

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penerimaan Karyawan**

Penerimaan karyawan merupakan kenyataan bahwa dalam suatu organisasi selalu terbuka kemungkinan untuk terjadinya berbagai lowongan dengan aneka ragam penyebabnya. Lowongan bisa terjadi karena berbagai alasan misalnya, karena perluasan kegiatan organisasi sehingga tercipta pekerjaan-pekerjaan dan kegiatan baru yang lain, ada pekerja yang berhenti atau telah mencapai usia pensiun bahkan bisa pula terjadi karena ada pekerja yang meninggal dunia.

Apapun alasan terjadinya lowongan dalam suatu organisasi, yang jelas ialah bahwa lowongan itu harus diisi, bahkan tidak mustahil ada lowongan yang harus diisi dengan segera. Dengan adanya lowongan pada perusahaan maka akan ada penerimaan karyawan baru pada perusahaan tersebut. Proses penerimaan karyawan merupakan cara memperoleh calon tenaga kerja baru untuk mengisi jabatan kosong di dalam perusahaan, proses penerimaan karyawan dilakukan dengan rekrutmen dan seleksi. Hal ini dilakukan guna untuk memperoleh tenaga kerja baru yang produktif dan sesuai dengan kebutuhan perusahaan.

##### **2.1.1 Rekrutmen**

Salah satu kegiatan penting dari kegiatan sumber daya manusia adalah proses rekrutmen yaitu menarik tenaga kerja yang memiliki kinerja yang baik. Hal ini dilakukan oleh organisasi kecil, menengah, dan skala besar. Sebelum organisasi menarik tenaga kerja, organisasi harus merekrut dan menempatkan individu yang memiliki keahlian sesuai yang dibutuhkan organisasi.

Menurut Rachmawaty (2008: 84), rekrutmen merupakan proses mencari, menemukan, dan menarik para calon karyawan untuk dipekerjakan dalam dan oleh organisasi.

Rivai (2004: 160-161), rekrutmen adalah serangkaian kegiatan yang dimulai ketika sebuah perusahaan atau organisasi memerlukan tenaga kerja dan membuka lowongan sampai mendapatkan calon karyawan yang diinginkan/*qualified* sesuai dengan jabatan atau lowongan yang ada.

Dari pengertian di atas disimpulkan bahwa rekrutmen adalah proses mendapatkan sejumlah calon tenaga kerja untuk jabatan atau pekerjaan tertentu dalam suatu organisasi atau perusahaan.

### **2.1.2 Prinsip-prinsip Rekrutmen**

Menurut Rivai (2004:161) prinsip-prinsip rekrutmen terdiri dari:

1. Mutu karyawan yang akan direkrut harus sesuai dengan kebutuhan yang diperlukan untuk mendapatkan mutu yang sesuai. Untuk itu sebelumnya perlu dibuat:
  - a. Analisis Pekerjaan
  - b. Deskripsi Pekerjaan, dan
  - c. Spesifikasi Pekerjaan
2. Jumlah karyawan yang diperlukan harus sesuai dengan job yang tersedia untuk mendapatkan hal tersebut perlu dilakukan:
  - a. Peramalan kebutuhan tenaga kerja, dan
  - b. Analisis terhadap kebutuhan tenaga kerja
3. Biaya yang diperlukan diminimalkan
4. Perencanaan dan keputusan-keputusan strategis tentang perekrutan.
5. *Fleksibility*
6. Pertimbangan-pertimbangan hukum

### 2.1.3 Tujuan Rekrutmen

Menurut Rivai (2004: 161) Tujuan dilakukannya rekrutmen adalah untuk menerima pelamar sebanyak-banyaknya sesuai dengan kualifikasi kebutuhan perusahaan dari berbagai sumber, sehingga memungkinkan akan terjaring calon karyawan dengan kualitas tertinggi dari yang terbaik.

### 2.1.4 Sumber-sumber Rekrutmen

Unsur lain dalam strategi rekrutmen dalam suatu organisasi adalah keputusan di mana mencari para calon pelamar. Bila suatu organisasi atau perusahaan akhirnya memutuskan untuk melakukan rekrutmen, tersedia dua pilihan sumber rekrutmen.

Menurut Rivai (2004, 161-166) Perekrutan tenaga kerja dapat dilakukan melalui dua cara, yaitu internal dan eksternal:

#### 1. Internal

Perekrutan dari sumber internal perusahaan, yakni SDM yang ditarik (diterima) adalah berasal dari perusahaan/lembaga itu sendiri. Dengan cara ini perusahaan mencari karyawan yang ada di dalam lingkungan sendiri untuk ditempatkan pada posisi tertentu.

#### 2. Eksternal

Dari sumber eksternal perusahaan dapat memperoleh melalui:

- a. *Walk-ins* yaitu pelamar datang dan melamar langsung ke perusahaan
- b. Rekomendasi yaitu penyampaian lowongan pekerjaan dilakukan oleh karyawan dalam perusahaan kepada teman, anggota keluarga atau karyawan perusahaan lain.
- c. Pengiklanan yaitu menyebarkan secara luas lowongan pekerjaan melalui majalah, surat kabar, televisi radio dan media lainnya.
- d. Agen-agen keamanan tenaga kerja negara
- e. Agen-agen penempatan tenaga kerja
- f. Lembaga pendidikan dan pelatihan yang menggunakan tenaga kerja khusus siap kerja
- g. Departemen tenaga kerja yang menampung SDM pencari kerja dan

menyalurkannya

- h. Tenaga-tenaga professional perusahaan. Dilakukan untuk mencari pelamar professional bagi jabatan eksklusif
- i. Organisasi-organisasi profesi/keahlian.
- j. Asosiasi-asosiasi pekerja.
- k. Organisasi militer. Banyak tenaga kerja ahli professional yang habis masa tugasnya
- l. Program pelatihan kerja yang dilakukan oleh pihak swasta
- m. Pekerja-pekerja sewaan. Tenaga kerja dibayar perhari atau perjam.
- n. Open house. Sistem penarikan yang relative baru dengan mengundang orang sekitar perusahaan untuk mengunjungi dan memperoleh penjelasan mengenai perusahaan.

#### **2.1.5 Seleksi**

Menurut Rachmawati (2008: 99) seleksi adalah serangkaian kegiatan yang digunakan untuk memutuskan apakah pelamar diterima atau ditolak.

Hal yang sama juga dikemukakan oleh Rivai (2004: 170) seleksi adalah kegiatan dalam manajemen SDM yang dilakukan setelah proses rekrutmen selesai dilaksanakan.

Dari pengertian diatas dapat ditarik kesimpulan bahwa seleksi merupakan proses yang dimulai ketika calon karyawan melamar kerja dan diakhiri dengan keputusan bahwa calon karyawan diterima atau di tolak.

#### **2.1.6 Tujuan Seleksi**

Menurut Rachmawati ( 2008: 100 ) Seleksi bertujuan memperoleh tenaga kerja yang diinginkan. Seleksi merupakan proses dua arah di mana organisasi menawarkan posisi kerja dengan kompensasi yang layak, sedangkan calon pelamar mengevaluasi organisasi dan daya tarik posisi serta imbalan yang ditawarkan organisasi. Dengan melakukan seleksi diharapkan dapat memperoleh karyawan yang memenuhi tuntutan organisasi

## 2.2 Teori Pendukung

### 2.2.1 Analytic Hierarchy Process

Metode *Analytic Hierarchy Process* (AHP) dikembangkan oleh Prof. Thomas Lorie Saaty dari Wharston Business school untuk mencari ranking atau urutan prioritas dari berbagai alternatif dalam pemecahan suatu permasalahan. Dalam kehidupan sehari-hari, seseorang senantiasa dihadapkan untuk melakukan pilihan dari berbagai alternatif. Disini diperlukan penentuan prioritas dan uji konsistensi terhadap pilihan-pilihan yang telah dilakukan. Dalam situasi yang kompleks, pengambilan keputusan tidak dipengaruhi oleh satu faktor saja melainkan multifaktor dan mencakup berbagai jenjang maupun kepentingan.

Pada dasarnya AHP adalah suatu teori umum tentang pengukuran yang digunakan untuk menemukan skala rasio baik dari perbandingan berpasangan yang diskrit maupun kontinu. Perbandingan-perbandingan ini dapat diambil dari ukuran aktual atau skala dasar yang mencerminkan kekuatan perasaan dan preferensi relatif. AHP memiliki perhatian khusus tentang penyimpangan dari konsistensi, pengukuran dan ketergantungan di dalam dan di luar kelompok elemen strukturalnya.

*Analytic Hierarchy Process* (AHP) mempunyai landasan aksiomatik yang terdiri dari:

1. *Resiprocal Comparison*, yang mengandung arti bahwa matriks perbandingan berpasangan yang terbentuk harus bersifat berkebalikan. Misalnya, jika A adalah  $k$  kali lebih penting daripada B maka B adalah  $1/k$  kali lebih penting dari A.
2. *Homogeneity*, yang mengandung arti kesamaan dalam melakukan perbandingan. Misalnya, tidak dimungkinkan membandingkan jeruk dengan bola tenis dalam hal rasa, akan tetapi lebih relevan jika membandingkan dalam hal berat.
3. *Dependence*, yang berarti setiap jenjang (level) mempunyai kaitan (*complete hierarchy*) walaupun mungkin saja terjadi hubungan yang tidak sempurna (*incomplete hierarchy*).

4. *Expectation*, yang artinya menonjolkan penilaian yang bersifat ekspektasi dan preferensi dari pengambilan keputusan. Penilaian dapat merupakan data kuantitatif maupun yang bersifat kualitatif.

Dalam menyelesaikan persoalan dengan metode *Analytic Hierarchy Process* (AHP) ada beberapa prinsip dasar yang harus dipahami antara lain:

1. *Decomposition*;
2. *Comparative judgment*;
3. *Synthesis of Priority*;
4. *Logical Consistency*.

#### 1. *Decomposition*

Pengertian *decomposition* adalah memecahkan atau membagi problema yang utuh menjadi unsur-unsurnya ke dalam bentuk hierarki proses pengambilan keputusan, dimana setiap unsur atau elemen saling berhubungan. Untuk mendapatkan hasil yang akurat, pemecahan dilakukan terhadap unsur-unsur sampai tidak mungkin dilakukan pemecahan lebih lanjut, sehingga didapatkan beberapa tingkatan dari persoalan yang hendak dipecahkan. Struktur hirarki keputusan tersebut dapat dikategorikan sebagai *complete* dan *incomplete*. Suatu hirarki keputusan disebut *complete* jika semua elemen pada suatu tingkat memiliki hubungan terhadap semua elemen yang ada pada tingkat berikutnya, sementara hirarki keputusan *incomplete* kebalikan dari hirarki yang *complete*.

#### 2. *Comparative judgment*

*Comparative Judgment* dilakukan dengan membuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkatan di atasnya. Penilaian ini merupakan inti dari AHP karena akan berpengaruh terhadap urutan prioritas dari elemen-elemennya. Hasil dari penilaian ini lebih mudah disajikan dalam bentuk matriks *pairwise comparisons* yaitu matriks perbandingan berpasangan memuat tingkat preferensi beberapa alternatif untuk tiap kriteria. Skala

preferensi yang digunakan yaitu skala 1 yang menunjukkan tingkat yang paling rendah (*equal importance*) sampai dengan skala 9 yang menunjukkan tingkatan yang paling tinggi (*erxtreme importance*).

### 3. *Synthesis of Priority*

*Synthesis of Priority* dilakukan dengan menggunakan *egine vector method* untuk mendapatkan bobot relatif bagi unsur-unsur pengambilan keputusan.

### 4. *Logical Consistency*

*Logical Consistency* merupakan karakteristik penting AHP. Hal ini dicapai dengan mengagregasikan seluruh *eigenvector* yang diperoleh dari berbagai tingkatan hirarki dan selanjutnya diperoleh suatu *vector composite* tertimbang yang menghasilkan urutan pengambilan keputusan.

## 2.2.2 Proses Penentuan Prioritas dengan Metode AHP

Tahapan-tahapan pengambilan keputusan dalam metode AHP pada dasarnya meliputi:

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan
2. Membuat struktur hirarki yang diawali dengan tujuan umum, dilanjutkan dengan kriteria-kriteria, sub kriteria dan alternatif-alternatif pilihan yang ingin di *ranking*
3. Membentuk matriks perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap masing-masing tujuan atau kriteria yang setingkat di atasnya. Perbandingan dilakukan berdasarkan pilihan atau "*judgment*" dari pembuat keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya
4. Menormalkan data yaitu dengan membagi nilai dari setiap elemen di dalam matriks

yang berpasangan dengan nilai total dari setiap kolom

5. Menghitung nilai *eigen vector* dan menguji konsistensinya, jika tidak konsisten pengambil data (preferensi) perlu diulangi. Nilai *eigen vector* yang dimaksud adalah nilai *eigen vector* maximum yang diperoleh dengan menggunakan matlab maupun manual

6. Mengulangi langkah 3,4, dan 5 untuk seluruh tingkat hirarki

7. Menghitung *eigen vector* dari setiap matriks perbandingan berpasangan. Nilai *eigen vector* merupakan bobot setiap elemen. Langkah ini mensintesis pilihan dan penentuan prioritas elemen-elemen pada tingkat hirarki terendah sampai pencapaian tujuan. Menguji konsistensi hirarki. Jika tidak memenuhi dengan  $CR < 0,100$  maka penilaian harus diulang kembali.

### 2.2.3 Penyusunan Prioritas

Menentukan susunan prioritas elemen adalah dengan menyusun perbandingan berpasangan yaitu membandingkan dalam bentuk berpasangan seluruh elemen untuk setiap sub hirarki. Perbandingan tersebut ditransformasikan dalam bentuk matriks. Contoh, terdapat  $n$  objek yang dinotasikan dengan  $(A_1, A_2, \dots, A_n)$  yang akan dinilai berdasarkan pada nilai tingkat kepentingannya antara lain  $A_i$  dan  $A_j$  dipresentasikan dalam matriks *Pairwise Comparison*.



	$A_1$	$A_2$	$\dots$	$A_n$
$A_1$	$a_{11}$	$a_{12}$	$\dots$	$a_{1n}$
$A_2$	$a_{21}$	$a_{22}$	$\dots$	$a_{2n}$
$A_n$	$a_{n1}$	$a_{n2}$	$\dots$	$a_{nn}$

**Tabel 2.1 Matriks Perbandingan Berpasangan**

Membuat matriks perbandingan berpasangan memerlukan besaran-besaran yang mampu mencerminkan perbedaan antara faktor satu dengan faktor lainnya. Untuk menilai perbandingan tingkat kepentingan satu elemen terhadap elemen lainnya digunakan skala 1 sampai 9. Pendekatan AHP menggunakan skala Saaty mulai dari bobot 1 sampai 9, seperti terlihat pada tabel berikut ini.

Tingkat Kepentingan	Definisi
1	Sama pentingnya dibanding yang lain
3	Moderat (cukup) pentingnya dibanding yang lain
5	Kuat pentingnya dibanding yang lain

7	Sangat kuat pentingnya dibanding yang lain
9	Ekstrem pentingnya dibanding yang lain
2,4,6,8	Nilai diantara dua nilai yang berdekatan

Resiprokal	Jika elemen $i$ memiliki salah satu angka diatas ketika dibandingkan elemen $j$ , maka $j$ memiliki kebalikannya ketika dibanding elemen $i$
------------	--

**Tabel 2.2 Skala Saaty**

Model AHP didasarkan pada *pairwise comparison matrix*, dimana elemen-elemen pada matriks tersebut merupakan *judgment* dari *decision maker*. Seorang *decision maker* akan memberikan penilaian, mempersepsikan, ataupun memperkirakan kemungkinan dari sesuatu hal/peristiwa yang dihadapi. Matriks tersebut terdapat pada setiap *level of hierarchy* dari suatu struktur model AHP yang membagi habis suatu persoalan.

Berikut ini contoh suatu *Pairwise Comparison Matrix* pada suatu level of

Hierarchy, yaitu.

	I	j	k
		1	8
i	1	—	—
		2	—
A= j	2	1	4
	1	1	1
		—	—
k	—	—	—
	8	4	—

Membacanya atau membandingkannya, dari kiri ke kanan.

Jika  $i$  dibandingkan dengan  $j$ , maka  $j$  *very strong importance* dari pada  $i$  dengan nilai *judgment* sebesar 4. Dengan demikian pada baris 1 kolom 2 diisi dengan kebalikan dari 4 yaitu 1/4 . Artinya,

$i$  dibanding  $j$  □  $j$  lebih penting dari  $i$

jika  $i$  dibandingkan dengan  $k$ , maka  $i$  *extreme importance* daripada  $k$  dengan nilai *judgment* sebesar 8. Jadi baris 1 kolom 3 diisi dengan 8, dan seterusnya.

### 2.2.4 Eigen value dan Eigenvector

Definisi. Jika  $A$  adalah matriks  $n \times n$  maka vektor tak nol  $x$  di dalam  $\mathbb{R}^n$  dinamakan *eigen vector* dari  $A$  jika  $Ax$  kelipatan skalar  $x$ , yakni :

$$Ax = \lambda x$$

Skalar  $\lambda$  dinamakan *eigenvalue* dari  $A$  dan  $x$  dikatakan *eigenvector* yang bersesuaian dengan  $\lambda$ . Untuk mencari *eigenvalue* dari matriks  $A$  yang berukuran  $n \times n$  maka dapat ditulis pada persamaan berikut :

$$Ax = \lambda x$$

atau secara  
ekivalen

$$(\lambda I - A)x = 0$$

Agar  $x$  menjadi *eigen value*, maka harus ada pemecahan tak nol dari persamaan ini. Akan tetapi, persamaan diatas akan mempunyai pemecahan tak nol jika dan hanya jika:

$$\det (\lambda I - A) = 0$$

Ini dinamakan persamaan karakteristik  $A$ , skalar  $\lambda$  yang memenuhi persamaan ini adalah eigen value dari  $A$ .

Bila diketahui bahwa nilai perbandingan elemen  $A_i$  terhadap elemen  $A_j$  adalah  $a_{ij}$ , maka secara teoritis matriks tersebut berciri positif berkebalikan, yakni  $a_{ij} = 1/a_{ji}$ . Bobot yang dicari dinyatakan dalam vektor  $w = (w_1, w_2, w_3, \dots, w_n)$ . Nilai  $w_n$  menyatakan bobot kriteria  $A_n$  terhadap keseluruhan set kriteria pada sub sistem tersebut.

Jika  $a_{ij}$  mewakili derajat kepentingan  $i$  terhadap faktor  $j$  dan  $a_{jk}$  menyatakan kepentingan dari faktor  $j$  terhadap faktor  $k$ , maka agar keputusan menjadi konsisten, kepentingan  $i$  terhadap faktor  $k$  harus sama dengan  $a_{ij} \cdot a_{jk}$  atau jika  $a_{ij} \cdot a_{jk} = a_{ik}$  untuk semua  $i, j, k$  maka matriks tersebut konsisten.

Untuk suatu matriks konsisten dengan faktor  $w$ , maka elemen  $a_{ij}$  dapat ditulis menjadi :

$$a_{ij} = \frac{w_i}{w_j} ; \quad i, j = 1, 2, 3, \dots, n \quad (1)$$

Akan diperoleh hubungan persamaan berikut:

$$a_{ij} \cdot w_j = 0 \quad \text{atau} \quad a_{ij} w_j = w_i \quad (2)$$

Jadi matriks konsisten adalah:

$$a_{ij} = \frac{w_i}{w_j} \cdot \frac{w_j}{w_k} = \frac{w_i}{w_k} \quad (3)$$

Seperti yang diuraikan diatas, maka untuk *pairwise comparison matrix* diuraikan seperti berikut ini:

$$a_{ji} = \frac{w_j}{w_i} \cdot \frac{w_i}{w_j} = \frac{w_j}{w_j} = 1 \quad (4)$$

Dari persamaan di atas dapat dilihat bahwa

$$\sum_{j=1}^n \frac{a_{ji} w_j}{w_i} = 1; \quad \forall i, j = 1, 2, 3, \dots, n \quad (5)$$

Dengan demikian untuk *pairwise comparison matrix* yang konsisten menjadi:

$$\sum_{j=1}^n \frac{a_{ij} w_j}{w_i} = 1; \quad \forall i, j = 1, 2, 3, \dots, n \quad (6)$$

$$\sum_{j=1}^n a_{ij} w_j = n w_i; \quad \forall i, j = 1, 2, 3, \dots, n \quad (7)$$

Persamaan di atas ekuivalen dengan bentuk persamaan matriks di bawah ini:

$$A.w = n.w \quad (8)$$

Dalam teori matriks, formulasi ini diekspresikan bahwa  $w$  adalah *eigenvector* dari matriks  $A$  dengan *eigen value*  $n$ . Perlu diketahui bahwa  $n$  merupakan dimensi matriks itu sendiri. Dalam bentuk persamaan matriks dapat ditulis sebagai berikut:

$$\frac{w_1}{w_2} = \frac{w_1}{w_1} = 1$$

$$\frac{w_2}{w_1}$$

$$\frac{W}{1} = \frac{w}{1}$$

$$\frac{w}{2} = \frac{w}{n} \cdot \frac{2}{n} = \frac{w}{n^2} \quad (9)$$

$$\frac{w}{2} = \frac{w}{2}$$

$$\frac{w_2}{n} = \frac{w}{n}$$

Pada prakteknya, tidak dapat dijamin bahwa :

$$a_{ij} = \frac{a_{ik}}{a_{jk}} \quad (10)$$

Salah satu faktor penyebabnya yaitu karena unsur manusia (*responden*) tidak selalu dapat konsisten mutlak (*absolte consistent*) dalam mengekspresikan preferensinya terhadap elemen-elemen yang dibandingkan. Dengan kata lain, *judgment* yang diberikan tidak untuk setiap elemen persoalan pada suatu *level hierarchy* dapat saja *inconsistent*.

### 2.2.5 Uji Konsistensi Indeks dan Rasio

Dalam teori matriks dapat diketahui kesalahan kecil pada koefisien akan menyebabkan penyimpangan kecil pada *eigenvalue*. Dengan mengkombinasikan apa yang telah diuraikan sebelumnya, jika diagonal utama dari matriks A bernilai satu dan jika A konsisten maka penyimpangan kecil dari  $a_{ij}$  akan tetap menunjukkan *eigenvalue* terbesar  $\lambda_{maks}$ , nilainya akan mendekati  $n$  dan *eigenvalue* sisanya akan mendekati nol. Penyimpangan dari konsistensi dinyatakan dengan indeks konsistensi dengan persamaan:

$$CI = \frac{(\lambda_{maks} - n)}{(n - 1)} \quad (11)$$

Di mana:  $CI$  = Rasio penyimpangan (deviasi) konsistensi (*consistency index*)

$\lambda_{maks}$  = *eigenvalue* maksimum

$n$  = ukuran matriks

Apabila  $CI$  bernilai nol, berarti matriks konsisten, batas ketidakkonsistensi (*inconsistency*) yang ditetapkan Saaty diukur dengan menggunakan Rasio Konsistensi ( $CR$ ), yakni perbandingan indeks konsistensi dengan nilai random indeks ( $RI$ ) yang diperlihatkan seperti tabel 2.3. Nilai ini bergantung pada ordo matriks  $n$ . Dengan demikian, Rasio Konsistensi dapat dirumuskan :

$$CR = \frac{CI}{RI} \quad (12)$$



Ukuran Matriks	1,2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0,00	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,58	1,59

**Tabel 2.3 Nilai Indeks Random (RI)**

### 2.3 Pengertian Mysql

Menurut Edi (2015:4), MySQL (My Structured Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL (database management sistem) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia dan tergolong yang bermanfaat untuk mengelola data dengan cara sangat fleksibel dan cepat. Itulah sebabnya istilah seperti tabel baris dan kolom di gunakan pada MySQL. Pada MySQL sebuah database mengandung satu atau sejumlah tabel.

Berdasarkan pengertian di atas, dapat disimpulkan MySQL adalah salah satu jenis database server yang termasuk jenis RDBMS. MySQL adalah sistem manajemen database SQL yang bersifat Open Source dan paling populer saat ini. Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multi-user, dan SQL database manajemen sistem (DBMS). Database ini di buat untuk keperluan sistem database yang cepat, handal dan mudah digunakan. Ulf Michael Widenius adalah penemu awal versi pertama MySQL yang kemudian pengembangan selanjutnya dilakukan oleh klinik MySQL AB. MySQL AB yang merupakan sebuah klinik komersial yang didirikan oleh para pengembang MySQL. Dalam mengakses sebuah database terdapat beberapa tahapan, yaitu koneksi ke database, query ke database, pemutusan koneksi dari database.

Kelebihan MySQL berikut ini beberapa kelebihan MySQL sebagai database server antara lain :

1. Source MySQL dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan database dapat dilakukan dengan mudah.
4. MySQL merupakan program yang multithreaded, sehingga dapat dipasang pada server yang memiliki multi CPU.
5. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dan lain sebagainya.
6. Bekerja pada berbagai platform (tersedia berbagai versi untuk berbagai sistem operasi).
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
8. Memiliki sistem keamanan yang cukup baik dengan verifikasi host.
9. Mendukung ODBC untuk sistem operasi Windows.
10. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

MySQL dan PHP merupakan sistem yang saling terintegrasi. Pembuatan database dengan menggunakan sintak PHP dapat dibuat. Sedangkan input yang dimasukkan melalui aplikasi web yang menggunakan script server side seperti PHP dapat langsung dimasukkan ke database MySQL yang ada di server dan tentunya web tersebut berada di sebuah web server.

Koneksi ke Database PHP menyediakan fungsi untuk melakukan koneksi ke database dengan sejumlah fungsi untuk pengaturan baik menghubungkan maupun

memutuskan koneksi dengan server database MySQL. Selain MySQL, PHP juga menyediakan beragam fungsi untuk melakukan hubungan dengan server database.

Biasanya fungsi koneksi ke server database menggunakan pola yang sama yaitu server database menggunakan pola yang sama yaitu server, port, user, password (tergantung model koneksi ke server database) dan menggunakan pola nama fungsi yang tidak jauh berbeda.

## **2.4 XAMPP**

XAMPP adalah perangkat lunak ( free software) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis

## **2.5 APACHE**

Apache merupakan server web yang dapat dijalankan di berbagai sistem operasi yang berbeda seperti Unix, BSD, Linux, Windows dan Novell Netware yang berfungsi untuk melayani fasilitas web dengan menggunakan protocol http.

Apache memiliki fungsi yang sama dengan fungsi web server pada umumnya, yaitu memperoleh berkas yang berisi permintaan (request) client melalui web browser, kemudian Apache akan memproses data tersebut dengan menghasilkan keluaran (output) yang diinginkan oleh client. Output didapat berdasarkan data yang tersimpan dalam database tersebut.

## 2.6 JAVA

Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Bahasa Pemrograman Java ini berorientasi objek (OOP-Object Oriented Programming), dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source. Dengan slogannya “*Write once, run anywhere*”. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose).

Paradigma OOP menyelesaikan masalah dengan merepresentasikan masalah ke model objek. Contoh Pemisalan Objek dalam OOP : Objek-objek dalam dunia nyata, mempunyai 2 karakteristik khusus : Status dan Perilaku. Contohnya, Laptop punya status (jumlah baterai, jumlah ram, sebuah layar) dan perilaku (mengeksekusi, menonton, mengerjakan).

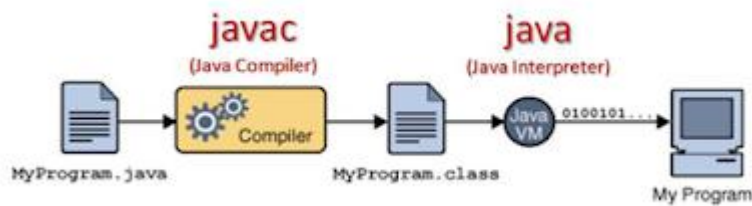
Bahasa yang berorientasi pada objek pun mempunyai karakteristik yang sama dengan objek-objek di dunia nyata. Yaitu status yang dalam bahasa pemrograman biasanya disimpan sebagai Variabel dan perilaku yang diimplementasikan sebagai Method.

### 2.6.1 Konsep Pemrograman Destop Java

Pemrograman adalah kegiatan menulis kode program yang akan dieksekusi oleh komputer. Komputer bekerja seperti switching dan hanya mengenali 0 dan 1. Manusia tidak (paham) berbicara dengan bahasa 0 dan 1.

Oleh karena itu diperlukan bahasa pemrograman yang dapat menjadi perantara percakapan antara komputer dan manusia. Bahasa pemrograman diubah ke dalam bahasa yang dipahami oleh komputer dengan menggunakan interpreter atau kompiler. Jadi, bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

### Compiler & Interpreter



**Compiler** : Mengkompilasi source code menjadi bentuk file yang bisa dieksekusi.

**Interpreter**: Mengkompilasi dan menjalankan source code secara langsung.

### 2.6.2 Sejarah Java

James Gosling, Mike Sheridan, dan Patrick Naughton memulai project bahasa java pada juni 1991. Bahasa ini mulanya disebut Oak dikarenakan adanya pohon Oak diluar jendela kantor Gosling. Kemudian diberi nama Green, dan pada akhirnya diberikan nama Java yang diambil dari daftar acak kumpulan kata-kata. Gosling mengimplementasi dari virtual machine dan gaya notasi bahasa C/C++. Sun Microsystems merilis Java 1.0 secara resmi kepada publik pada tahun 1995. Pada 8 mei 2007, Sun menyelesaikan prosesnya, untuk membuat semua kode Java tersedia dibawah GNU Public License.

## 2.7 UML (Unified Model Language)

UML (*Unified Model Language*) ada sebuah “bahasa yang telah menjadi standar dalam industry untuk visualisasi, merancang, dan mendokumentasikan system piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. UML dapat juga didefinisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek – aspek dari sebuah system. (Sugiarti, 2012). UML biasa digunakan untuk :

1. Menggambarkan batasan system dan fungsi – fungsi system secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur static sebuah system dalam bentuk *class diagrams*.
4. Membuat model behavior yang menggambarkan kebiasaan atau sifat sebuah system dengan *state transition digrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas fungsionalitas dengan *stereotypes*.


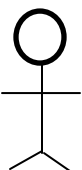

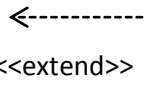
Terdapat berbagai diagram dalam merancang system menggunakan UML, diantaranya adalah:

### 2.7.1 Use case Diagram

*Use casediagram* merupakan pemodelan untuk menggambarkan kelakuan system yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan system yang akan dibuat. Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah system dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. (Sugiarti, 2012).

Simbol – simbol *Use case* :

*Tabel 2. 1 Simbol Use Case Diagram*

Simbol	Keterangan
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan system sebagai unit – unit yang saling bertukar pesan antar unit atau <i>actor</i>. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p><i>Actor</i></p> 	<p>Orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang tapi <i>actor</i> belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>
<p>asosiasi</p> 	<p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i></p>
<p><i>extend</i></p> 	<p>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 tersebut. Mirip dengan prinsip inheritance pada pemrograman berorientasi object, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambhkan, arah panah menunjuk pada <i>use case</i> yang dituju.</p>

<p><i>Include</i></p> <p>-----&gt;</p> <p>&lt;&lt;include&gt;&gt;</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>, include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p>
---	--

### 2.7.2 Class diagram

*Class diagram* menggambarkan struktur system dari segi pendefinisian *class* – *class* yang akan dibuat untuk membangun system. *Class* memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel – variabel yang dimiliki oleh suatu *class*.
2. Atribut mendeskripsikan property dengan sebaris teks didalam kotak *class* tersebut.
3. Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu *class*.


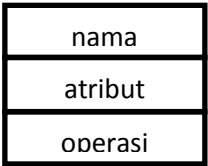

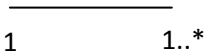

*Class diagram* mendeskripsikan jenis – jenis objek dalam system dan berbagai hubungan statis yang terdapat di antara mereka. *Class diagram* juga menunjukkan property dan operasi sebuah *class* dan batasan – batasan yang terdapat dalam hubungan – hubungan objek tersebut. *Class diagram* menggambarkan struktur dan deskripsi *class*, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain – lain. (Sugiarti, 2012). *Class diagram* memiliki tiga area pokok :

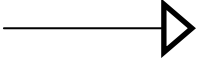
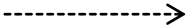
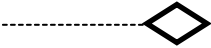
1. Nama
2. Atribut
3. Operasi



Simbol – simbol *class diagram* :

*Tabel 2. 2 Simbol Class Diagram*

Simbol	Deskripsi
<p><i>Package</i></p> 	<p>Package merupakan sebuah bungkusan dari satu atau lebih <i>class</i>.</p>
<p>Operasi</p> 	<p><i>Class</i> pada struktur system.</p>
<p>Antarmuka/interface</p> 	<p>Sama dengan konsep interface dalam pemrograman berorientasi objek.</p>
<p>Asosiasi</p> 	<p>Relasi antar <i>class</i> dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity</p>
<p>Asosiasi berarah</p> 	<p>Relasi antar <i>class</i> dengan makna <i>class</i> yang satu digunakan oleh</p>

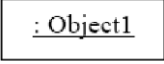



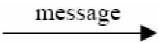
	<i>class</i> yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antar <i>class</i> dengan makna generalisasi – spesialisasi
Kebergantungan/ <i>dependency</i> 	Relasi antar <i>class</i> dengan makna kebergantungan antar <i>class</i>
Agregasi 	Relasi antar <i>class</i> dengan makna semua – bagian

### 2.7.3 Sequence diagram

*Sequence diagram* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki *class* yang diinstansiasi menjadi objek tersebut. Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan

interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak. (Sugiarti, 2012) Simbol – simbol *Sequence diagram* adalah sebagai berikut:


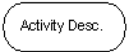



Tabel 2. 3 Simbol Sequence Diagram

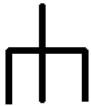


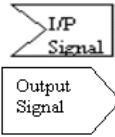
Simbol	Keterangan
<p><i>Object</i></p> 	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal.
<p><i>Actor</i></p> 	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
<p><i>Lifeline</i></p> 	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu.
<p><i>Activation</i></p> 	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> .
<p><i>Message</i></p> 	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>activation</i> .

### 2.7.4 Activity diagram

*Activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah system atau proses bisnis. *Activity diagram* menggambarkan berbagai alir aktivitas dalam system yang sedang dirancang, bagaimana masing – masing alir berawal, decision yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi. (Sugiarti, 2012). Simbol – simbol *activity diagram* :

Tabel 2. 4 Simbol Activity Diagram

Simbol	Keterangan
<p><i>Initial Activity</i></p> 	sebagai awal dari aktivitas modul sistem aplikasi.
<p><i>Activity</i></p> 	menunjukkan aktivitas yang dilakukan.
<p><i>Final Activity</i></p> 	menunjukkan akhir dari aktivitas.
<p><i>Decisions</i></p> 	menunjukkan aktivitas yang harus dipilih apakah pilihan pertama atau kedua.
<p><i>Fork</i></p> 	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
<p><i>Rake</i></p>	Menunjukkan adanya dekomposisi

	
	Tanda waktu
Flow Final 	Aliran akhir
Signal 	sebagai pengirim dan penerima pesan dari aktivitas yang terjadi. Sinyal terdiri dari sinyal penerima yang digambarkan dengan poligon terbuka dan sinyal pengirim dengan yang digambarkan dengan <i>convex</i> poligon.

## 2.8 Testing

Faktor pengujian software adalah teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Perangkat lunak dapat diuji dengan dua cara, yaitu :

### 1. *Black Box*

Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Selain itu, pengujian *Black Box* juga merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Dalam Pengujian perangkat lunak (*Software*) *black box* berfokus kepada persyaratan fungsional perangkat lunak. Pengujian *black box* memungkinkan perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternative dari teknik *white box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white box*.

## **2. White Box**

*White box* testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur control dari desain secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box* testing merupakan petunjuk untuk mendapatkan program yang benar secara 100 %.

## **2.9 PHPMyadmin**

Menurut Sukarno (2006:223) mengemukakan bahwa “phpMyadmin merupakan halaman yang terdapat pada web server”. Fungsi dari halaman ini adalah sebagai pengendalian database MySQL. Menggunakan web server MySQL mempunyai tampilan client yang mempermudah anda dalam mengakses database dengan kata sandi untuk mengijinkan proses yang boleh anda lakukan. Untuk masuk kedalam database disediakan user default, yaitu root dengan password phpMyadmin.