

Yii adalah *framework* pemrograman *web* generik, yang berarti yii dapat digunakan untuk mengembangkan berbagai macam aplikasi *web* dengan menggunakan PHP. Yii 2 *framework* adalah versi terbaru dari Yii yang merupakan *complete rewrite* (Sari & Dwiyani, 2019).

2.9. Model Sekuensial Linier atau *Waterfall Development Model*

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Model ini sering disebut juga dengan "*classic life cycle*" atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara

sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Widiyanto, 2018)

2.10. Model *Prototype*

Proses pembuatan model sederhana *software* yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Kunci agar model *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa *prototype* dibangun untuk mendefinisikan kebutuhan. (Widiyanto, 2018)

2.11. *Rapid Application Development* (RAD)

Rapid Application Development atau RAD adalah bentuk metodologi *agile software development*, yang ditemukan oleh James Martin pada tahun 1991. Metode RAD (*Rapid Application Development*) mendukung untuk pengembangan *prototype* secara cepat dan memampukan *developers* untuk mendapat *multiple user feedback*. Konsep dari RAD (*Rapid Application Development*). Proses RAD (*Rapid Application Development*) dibagi menjadi 4 tahapan (Agrawal, 2019) yaitu:

1. *Requirements planning phase*

Pada tahapan ini mengidentifikasi segala kebutuhan yang diperlukan oleh *user* dan melakukan perencanaan untuk memenuhi kebutuhan yang diperlukan dengan memastikan semua kebutuhan sudah benar dan konsisten. (Hassan, Qamar, & Idris, 2015).

2. *User design phase*

Pada tahapan *User Design*, diperlukan *design* untuk sistem yang didapat dengan cara menganalisa kebutuhan. Pada fase ini memastikan bahwa *design* dibuat berdasarkan kebutuhan. Tahap *User design*dapat berjalan secara *parallel* dengan tahap *Construction* (Hassan et al., 2015).

3. *Construction phase*

Tahapan *Construction* merupakan tahap terpenting dalam model pengembangan ini. Karena pada tahap ini dilakukan implementasi design yang telah dibuat dan melakukan *coding* berdasarkan *design* (Hassan et al., 2015).

4. *Cut-over phase*

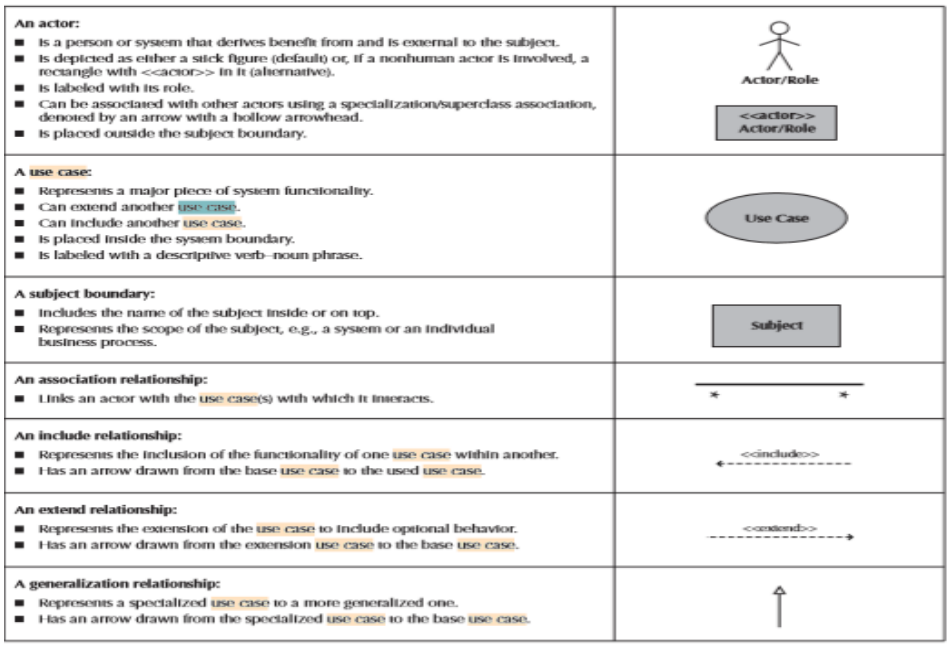
Tahapan yang terakhir adalah melakukan *testing*, *maintenance*, dan *deployment* (Hassan et al., 2015).

2.12. *Unified Modelling Language* (UML)

UML merupakan suatu alat untuk memvisualisasikan dan mendokumentasikan desain sistem *software*. Terdapat beberapa cara untuk memudahkan dalam penganalisisan dan mendesain suatu sistem, seperti *use case diagram*, *activity diagram*, dan *class diagram* (Dennis, Alan;Wixom, 2015).

2.12.1. *Use Case Diagram*

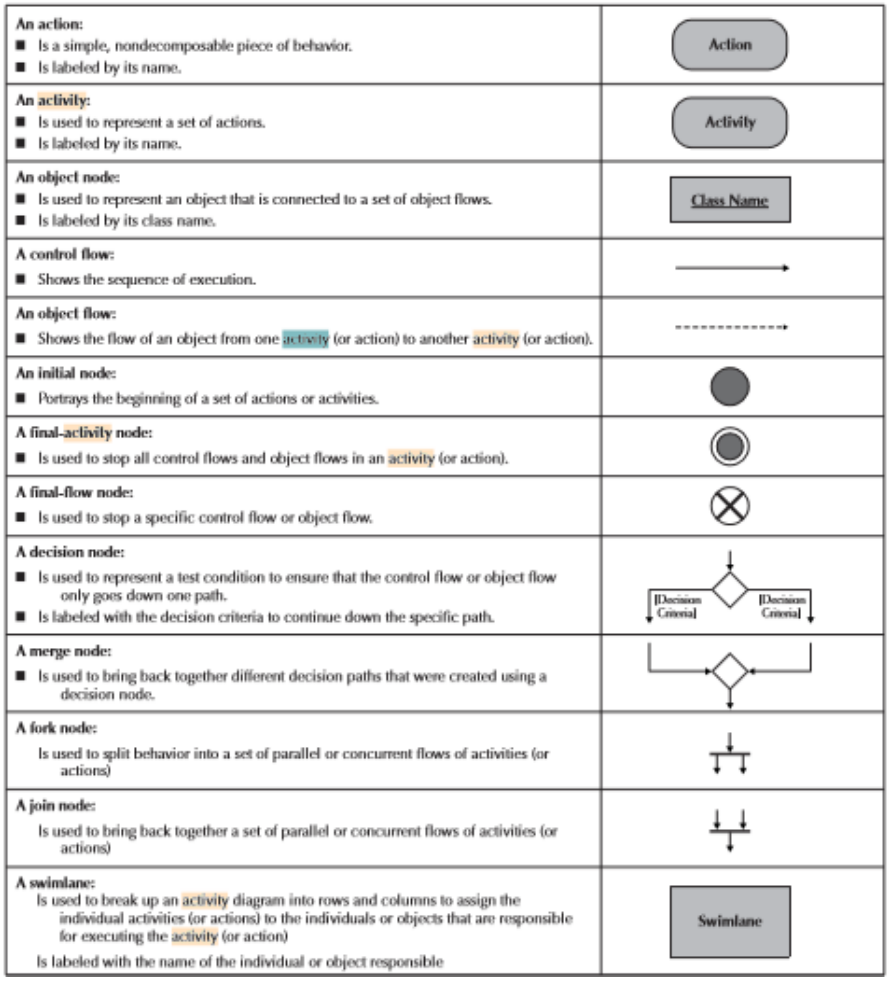
Sebuah *use case diagram* secara visual menampilkan interaksi antara user dengan sebuah sistem. Pada *use case diagram user* digambarkan sebagai aktor, yang menjelaskan secara spesifik apa yang dilakukan *user* terhadap sistem (Dennis, Alan;Wixom, 2015). Pada gambar 2.1. merupakan notasi pada *use case diagram*.



Gambar 2.1. Notasi *Use Case Diagram*

2.12.2. *Activity Diagram*

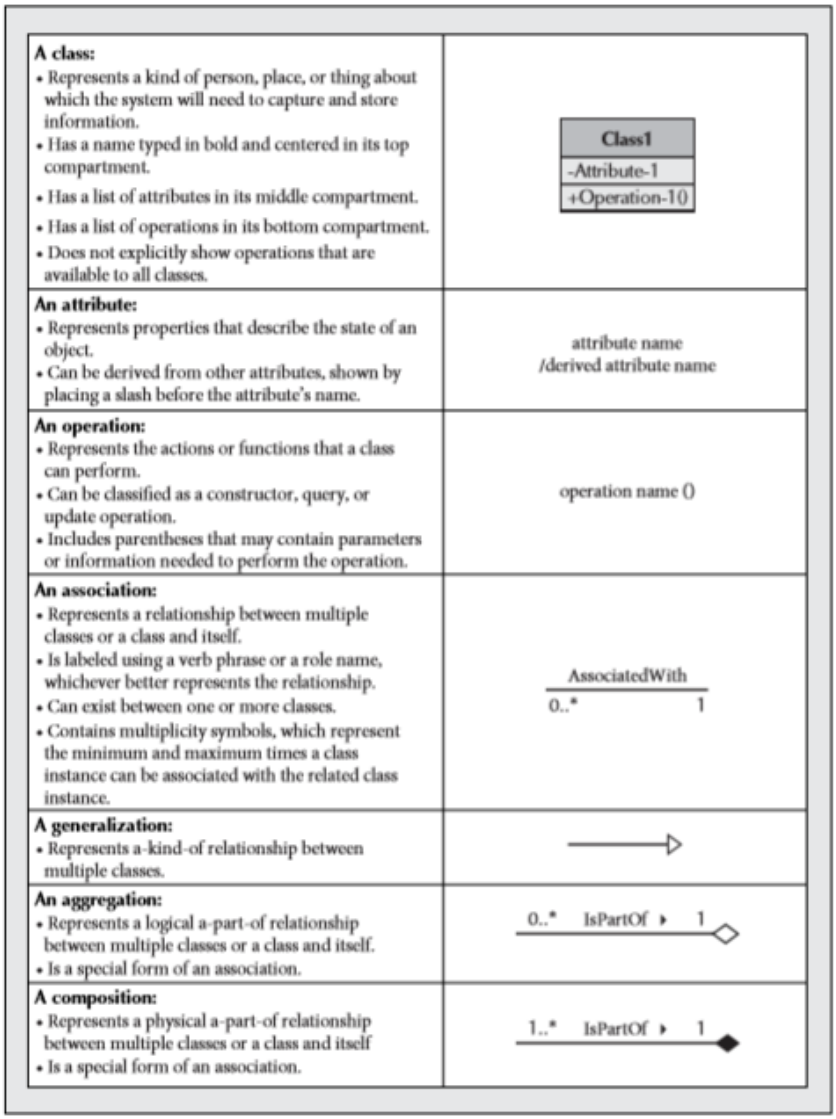
Activity Diagram menyerupai alur horizontal yang menggambarkan urutan aktivitas yang terjadi selama sistem berlangsung. Aktivitas yang diagram ini dijelaskan secara berurutan mulai dari awal hingga akhir aktivitas.*Activity diagram* biasa digunakan untuk menambah pemahaman tentang *use case* yang telah dibuat (Dennis, Alan;Wixom, 2015). Pada gambar 2.2. merupakan notasi dari *activity diagram*.



Gambar 2.2. Notasi *Activity Diagram*

2.12.3. *Class Diagram*

Class diagram adalah model statis yang menampilkan kelas dan relasi antar kelas. Pada diagram ini menggambarkan kelas perilaku dan status antar kelas tersebut. (Dennis, Alan;Wixom, 2015). Pada gambar 2.3 merupakan notasi dari *class diagram*



Gambar 2.3. Notasi *Class Diagram*

2.13. *Black Box Testing*

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* cenderung untuk menemukan hal-hal berikut. (Mustaqbal, Firdaus, & Rahmadi, 2015):

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.14. *Mean Opinion Score*

Mean Opinion Score (MOS) adalah satuan kualitas suara yang biasa digunakan. Metode MOS adalah hasil *survey* dari percakapan dimana nilai rata-rata kualitas suara antara 1 sampai 5, dimana 1 berarti buruk dan 5 adalah yang paling baik.(Fitriyanti, Lindawati, & Aryanti, 2018).

2.15. *Eight Golden Rules*

Prinsip *Eight Golden Rules* berasal dari pengalaman dan disempurnakan selama lebih dari tiga dekade. Dapat digunakan dengan baik bagi siswa ataupun desainer dalam merancang sebuah *user interface*. Berikut delapan aturan dari *Eight*

Golden Rules (Shneiderman, Ben; Plaisant, Catherine; Cohen, Maxine; Jacobs, Maxine; Niklas, Elmqvist; Diakopoulos, 2015) :

1. *Strive for consistency*, konsistensi dalam menampilkan warna, *font*, *layout*, menu, huruf kapital, dan sebagainya.
2. *Seek universal usability*, dapat digunakan secara universal dari berbagai kalangan, mulai dari umur, disabilitas dan sebagainya.
3. *Offer informative feedback*, memberikan informasi *feedback* kepada *user* untuk tindakan *minor* ataupun yang sering dilakukan.
4. *Design dialogs to yield closure*, mengurutkan aksi secara sekuensial mulai dari awal hingga akhir. Sebagai contoh adalah proses transaksi pada *website e-commerce*.
5. *Prevent errors*, mengurangi bahkan menghilangkan error yang dapat dilakukan oleh *user*. Apabila terjadi kesalahan, *interface*mengarahkan *user* untuk memperbaikan bagian yang *error* tersebut.
6. *Permit easy reversal of actions*, *user* bisa saja melakukan kesalahan dan maka dari itu sistem harus bisa melakukan *reversal* untuk kembali ke langkah yang telah dilakukan *user*agar bisa dilakukan perbaikan.
7. *Keep user in control*, aplikasi dapat dikontrol sepenuhnya oleh *user*.
8. *Reduce short term-memory load*, untuk memudahkan user dalam hal mengingat dikarenakan pada umumnya manusia memiliki keterbatasan dalam mengingat.

2.16. Penelitian Terdahulu

Berikut tabel 2.1. menunjukkan jurnal terdagulu dan hasil penelitian jurnal-jurnal tersebut

Tabel 2.1.Penelitian Terdahulu

Judul Jurnal	Hasil Penelitian
Sistem Pendukung Keputusan Berbasis Website untuk Pemilihan Destinasi Pariwisata Kalimantan Timur dengan Metode *Elimination and Choice Expressing Reality* (ELECTRE*)* **Peneliti** Muhammad Salahudin, Indah Fitri Astuti,Awang Harsa Kridalaksana **Lokasi** Kalimantan Timur **Tahun** 2016 **Nama Jurnal** Prosiding Seminar Ilmu Komputer dan Teknologi Informasi Vol. 1, No. 1, September 2016 **Volume** 1	Pada artikel ini *Elimination and Choice Expressing Reality* (ELECTRE) berhasil diterapkan dalam sistem penunjang keputusan pemilihan destinasi pariwisata di Kalimantan Timur. Kriteria yang dipilih pada penelitian ini yaitu biaya, jarak, dan fasilitas dapat diproses dan kemudian menghasilkan rekomendasi melalui tahap-tahap yang telah ditetapkan dalam metode ELECTRE.
Sistem Pendukung Keputusan Pemilihan Pemilihan Rumah di Kota Medan Menggunakan Metode Elimination and *Elimination and Choice Expressing Reality (*ELECTRE*)*	Pada artikel ini metode ELECTRE dapat digunakan untuk membantu melakukan proses pemilihan perumahan sesuai dengan kriteria yang telah ditentukan.

Peneliti Arina Prima Silalahi **Lokasi** Medan, Sumatera Utara **Tahun** 2016 **Nama Jurnal** Jurnal METHODIKA, Vol. 2 No. 2 NOPEMBER 2016 **Volume** 2	
Utilization* ELECTRE *Method with Decision Support System in Select Locations Warehouse Production **Peneliti** Frans Ikorasaki, Muhammad Barkah Akbar **Lokasi** Tanjung Mulia, Medan **Tahun** 2019 **Nama Jurnal** 6th International Conference on Cyber and IT Service Management, CITSM 2018 **Volume** -	Pada artikel ini metode ELECTRE dapat digunakan untuk membantu proses pemilihan memilih lokasi pergudangan dengan kriteria yang telah ditentukan sebelumnya.
Rancang Bangun Aplikasi Rekomendasi Wisata Kuliner dengan Metode *Elimination Et Choix Traduisant La Realite (*ELECTRE*)* **Peneliti** Thomas Nugroho Kurniawan **Lokasi** Tangerang	Pada penelitian ini dilakukan rancang bangun aplikasi untuk melakukan rekomendasi wisata kuliner untuk studi kasus Kota Surakarta menggunakan metode ELECTRE. (UMN).

Tahun 2019 **Nama Jurnal** - **Volume** -	
Pemilihan Paket Wisata Menggunakan Metode *Analytical Hierarchy Process* (AHP) **Peneliti** Mulya Ade Irma Suryani , Zainal Arifin, Heliza Rahmania Hatta **Lokasi** Samarinda **Tahun** 2017 **Nama Jurnal** Jurnal Informatika Mulawarman e-ISSN 2597-4963 dan p-ISSN 1858-4853 **Volume** 12	Pada penelitian ini digunakan metode AHP untuk menentukan paket wisata terbaik. Penelitian diterapkan dalam bentuk *website*, kriteria pemilihan yang ditentukan ialah harga, lama perjalanan, dan hotel. Artikel ini dijadikan referensi penelitian dengan metode dan hasil yang berbeda namun kriteria yang digunakan sama. Metode yang digunakan untuk penelitian ini adalah ELECTRE dan hasilnya ialah berupa aplikasi *mobile* yaitu android

Penerapan penelitian terdahulu pada penelitian ini:

1. Artikel Sistem Pendukung Keputusan Berbasis Website untuk Pemilihan Destinasi Pariwisata Kalimantan Timur dengan Metode *Elimination and Choice Expressing Reality* (ELECTRE) ini menjadi referensi untuk mempelajari metode ELECTRE digunakan pada sektor pariwisata, begitu juga penelitian ini berada di sektor pariwisata.

2. Artikel Sistem Pendukung Keputusan Pemilihan Pemilihan Rumah di Kota Medan Menggunakan Metode Elimination and *Elimination and Choice Expressing Reality (*ELECTRE*)* ini dijadikan referensi pada penelitian ini karena metode ini dapat menghasilkan keputusan yang akurat dan cepat.
3. Artikel *Utilization* ELECTRE *Method with Decision Support System in Select Locations Warehouse Production* ini dijadikan referensi dalam penelitian sebagai pembelajaran untuk penerapan metode ELECTRE.
4. Pada artikel Rancang Bangun Aplikasi Rekomendasi Wisata Kuliner dengan Metode *Elimination Et Choix Traduisant La Realite* (ELECTRE), dijadikan sebagai referensi untuk penelitian yang sedang dilakukan ini. Dalam hal perancangan sistem pendukung keputusan dengan metode ELECTRE pada aplikasi *handphone* yaitu android.
5. Artikel Pemilihan Paket Wisata Menggunakan Metode *Analytical Hierarchy Process* (AHP) ini dijadikan referensi penelitian dengan metode dan hasil yang berbeda namun kriteria yang digunakan sama. Metode yang digunakan untuk penelitian ini adalah ELECTRE dan hasilnya ialah berupa aplikasi *mobile* yaitu android.