



BAB II

TINJAUAN PUSTAKA

2.1 Penerapan

Badudu dan Zain (1996:1487), mengungkapkan “penerapan adalah hal, cara atau hasil. Dan menurut Ali (1995:1044), penerapan adalah mempraktekkan, memasang.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa penerapan merupakan sebuah perilaku untuk mencapai tujuan yang telah direncanakan baik dilakukan secara individu atau kelompok.

2.2 Metode *Profile Matching*

Menurut Junaidi dan Visella dalam jurnal yang berjudul Pemilihan Penerima Beasiswa Menggunakan Metode *Profile Matching*. “*Profile Matching* secara garis besar merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk karyawan menempati posisi tersebut”.

Menurut Fadillah dalam jurnal yang berjudul Implementasi Metode Profile Matching Terhadap Sistem Pendukung Keputusan Penerimaan Dana Zakat pada Badan Amil Zakat Pertamina (BAZMA). “Metode *Profile Matching* atau pencocokan profil adalah metode yang sering digunakan sebagai mekanisme dalam pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati”.

Menurut Sudrajat dalam jurnal yang berjudul Pemilihan Pegawai Berprestasi dengan Menggunakan Metode *Profile Matching*. “*Profile matching* merupakan suatu proses yang sangat penting dalam manajemen SDM dimana terlebih dahulu ditentukan kompetensi (kemampuan) yang diperlukan oleh suatu jabatan.

Kompetensi atau kemampuan tersebut haruslah dapat dipenuhi oleh pemegang atau calon pemegang jabatan”.

Dari ketiga pendapat di atas dapat disimpulkan bahwa Metode *Profile Matching* merupakan salah satu metode yang sederhana dalam sistem pendukung keputusan yang membandingkan gap antara nilai alternatif dan kriteria.

2.2.1 Tahapan Perhitungan *Profile Matching*

Ada beberapa hal yang diketahui tentang analisis GAP, salah satunya diantaranya adalah table nilai bobot GAP. Selain itu analisis GAP ini juga harus memahami konsep skala prioritas, karena di dalam pembuatan bobot dengan range 0-5 berdasarkan prioritas setiap kriteria. Berikut ini adalah bobot nilai GAP yaitu:

Tabel 2.1 Bobot nilai GAP

No.	Selisih (GAP)	Nilai Bobot	Keterangan
1	0	6	Tidak ada GAP (Kompetensi sesuai yang dibutuhkan)
2	1	5,5	Kompetensi individu kelebihan 1 tingkat
3	-1	5	Kompetensi individu kekurangan 1 tingkat
4	2	4,5	Kompetensi individu kelebihan 2 tingkat
5	-2	4	Kompetensi individu kekurangan 2 tingkat
6	3	3,5	Kompetensi individu kelebihan 3 tingkat
7	-3	3	Kompetensi individu kekurangan 3 tingkat
8	4	2,5	Kompetensi individu kelebihan 4 tingkat
9	-4	2	Kompetensi individu kekurangan 4 tingkat
10	5	1,5	Kompetensi individu kelebihan 5 tingkat

Adapun algoritma penyelesaian metode *Profile Matching* terdapat 2 konsep ialah.

Konsep I:

Langkah 1 : mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan sebagai tolak ukur penyelesaian masalah.



Langkah 2 : menghitung nilai GAP antara *profile* subjek dengan *profile* yang dibutuhkan.

Langkah 3 : menghitung nilai *Mapping* GAP yang bersumber dari analisis GAP.

$$\text{GAP} = \text{Profile alternatif} - \text{profile kriteria}$$

Langkah 4 : menghitung nilai akhir.

$$\text{Nilai akhir} = (\text{nama kriteria} * \text{nilai bobot})$$

Langkah 5 : melakukan perbandingan.

Konsep II:

Langkah 1 : mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan sebagai tolak ukur penyelesaian masalah dan menentukan prioritas dari masing-masing kriteria (profil kriteria).

Langkah 2 : menghitung nilai GAP dan *mapping* GAP.

Langkah 3 : menghitung nilai rata-rata *core factor* dan nilai rata *secondary factor*.

Adapun rumus menghitung nilai rata-rata *core factor* dan *secondary factor* ialah.

Rumus nilai rata-rata *core factor*

$$NCF = \frac{\sum NC}{\sum IC}$$

Keterangan :

NCF : nilai rata-rata *core factor*

NC : jumlah total nilai *core factor*

IC : jumlah total nilai *item core factor*

Rumus nilai rata-rata *secondary factor*

$$NSF = \frac{\sum NS}{\sum IS}$$

Dimana:

NSF : nilai rata-rata *secondary factor*

NS : jumlah total nilai *secondary factor*

IS : jumlah total nilai *item secondary factor*

Langkah 5 : Melakukan perbandingan.



2.3 Sistem Pendukung Keputusan

Turban, 2005:137 menurut Bonczek et.al (1980) dalam buku *Decision Support System and Intelligent System* mendefinisikan “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 (repositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan)”.

Menurut Yulianti Liza dkk dalam jurnal yang berjudul “Sistem Pendukung Keputusan Peserta KB Teladan di Bkkbn Bengkulu Menggunakan Pemrograman Visual Basic 6.0” Sistem Pendukung Keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Sedangkan menurut Wahyudi Yusuf dkk dalam jurnal yang berjudul “Sistem Pendukung Keputusan Pengangkatan Pegawai Negeri Sipil dalam Jabatan Struktural pada Badan Kepegawaian Daerah Provinsi Bengkulu” Sistem pendukung keputusan adalah suatu sistem informasi ineraktif yang menyediakan informasi, permodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan terstruktur dan situasi yang tidak terstruktur.

Dari ketiga pendapat di atas dapat disimpulkan bahwa sistem pendukung keputusan adalah sistem informasi yang berfungsi untuk membantu seseorang atau manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang ada untuk menghasilkan berbagai alternatif.

Karakteristik dari sistem pendukung keputusan yaitu:

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.



2. Adanya *interface* mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan keputusan.
5. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
6. Memiliki dua komponen utama yaitu data dan model.

2.3.1 Komponen Sistem Pendukung Keputusan

Menurut Latif komponen-komponen sistem pendukung keputusan terdiri dari:

1. *Data management*. Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management Systems (DBMS)*.
2. *Model Management*. Melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan kesistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.
3. *Communication (dialog subsystem)*. *User* dapat berkomunikasi dan memberikan perintah pada DSS melalui sub sistem ini. Ini berarti menyediakan antarmuka.
4. *Knowledge Management*. Subsistem opsional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.3.2 Proses Pengambilan Keputusan

Menurut Simon ada tiga fase dalam proses pengambilan keputusan diantaranya sebagai berikut:

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendekteksian dari ruang lingkup problematika secara 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 menguji kelayakan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplmentasikan dalam proses pengambilan keputusan.

2.4 **Fakir Miskin**

Menurut Undang-Undang Republik Indonesia nomor 13 tahun 2011 pasal 1, “Fakir miskin adalah orang yang sama sekali tidak mempunyai sumber mata pencaharian dan/atau mempunyai sumber mata pencaharian tetapi tidak mempunyai kemampuan memenuhi kebutuhan dasar yang layak bagi kehidupan dirinya dan/atau keluarganya”.

Menurut Musthafa Husni Assiba’I “Fakir miskin adalah orang yang sama sekali tidak mempunyai sumber mata pencarian dan atau mempunyai sumber mata pencarian tetapi tidak mempunyai kemampuan memenuhi kebutuhan dasar yang layak bagi kehidupan dirinya dan atau keluarganya.

2.5 **PHP**

Menurut Gunawan (2010:12), “PHP merupakan bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi *web* yang dinamis dengan cepat dan mudah”.

Menurut Winarno, dkk. (2014:49), “PHP atau PHP *Hypertext Preprocessor* adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode PHP dari kode *web* dengan



ekstensi .php, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client (browser)*”.

Dari pengertian di atas dapat disimpulkan bahwa PHP merupakan bahasa pemrograman untuk membuat *website* yang dinamis.

2.6 MySql

Menurut Mundzir (2018:217) dalam buku *Pemrograman Web Seri PHP* mengatakan bahwa “MySQL adalah sistem manajemen *database* SQL yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database* MySQL mampu mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system (DBMS)*”.

Sedangkan menurut Winarno (2014:102), “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan”.

Dari pengertian di atas dapat disimpulkan bahwa MySQL merupakan sebuah tempat untuk menyimpan data yang saling berhubungan dan sifatnya *open source*.

2.7 XAMPP

Menurut Enterprise (2017:93), “XAMPP merupakan *tool* yang digunakan untuk membantu kita bekerja menggunakan *MySQL Server*, namun dengan tampilan antarmuka grafis yang lebih ramah untuk siapa pun, terlebih bagi kalangan pemula”.

Menurut Gunawan (2010:17), “XAMPP adalah aplikasi *web server* instan yang dibutuhkan untuk membangun aplikasi berbasis *web*. Fungsi XAMPP adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl”.

Dari pengertian di atas dapat disimpulkan bahwa XAMPP merupakan alat untuk membantu dalam menggunakan *MySql Server* dan penterjemah bahasa yang ditulis dengan bahasa pemrograman.



2.8 Unified Modelling Language (UML)

Menurut Rosa (2018:137), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

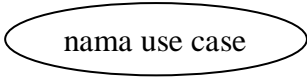
Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Sehingga penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan untuk metodologi berorientasi objek.

2.9 Use Case Diagram

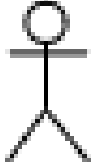

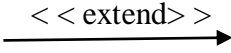
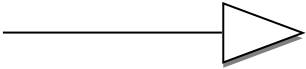
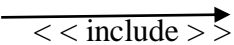
Menurut Rosa (2018:155), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.2 Simbol *Use Case* Diagram

No	Simbol	Deskripsi
1		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i>







Lanjutan Tabel 2.2 Simbol Use Case Diagram

2	Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal fase nama aktor.
3	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4	Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i>
5	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / <i>include</i> / <i>uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2.10 Activity Diagram

Menurut Rosa (2018:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.3 Simbol *Activity Diagram*

No	Simbol	Deskripsi
1	Start / status awal (<i>Intial State</i>) 	<i>Start</i> atau <i>intial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup
2	End / status akhir (<i>final state</i>) 	<i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem.
3	Event 	<i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin.
4	State 	Sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu yang memicu perubahan tersebut.

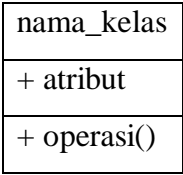
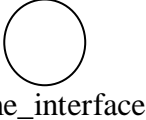


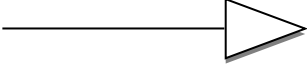


2.11 Class Diagram

Menurut Rosa (2018:143), “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan



dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi”.

Tabel 2.4 Simbol *Class Diagram*

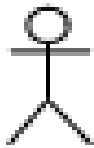
No	Simbol	Deskripsi
1	<p>Kelas</p> 	Kelas pada struktur sistem
2	<p>Antarmuka / <i>interface</i></p> 	sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
3	<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang saat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	<p>Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6	<p>Kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	<p>Agregasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)



2.12 Sequence Diagram

Menurut Rosa (2018:165), “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu dan juga dibutuhkan untuk melihat skenario yang ada pada use case.

Tabel 2.5 Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	<p>Aktor</p>  <p>atau</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-left: 20px;">Nama_aktor</div>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>