

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian yang di rancang untuk memperkenalkan Metode *Technique For Others Reference By Si Milarity Toideal Solution*(TOPSIS). Salah satu metode yang bisa membantu proses pengambilan keputusan yang optimal untuk menyelesaikan masalah keputusan secara praktis. Hal ini disebabkan karena konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana(Khusumadewi, 2006).

Penelitian suatu sistem pendukung keputusan penyeleksian calon siswa baru di SMA Negeri 3 Garut ini contohnya. Hasil dari pada penelitian menunjukkan bahwa penggunaan sistem pendukung keputusan penyeleksian calon siswa baru di SMA Negeri 3 garut ini dapat membantu, mempermudah pekerjaan dan meminimalisir kesalahan yang dilakukan oleh panitia penyeleksi calon siswa baru dalam pengambilan keputusan penerimaan calon siswa baru.

Sistem Informasi Penilaian Kinerja Karyawan Menggunakan Metode *technique For Others Reference By Si Milarity Toideal Solution* (TOPSIS). Hasil dari penelitian ini juga menunjukkan bahwa penggunaan system informasi penilaian kinerja karyawan menggunakan metode TOPSIS ini dapat membantu, mempermudah proses penilaian terhadap karyawan dalam sebuah perusahaan yang dilakukan dengan kelengkapan administrasi, seleksi tertulis dan wawancara.

II.2. Landasan Teori

II.2.1. Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Hendra, 2012, 157).

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisir, saling tergantung satu sama lain dan terpadu (Kumorotomo dan Margono, 2016).

II.2.2. Sistem Pendukung Keputusan (SPK)

Menurut Mat dan Watson, Sistem Pendukung Keputusan (SPK) merupakan suatu sistem interaktif yang membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur. Sedangkan menurut Alter, Sistem Pendukung Keputusan (SPK) adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tidak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat. Dari definisi tersebut dapat disimpulkan bahwa Sistem Pendukung Keputusan (SPK) adalah aktivitas manajemen berupa pemilihan tindakan dari sekumpulan alternatif yang telah dirumuskan sebelumnya untuk memecahkan suatu masalah atau suatu konflik dalam manajemen. Sistem pendukung keputusan adalah suatu pendekatan

sistematis pada hakekat suatu masalah, pengumpulan fakta-fakta penentu yang matang dari alternatif yang dihadapi dan pengambilan tindakan yang paling tepat (Lia Ayu Ivanjelita, 2015).

Sistem Pendukung Keputusan merupakan suatu sistem interaktif yang membantu pengambil keputusan melalui penggunaan data dan model model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur. Dasar-dasar keputusan adalah sebagai berikut:

- a. Intuisi, yaitu keputusan diambil berdasarkan perasaan dan pemikiran si pengambil keputusan.
- b. Pengalaman, yaitu keputusan diambil berdasarkan kejadian-kejadian yang pernah dialami sebelumnya oleh si pengambil keputusan.
- c. Fakta, yaitu keputusan diambil berdasarkan data dan informasi yang telah dikumpulkan.
- d. Wewenang, yaitu keputusan diambil oleh pihak yang memiliki kekuasaan dan wewenang yang lebih tinggi.

Rasional, yaitu keputusan yang diambil harus logis atau dapat diterima akal sehat. Konsep Sistem Pendukung Keputusan (SPK) pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael S.Scott Morton dengan istilah *Decision System* (M. Arfan Rinaldi, 2013).

II.2.3. Metode *Technique For Other Reference by Similarity to Ideal Solution* (TOPSIS)

Technique for Order Performance 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 jarak terpanjang (terjauh) dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* (jarak antara dua titik) 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. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif (Siti Kholijah Ritonga, 2013).

Secara umum, prosedur atau langkah-langkah dalam metode TOPSIS (*Technique For Order Preference By Similarity To Ideal Solition*) meliputi:

1. Membuat matriks keputusan yang ternormalisasi.
2. Membuat matriks keputusan yang ternormalisasi terbobot.
3. Membuat matriks solusi ideal positif dan matriks solusi ideal negatif.
4. Menentukan jarak anatara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif.
5. Menentukan nilai nilai preferensi untuk setiap alternatif.

TOPSIS membutuhkan rating kinerja setiap alternatif A_i pada setiap kriteria C_j yang ternormalisasi, yaitu:

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

dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (y_{ij}) sebagai:

$$y_{ij} = w_i r_{ij} \quad \dots \dots \dots (2)$$

dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

$$A^+ = (y_{1^+}, y_{2^+}, \dots, y_{n^+}); \quad \dots \dots \dots (3)$$

$$A^- = (y_{1^-}, y_{2^-}, \dots, y_{n^-}); \quad \dots \dots \dots (4)$$

dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan sebagai:

$$D_i^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})^2}; \quad \dots \dots \dots (5)$$

Dengan $i = 1, 2, \dots, m$

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2}; \quad \dots \dots \dots (6)$$

Dengan $i = 1, 2, \dots, m$

Nilai dari preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; \quad \dots \dots \dots (7)$$

dengan $i = 1, 2, \dots, m$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i lebih dipilih.

II.2.4. MySQL

MySQL (*My Structured Query Language*) adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase* (Ratna, 2014).

II.2.5. Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Microsoft Visual Studio.Net 2003* yang diproduksi oleh *Microsoft* pada bulan februari 2002 *Microsoft* memproduksi teknologi net *Framework 1.0*, teknologi ini didasarkan atas susunan berupa net *Framework* sehingga setiap produk baru yang terkait dengan teknologi net akan selalu

berkembang mengikuti perkembangan net *Framework* tersebut. Pada perkembangan berikutnya mungkin untuk membuat program dengan teknologi net, sehingga memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi *cross platform*, yaitu dapat dikembangkan di sistem operasi *windows*, juga dapat dijalankan pada sistem operasi lain, misalnya pada sistem operasi *Linux*, seperti yang telah dilakukan oleh pemrograman Java oleh *sun microsystem*.

Visual Basic dibuat oleh *Microsoft*, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi *database*. *Visual Basic* merupakan bahasa pemrograman *event drive*, dimana program akan menunggu sampai ada respons dari *user* / pemakai program aplikasi yang dapat berupa kejadian atau event, misalnya ketika *user* mengklik tombol atau menekan Enter (Edy Winarno : 2010 : 83)

II.2.6. Unified Modelling Language (UML)




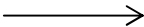
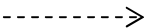
Hasil pemodelan pada OOAD (*Object Oriented Analysis And Design*) terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan

umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

a. *Usecase Diagram*

Usecase diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

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 control 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> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya




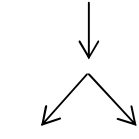
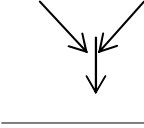
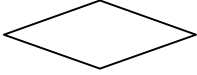
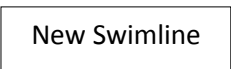
	adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

b. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Tabel II.2. Simbol *Activity Diagram*

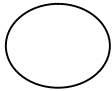

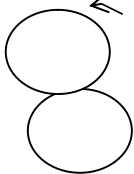

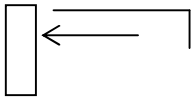


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel 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</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

c. *Sequence Diagram*

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

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 aplikasi 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 : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

d. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. (Windu Gata ; 2013 : 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 : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)