

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

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

Penelitian yang dijadikan rujukan yaitu penelitian mengenai “Sistem Pendukung Keputusan Seleksi Penerimaan Tenaga Kerja Kontrak Dengan Metode MOORA Pada Kantor Kepolisian Daerah Sumatera Utara” Hasil dari penelitian tersebut mendapatkan kesimpulan bahwa Metode MOORA dapat membantu pengambilan keputusan melalui seleksi penerimaan tenaga kerja kontrak yang ditentukan berdasarkan kriteria dengan menggunakan rumus yang hasilnya lebih akurat dan tepat sasaran. (Tanty Anggraeni¹, R. Mahdalena Simanjorang², 2019)

Selanjutnya penelitian mengenai “Sistem Pendukung Keputusan Penerimaan Jurnal Menerapkan Multi-Objective Optimization On The Basis Of Ratio Analysis (MOORA)” hasil yang didapat dari penelitian tersebut adalah dengan metode MOORA dapat membantu pihak terkait untuk mendapatkan pelamar terbaik dalam penerimaan jurnal pada PT.Waspada Medan. Karena sebelumnya dalam proses penerimaan jurnal data yang di peroleh dari hasil seleksi masih menggunakan cara manual yang sangat menguras waktu dan tenaga bagi Human Resources Department (HRD) perusahaan. (Aldi Muharsyah¹, Soraya

Rahma Hayati², M.Ikhsan Setiawan³, Heri Nurdiyanto⁴, Yuhandri⁵,
2018)

Penelitian selanjutnya mengenai “Sistem Pendukung Keputusan Pemilihan Alternatif Jaminan Kesehatan Masyarakat (Jamkesmas) Menerapkan Metode MOORA” Hasil dari penelitian tersebut mendapatkan kesimpulan bahwa Metode MOORA dapat membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah yang sifatnya semi terstruktur maupun yang tidak terstruktur dan dalam menentukan yang berhak atau tidak menjadi para alternatif Jamkesmas yang ditentukan berdasarkan kriteria dengan menggunakan rumus yang hasilnya lebih akurat dan tepat sasaran. (Mesran¹, Swandi Dedi Arnold Pardede², Arahman Harahap³, Andysah Putera Utama Siahaan⁴, 2018)

Penelitian lain yang dijadikan sebagai rujukan adalah penelitian mengenai “Sistem Pendukung Keputusan Pemilihan Sekolah Terbaik Dengan Metode MOORA” hasil dari penelitian tersebut menghasilkan informasi mengenai pemilihan sekolah terbaik, Dengan adanya sistem ini nantinya akan berguna bagi siswa siswi dalam mengambil keputusan melakukan pemilihan sekolah terbaik sesuai dengan kriteria-kriteria sekolah yang siswa siswi inginkan dan diharapkan dapat meningkatkan minat belajar siswa siswi. (Ermayanti Astuti¹, Nidia Anjelita Saragih², 2020)

II.2. Landasan Teoritis

II.2.1. Sistem pendukung keputusan

Sistem pendukung keputusan didefinisikan sebagai sistem berbasis komputer yang terdiri dari komponen-komponen yang saling berinteraksi, yaitu: sistem bahasa, sistem pengetahuan, dan sistem pemrosesan masalah (Turban, 2010).

Sistem Pendukung Keputusan (SPK) bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. SPK ditujukan untuk membantu para pengambil keputusan untuk memecahkan masalah semi dan atau tidak terstruktur dengan fokus menyajikan informasi yang nantinya bisa dijadikan sebagai bahan alternatif pengambilan keputusan yang terbaik.

II.2.2 Metode Multi-Objective Optimization by Ratio Analysis (MOORA)

Metode Multi-Objective Optimization by Ratio Analysis (MOORA) adalah metode yang diperkenalkan oleh Brauers dan Zavadkas (2006). Metode yang relatif baru ini pertama kali digunakan oleh Brauers dalam suatu pengambilan dengan multi- kriteria. Metode MOORA memiliki tingkat fleksibilitas dan kemudahan untuk dipahami dalam memisahkan bagian subjektif dari suatu proses evaluasi kedalam kriteria bobot keputusan dengan beberapa atribut pengambilan keputusan. Metode MOORA mudah dipahami dan fleksibel dalam memisahkan objek hingga proses evaluasi kriteria bobot keputusan. Metode

MOORA juga memiliki tingkat selektifitas yang baik karena dapat menentukan tujuan dan kriteria yang bertentangan, yaitu kriteria yang bernilai menguntungkan (Benefit) atau yang tidak menguntungkan (Cost). Langkah – langkah penyelesaian masalah menggunakan metode MOORA

II.2.3. 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 bahasa pemrograman berjenis service-side. Dengan demikian, PHP akan diproses oleh server yang hasil olahannya akan dikirim kembali ke browser. Oleh karena itu, salah satu tool yang harus tersedia sebelum memulai pemrograman PHP adalah server.

II.2.4. Makanan Kucing

Makanan kucing adalah makanan khusus yang diberikan dan dikonsumsi oleh kucing domestik. Makanan kucing harus memiliki kebutuhan gizi yang dibutuhkan oleh kucing, yaitu makanan yang banyak mengandung vitamin dan asam amino.

Makanan yang mengandung asam amino taurin contohnya adalah daging. Kekurangan zat taurin dalam jangka waktu panjang, misalnya jika kucing diberi makanan anjing yang mengandung sedikit taurin, dapat mengakibatkan degenerasi retina, kehilangan penglihatan, dan kerusakan jantung

II.2.4. Sublime Text

Sublime text merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi. Sublime text mempunyai fitur plugin tambahan yang memudahkan programmer (Supono dan Putratama, 2016:14).

II.2.5. Database

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan

satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Database mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah:

1. Redudansi dan inkonsistensi data
2. Kesulitan dalam pengaksesan data
3. Isolasi data untuk standarisasi
4. Multiuser
5. Keamanan data
6. Integritas data
7. Kebebasan data (Urva dan Siregar, 2016:93)

II.2.6. 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.

II.2.7. 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 merupakan 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:

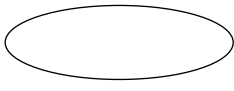
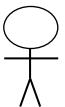
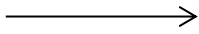
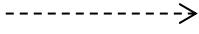
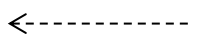
1. *Use case* Diagram

Menurut Sri Mulyani (2018 : 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 actor 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.
	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> , 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 : Sri Mulyani : 2018: 245)

2. Diagram Aktivitas (*Activity Diagram*)

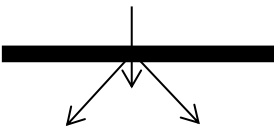
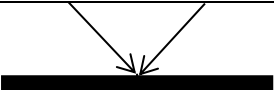
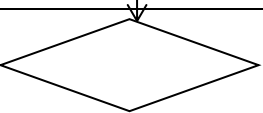
Activity diagram menggambarkan rangkaian aliran dari aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Menurut Sri Mulyani (2018 : 249) : “*Activity diagram* adalah diagram UML yang digunakan untuk menggambarkan alur aktivitas dari suatu proses”

Menurut Muhammad Muslihudin dan Oktafianto (2018 :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 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 paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, 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 : 2018: 249)

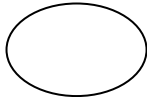
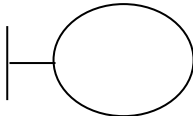
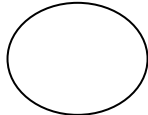
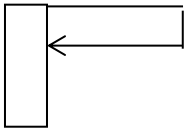
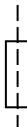
3. Diagram Urutan (*Sequence Diagram*)

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. *Sequence Diagram* yang dibuat yaitu berhubungan langsung dengan kegiatan utama dari sistem anggaran pendapatan dan belanja desa berbasis objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</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 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)

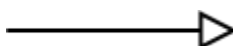
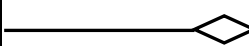
4. Diagram Kelas (*Class Diagram*)

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 merupakan penjelasan mengenai class diagram:

Menurut Sri Mulyani (2018 : 247) menyatakan bahwa : “Class Diagram adalah diagram yang digunakan untuk mempresentasikan kelas, komponen-komponen kelas dan hubungan antara masing-masing kelas” Simbol *class diagram* dan *multiplicity class diagram* dapat dilihat pada Tabel II.4 dan TabelII.5.:

Tabel II.4. Simbol Class Diagram

Gambar	Keterangan			
	<i>Generalization</i> , untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas dengan arti umum-khusus.			
<table border="1"><tr><td>Nama_kelas</td></tr><tr><td>+atribut</td></tr><tr><td>+operasi</td></tr></table>	Nama_kelas	+atribut	+operasi	<i>Class</i> , untuk sebuah kelas pada struktur sistem. Penulisan tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi.
Nama_kelas				
+atribut				
+operasi				
	<i>Interface</i> , untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP).			
	<i>Association</i> , digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum.			
	<i>Directed Association</i> , adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.			
	<i>Aggregation</i> , adalah relasi antar kelas dengan makna semua bagian.			
	<i>Dependency</i> , adalah relasi antar kelas dengan makna kebergantungan antar kelas.			

(Sumber : Sri Mulyani : 2018 : 247)

Tabel II.5. 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 : 2018 : 247)