

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Dalam penelitian terdahulu, penulis mengambil beberapa referensi yang berkaitan dengan latar belakang masalah yang penulis ambil. Salah satunya yaitu penelitian yang dilakukan oleh Darniyati, Riya, Emilya Ully Artha, Agus Setiawan, 2018 “Sistem Pendukung Keputusan Seleksi Pemain Pada Cabang Olahraga Futsal Dengan Metode *Profile Matching*” yang berkesimpulan bahwa sistem yang dibuat dan dihasilkan ini dapat mempermudah dalam membantu seorang pelatih dalam menyeleksi pemain yang akan di masukkan dalam tim inti club dengan menggunakan kriteria – kriteria yang telah ditentukan.

Penelitian yang dilakukan oleh Mervin Angeline, Feriana Astuti, 2018 “Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Menggunakan Metode *Profile Matching*” berkesimpulan bahwa dengan menggunakan sistem pendukung keputusan pemilihan karyawan terbaik dengan menggunakan metode *profile matching* dapat mempermudah perusahaan dalam mengambil keputusan.

Penelitian yang dilakukan oleh Putu Sugiartawan, Heruzulkifli Rowa, Nurul Hidayat, 2018 “Sistem Pendukung Keputusan Kenaikan Jabatan Menggunakan Metode *Profile Matching*” berkesimpulan dengan menggunakan sistem pendukung keputusan menggunakan metode *profile matching* dalam memutuskan kenaikan jabatan menjadi lebih mudah dan efisien.

Penelitian yang dilakukan oleh Dedek Indra Gunawan HTS “Analisa Metode *Profile Matching* Untuk Pengangkatan Kepala Sekolah (Studi Kasus Yayasan Perguruan Al-Azhar Medan)” berkesimpulan bahwa *Profile Matching* sangat baik untuk memberikan ranking kepada Yayasan dalam menentukan kepala sekolah yang layak untuk memimpin suatu unit.

II.2. Uraian Teoritis

II.2.3. Sistem Pendukung Keputusan

Adapun Sistem Pendukung Keputusan (SPK) banyak di definisikan dari sudut pandang yang berbeda. Berikut ini beberapa definisi yang di kemukakan oleh beberapa ahli.

1. Sistem Pendukung Keputusan (SPK) merupakan sistem yang mampu memberikan kemampuan pemecahan masalah dengan kondisi semi terstruktur dan tak terstruktur. (Erna Ningsih, dkk : 2017)
2. Sistem Pendukung Keputusan (SPK) adalah bagian dari sistem inFormasi berbasis komputer termasuk sistem berbasis pengetahuan atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan(Sri Primaini, dkk : 2015).

II.1.1. Metode Profile Matching

Profile Matching adalah sebuah mekanisme pengambilan keputusan dengan mengansumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Dalam proses *profile matching*, akan dilakukan proses perbandingan antara

kompetensi individu ke dalam kompetensi standar, dalam hal ini *profile* asisten praktikum yang ideal sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*). Semakin kecil *gap* yang dihasilkan maka bobot nilainya semakin besar. Calon yang memiliki bobot nilai yang besar berarti memiliki peluang lebih besar untuk dapat menempati posisi sebagai asisten praktikum. Dan berikut merupakan langkah-langkah perhitungan dalam *profile matching* (Budi Darma Setiawan, dkk:2015).

II.1.2. Proses Perhitungan Metode Profile Matcing

Langkah-langkah dalam penyelesaian metode *Profile Matching* adalah sebagai berikut:

1. Pemetaan *Gap* Kompetensi

Yang dimaksud dengan *gap* disini adalah beda antara profil pegawai dengan profil jabatan atau dapat ditunjukkan pada rumus berikut ini: (Budi Darma Setiawan, dkk:2015).

$$\text{Gap} = \text{Profil Pegawai} - \text{Profil Jabatan} \dots\dots\dots (1)$$

2. Pembobotan *GAP* kompetensi.

Setelah diperoleh *gap* pada masing-masing karyawan, setiap profil karyawan diberi bobot nilai dengan patokan tabel bobot nilai *gap* (Budi Darma Setiawan, dkk:2015).

Tabel II. 1. Tabel bobot nilai gap

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih (kompetensi sesuai dengan yang dibutuhkan)
2	1	4,5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kekurangan 1 tingkat/level

4	2	3,5	mpetensi individu kelebihan 2 tingkat/level
5	-2	3	mpetensi individu kelebihan 2 tingkat/level
6	3	2,5	mpetensi individu kelebihan 3 tingkat/level
7	-3	2	mpetensi individu kelebihan 3 tingkat/level
8	4	1,5	mpetensi individu kelebihan 4 tingkat/level
9	-4	1	mpetensi individu kelebihan 4 tingkat/level

(Sumber: Budi Darma Setiawan,dkk:2015)

3. Perhitungan dan Pengelompokan *Core* dan *Secondary Factor*

Setelah diperoleh nilai *gap* selanjutnya diberikan bobot untuk masingnilai *gap*.

Perhitungan dan pengelompokan *Core factor* (faktor utama)dan *Secondary factor*

(faktor pendukung).Untuk perhitungan *core factor* dapat ditunjukkan pada rumus

berikut: (Budi Darma Setiawan,dkk:2015).

$$NSC = \frac{\sum NC (i,s,p)}{\sum IC}$$

Keterangan :

NCF : Nilai rata-rata core factor

NC(I, s, p) : Jumlah total nilai core factore

IC : Jumlah item core factor

Sedangkan untuk perhitungan secondary factor dapat ditunjukan pada

rumus di bawah ini :

$$NCS = \frac{\sum NS (i,s,p)}{\sum IS}$$

Keterangan :

NSF : Nilai rata-rata secondary factor

NC(i, s, p) : Jumlah total nilai secondary factore

ICS : Jumlah item secondary factor

4. Perhitungan Nilai Total

Dari hasil perhitungan dari tiap aspek di atas kemudian dihitung nilai total berdasarkan presentasi dari core dan secondary yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil. Contoh perhitungan dapat dilihat pada rumus di bawah ini

$$N(i, s, p) = (x)\%NCF(i, s, p)(x)\%NSF(i, s, p)$$

Keterangan :

(i,s,p) : (Intelektual, Sikap Kerja, Perilaku)

N(i,s,p) : Nilai total dari aspek

NCF(i,s,p) : Nilai rata-rata core factor

NSF(i,s,p) : Nilai rata-rata secondary factor

(x)% : Nilai persen yang diinputkan

5. Perhitungan Penentuan Hasil Akhir atau Ranking

Hasil akhir dari proses ini adalah ranking dari kandidat yang diajukan.

Adapun perhitungan tersebut ditunjukkan dengan persamaan (4).

$$Ha = (x)\%Ni + (x)\%Ns + (x)\%Np$$

Keterangan:

Ha : Hasil Akhir

Ni : Nilai Kapasitas Intelektual

Ns : Nilai Sikap Kerja

Np : Nilai Perilaku

(x)% : Nilai Persen yang diinputkan

II.1.3. Basis Data

Menurut Abdulloh (2018:103), Database atau basis data adalah kumpulan inFormasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh inFormasi.

Menurut Andry Andaru (2018), secara konsep basis data (database) merupakan kumpulan dari data yang saling berhubungan (relasi) antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu. Pada komputer, basis data disimpan dalam perangkat hardware penyimpanan, dan dengan software tertentu dimanipulasi untuk kepentingan atau kegunaan tertentu.

II.1.4. UML

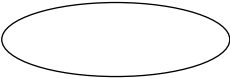
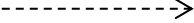
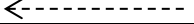
Menurut Rosa Dan Shalahuddin (2015:133) *UML (Unified Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek. Alat bantu yang digunakan 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. (Gellysa Urva, 2015 : 3). Simbol-simbol *Use Case Diagram* dapat dilihat pada Tabel II.16 berikut :

Tabel II. 2. Simbol Use Case Diagram

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,
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain.

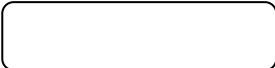
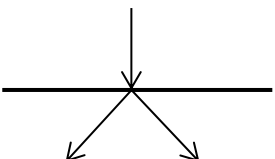
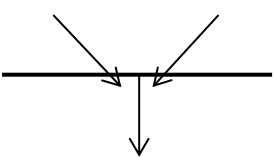
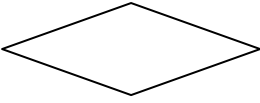
(Sumber :Urva ; 2015)

2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Adapun simbol-simbol *activity diagram* yang dapat di lihat pada Tabel II.17 berikut :

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 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.

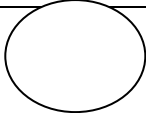
(Sumber :Urva ; 2015)

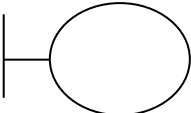
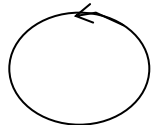
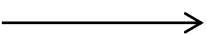
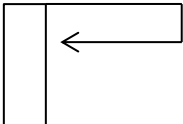
3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence Diagram* dapat dilihat pada Tabel II.4 berikut

:

Tabel II. 4. 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 <i>Formentry</i> dan <i>Formcetak</i> .
	<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> .

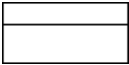
(Sumber :Urva ; 2015)

4. 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. *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. Adapun simbol-simbol *Multiplicity class diagram* yang dapat di lihat pada Tabel II.5 berikut:

Tabel II. 5. Tabel Class Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objeklainnya

Tabel II.6 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 :Urva ; 2015)

II.3. *Software*

II.3.1. *MySql*

Menurut (Risdiyansyah, 2017) “MySQL merupakan *database server* yang bersifat *multiuser* dan *multi-threaded*. SQL adalah bahasa *database standar* yang memudahkan penyimpanan, pengubahan dan akses inFormasi.

II.3.2 *PHP*

Menurut Hidayatullah dan Kawistara dalam bukunya yang berjudul Pemrograman Web (2017:223) mengemukakan bahwa “*PHP Hypertext Preprocessor* adalah suatu bahasa scripting khususnya digunakan untuk *web development*”. PHP memiliki sifat *server side scripting* sehingga untuk menjalankan *PHP* harus menggunakan web server.