

BAB II

LANDASAN TEORI

2.1 Sistem pakar

Menurut (Martin dan Oxman,1988) Sistem Pakar adalah sistem yang berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya dapat diselesaikan oleh seorang pakar dalam bidang tertentu. Sistem Pakar (*Expert System*) dibuat bertujuan untuk dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya bisa diselesaikan oleh para ahli. Pembuatan sistem pakar bukan untuk menggantikan ahli itu sendiri melainkan dapat digunakan sebagai asisten yang sangat berpengalaman (Kusumadewi, 2003).

Adapun manfaat atau keuntungan yang dapat diperoleh dengan mengembangkan sistem pakar, antara lain (Kusumadewi, 2003) :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Dapat melakukan proses berulang secara otomatis
3. Menyimpan pengetahuan dan keahlian dari para pakar
4. Meningkatkan output dan produktivitas
5. Meningkatkan kualitas
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama termasuk yang keahlian langka)
7. Memiliki kemampuan untuk mengakses pengetahuan
8. Memiliki reabilitas
9. Meningkatkan kapabilitas sistem

10. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidak pastian.
11. Menghemat waktu dalam mengambil keputusan.

Selain banyak manfaat yang diperoleh, ada juga kelemahan pengembangan sistem pakar, yaitu (Kusumadewi, 2003) :

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem pakar tidak 100% bernilai benar.

2.1.1 Ciri – ciri Sistem pakar

Adapun ciri – ciri sistem pakar yang baik yaitu :

1. Memiliki fasilitas informasi yang handal
2. Mudah dimodifikasi
3. Dapat digunakan dalam berbagai jenis komputer
4. Memiliki kemampuan untuk belajar beradaptasi.

2.1.2 Struktur sistem pakar

Sistem pakar terdiri dari 2 bagian pokok, yaitu : lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consulation environment*) (Kusumadewi, 2003). Lingkungan pengembangan digunakan sebagai pembangunan sistem pakar baik dari segi pembangunan komponen maupun basis

pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi.

2.2 Pengertian Metode *Fuzzy*

Fuzzy logic atau sistem *fuzzy* merupakan suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output untuk sistem yang rumit. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. Namun berapa besar kebenaran dan kesalahan tergantung pada bobot nilai keanggotaan yang dimilikinya (Kusumadewi S, Purnomo H, 2004).

Beberapa alasan digunakan *fuzzy logic* antara lain :

1. Konsep *fuzzy logic* mudah dimengerti, karena di dalam logika *fuzzy* terdapat konsep matematis sederhana dan mudah dimengerti yang mendasari penalaran fuzzy
2. *Fuzzy logic* sangat fleksibel
3. *Fuzzy logic* memiliki toleransi terhadap data yang tidak tepat.
4. *Fuzzy logic* mampu memodelkan fungsi – fungsi nonlinier yang sangat kompleks
5. *Fuzzy logic* didasarkan pada bahasa alami
6. *Fuzzy logic* dapat membangun dan mengaplikasikan pengalaman – pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan (Kusumadewi, 2003).

Ada beberapa hal menjadi lingkup dari sistem *fuzzy*, yaitu :

1. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*.

Contoh : umur, temperatur, permintaan.

2. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

Contoh : Variabel jarak, terbagi menjadi 3 himpunan *fuzzy*, yaitu : DEKAT, SEDANG dan JAUH.

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan atau sebaliknya. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh: Semesta pembicaraan untuk variabel umur: $(0+\infty)$

4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif. Contoh: Muda= $[0-45]$, Tua = $[45-\infty]$.

2.2.1 Fungsi keanggotaan

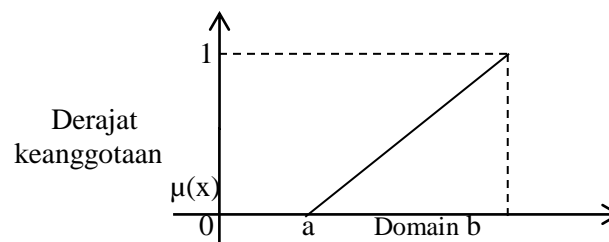
Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik – titik input ke dalam nilai keanggotaan yang memiliki interval 0 sampai 1. (Falopi T, 2011)

Ada beberapa fungsi yang dapat digunakan, yaitu :

1. Representasi linier :

Pada representasi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai garis lurus. Ada 2 keadaan himpunan *fuzzy* yang linier.

Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi. (Kusumadewi S, Purnomo H, 2004)

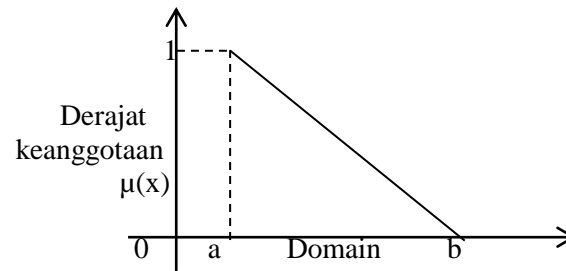


Gambar 2.1 Representasi Linier Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)} ; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (1)$$

Kedua, Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



Gambar 2.2 Representasi Linier Turun

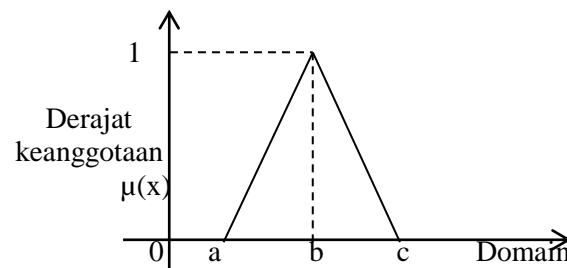
Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 1 & x \leq a \\ \frac{(b-x)}{(b-a)}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2)$$

2. Representasi kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linier*) seperti terlihat pada gambar 2.4 dibawah ini.

keanggotaan lebih rendah.



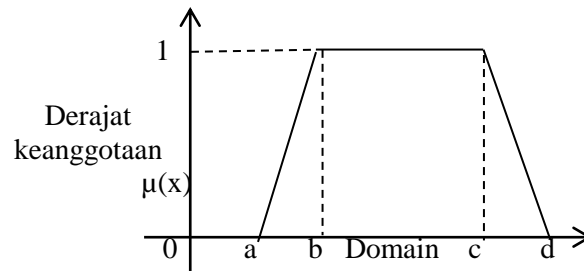
Gambar 2.3 Representasi Kurva Segitiga

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{b-a}; & a \leq x \leq b \\ \frac{b-x}{c-b}; & b \leq x \leq c \end{cases} \quad (3)$$

3. Representasi kurva trapezium

Kurva trapezium pada dasarnya seperti bentuk segitiga, hanya saja pada rentang tertentu ada beberapa titik yang memiliki nilai keanggotaan 1.(Kusumadewi S, Purnomo H, 2004)



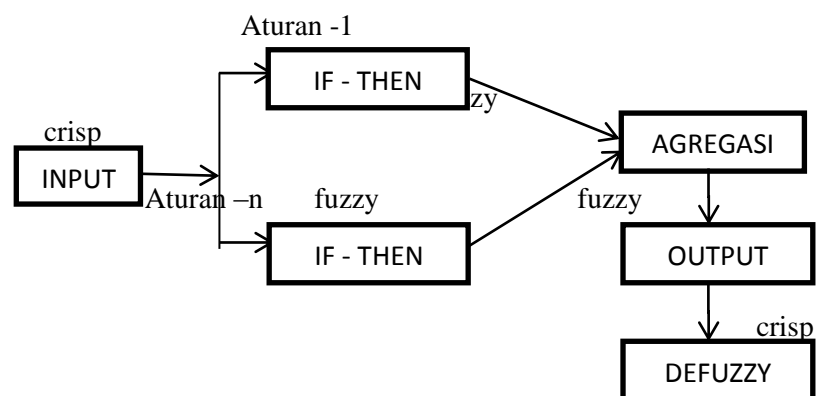
Gambar 2.4 Representasi Kurva Trapesium

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } \geq d \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & x \geq d \end{cases} \quad (4)$$

2.2.2 Metode Fuzzy Inferensi Sistem Tsukamoto

Inferensi adalah proses penggabungan banyak aturan berdasarkan data yang tersedia. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Menurut (Sri Kusumadewi, 2003) sistem inferensi *fuzzy* merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* yang berbentuk IF-THEN, dan penalaran *fuzzy*.



Gambar 2.5 Diagram Blok Sistem Inferensi Fuzzy

Sistem inferensi *fuzzy* menerima input *crisp*. Input ini kemudian dikirim ke basis pengetahuan yang berisi *n* aturan *fuzzy*

dalam bentuk IF-THEN. *Fire strength* (nilai keanggotaan anteseden atau α) akan dicari pada setiap aturan. Apabila aturan lebih dari satu, maka akan dilakukan agregasi semua aturan. Selanjutnya pada hasil agregasi akan dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai output sistem. Salah satu metode FIS yang dapat digunakan untuk pengambilan keputusan adalah metode FIS *Tsukamoto*.

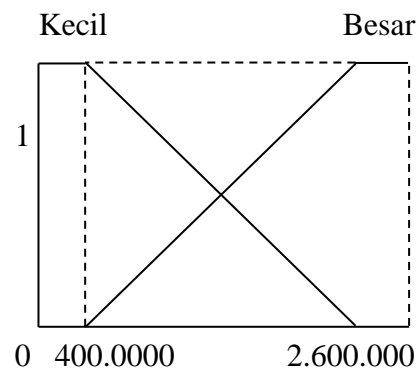
Contoh kasus metode *tsukamoto* :

Hitung jumlah penanaman pohon jati jika diketahui

Dana = 1400000 dan Jumlah jati per $m^3 = 7,485$

Tahap Ke-1: Fuzzifikasi

1. Dana Kecil dan Besar



Gambar 2.6 Fungsi Keanggotaan Variabel Dana

$$\mu[\text{kecil}] = \begin{cases} 1; & x \leq 400000 \\ \frac{2600000-x}{2200000} & 400000 < x \leq 2600000 \\ 0; & x \geq 2600000 \end{cases}$$

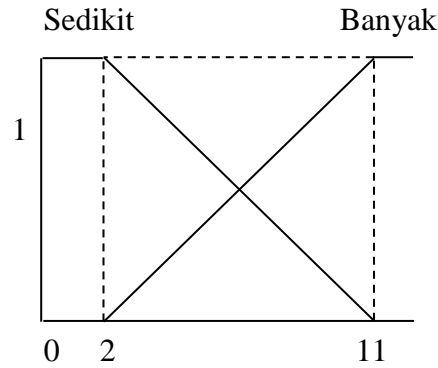
$$\mu[\text{besar}] = \begin{cases} 0; & x \leq 400000 \\ \frac{x-400000}{2200000} & 400000 < x \leq 2600000 \\ 1; & x \geq 2600000 \end{cases}$$

Derajat keanggotaan untuk dana 1400000 adalah:

$$\mu \text{ KECIL } [1400000] = (2600000-1400000)/2200000 = 0,5455$$

$$\mu \text{ BESAR } [1400000] = (1400000 - 400000) / 2200000 = 0,45455$$

2. Jati { Sedikit, Banyak }



Gambar 2.7 Fungsi Keanggotaan Variabel jati

$$\mu[\text{sedikit}] = \begin{cases} 1; & x \leq 2 \\ \frac{11-x}{9} & ; \quad 2 < x \leq 11 \\ 0; & x \geq 11 \end{cases}$$

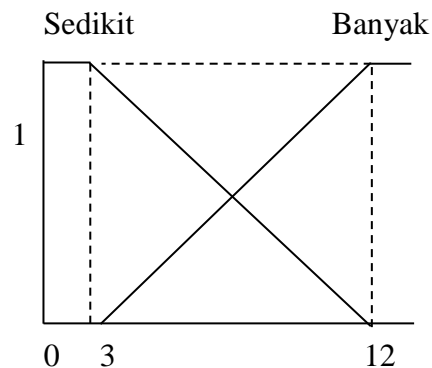
$$\mu[\text{banyak}] = \begin{cases} 0; & x \leq 2 \\ \frac{x-2}{9} & ; \quad 2 < x \leq 11 \\ 1; & x \geq 11 \end{cases}$$

Derajat keanggotaan untuk Jati 7,485 adalah:

$$\mu \text{ SEDIKIT } [7,485] = (11 - 7,485) / 9 = 0,3905$$

$$\mu \text{ BANYAK } [7,485] = (7,485 - 2) / 9 = 0,6094$$

3. Tanam jati { Sedikit, Banyak }



Gambar 2.8 Fungsi keanggotaan variabel tanaman

$$\mu[\text{sedikit}] = \begin{cases} 1; & x \leq 3 \\ \frac{12-x}{9} & 3 < x \leq 12 \\ 0; & x \geq 12 \end{cases}$$

$$\mu[\text{banyak}] = \begin{cases} 0; & x \leq 3 \\ \frac{x-3}{9} & 3 < x \leq 12 \\ 1; & x \geq 12 \end{cases}$$

Tahap ke-2: Pembentukan Rule :

[R1] IF dana BESAR And jati BANYAK THEN tanam jati

BANYAK

[R2] IF dana BESAR And jati SEDIKIT THEN tanam jati

SEDIKIT

[R3] IF dana KECIL And jati BANYAK THEN tanam jati

BANYAK

[R4] IF dana KECIL And jati SEDIKIT THEN tanam jati

SEDIKIT

Tahap ke-3: Mesin Inferensi

Pada mesin inferensi, penerapan fungsi MIN untuk setiap aturan.

$$[R1] \alpha - \text{predikat1} = \min(0,45455; 0,6094) = 0,45455$$

Himpunan BANYAK pada variabel Tanam jati

$$(z - 3) / 9 = 0,45455 \rightarrow z1 = 7,091$$

$$[R2] \alpha - \text{predikat2} = \min(0,45455; 0,3906) = 0,3906$$

Himpunan SEDIKIT pada variabel Tanam jati

$$(12 - z) / 9 = 0,3906 \rightarrow z2 = 8,4846$$

$$[R3] \alpha - \text{predikat3} = \min(0,5455; 0,6094) = 0,5455$$

Himpunan BANYAK pada variabel Tanam jati

$$(z - 3) / 9 = 0,5455 \rightarrow z3 = 7,9095$$

$$[R4] \alpha - \text{predikat4} = \min (0,5455; 0,3906) = 0,3906$$

Himpunan SEDIKIT pada variabel Tanam jati

$$(12 - z) / 9 = 0,3906 \rightarrow z4 = 8,4846$$

Tahap Ke-4: Defuzzifikasi

nilai tegas (defuzzyfikasi) z dapat dicari menggunakan rata-rata terbobot, yaitu:

$$z = \frac{\alpha_{\text{pred1}} * z1 + \alpha_{\text{pred2}} * z2 + \alpha_{\text{pred3}} * z3 + \alpha_{\text{pred4}} * z4}{\alpha_{\text{pred1}} + \alpha_{\text{pred2}} + \alpha_{\text{pred3}} + \alpha_{\text{pred4}}}$$

$$z = \frac{0,45455 * 7,091 + 0,3906 * 8,4846 + 0,5455 * 7,9095 + 0,3906 * 8,4846}{0,45455 + 0,3906 + 0,5455 + 0,3906}$$

$$Z = 7.9529 \text{ m}^3$$

Diketahui rata – rata pohon jati sebesar 0,5m³.

Jadi untuk penanaman jati adalah $7.9529/0,5 = 15,9058$

Jumlah penanaman pohon jati setelah dibulatkan adalah 16

pohon jati (RF Muhammad 2013)

2.2.3 Rule IF – THEN

Rule adalah struktur *knowledge* yang menghubungkan beberapa informasi yang sudah diketahui ke informasi lain sehingga dapat disimpulkan. Sebuah rule adalah sebuah bentuk *knowledge* yang *procedural*. Dengan demikian yang dimaksud dengan sistem pakar berbasis *rule* adalah sebuah program komputer untuk memproses masalah dari informasi spesifik yang terdapat dalam memori aktif dengan sebuah set dari *rule* dalam

knowledge base, dengan menggunakan *inference engine* untuk menghasilkan informasi baru.

Struktur rule secara logika menghubungkan satu atau lebih *antaseden* (juga disebut premis) yang terletak dalam bagian IF dengan satu lebih konsekuensi (juga disebut konklusi) yang terletak dalam bagian THEN. Secara umum, sebuah rule dapat mempunyai premis jamak dihubungkan dengan pernyataan AND (konjungsi) pernyataan OR (disjungsi) atau kombinasi dari keduanya.

Dalam sistem pakar berbasis *rule domain knowledge* ditampung dalam sebuah set dari *rules* dan dimasukkan dalam basis sistem pengetahuan. Sistem selama berada dalam memori aktif untuk memecahkan masalah.

Sistem pakar berbasis *rule* mempunyai arsitektur yang dapat dijelaskan sebagai berikut :

1. *User inferensi*

Digunakan sebagai media oleh user untuk melihat dan berinteraksi dengan sistem.

2. *Developer interface*

Media yang digunakan untuk mengembangkan sistem oleh engineer.

3. Fasilitas penjelasan

Sub sistem yang berfungsi untuk menyediakan penjelasan dalam sistem *reasoning*.

4. Program eksternal

Program seperti *database*, *spreadshet*, yang bekerja dalam mendukung keseluruhan sistem.

2.2.4 Penerapan *Fuzzy Tsukamoto* sebelumnya

Metode *Fuzzy Tsukamoto* dapat diterapkan dalam diagnosa suatu penyakit. Metode ini digunakan karena pada metode tsukamoto setiap konsekuen pada aturan yang berbentuk *if then* harus dipresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap – tiap aturan diberikan secara tegas berdasarkan α -Predikat. Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot. Metode ini pernah diterapkan pada beberapa permasalahan diantaranya adalah “Sistem Pendukung Keputusan Investasi Penanaman Pohon Pada UD.Murah Rejeki Jepara Dengan Algoritma *Fuzzy Tsukamoto*” (RF Muhammad, 2013), “Aplikasi *Fuzzy Inference System (FIS) Tsukamoto* untuk Menganalisa Tingkat Resiko Penyakit Dalam” (Tria Falopi, 2011).

2.3 UML (*Unified Modeling Language*)


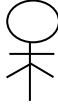

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object – Oriented*).

2.3.1 Diagram – diagram UML

1. Use Case Diagram

Merupakan model digram UML yang digunakan untuk menggambarkan requirement fungsional yang diharapkan dari sebuah sistem. *Use case diagram* menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak akan dibangun.

Tabel 2.1 *Use Case Diagram*

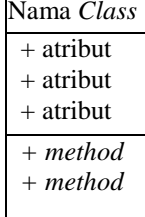
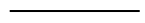
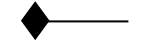
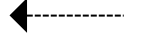
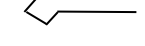
Nama komponen	Keterangan	Simbol
<i>Use Case</i>	<i>Use case</i> digunakan sebagai lingkaran elips dengan nama use case dituliskan didalam elips tersebut.	
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .	
<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>actor</i> dengan <i>Use case</i> .	

2. Class Diagram

Class Diagram digunakan untuk menampilkan kelas – kelas dan paket – paket di dalam system. *Class diagram* memberikan gambaran system secara statis dan relasi antar mereka. Biasanya, dibuat beberapa class diagram untuk system tunggal.

Beberapa diagram akan menampilkan subset dari kelas – kelas dan relasinya.

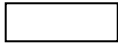



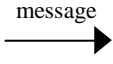
Tabel 2.2 *Class Diagram*

Nama Komponen	Keterangan	Simbol
<i>Class</i>	<i>Class</i> adalah blok – blok pembangunan pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan <i>method - method</i> dari sebuah <i>class</i> .	
<i>Association</i>	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> .(contoh : <i>One-to-one, one-to-many, many-to-many</i>)	
<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut.	
<i>Dependency</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.	
<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi	

3. *Sequence Diagram*

Merupakan suatu diagram yang memperlihatkan atau menampilkan interaksi – interaksi antar objek didalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagiannya berupa pesan.





Tabel 2.3 *Sequence Diagram*



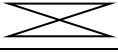

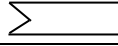

Nama komponen	Keterangan	Simbol
<i>Object</i>	Simbol yang menggambarkan suatu objek yang saling berinteraksi	
<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi	
<i>Lifeline</i>	Menyatakan kehidupan suatu objek	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah aobyek yang akan melakukan sebuah aksi	
<i>Message</i>	<i>Mesaage</i> , digambarkan dengan anak panah horizontal antara <i>activation message</i> mengindikasikan komunika komunikasi antara <i>object-object</i> .	

4. *Activity Diagram*

Merupakan representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut.

Tabel 2.4 *Activity Diagram*

Nama komponen	Keterangan	Simbol
<i>Start state</i>	Menunjukkan dimana aliran kerja itu dimulai	
<i>End state</i>	Menunjukkan dimana aliran kerja itu berakhir	
<i>Action state</i>	<i>Action state</i> adalah langkah – langkah dalam sebuah <i>activity</i> . <i>Action</i> bisa terjadi saat memasuki <i>activity</i> , meninggalkan <i>activity</i> , atau pada <i>event</i> yang spesifik	
Percabangan/	Pilihan untuk pengambilan	

<i>decision</i>	keputusan	
Fork node	Untuk menunjukkan kegiatan yang dilakukan secara paralel	
Rake	Menunjukkan adanya dekomposisi	
Time	Tanda waktu	
Send	Tanda pengirim	
Receive	Tanda penerimaan	
Flow Final	Aliran akhir	

2.4 Hypertext Preprocessor (PHP)

Pada awalnya PHP merupakan singkatan dari *Personal Home Page tools*, yang gunanya untuk memonitor pengunjung web. PHP mula – mula dikembangkan oleh Rasmus Lerdorf. Kemudian istilah PHP mengacu pada *Hypertext Preprocessor*. PHP merupakan bahasa berbentuk skrip yang di tempatkan dalam server dan di proses di server. Hasilnya akan dikirimkan ke client, tempat pemakai menggunakan browser. PHP dikenal sebagai sebuah bahasa scripting, yang menyatu dengan tag – tag HTML, dieksekusi di server, dan digunakan untuk membuat halaman web yang dinamis seperti halnya *Active Server Pages (ASP)* atau *Java Server Pages (JSP)*. PHP merupakan sebuah *software open source*. Dan hal inilah yang membedakan ASP dengan PHP.

Kelebihan PHP dari bahasa pemrograman lain yaitu :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya,
2. Web server yang mendukung PHP dapat ditemukan dimana – mana dari mulai *apache, IIS, Lighttpd, hingga Xitami* dengan konfigurasi yang relatif mudah.

3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis – milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.

2.5 MySQL

Merupakan sebuah perangkat lunak sistem manajemen basis data SQL / DBMS (*Database Management System*) yang multi user, dengan sekitar 6 juta instalasi diseluruh Indonesia. Didistribusikan secara gratis dibawah lisensi GPL (*General Public Lisence*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial.

Keistimewaan MySQL yakni :

1. Portabilitas MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
2. *Open Source* MySQL didistribusikan secara *open source* dibawah lisensi GPL sehingga dapat digunakan secara Cuma - cuma.
3. *Multiuser*, MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain – lain.

5. Perintah dan fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
6. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.