

BAB III

LANDASAN TEORI

3.1. SISTEM PENDUKUNG KEPUTUSAN (SPK)

Sistem Pendukung Keputusan merupakan sebuah informasi yang berbasis komputer interaktif, fleksibel dan dapat di adaptasikan, dikembangkan secara khusus untuk mendukung solusi masalah manajemen yang tidak terstruktur untuk peningkatan pengambilan keputusan yang lebih baik. (wijaya, 2015)

Pada dasarnya sistem pendukung keputusan dirancang untuk mendukung seluruh tahapan dalam pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam pengambilan keputusan sampai mengevaluasi pemilihan alternatif.

3.2. MANFAAT SISTEM PENDUKUNG KEPUTUSAN (SPK)

Sistem pendukung Keputusan mempunyai beberapa manfaat antara lain:

1. Dapat memperluas kemampuan dalam mengambil keputusan dalam memproses data dan informasi bagi pemakainya
2. Membantu dalam pengambilan keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak teratur.

3. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat dipercaya.
4. Dapat dijadikan referensi dalam pengambilan keputusan dalam memahami persoalan atau masalah yang ada karena mampu menyajikan berbagai alternatif pemecahan.

3.3. Technique for Order Preference by Similarity to Ideal Solution(TOPSIS)

Technique for Order Preference by Similarity to Ideal Solution(TOPSIS) adalah salah satu metode yang digunakan untuk pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang. TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari nilai solusi ideal negatif. Dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi yang optimal. Solusi Ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut. Sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut TOPSIS dengan mempertimbangkan keduanya. Jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya susunan prioritas alternatif bisa dicapai. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan. Hal ini disebabkan konsepnya yang cukup sederhana mudah dipahami, komputasinya

efisien dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan(Kurniasih, 2017).

3.4. LANGKAH- LANGKAH *TOPSIS*

Untuk melakukan perhitungan TOPSIS ada beberapa langkah yang harus dilakukan, antara lain:

1. Menentukan matriks keputusan yang ternormalisasi

Pada perhitungan TOPSIS memerlukan adanya rating kinerja setiap alternatif (A_i) pada setiap kriteria(C_j)yang ternormalisasi, yaitu:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_i^m x_{ij}^2}}$$

Keterangan:

X_{ij} : Merupakan rating kinerja alternatif ke-i terhadap atribut ke-j

R_{ij} : adalah elemen dari matrik keputusan yang ternormalisasi

2. Menghitung matriks keputusan ternormalisasi terbobot

$$y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1j} \\ y_{21} & y_{22} & \dots & y_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i1} & y_{i2} & \dots & y_{ij} \end{bmatrix} \text{ untuk } y_{ij} = w_j r_{ij}$$

Keterangan:

W_j : adalah bobot dari kriteria ke-j

Y_{ij} : adalah elemen dari matriks keputusan yang ternormalisasi terbobot

3. Menghitung matriks solusi Ideal Positif dan matriks solusi ideal negatif

$$A_i^- = (y_{1+}, y_{2+}, y_{3+}, \dots, y_{9+})$$

$$A_i^+ = (y_{1-}, y_{2-}, y_{3-}, \dots, y_{9-})$$

Keterangan:

A_i^- : Hasil nilai alternatif ideal negatif

A_i^+ : Hasil nilai alternatif ideal positif

Y_i : nilai alternatif hasil dari perhitungan

4. Menghitung jarak antara nilai terbobot setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_{ij}^+ - y_j^+)^2};$$

Keterangan:

Y_i^+ : elemen dari matriks solusi ideal positif

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij}^- - y_j^-)^2};$$

Keterangan:

Y_j^- : elemen dari matriks solusi ideal negatif

5. Menghitung nilai preferensi

$$V_i = \frac{D_i^-}{D_i^- + D_i^+};$$

Keterangan:

Nilai ci yang lebih besar menunjukkan prioritas alternative

3.5. PHP

PHP adalah singkatan dari Hypertext Processor yang digunakan sebagai bahasa script server-side dalam pengembangan web disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan web dapat dibuat dinamis sehingga maintenance situs web tersebut lebih mudah dan lebih efisien. Php ditulis dengan bahasa C(Suhartanto, 2017).

3.6. MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB sekitar pada tahun 1994-1995. Tujuan MySQL untuk mengembangkan aplikasi web untuk klien. MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang database sebagai pengelolaan datannya. Kepopuleran MySQL karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakannya dan bersifat open source dan free pada berbagai platform kecuali pada windows yang bersifat shareware(Suhartanto, 2017).

3.7. UML(Unified Modelling Language)

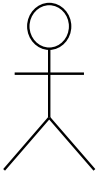
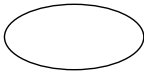
UML(*Unified Modeling Language*) merupakan sebuah bahasa yang digunakan untuk menentukan, visualisasi, kontribusi dan mendokumentasi bagian dari informasi yang digunakan dalam suatu proses perancangan perangkat lunak. UML menawarkan sebuah standar untuk merancang model


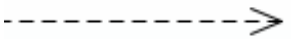
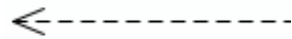
sebuah sistem. Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras dan sistem operasi dan jaringan apapun. Tetapi karena *UML* menggunakan class dan operation dalam konsep dasarnya. Maka lebih cocok untuk penulisan piranti lunak dalam bahasa- bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C . *UML* berorientasi objek menerapkan beberapa diagram diantaranya *Use Case Diagram*, *Class Diagram*, *Sequence Diagram* dan *Activity Diagram*.(Permatasari, 2016).

3.7.1. Diagram Use Case

Secara umum Use Case Diagram merupakan sebuah diagram yang menggambarkan interaksi antara aktor dengan sistem.

Tabel 3.1. *Use Case* (Dharwiyanti, 2012)

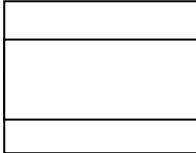
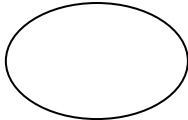

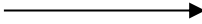

No	Gambar	Nama	Keterangan
1.		Actor	Actor berfungsi untuk mempresentasikan suatu yang berinteraksi dengan sistem
2.		Use Case	Use case berguna untuk berinteraksi

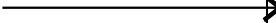
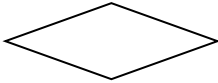
			dari actor dengan sistem
3.		Associate	Menghubungkan objek yang satu dengan objek yang lain
4.		Include	Use case yang dituju harus melewati sebuah proses lain
5.		Extend	Use case yang dituju dapat berdiri sendiri tanpa harus melewati sebuah proses lainnya

3.7.2. Class Diagram

Class Diagram berguna untuk mendiskripsikan suatu struktur object sistem dan menggambarkan kelas-kelas dalam sistem yang dapat saling berhubungan dengan kelas lain yang mempunyai atribut dan operation.

Tabel 3.2. *Class Diagram* (Dharwiyanti, 2012)

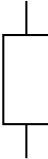
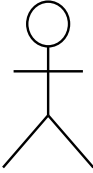

No	Gambar	Nama	Keterangan
1.		<i>Class</i>	Terdiri dari nama, atribut dan operasi, yang berfungsi memberi karakteristik pada data yang memiliki suatu objek di dalam class
2.		<i>Interface</i>	Konsep interface dalam pemograman berorientasi objek
3.		Association	Menghubungkan kelas yang satu dengan kelas yang lain
4.		<i>Directed Association</i>	Relasi antara kelas dengan kelas yang satu dengan kelas yang lainnya
5.		<i>Generalisasi</i>	Fitur warisan dari konsep yang berorientasi objek

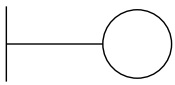

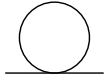
6.		<i>Depedency</i>	Relasi antar kelas yang ketergantungan antar kelas lainnya
7.		<i>Aggregation</i>	Relasi semua kelas dengan makna semua bagian

3.7.3. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek dan memberi tanda atau mengindikasikan komunikasi antara satu objek dengan objek lainnya.

Tabel 3.3. *Sequence Diagram* (Dharwiyanti, 2012)




No	Gambar	Nama	Keterangan
1.		<i>Life Liine</i>	Garis yang bersifat individu dalam sebuah interaksi
2.		<i>Interface</i>	Konsep interface dalam pemograman berorientasi objek
3.		<i>Message</i>	Pesan berupa permintaan dari operasi target


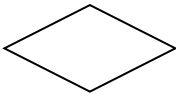
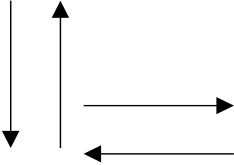
4.		<i>Boundary</i>	Penggambaran dari form
5.		<i>Control Class</i>	Penghubung antara boundary dengan table
6.		<i>Entity Class</i>	Menyimpan data dan informasi

3.7.4. Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan sebuah aktivitas dari sistem atau proses bisnis . Dengan catatan diagram aktivitas hanya menggambarkan aktivitas sistem bukan aktivitas actor.

Tabel 3.4 Activity Diagram

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Kelas yang saling berinteraksi satu dengan yang lain
2.		<i>Action</i>	Eksekusi atau tindakan yang dilakukan dari suatu aksi
3.		<i>Initial Node</i>	Awal dari bentuk Objek

4.		<i>Activity</i> <i>Final Node</i>	Bentuk objek akhir
5.		<i>Decision</i>	Digunakan untuk mengambil suatu keputusan/tindakan
6.		<i>Line</i> <i>Connector</i>	Digunakan untuk menghubungkan symbol