

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

Berdasarkan penelitian yang dilakukan oleh (Borman et al., 2020) dengan judul ” Implementasi Metode TOPSIS Pada Sistem Pendukung Keputusan Pemilihan Biji Kopi Robusta Yang Bernilai Mutu Ekspor (Studi Kasus: PT. Indo Cafco Fajar Bulan Lampung)” tujuan penelitian ini adalah untuk mengetahui pemilihan biji kopi robusta yang bernilai mutu,

Berdasarkan penelitian yang dilakukan oleh (Asrul & Zuhriyah, 2021) dengan judul ”Sistem Pendukung Keputusan Pendistribusian Air Bersih Menggunakan Mobil Tangki Pada Pdam Kota Makassar Dengan Menggunakan Metode Topsis” penelitian yang dilakukan oleh penelitian ini bertujuan untuk menentukan distribusi air bersih di Pdam Kota Makasar dengan menggunakan metode topsis

Berdasarkan penelitian yang dilakukan oleh (Putra et al., 2020) dengan judul “metode topsis dalam sistem pendukung keputusan pemilihan objek wisata” Penelitian ini bertujuan untuk menentukan objek wisata populer di kabupaten lima puluh, Sumatera Barat dengan menggunakan metode topsis.

Berdasarkan penelitian yang dilakukan oleh (Gunawan, 2020) dengan judul “Penerapan Metode Topsis Untuk Pengangkatan Karyawan Kontrak Menjadi Karyawan Tetap (Studi Kasus: Pt Hanuraba Sawit Kencana)” Penelitian ini bertujuan untuk menentukan pengangkatan karyawan kontrak menjadi karyawan tetap dengan menggunakan metode tophis.

Berdasarkan penelitian yang dilakukan oleh (Nalatissifa & Ramdhani, 2020) dengan judul “Sistem Penunjang Keputusan Menggunakan Metode Topsis Untuk Menentukan Kelayakan Bantuan Rumah Tidak Layak Huni (Rtlh) Pada Desa Sumbaga” Penelitian ini bertujuan untuk menentukan Kelayakan Bantuan Rumah Tidak Layak Huni (Rtlh) Pada Desa Sumbaga

II.2. Uraian Teoritis

II.2.1. Sistem Pendukung keputusan

Sistem pendukung keputusan merupakan alternatif bagian dalam sistem informasi berbasis komputer. tujuan dari penerapan sistem pendukung keputusan yaitu untuk mendukung pihak pengambil keputusan, memilih alternatif yang terbaik melalui proses pengolahan data dengan menggunakan model pengambil keputusan (Mesran, Pristiwanto, Sihombing, 2018).

Sistem Pendukung Keputusan adalah suatu sistem yang ditujukan untuk mendukung manajemen pengambilan keputusan. Sistem berbasis model yang terdiri dari prosedur-prosedur dalam pemrosesan data dan pertimbangannya untuk membantu manajer dalam mengambil keputusan agar berhasil mencapai Tujuannya.(Nofriansyah dan Dicky. 2014)

II. 2.2 Metode Topsis (*Technique For Others Preference By Similarity To Ideal Solution*)

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi 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 (Borman et al., 2020)

TOPSIS 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 sederhana, mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan (Gunawan, 2020).

II. 2.2.3 Langkah – Langkah Metode Topsis

Menurut (Gunawan, 2020) terdapat langkah –langkah dalam menggunakanmetode topsis yaitu :

1. Membangun normalized decision matrix Elemen rij hasil dari normalisasi decision matrix R dengan metode Euclidean length of a vector adalah :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \dots\dots\dots(1)$$

Dimana :

rij= hasil dari normalisasi matriks

keputusan Ri = 1,2,3,...,m;

j = 1,2,3,...,n;

Membangun weighted normalized decision matrix Dengan bobot W =

(w1,w2,...,wn), maka normalisasi bobot matriks V adalah :

$$V = \begin{bmatrix} w_{11}r_{11} & \cdots & w_{1n}r_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1}r_{m1} & \cdots & w_{nm}r_{nm} \end{bmatrix} \dots\dots\dots(2)$$

2. Menentukan solusi ideal positif dan solusi ideal negatif Solusi ideal positif dinotasikan dengan A+ dan solusi ideal negatif dinotasikan dengan A, sebagai berikut : Menentukan solusi ideal (+) dan (-)

$$A^+ = \{(\max v_{ij})(\min v_{ij} | j \in J'), i = 1,2,3, \dots m\} = \{v_1^+, v_2^+, \dots v_m^+\} .(3)$$

$$A^- = \{(\max v_{ij})(\min v_{ij} | j \in J'), i = 1,2,3, \dots m\} = \{v_1^-, v_2^-, \dots v_m^-\} .(4)$$

Dimana :

V_{ij} =elemen matriks V baris ke-i dan kolom ke- j

$J = \{j=1,2,3,...,n \text{ dan } j \text{ berhubungan dengan benefit criteria}\}$
 $J' = \{j=1,2,3,...,n \text{ dan } j \text{ berhubungan dengan cost criteria}\}$

3. Menghitung separasi

Separation measure ini meruAlternatif pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif. Perhitungan matematisnya adalah sebagai berikut :

Separation measure untuk solusi ideal positif

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \text{ dengani } = 1,2,3,...,m. \dots\dots\dots(5)$$

Dimana :

$J = \{j=1,2,3,..,n \text{ dan } j \text{ meruAlternatif benefitcriteria}\}$

$J' = \{j=1,2,3,..,n \text{ dan } j \text{ meruAlternatif cost criteria}\}$

Separation measure untuk solusi ideal negatif

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \text{ dengani } = 1,2,3,...,n(6)$$

Dimana :

$J = \{j=1,2,3,...,n \text{ dan } j \text{ meruAlternatif benefit criteria}\}$
 $J' = \{j=1,2,3,...,n \text{ dan } j \text{ meruAlternatif cost criteria}\}$

4. Menghitung kedekatan relatif terhadap solusi ideal

Kedekatan relatif dari alternatif dengan solusi ideal direpresentasikan dengan :

$$C_i = \frac{S_i^-}{S_i^- - S_i^+}$$

,\n
dengan $0 < \alpha < 1$ dan $i = 1, 2, 3, \dots, m$ (7)

a. Meranking alternatif

Alternatif dapat diranking berdasarkan urutan. Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal dan berjarak terjauh dengan solusi ideal negatif.

II.2.4. PHP (*Hypertext Preprocessor*)

Menurut Jubille Enterprise (2017:1) PHP adalah bahasa pemrograman yang digunakan untuk membuat sistem berbasis *website*. Sebagai sebuah sistem, *website* tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis, artinya *website* tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya, menampilkan produk yang berbeda-beda untuk setiap pengunjung). Artinya interaktif *website* tersebut dapat memberi *feedback* bagi *user* (misalnya, menampilkan hasil pencarian produk).

PHP merupakan alternatif bahasa pemrograman berjenis *service-side*. Dengan demikian, PHP akan diproses oleh *server* yang hasilnya akan dikirim kembali ke *browser*. Oleh karena itu, salah satu *tool* yang harus tersedia sebelum memulai pemrograman PHP adalah *server*.

II.2.5. MySQL

Menurut Solichin (2016:138), MySQL merupakan salah satu perangkat lunak basis data yang sangat populer. Saat ini tersedia versi MySQL yang berbayar

(MySQL *Enterprise Edition*), namun tetap tersedia versi MySQL yang gratis (MySQL *Community Edition*). MySQL *Community Edition* dapat diunduh secara gratis, dan bebas digunakan dalam berbagai keperluan. Walaupun demikian, secara fitur dan kemampuan, MySQL versi tidak terlalu jauh berbeda dengan versi berbayar.

Kelebihan MySQL adalah sebagai berikut:

1. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan alternatif standar bahasa dunia dalam pengolahan data/.
3. *Super performance* dan *realible*, tidak bisa diragukan, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*easy of use*).
5. Memiliki dukungan *support (group)* penggunaan MySQL.
6. *Multiuser*, dimana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik.

II.2.6. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) digunakan sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem perangkat lunak. Dengan adanya bahasa yang bersifat standar, komunikasi perancang dengan pemrogram (komunikasi antar anggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus. Adapun pengertian UML menurut

para ahli dapat dipaparkan sebagai berikut:

Menurut Shofwan Hanief & Dian Pramana (2018 : 166) menyatakan bahwa *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisas, merancang dan mendokumentasikan sistem piranti lunak.

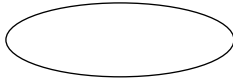
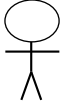
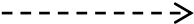
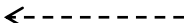
Menurut Eng RH. Sianipar (2016 :75) menyatakan bahwa UML meruAlternatif singkatan *Unifed Modelling Language*, yang telah menjadi notasi populer untuk mempresentasikan perancangan atas sebuah program berorientasi objek.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1. *Use case Diagram*

Menurut Sri Mulyani (2016 : 245) menyatakan bahwa *Use Case Diagram* yaitu *diagram* yang menggambarkan dan merepresentasikan aktor, *use case* dan *dependencies* suatu proyek dimana tujuan dari diagram ini adalah menjelaskan konsep hubungan antara sistem dengan dunia luar. Jadi, dapat disimpulkan *use case* adalah langkah-langkah atau urutan kegiatan yang dilakukan aktor dan sistem informasi yang akan dibuat. Secara singkat, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan.

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<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 adalah <i>abstraction</i> dari orang atau 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. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi 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 antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , meruAlternatif 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> , meruAlternatif perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Sri Mulyani : 2016: 245)

2. Diagram Aktivitas (*Activity Diagram*)

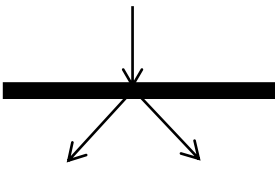
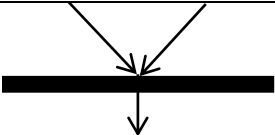
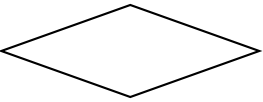
Activity diagram menggambarkan rangkaian aliran dari aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Menurut Sri Mulyani (2016 : 249) *activity diagram* adalah diagram UML yang digunakan untuk menggambarkan alur aktivitas dari suatu proses. Menurut Muhammad Muslihudin dan Oktafianto (2016 :63) mengungkapkan diagram

aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.

Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2:

Tabel II.2. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan meruAlternatif 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 paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , 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 diagram</i> untuk menunjukkan siapa melakukan apa.

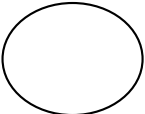
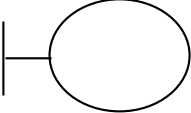
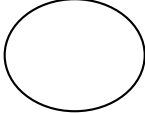
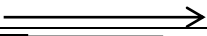
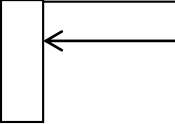
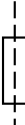
(Sumber : Sri Mulyani : 2016: 249)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

Menurut (Irmayani & Susyati, 2017), *Sequence Diagram* menggambarkan bagaimana sistem merespon kegiatan *user*. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , meruAlternatif 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 <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika sistem 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> , <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 : Irmayani & Susyati, 2017)

4. Diagram Kelas (*Class Diagram*)

Diagram kelas (*Class Diagram*) adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang kita kembangkan. Dan berikut ini meruAlternatif penjelasan mengenai *class diagram*:

Menurut Sri Mulyani (2016 : 247) menyatakan bahwa *class diagram* adalah diagram yang digunakan untuk mempresentasikan kelas, komponen-komponen kelas dan hubungan antara masing-masing kelas. Simbol *multiplicity class diagram* dapat dilihat pada Tabel II.4.:

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Sri Mulyani : 2016 : 247)