

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Pakar**

Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah. Sebuah sistem pakar umumnya tidak memiliki pengetahuan lain diluar area pengetahuannya kecuali jika diprogram dan dimuat ke dalam sistem. (Rika Rosnelly; 2012: 2)

##### **II.1.1 Karakteristik Sistem Pakar**

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

1. Kinerja sangat baik (*high performance*)

Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.

2. Waktu respon yang baik (*adequate respon time*)

Sistem juga harus mampu bekerja dalam waktu yang sama baiknya (*reasonable*) atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (*real-time*).

3. Dapat dipahami (*understandable*)

Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.

4. Dapat diandalkan (*good reliability*)

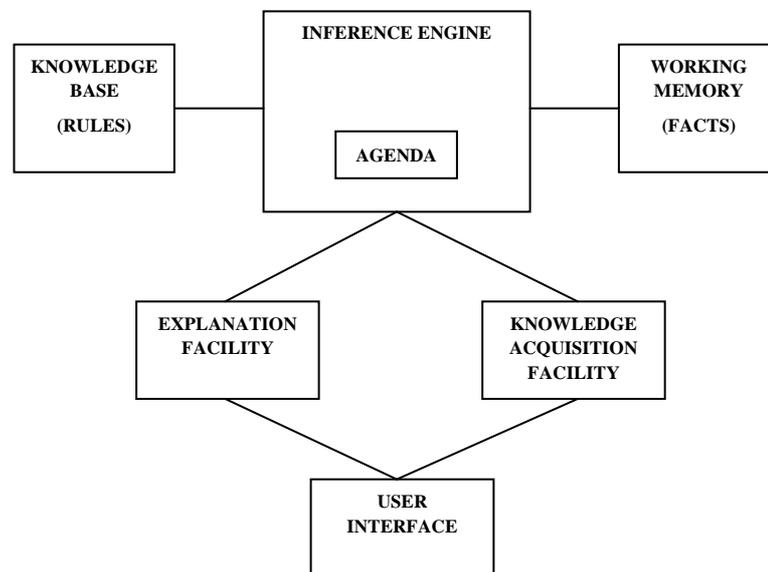
Sistem harus dapat diandalkan dan tidak mudah rusak / *crash*.

5. Fleksibel (*flexibility*)

Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan. (Rika Rosnelly; 2012: 20-21)

### II.1.2 Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada gambar II.1. berikut ini :



**Gambar II.1. Struktur Sistem Pakar**  
(Sumber : Rika Rosnelly, 2013 : 13)

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base (rules)*, *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility*, *user interface*.

1. *Knowledge Base* (Basis Pengetahuan)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui. Pada struktur sistem pakar diatas, *knowledge base* disini untuk menyimpan pengetahuan dari pakar berupa rule / aturan (if <kondisi> then <aksi> atau dapat juga disebut condition-action rules).

2. *Inference Engine* (Mesin Inferensi)

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* (struktur control) atau rule interpreter (dalam sistem pakar berbasis kaidah). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah processor pada sistem pakar yang mencocokkan bagian kondisi dari rule yang tersimpan di dalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

### 3. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global *database* dari fakta yang digunakan oleh rule-rule yang ada.

### 4. *Explanation Facility*

Menyediakan kebenaran dari solusi yang di hasilkan kepada *user* (*reasoning chain*).

### 5. *Knowledge Acquisition Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program komputer, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

### 6. *User Interface*

Mekanisme untuk memberi kesempatan kepada *user* dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.  
(Rika Rosnelly; 2012: 13-15)

### **II.1.3 Manfaat Sistem Pakar**

Menurut T. Sutojo, et al. (2011: 160-161) Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat diberikannya, diantaranya :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal.
8. Meningkatkan kapabilitas sistem komputer.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

### **II.1.4 Kekurangan Sistem Pakar**

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar menurut T. Sutojo, et al. (2011: 161), diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.

2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

## II.2. Morfologi Jamur

Menurut Achmad, et al (2013: 3), Jamur merupakan fungi yang memiliki bentuk luar berupa tubuh buah berukuran besar sehingga dapat diamati mata secara langsung. Umumnya bentuk tubuh buah jamur yang tampak di permukaan media tumbuh seperti payung. Tubuhnya terdiri dari bagian tegak yang berfungsi sebagai batang penyangga tudung serta tudung yang berbentuk mendatar atau membulat. Bagian tubuh lainnya adalah jaring-jaring di bawah permukaan media tumbuh berupa *miselia* yang tersusun dari berkas-berkas *hifa*. Morfologi jamur sangat bervariasi, terutama bentuk tudungnya.

### II.2.1 Pertumbuhan dan Reproduksi Jamur

Perkembangbiakan jamur dibagi menjadi 2 tahap, yakni aseksual dan seksual.

#### 1. Tahap Aseksual

Tahap aseksual berarti berkembangnya *miselium* melalui tiga tahapan sebagai berikut :

- a. *Miselium primer*, yaitu yang hanya memiliki satu inti, disebut monokarion dan bersifat infertil. *Miselium* ini tumbuh dari perkecambahan basidiospora.

- b. *Miselium sekunder*, yaitu *miselium* yang terbentuk dari penggabungan dua *miselium* dengan tipe berbeda. *Miselium sekunder* ini memiliki dua inti yang disebut dengan dikarion dan bersifat fertil, sehingga dapat menghasilkan tubuh buah jamur.
- c. *Miselium tersier*, yaitu *miselium* yang terbentuk dari tubuh buah terutama dari jaringan seperti tudung, tangkai, atau *lamella*. Sama seperti *miselium sekunder*, *miselium* ini juga memiliki dua inti dan dapat menghasilkan tubuh buah. Tahapan perkembangan ini biasa terjadi pada jamur *shiitake*.

Perlu diketahui, bentuk dan ukuran *spora* jamur itu berbeda. Ada *spora* jamur berbentuk *uniseluler*, tetapi ada juga yang *multiseluler*. Pada kondisi lingkungan yang sesuai, jamur dapat memperbanyak diri dengan memproduksi sejumlah besar *spora*. *Spora* yang tumbuh ini selanjutnya terbawa oleh air atau angin, dan tumbuh menjadi jamur yang dewasa jika mendapat tempat yang sesuai.

## 2. Tahap Seksual

Reproduksi jamur secara seksual atau generatif biasanya terjadi melalui kontak gametangium dan konjugasi. Kontak ini menyebabkan terjadinya singami atau menyatunya sel dari dua individu. Singami terjadi melalui dua tahap, yaitu *plasmogami* dan *kariogami*. Tahap pertama (*plasmogami*) merupakan tahap peleburan sitoplasma, sedangkan tahap kedua (*kariogami*) merupakan tahap peleburan inti. (Redaksi Agromedia; 2010: 5-6)

## II.2.2 Jamur Edibel dan Jamur Non-Edibel

Menurut Achmad, et al (2013: 10), jamur dikelompokkan menjadi 3 golongan berdasarkan manfaatnya, yaitu jamur yang dapat dikonsumsi , jamur yang digunakan sebagai obat, dan jamur yang beracun.

### 1. Jamur Edibel

Jamur edibel adalah semua jenis jamur yang dapat dimakan dan atau dapat diolah sebagaimana lazimnya bahan pangan lainnya. Jamur edibel yang telah dikonsumsi secara luas oleh masyarakat di seluruh dunia, antara lain jamur tiram (*Pleurotus ostreatus*), jamur kuping (*Auricularia polytrica*), jamur kancing (*Agaricus bisporus*), jamur shiitake (*Lentinula edodes*), jamur merang (*Volvariella volvacea*).

### 2. Jamur Non-Edibel

Non-edibel merupakan lawan dari kata edibel. Jamur non-edibel artinya jamur tidak dapat makan. Istilah ini juga sering disandingkan dengan istilah beracun (poisonous). Oleh karena itu, kategori non-edibel adalah jamur obat dan jamur beracun.

#### a. Jamur obat

Jamur kerap kali digunakan dalam pengobatan dan upaya peningkatan kesehatan. Beberapa jenis jamur obat yang telah dikenal luas, antara lain reishi/lingzhi (*Ganoderma lucidum* dan *Ganoderma tsugae*), maitake (*Grifola frondosa*), enokitake (*Flammulina velutipes*), dan *Agaricus blazei* (*Agaricus subrufescens*).

b. Jamur beracun

Jamur non-edibel ada yang berbahaya jika dikonsumsi karena beracun. Beberapa jenis jamur beracun memiliki tampilan yang mirip dengan jamur yang bisa dimakan. Kemiripan inilah yang sering menjadi sebab kesalahan mengidentifikasi sehingga jamur beracun disangka jamur pangan. Beberapa jenis jamur yang beracun adalah jamur *Amanita muscaria*, jamur Papan (*Polyporus*), dan jamur *Omphalotus olarius*.

### II.3. Teorema Bayes

Menurut Rika Rosnelly (2012: 79-81), Probabilitas bayes merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan formula bayes yang dinyatakan :

$$P(H | E) = \frac{P(E | H) * P(H)}{P(E)} \dots \dots \dots (1)$$

Dimana :

$P(H | E)$  = Probabilitas hipotesis H benar jika diberi *evidence* E.

$P(E | H)$  = Probabilitas munculnya *evidence* E jika diketahui hipotesis H.

$P(H)$  = Probabilitas hipotesis H tanpa memandang *evidence* apapun.

$P(E)$  = Probabilitas *evidence* E.

Secara umum, teorema bayes dengan E kejadian dan hipotesis H dapat dituliskan dalam bentuk :

$$\begin{aligned}
 P(H_i | E) &= \frac{P(E \cap H)}{\sum p(E \cap H_i)} \dots\dots\dots(2) \\
 &= \frac{p(E | H_i) P(H_i)}{\sum (P(E | H_i)P(H_i))} \\
 &= \frac{P(E | H_i) P(H_i)}{P(E)}
 \end{aligned}$$

Teorema Bayes dapat dikembangkan jika setelah dilakukan pengujian terhadap hipotesis kemudian muncul lebih dari sebuah *evidence*. Dalam hal ini maka persamaannya akan terjadi :

$$P(H | E,e) = P(H | E) \frac{P(e | E,H)}{P(e | E)} \dots\dots\dots(3)$$

Dimana :

e = *Evidence* lama.

E = *Evidence* baru.

$P(H | E,e)$  = Probabilitas hipotesis H benar jika muncul *evidence* baru E dari *evidence* lama e.

$P(H | E)$  = Probabilitas hipotesis H benar jika diberikan *evidence* E.

$P(e | E,H)$  = Kaitan antara e dan E jika hipotesis H benar.

$P(e | E)$  = Kaitan antara e dan H tanpa memandang hipotesis apapun.

#### II.4. Basis Data

Menurut Bunafit Nugroho (2005: 72), Basisdata dapat juga disebut *database*, adalah sekumpulan informasi yang sangat kompleks yang berguna untuk mengatur semua data yang ada di dalamnya sehingga dapat diakses oleh pengguna dengan mudah dan cepat.

Dalam basisdata, data yang ada tidak hanya diletakkan dan disimpan begitu saja dalam sebuah media penyimpanan, akan tetapi dikelola dengan sebuah sistem pengaturan basisdata yang sering disebut dengan *Database Management System* (DBMS).

#### II.5. ERD

*Entity Relationship Diagram* (ERD) adalah sarana untuk menggambarkan hubungan antar data di dalam sebuah sistem, ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. (Elfani., Ardi Pujiyanta; 2013)

Berikut adalah simbol-simbol yang ada pada *Entity Relationship Diagram* (ERD) :

**Tabel II.1. Simbol-Simbol ERD**

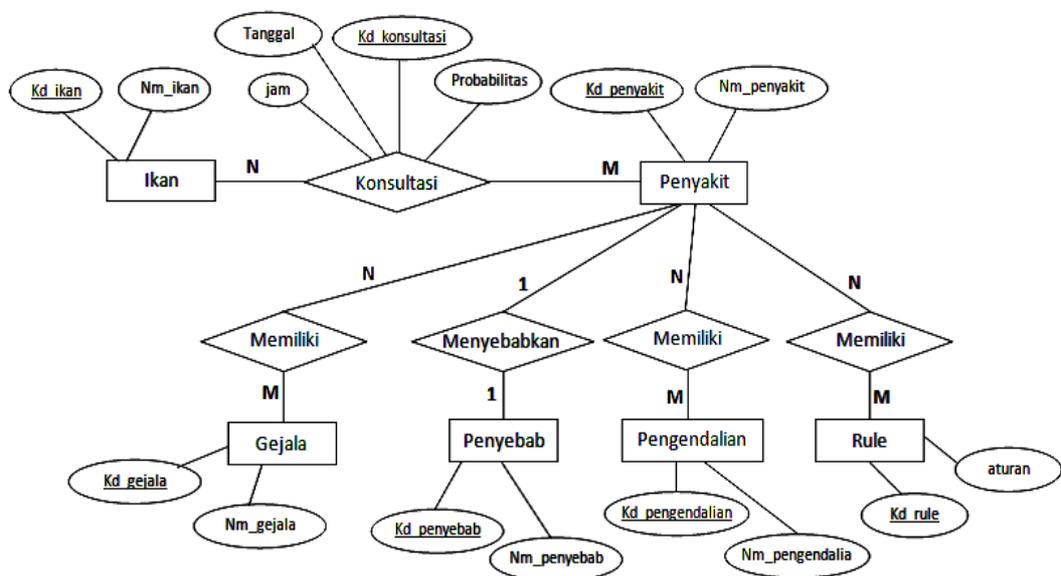
Simbol	Deskripsi
 Entitas	Persegi panjang mewakili kumpulan entitas
 Atribut	Elips mewakili atribut

	Belah ketupat mewakili relasi
—————	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

(Sumber : Janner Simarmata, Iman Paryudi; 2006: 60)

Di bawah ini adalah contoh dari ERD pada studi kasus Sistem Pakar

Mendiagnosa Penyakit Pada Ikan Konsumsi Air Tawar :



**Gambar II.2. ERD Sistem Pakar Mendiagnosa Penyakit Pada Ikan Konsumsi Air Tawar**

(Sumber : Elfani, Ardi Pujiyanta; 2013)

## II.6. Kamus Data

Menurut Rosa A.S., M. Shalahuddin (2014: 73), Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur.

Berikut adalah contoh dari kamus data pada studi kasus Sistem Pakar Mendeteksi Kerusakan Telepon Selular :

**Tabel II.2. Kamus Data Diagnosa**

<b>Field Name</b>	<b>Keterangan</b>
Idnomor	Idnomor diagnosa dalam website sistem pakar
Kategori	Kategori kerusakan telepon selular
Pertanyaan	Pertanyaan yang akan ditampilkan
JawabYa	True dan False
KesimpulanYa	Kesimpulan akhir dari pertanyaan, jika pilihan ya
KesimpulanTidak	Kesimpulan akhir dari pertanyaan, jika pilihan tidak

*(Sumber : Dinny Wahyu Widarti, Endah Setyowati)*

**Tabel II.3. Kamus Data Kesimpulan**

<b>Field Name</b>	<b>Keterangan</b>
Idkesimpulan	Id kesimpulan sistem pakar
Kesimpulan	Kesimpulan akhir berupa kerusakan yang terjadi pada telepon selular

*(Sumber : Dinny Wahyu Widarti, Endah Setyowati)*

## **II.7. Normalisasi**

Menurut Janner Simarmata dan Iman Paryudi (2006: 77)

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai

pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional.

### II.7.1 Tahapan Normalisasi

Ada beberapa tahap yang harus dilakukan dalam menormalisasi tabel. Tahapan-tahapan ini diperlukan untuk menyesuaikan tabel dengan data asli yang ada. Tahapan itu adalah :

#### 1. Bentuk Tidak Normal

Menurut Bunafit Nugroho (2005: 85), Bentuk tidak normal merupakan bentuk tabel yang tidak dapat menampung data secara lengkap, ditandai dengan adanya data yang tidak lengkap sehingga informasinya tidak dapat terbaca utuh.

**Tabel II.4. Contoh Bentuk Data Tidak Normal**

No	No_mhs	Nama	Jurusan	Jen
1	025410108	Bunafit Nugroho	Tehnik Informatika	S1
2	01541005	Bambang Mariaci	Tehnik Informatika	S1
3	013110110	Esti Farida		
4	015410073	Mawardi Budiono		
5	023110171	Iis Suwindri	Manajemen Informatika	D3
6			Sistem Informasi	
7	015410122	Sugeng Fitriyadi		

(Sumber : Bunafit Nugroho; 2005: 86)

## 2. Bentuk Normal ke-1

Menurut Bunafit Nugroho (2005: 89), Bentuk normal ke-1 memiliki ketentuan sebagai berikut :

- Setiap record berisi data lengkap atau tidak ada isian yang kosong
- Satu pengetahuan, satu atribut/kolom/field

Jadi, pada bentuk normal ke-1 ini juga telah terbentuk tabel yang datanya sudah benar-benar memenuhi kedua kriteria di atas. Berikut adalah contohnya bentuk normal ke-1 :

**Tabel II.5. Contoh Bentuk Normal Ke-1**

No_mhs*	Nama	Kd_mk	Nama_mk	Nilai
025410108	Bunafit Nugroho	Alg001	Algoritma	A
01541005	Bambang Mariaci	Bas001	Basisdata1	A
013110110	Esti Farida	Fis001	Fisika	B
015410073	Mawardi Budiono	Fis001	Fisika	C
023110171	Iis Suwindri	Pas001	Pascal	A
025410101	Yuan Wiana	Bas002	Basis Data	C
015410122	Sugeng Fitriyadi	Alg001	Algoritma	B

*(Sumber : Bunafit Nugroho; 2005: 92-93)*

## 3. Bentuk Normal ke-2

Menurut Bunafit Nugroho (2005: 90), Ciri-ciri bentuk normal ke-2 adalah sebagai berikut :

- Datanya berasal dari normal ke-1.
- Setiap entitas/tabel harus memiliki kunci primer, apabila belum ada harus dibuat pada tahap ini. Bentuk kunci primer dibuat

berdasarkan nama tabel atau yang mewakili seluruh data tabel tersebut.

- Atribut-atribut ditempatkan ke dalam tabel relasi yang terkait dengan entitas/kunci primernya. Penempatan kunci primer ini akan dianggap sebagai kunci tamu pada tabel yang didatangi.

Berikut ini adalah contoh bentuk normal ke-2 :

**Tabel II.6. Contoh Bentuk Normal Ke-2**

<b>No_mhs*</b>	<b>Nama</b>	<b>Nilai</b>
025410108	Bunafit Nugroho	?
01541005	Bambang Mariaci	?
013110110	Esti Farida	?
015410073	Mawardi Budiono	?
023110171	Iis Suwindri	?
025410101	Yuan Wiana	?
015410122	Sugeng Fitriyadi	?

<b>Kd_mk*</b>	<b>Nama_mk</b>	<b>Nilai</b>
Alg001	Algoritma	?
Bas001	Basisdata1	?
Fis001	Fisika	?
Pas001	Pascal	?
Bas002	Basis Data	?

*(Sumber : Bunafit Nugroho; 2005: 93)*

#### 4. Bentuk Normal ke-3

Menurut Bunafit Nugroho (2005: 90), bentuk normal ke-3 memiliki ciri sebagai berikut :

- Berasal dari bentuk normal ke-2.
- Semua atribut tergantung penuh pada kunci primer.

Adapun langkah-langkah dalam bentuk normalisasi ke-3 :

- Memecah atribut yang mengalami redudansi. Apabila kedua identitas tersebut sama-sama mengalami redudansi, tak perlu ada kunci relasi.
- Atribut / *field* yang mengalami kesulitan dalam memasukkan data atau redudansi harus dipisah menjadi tabel tersendiri.
- Kunci primer tabel merupakan gabungan kunci primer dari masing-masing tabel tempat atribut ini ada.

Berikut adalah contoh bentuk normal ke-3 :

**Tabel II.7. Contoh Bentuk Normal Ke-3**

No_mhs*	Nama
025410108	Bunafit Nugroho
01541005	Bambang Mariaci
013110110	Esti Farida
015410073	Mawardi Budiono
023110171	Iis Suwindri
025410101	Yuan Wiana
015410122	Sugeng Fitriyadi

<b>Kd_mk*</b>	<b>Nama_mk</b>
Alg001	Algoritma
Bas001	Basisdata1
Fis001	Fisika
Fis001	Fisika
Pas001	Pascal
Bas002	Basis Data
Alg001	Algoritma

<b>No_mhs**</b>	<b>Kd_mk**</b>	<b>Nilai</b>
025410108	Alg001	A
01541005	Bas001	A
013110110	Fis001	B
015410073	Fis001	C
023110171	Pas001	A
025410101	Bas002	C
015410122	Alg001	B

(Sumber : Bunafit Nugroho; 2005: 94)

## II.8. MySQL

Menurut Bunafit Nugroho (2005: 1), MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL* (*Structred Query Language*). MySQL memiliki dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah

MySQL *FreeSoftware* yang berada di bawah Lisensi GNU/GPL (*General Public Licence*).

### II.8.1 Kelebihan MySQL

Menurut Bunafit Nugroho (2005: 3-4), MySQL memiliki beberapa kelebihan dibanding *database* lain, diantaranya adalah :

1. MySQL sebagai *Database Management System* (DBMS)
2. MySQL merupakan sebuah *database server*, jadi dengan menggunakan *database* ini anda dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
3. MySQL merupakan sebuah *database* yang mampu menyimpan ata berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
4. MySQL memiliki kecepatan dalam pembuatan *table* maupun *update-an table*.
5. MySQL adalah *database* menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.

### II.9. PHP

Menurut Sibero (2013 : 49), PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. PHP adalah suatu

bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya.

## II.10. *Unified Modelling Language (UML)*

Menurut Munawar (2005: 17), UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Berikut ini adalah beberapa diagram UML (Rosa A. S, M. Shalahuddin; 2014: 141-167) :

### 1. *Class Diagram*

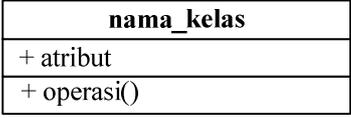
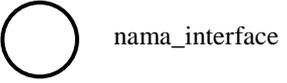
Diagram kelas atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas

- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas :

**Tabel II.8. Simbol-Simbol Diagram Kelas**

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi / <i>Generalization</i></p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas
<p>Agregasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semua bagian ( <i>whole-part</i> )

(Sumber : Rosa A.S, M. Shalahuddin, 2014 : 141-147)

## 2. Use Case Diagram

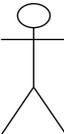
*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- Aktor merupakan 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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel II.9. Simbol-simbol Diagram Use Case**

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor / Actor</p> 	<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</p>

	benda diawal frase nama aktor
Asosiasi / <i>Association</i> _____	Komuniikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / <i>Extends</i> «extends» —————>	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> yang ditambahkan
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.
Menggunakan / <i>Include / Uses</i> «uses» —————>	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

(Sumber : Rosa A.S, M. Shalahuddin; 2014 : 156-158)

### 3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel II.10. Simbol-simbol Diagram Aktivitas**

<b>Simbol</b>	<b>Keterangan</b>
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	<i>Decision</i> , atau pilihan untuk mengambil keputusan.
Penggabungan / <i>Join</i> 	Arah tanda panah alur proses.
Status Akhir 	Titik akhir atau akhir dari aktivitas.

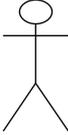
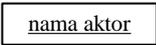
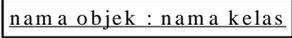
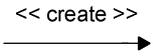
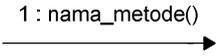
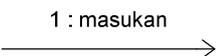
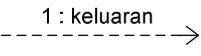
(Sumber : Rosa A.S, M. Shalahuddin; 2014 : 162)

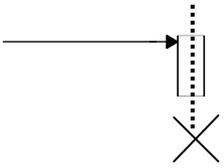
#### 4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

**Tabel II.11. Simbol-simbol Diagram Sekuen**

Simbol	Keterangan
<p>Aktor / <i>Actor</i></p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<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 diawal frase nama aktor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek</p>

	yang menerima kembalian
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

(Sumber : Rosa A.S, M. Shalahuddin, 2014 : 165-167)