

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1 Konsep Sistem Informasi**

Sistem merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan. Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta – fakta yang ada (Budi Sutejo : 2006 : 168).

##### **II.1.1. Sistem Informasi**

Sistem informasi secara sederhana dapat diartikan sebagai kumpulan dari beberapa komponen yang saling berinteraksi untuk mencapai hasil dari satu tujuan. Pengertian sederhana ini sesuai dengan pendapat O'Brien (2006,p5)" Sistem Informasi dapat merupakan kombinasi teratur apapun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi". Menurut Whitten (2004,p12) "*information system is an arrangement of people, data, process and information technology that interact to collect, process, store, and provide as output the information needed to support an organization*". Defenisi tersebut dapat dijelaskan sistem informasi adalah susunan dari orang, data, pemrosesan dan teknologi informasi yang dibutuhkan untuk mendukung sebuah organisasi ( Dr. Hj. Henny Hendarti, S.Kom, MM : 2011:1)

### II.1.2. Sistem Pakar

Dalam ilmu komputer, banya ahli yang berkonsentrasi pada pengembangan kecerdasan buatan atau *Artificial Intelligence* (AI). AI adalah suatu studi khusus di mana tujuannya adalah membuat komputer berpikir dan bertindak seperti manusia. Banyak implementasi AI dalam bidang komputer, misalnya *Decision Support System* (Sistem Pendukung Keputusan), *Robotic*, *Natural Language* (Bahasa Alami), *Neural Network* (Jaringan Saraf), dan lain-lain.

Contoh bidang lain pengembangan kecerdasan buatan adalah sistem pakar yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak.

Ada banyak manfaat yang dapat diperoleh dengan mengembangkan system pakar, antara lain :

1. Masyarakat awam non-pakar dapat memanfaatkan keahlian di dalam bidang tertentu tanpa kehadiran langsung seorang pakar.
2. meningkatkan produktivitas kerja, yaitu bertambah efisiensi pekerjaan tertentu serta hasil solusi kerja.
3. penghematan waktu dalam meyelesaikan masalah yang kompleks.
4. Memberikan penyederhanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang.
5. pengetahuan dari seorang pakar dapat didokumentasikan tanpa batas waktu.

6. Memungkinkan penggabungan berbagai bidang pengetahuan dari berbagai pakar untuk dikombinasikan.

Berikut ini merupakan perbandingan antara kemampuan pakar manusia dan sistem komputer yang menjadi pertimbangan pengembangan sistem pakar (Tim Penerbit Andi : 2009 : 3).

**Tabel II.1. Perbandingan Pakar Manusia Dengan Sistem Pakar**

<b>Pakar Manusia</b>	<b>Sistem Pakar</b>
Terbatas waktu karena manusia membutuhkan istirahat.	Tidak terbatas karena dapat digunakan kapanpun juga.
Tempat akses bersifat lokal pada suatu tempat saja dimana pakar berada.	Dapat digunakan diberbagai tempat.
Pengetahuan bersifat variabel dan dapat berubah-ubah tergantung situasi.	Pengetahuan bersifat konsisten.
Kecepatan untuk menentukan solusi sifatnya bervariasi.	Kecepatan untuk memberikan solusi konsisten dan lebih cepat daripada manusia.
Biaya yang harus dibayar untuk konsultasi biasanya sangat mahal.	Biaya yang dikeluarkan lebih murah.

Selain dari beberapa manfaat yang diperoleh, ada juga kelemahan pengembangan sistem pakar, yaitu :

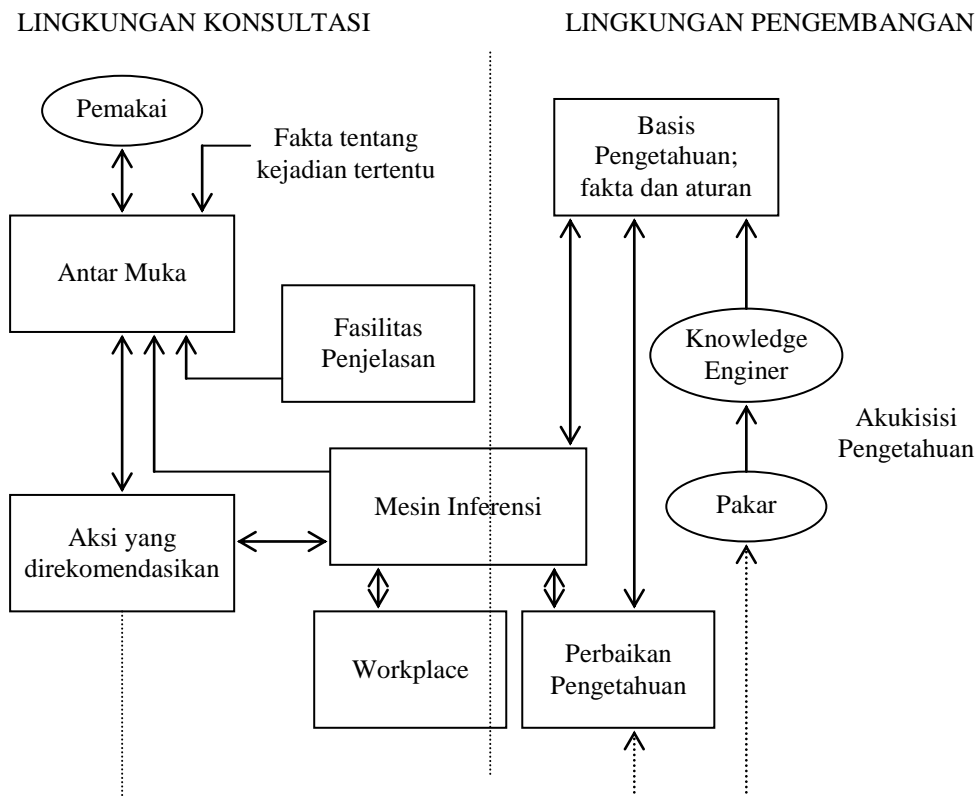
1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional. Hal ini dapat dilihat dari tabel perbandingan berikut ini :

**Tabel II.2: Kelemahan Pengembangan Sistem Pakar**

<b>Perangkat Lunak Konvensional</b>	<b>Perangkat Lunak Sistem Pakar</b>
Fokus pada solusi.	Fokus pada permasalahan.
Pengembangan dapat dilakukan secara individu.	Pengembangan dilakukan oleh tim kerja.
Pengembangan secara sekuensial.	Pengembangan secara interaktif.

### **II.1.3. Struktur Sistem Pakar**

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*) (Turban, 1995). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam gambar II.1. berikut ini :



**Gambar II.1** Arsitektur sistem pakar (Sumber: Turban 1995)

Komponen-komponen yang terdapat dalam sistem pakar adalah seperti yang terdapat pada gambar II.1, yaitu *User Interface* (antar muka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan (Arhami; 2005: 14).

#### II.1.4. Metode Forward Chaining

Suatu perkalian inferensi yang menghubungkan suatu permasalahan dengan solusinya disebut dengan rantai (*chain*). Suatu rantai yang dicari atau dilewati/dilintasi dari suatu permasalahan untuk memperoleh solusinya disebut dengan *forward chaining*. Cara lain menggambarkan *forward chaining* ini adalah

dengan penalaran dari fakta menuju konklusi yang terdapat dari fakta (Arhami; 2005 : 111).

## **II.2. Tahapan Tumbuh Kembang Anak**

Tahapan pertumbuhan dan perkembangan anak dapat ditentukan oleh masa atau waktu kehidupan anak. Secara umum terdiri atas masa prenatal dan masa postnatal.

### **1. Masa Prenatal**

Masa prenatal terdiri atas dua fase, yaitu fase embrio dan fase fetus. Pada fase embrio, pertumbuhan dapat diawali mulai dari konsepsi hingga 8 minggu pertama yang dapat terjadi perubahan cepat dari ovum menjadi suatu organisme dan terbentuknya manusia. Pada minggu ke-2, terjadi pembelahan sel dan pemisahan jaringan antara endosterm dan ekstoderm. Pada minggu ke-3 terbentuk lapisan mesoderm. Pada masa ini sampai usia 7 minggu belum tampak adanya gerakan yang berarti melainkan hanya terdapat denyut jantung janin, yaitu sudah mulai dapat berdenyut sejak 4 minggu. Pada fase fetus terjadi sejak usia 9 minggu hingga kelahiran, sedangkan minggu ke-12 sampai ke-40 terjadi peningkatan fungsi organ, yaitu bertambah ukuran panjang dan berat badan terutama pertumbuhan serta penambahan jaringan subkutan dan jaringan otot.

### **2. Masa Postnatal**

Masa postnatal terdiri atas masa neonatos, masa bayi, masa prasekolah, masa sekolah, dan masa remaja.

a. Masa Neonatos (0-28 hari)

Pertumbuhan dan perkembangan postnatal atau dikenal dengan pertumbuhan dan perkembangan setelah lahir ini diawali dengan masa neonatos (0-28). Masa ini merupakan masa terjadinya kehidupan yang baru dalam akstreuteri, yaitu adanya proses adaptasi semua sistem organ tubuh. Proses adaptasi dari organ tersebut dimulai dari aktivitas pernapasan yang disertai dengan pertukaran gas dengan frekuensi pernapasan antara 35-50 kali per menit, penyesuaian denyut jantung antara 120-160 kali per menit dengan ukuran jantung lebih besar apabila dibandingkan dengan rongga dada.

b. Masa Bayi

Masa bayi ini dibagi menjadi dua tahap perkembangan. Tahap pertama (antara usia 1-12 bulan) pertumbuhan dan perkembangan pada masa ini dapat berlangsung secara terus-menerus, khususnya dalam peningkatan susunan saraf. Tahap kedua (usia 1-12 tahun) kecepatan pertumbuhan pada masa ini mulai menurun dan terdapat percepatan pada perkembangan motorik.

c. Masa Prasekolah

Perkembangan pada masa ini dapat berlangsung stabil dan masih terjadi peningkatan pertumbuhan serta perkembangan, khususnya pada aktivitas fisik dan kemampuan kognitif.

d. Masa Sekolah

Perkembangan masa sekolah ini lebih cepat dalam kemampuan fisik dan kognitif dibandingkan dengan masa prasekolah.

#### e. Masa Remaja

Pada tahap perkembangan remaja terjadi perbedaan pada perempuan dan laki-laki. Pada umumnya wanita 2 tahun lebih cepat untuk masuk ke dalam tahap remaja/pubertas dibandingkan dengan anak laki-laki dan perkembangan ini ditunjukkan pada perkembangan pubertas (A. Aziz Alimul Hidayat : 2008: 14).

### 3. Penyakit

Ada beberapa penyakit yang biasa diderita oleh anak – anak pada masa pertumbuhannya, berikut adalah beberapa penyakit tersebut :

- a. Bronkiolitis : Bronkiolitis adalah suatu peradangan pada bronkiolus (saluran udara yang merupakan percabangan dari saluran udara utama), yang biasanya disebabkan oleh infeksi virus.
- b. Difteria : Difteria adalah suatu penyakit infeksi mendadak yang disebabkan oleh kuman *Corynebacterium Diphtheriae*. Mudah menular dan menyerang terutama saluran nafas bagian atas.
- c. Ensephalitis : Ensephalitis adalah peradangan akut otak yang disebabkan oleh infeksi virus (Yuniraharjo, 2009). Terkadang ensephalitis dapat disebabkan oleh infeksi bakteri, seperti meningitis, atau komplikasi dari penyakit lain seperti rabies(disebabkan oleh virus) atau sifilis (disebabkan oleh bakteri). Penyakit parasit dan protozoa seperti toksoplasmosis, malaria, atau primary amoebic meningoencephalitis, juga dapat menyebabkan ensephalitis pada orang yang sistem kekebalan tubuhnya kurang.

- d. Hepatitis : Hepatitis adalah suatu penyakit hati yang disebabkan oleh virus Hepatitis, suatu anggota famili Hepadnavirus yang dapat menyebabkan peradangan hati akut atau menahun.
- e. Meningitis : Morbili/campak adalah suatu infeksi virus yang sangat menular, yang ditandai dengan demam, batuk, konjungtivitis (peradangan selaput ikat mata/konjungtiva), dan ruam kulit. Penyakit ini disebabkan karena infeksi virus campak golongan Paramyxovirus (Masjoer, 2010).

### **II.3. UML (*Unified Modeling Language*)**

UML (*Unified Modeling Language*) adalah suatu alat Bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar ; 2005 : 17). 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.

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun tipe diagram UML yang ada seperti pada Tabel II.3.

Tabel II.3 Tipe Diagram UML

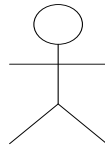
Diagram	Tujuan	Keterangan
Activity	Prilaku prosedural dan paralel	Sudah ada di UML 1
Class	Class, fitur dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara objek. Lebih menekankan kepada link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	Sudah ada di UML 1
Composite Structure	Dekomposisi sebuah class saat runtime	Baru untuk UML 2
Deployment	Penyebaran/instalasi ke klien	Sudah ada di UML 1
Interaction Overview	Gabungan dari activity dan sequence diagram	Baru untuk UML 1
Object	Contoh konfigurasi instance	Tidak resmi ada di UML 1
Package	Struktur hierarki saat kompilasi	Tidak resmi ada di UML 1
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.	Sudah ada di UML 1
State Machine	Bagaimana event mengubah sebuah objek	Sudah ada di UML 1
Timing	Interaksi antar objek. Lebih menekankan pada waktu	Sudah ada di UML 1
Use Case	Bagaimana user berinteraksi dengan sebuah sistem	Sudah ada di UML 1

Sumber : " Pemodelan Visual dengan UML (Munawar ; 2005 : 23)"

### II.3.1. Notasi Dasar UML

#### 1. Actor

Actor adalah *abstraction* dari orang dan *system* yang lain yang mengaktifkan fungsi dari target *system*. Orang atau system bisa muncul dalam bebrapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Berikut notasi actor dalam UML:

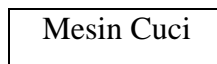


**Gambar II.2 : Notasi Actor pada UML**

*Sumber : "Pemodelan Visual dengan UML (Munawar ; 2005: 64)"*

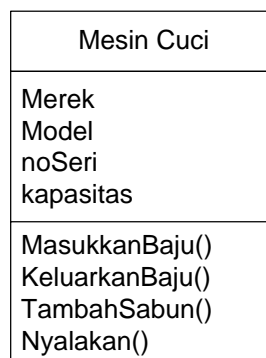
## 2. Class

Class, dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar diawal kalimatnya dan diletakkan diatas kotak. Bila class mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. Berikut notasi class dalam UML:



**Gambar II.3 : Notasi Class di UML**

*Sumber : " Pemodelan Visual dengan UML (Munawar ; 2005 : 35)"*

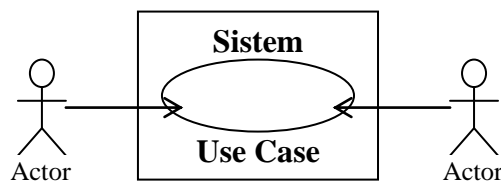


**Gambar II.4 : Contoh Class Diagram**

## 3. Use Case

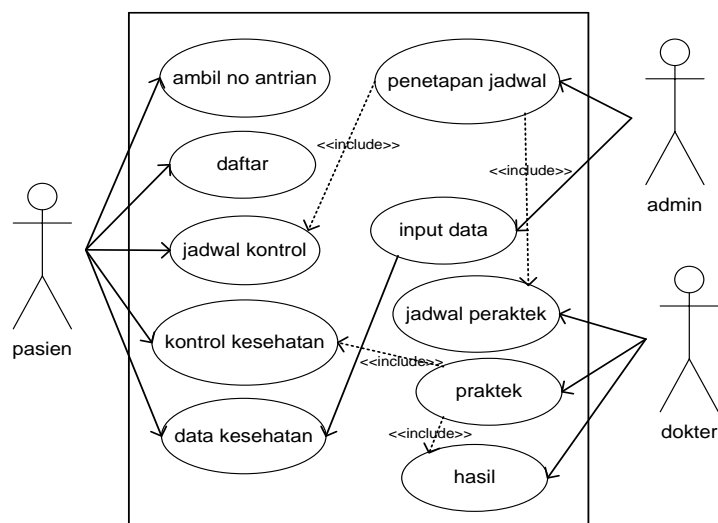
*Use Case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu *system* dari sudut pandangnya. Tidak selalu

mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah *system*. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu. Notasi *use case* dapat dilihat pada gambar II.5 :



**Gambar II.5 : Notasi Use Case pada UML**

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2005 : 64)"

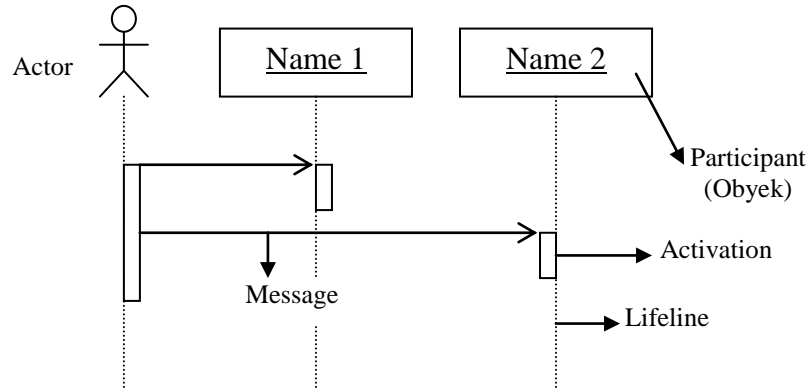


**Gambar II.6 : Contoh Diagram Use Case**

#### 4. Sequence Diagram

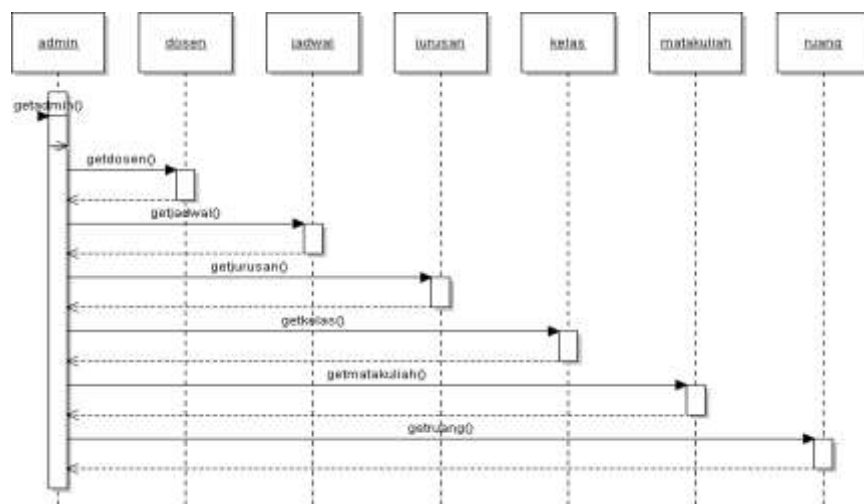
*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini dalam *use case*. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat

bernama. Message dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical. Berikut Contoh *sequence diagram* :



**Gambar II.7 : Simbol-simbol yang ada pada sequence diagram**

Sumber : " *Pemodelan visual dengan UML (Munawar ; 2005 : 89)*"







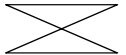

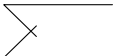



**Gambar II.8 : Contoh Squence Diagram**

## 5. Activity Diagram

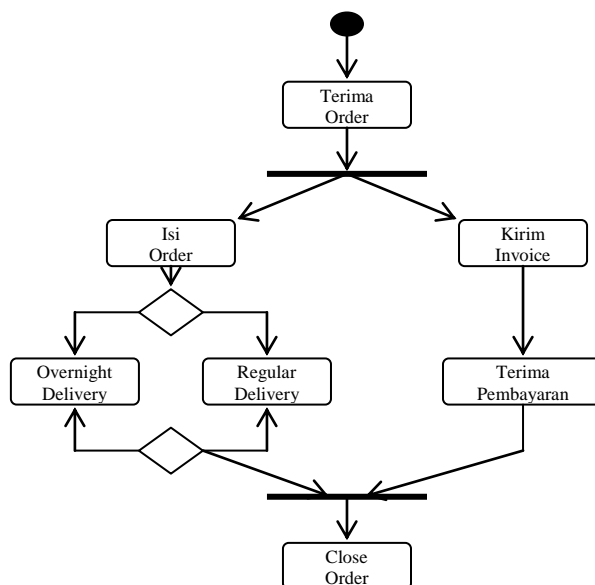
*Activity diagram* adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan activity diagram.

**Tabel II.4 Simbol-simbol yang sering dipakai pada activity diagram**

Simbol	Keterangan
	Titik awal
	Titik akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (Flow Final)

*Sumber : " Pemodelan visual dengan UML (Munawar ; 2009 : 109)"*

Adapun contoh dari Activity Diagram dapat di lihat pada Gambar II.9.



**Gambar II.9 : Contoh Activity Diagram Sederhana**

Sumber : " *Pemodelan visual dengan UML (Munawar ; 2005 : 111)*"

## II.4. Pengertian Database

*Database* merupakan komponen terpenting dalam pembangunan SI, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun-menyusun informasi-informasi dalam berbagai bentuk. *Database* merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi (Budi Sutedjo Dharma Oetomo, S.kom., MM : 2006: 99).

### II.4.1. Hierarki Data Dalam Database

Data dalam sebuah *database* disusun berdasarkan sistem hierarki yang unik, yaitu:

1. **Database**, merupakan kumpulan file yang saling terkait satu sama lain, misalnya file data induk karyawan, file jabatan file penggajian dan lain sebagainya. Kumpulan file yang tidak saling terkait satu sama lain tidak dapat disebut *database*, misalnya file data induk karyawan, file tamu undangan perkawinan, file barang retail pasar swalayan.
2. **File**, yaitu kumpulan dari *record* yang saling terkait dan memiliki format *field* yang sama dan sejenis.
3. **Record**, yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.
4. **Field**, yaitu atribut dari *record* yang menunjukkan suatu item dari data, seperti nama, alamat, dan lain sebagainya.
5. **Byte**, yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus
6. **Bit**, yaitu bagian terkecil dari data secara keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte* (Budi Sutedjo Dharma Oetomo, S.kom., MM : 2006: 102).

#### II.4.1. MySQL (phpMyAdmin)

*PhpMyAdmin* adalah aplikasi berbasis web yang dibuat dari pemrograman *PHP* dan *JavaScript*. *PhpMyAdmin* juga dapat disebut sebagai tools yang berguna untuk mengakses database MySQL Server dalam bentuk tampilan *web*. dengan adanya *phpMyAdmin*, semua pekerjaan menjadi mudah, karena tanpa harus

mengerti perintah-perintah dasar *SQL* namun sudah dapat memanajemen database dan data yang ada didalamnya (Sumber: Bunafit Nugroho: 2009: 13).

## II.5. Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perencanaan sistem (Jogiyanto: 2005: 725).

**Tabel II.5 Notasi Kamus Data**

Notasi	Arti
=	Terbentuk dari (is composed) atau terdiri dari (consist of) atau sama dengan (is equivalent of)
+	AND
[]	Salah satu dari (memilih salah satu dari elemen-elemen data di dalam kurung bracket ini)
	Sama dengan simbol []
M{ }M	Intensi (elemen data didalam kurung brace berinterasi mulai minimum N kali dan maksimum M kali)
()	Optional (elemen data di dalam kurung parenthesis sifatnya optional, dapat ada dan dapat tidak ada )
*	Keterangan setelah tanda ini adalah komentar

Sumber : " Pengantar Sistem Informasi (Jogiyanto: 2005: 730)"

## II.6. Entity Relationship Diagram – ERD

### II.6.1. Model-model Data

Struktur yang mendasari suatu basisdata adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data semantik, dan batasan konsistensi. Untuk mengilustrasikan konsep model data, berikut disajikan dua model data, yaitu *entity relationship model* dan *relational*

*model*. Kedua model menyediakan cara mendeskripsikan rancangan basisdata pada tingkatan logis.

### **II.6.2. Entity Relationship Model**

*Entity Relationship* (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

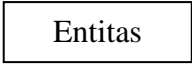
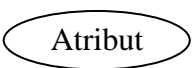


Entitas digambarkan dalam basisdata dengan kumpulan atribut. Misalnya atribut nim, nama, alamat dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terhadap dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang di ambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relationship set*).

Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut :

**Tabel II.6 Notasi ERD (Entity Relationship Diagram)**

 Entitas	Persegi panjang mewakili kumpulan entitas
 Atribut	Elips mewakili atribut
 Relasi	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

Sumber : "Basis Data (Janner Simarmata & Imam Prayudi: 2006: 59)"

## II.7. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data 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 relational ([www.utexas.edu](http://www.utexas.edu)).

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

1. Bentuk Normal Pertama (1NF)
2. Bentuk Normal Kedua (2NF)
3. Bentuk Normal Ketiga (3NF)
4. Bentuk Normal Boyce-Code (BCNF)
5. Bentuk Normal Keempat (4NF)
6. Bentuk Normal Kelima (5NF)

Tujuan normalisasi adalah membuat kumpulan tabel relasional yang bebas dari data berulang yang dapat dimodifikasi secara benar dan konsisten. Ini berarti bahwa semua tabel pada basisdata relasional harus berada pada bentuk normal ketiga (3NF). Sebuah tabel relasional berada pada 3NF jika dan hanya jika semua kolom bukan kunci adalah (a) saling independen dan (b) sepenuhnya tergantung pada kunci utama. Saling independen berarti bahwa tidak ada kolom bukan kunci yang tergantung pada senbarang kombinasi kolom lainnya. Dua bentuk normal pertama adalah langkah antara untuk mencapai tujuan, yaitu mempunyai semua tabel dalam 3NF (Stephens and Plew, 2000) (Janner Simarmata & Imam Prayudi: 2006: 77).

## **II.8. Bahasa Pemrograman PHP (*Hypertext Preprocessor*)**

Untuk membuat sebuah *website* yang dinamis dan mudah untuk *update* setiap saat dari *browser*, dibutuhkan sebuah program yang mampu mengolah data dari komputer *client* atau dari komputer *server* itu sendiri sehingga mudah dan nyaman untuk disajikan di *browser*.

Salah satu program yang dapat dijalankan di *server* dan cukup handal adalah PHP. PHP adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah server. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website*/komputer *client* akan diolah dan disimpan pada *database web server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP ini, file harus *upload* ke dalam *server* (Mei Lenawati : 2007: 3).