

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Dalam penyusunan skripsi ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini.

Tabel II.1. Penelitian Terdahulu

No	Nama	Judul	Hasil
1	Eka Andrita Gusdha ; 2012	Sistem Promosi Jabatan Karyawan dengan Metode Analytical Hierarchy Process (AHP) dan Multi-Attribute Utility Theory (MAUT) (Studi Kasus pada PT. Ginsa Inti Pratama)	Dari hasil penelitian dengan menggunakan metode MAUT untuk 20 sample data nilai yang degenerate secara random dengan menggunakan sistem pendukung keputusan promosi dan mutasi karyawan (SPKPMK), ternyata memberikan 3 rekomendasi promosi jabatan untuk karyawan. Sehingga dengan adanya SPK PMK dengan metode MAUT ini dapat membantu dan memudahkan manager dalam perencanaan karir (promosi atau mutasi) dengan lebih menghemat waktu, biaya, dan lebih objektif
2	Didik Trisianto ; 2012	Sistem Pendukung Keputusan Untuk Menentukan Tingkat Kesejahteraan Masyarakat	Penilaian terhadap hasil ujian pelamar sudah ditentukan dengan nilai pasti dan variabel yang telah ditentukan oleh Ketua Yayasan dan Ketua STIKes Hang Tuah Pekanbaru sebagai dasar sistem penilaian. Sistem dapat memudahkan bagian personalia dalam memberikan penilaian terhadap hasil ujian masing-masing pelamar secara cepat dan efektif. Sistem ini dapat memudahkan

			bagian personalia dan pelamar dalam melihat hasil ujian karena hasil ujian masing-masing pelamar langsung ditampilkan setelah pelamar selesai melaksanakan ujian
--	--	--	--

II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer yang membantu para pengambil keputusan mengatasi berbagai masalah melalui interaksi Pengambilan keputusan (*Decision Making*) adalah melakukan penilaian dan menjatuhkan pilihan. Keputusan ini diambil setelah melalui beberapa perhitungan dan pertimbangan alternatif. *Decision support system* atau sistem pendukung keputusan secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, Sistem pendukung keputusan didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manager maupun sekelompok manager dalam memecahkan masalah semi terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Muhammad Yudin Ritonga ; 2014 : 19).

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang *fleksibel, interaktif*, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur) dan sistem pemrosesan atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Dicky Nofriansyah ; 2014 : 1).

II.2.1. Tujuan Sistem Pendukung Keputusan

Tujuan yang ingin dicapai dalam SPK adalah :

1. Mendukung pengambil keputusan dalam menyelesaikan porsi permasalahan yang dapat distrukturkan.
2. Membantu para pengambil keputusan agar mereka memfokuskan diri terhadap porsi permasalahan yang tidak dapat distrukturkan.
3. Sistem pendukung keputusan tidak ditekankan untuk membuat keputusan, melainkan melengkapi kemampuan untuk mengolah informasi yang diperlukan untuk membuat keputusan. Dengan kata lain, sistem pendukung keputusan membantu manusia dalam proses membuat keputusan, bukan menggantikan perannya dalam mengambil keputusan (Suhermin ; 2012 : 4).

II.2.2. Manfaat Sistem Pendukung Keputusan

Manfaat yang dihasilkan dari SPK antara lain :

1. Memperbesar kemampuan pengambil keputusan untuk memproses informasi dan pengetahuan.
2. Memperbesar kemampuan pengambil keputusan dalam menangani permasalahan yang kompleks, berskala besar, dan menggunakan banyak waktu.
3. Memperpendek waktu pengambilan keputusan.
4. Mendorong pelaksanaan eksplorasi bagi pengambil keputusan.
5. Memberikan pendekatan baru dalam proses berpikir mengenai lingkup permasalahan dan konteks keputusan.
6. Membangkitkan bukti baru dalam mendukung sebuah keputusan atau konfirmasi dari asumsi yang sudah ada.
7. Menghasilkan keunggulan strategis dan kompetitif di dalam persaingan antar organisasi.
8. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan (Suhermin ; 2012 : 4).

Pengambilan keputusan meliputi beberapa tahap dan melalui beberapa proses. Pengambilan keputusan meliputi empat tahap yang saling berhubungan dan berurutan. Empat proses tersebut adalah :

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi dan menguji kelayakan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan.

4. *Implementation*

Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana proses pengambilan keputusan (Suhermin ; 2012 : 4).

II.3. Metode MAUT

Multi Attribute Utility Theory (MAUT) merupakan suatu skema yang evaluasi akhir, $v(x)$, dari suatu objek x didefinisikan sebagai bobot yang dijumlahkan dengan suatu nilai yang relevan terhadap nilai dimensinya. Ungkapan yang biasa digunakan untuk menyebutnya adalah nilai utilitas. *MAUT* digunakan untuk merubah dari beberapa kepentingan kedalam nilai numerik dengan skala 0-1 dengan 0 mewakili pilihan terburuk dan 1 terbaik. Hal ini

memungkinkan perbandingan langsung yang beragam ukuran (Riadhil Jannah ; 2015 : 80)

Analisis *MAUT* secara eksplisit mengidentifikasi langkah-langkah yang digunakan untuk mengevaluasi alternatif dan membantu untuk mengidentifikasi alternatif-alternatif yang berkinerja baik dengan penekanan khusus pada langkah-langkah yang dianggap relatif lebih penting. Fungsi utilitas *MAUT* terdiri dari :

1. *MAUT additive* :

$$U(x_1) = \sum_{i=1}^n w_i u_i(x_i) \dots\dots\dots(1)$$

2. *MAUT non-additive* :

$$U(x) = \sum_{i=1}^n w_i u_i(x_i) + \sum_{i=1}^n \sum_{j>1} w_{ij} u_i(x_i) u_j(x_j) \dots\dots\dots(2)$$

Keterangan :

$U(x_1)$: Total utilitas alternatif ke- 1

W_i : Bobot atribut ke- i ($\sum w = 1$)

$U_i(x_i)$: Fungsi atribut atribut ke- i

I : Atribut 1, 2, 3., n

x : Kriteria ke –i

Langkah-langkah yang dilakukan dalam metode *MAUT* secara ringkas adalah sebagai berikut :

1. Memecah sebuah keputusan ke dalam dimensi yang berbeda
2. Menentukan bobot relatif pada masing-masing dimensi
3. Mendaftar semua alternatif
4. Menghitung utilitas untuk masing-masing alternatif sesuai atributnya menggunakan rumus.

5. Mengalikan utilitas dengan bobot untuk menemukan nilai masing-masing alternatif menggunakan rumus (Usman Effendi ; 2017 : 34)

II.4. Perekrutan Pegawai

Rekrutmen merupakan komunikasi dua arah. Para pelamar menghendaki informasi yang akurat mengenai seperti apa rasanya bekerja di dalam sebuah organisasi. Sedangkan organisasi sangat menginginkan informasi yang akurat tentang pelamar tersebut jika kelak mereka menjadi karyawan. Rekrutmen memiliki beberapa tujuan, antara lain sebagai berikut :

1. Untuk memikat sebagian besar pelamar kerja sehingga organisasi akan mempunyai kesempatan yang lebih besar untuk melakukan pemilihan terhadap calon-calon pekerja yang dianggap memenuhi standar kualifikasi organisasi.
2. Tujuan pasca pengangkatan adalah penghasilan karyawan-karyawan yang merupakan pelaksana-pelaksana yang baik dan akan tetap bersama dengan perusahaan sampai jangka waktu yang masuk akal.
3. Meningkatkan citra umum organisasi, sehingga para pelamar yang gagal mempunyai kesan-kesan positif terhadap organisasi atau perusahaan (Shafira Adrizayani ; 2012 : 1).

II.5. Visual Basic 2010

Microsoft Visual Basic 2010 adalah salah satu komponen Microsoft Visual Studio 2010. Software ini diluncurkan Microsoft pada tanggal 12 April

2010 dengan nama kode Dev10 dan menggunakan .Net Framework 4.0. *Integrated Development Environment* (IDE) pada Visual studio 2010 telah didesain ulang sehingga lebih enak dipandang dan digunakan programmer.

Untuk kode editor-nya, Visual Basic 2010 telah menambah fitur *highlights reference*. Ketika satu simbol/kode dalam bahasa pemrogramannya dipilih, maka simbol/kode yang sama, meskipun penggunaannya berbeda akan terlihat berwarna sama. Misal jika kode math dipilih, seluruh kode math akan terlihat berwarna sama (Fadillah ; 2014 : 10).

II.6. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database. SQL Server adalah sebuah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Komputer ; 2013 : 2)

II.7. Teknik Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

- a. Berisi data yang diperlukan.
- b. Memiliki sesedikit mungkin redundansi.
- c. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
- d. Mengefisienkan update.
- e. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal. Adapun tahap normalisasi sebagai berikut :

1. Tahap tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

2. Tahap normal tahap pertama (1st Normal Form)

Sebuah table disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing cell bernilai tunggal

3. Tahap normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

4. Tahap normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \twoheadrightarrow Y \twoheadrightarrow Z$, dimana Y mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X , maka :

- a. X haruslah superkey pada tabel tersebut.
- b. Atau Y merupakan bagian dari primary key pada tabel tersebut.

5. Boyce Code Normal Form (BCNF)

- a. Memenuhi 1st NF
- b. Relasi harus bergantung fungsi pada atribut superkey

6. Tahap Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF) (Tawar ; 2012 : 7)

II.8. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

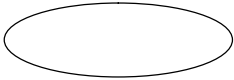
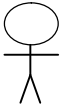
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

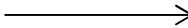
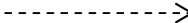
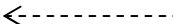
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case Diagram*

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

Tabel II.2. 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

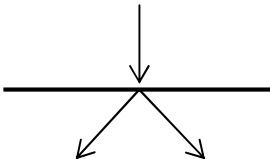
	<i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <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.

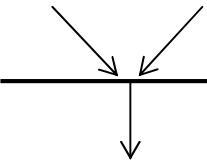
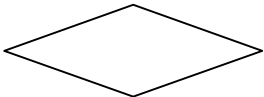
(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (Activity Diagram)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.3. 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.

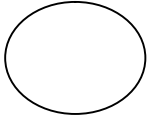
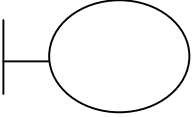
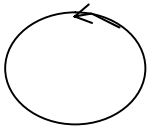
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

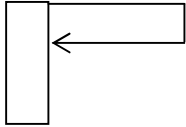
(Sumber : Windu Gata ; 2013 : 6)

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. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</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 : Windu Gata ; 2013 : 7)

4. Class Diagram (Diagram Kelas)

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.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

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 : Windu Gata ; 2013 : 9)