

## 2. LANDASAN TEORI

Pada bab ini berisikan teori-teori penunjang yang menjadi dasar bagi penulis dalam pembuatan Perancangan dan Pembuatan Sistem Informasi Administrasi untuk Penjualan, Pembelian, dan Inventori berbasis Web pada PT. X. Penjelasannya akan diuraikan pada sub bab berikut.

### 2.1 Sistem Informasi Manajemen

Definisi dari Sistem Informasi Manajemen atau disebut SIM ini dibagi dalam 3 kata yaitu : Sistem, Informasi, dan Manajemen.

1. **Sistem** adalah suatu susunan yang teratur dari kegiatan-kegiatan yang saling berkaitan dan susunan prosedur-prosedur yang saling berhubungan, yang melaksanakan dan mempermudah kegiatan-kegiatan utama organisasi/institusi.
2. **Informasi** adalah data yang telah diproses/diolah sehingga memiliki arti atau manfaat yang berguna
  - a. Data = fakta-fakta, angka atau statistik yang daripadanya dapat menghasilkan kesimpulan
3. **Manajemen** adalah sebagai **Proses**, kegiatan yang dilakukan untuk menyelesaikan suatu pekerjaan secara bersama-sama atau melibatkan orang lain demi mencapai tujuan yang sama

Sehingga dapat disimpulkan dari definisi diatas bahwa sistem informasi manajemen adalah sistem yang berbasis pada komputer yang menyediakan informasi bagi para pengguna yang memiliki kebutuhan yang sama (McLeod, Yulianto & Schell, 2007).

Tujuan dari mempelajari SIM ini adalah agar orang dapat memandang bahwa, nilai dari informasi amatlah berharga oleh karena itu harus dikelola dengan baik. Bila seseorang sebagai seorang wirausaha atau staff ataupun sebagai manager harus dapat menghargai dan mampu mengelola informasi bagi kemajuan perusahaan atau usahanya.

## 2.2 Harga Pokok Penjualan

### 2.2.1. Pengertian

Harga pokok penjualan adalah seluruh biaya yang dikeluarkan untuk memperoleh barang yang dijual atau harga perolehan dari barang yang dijual (Romney, 2006).

Ada dua manfaat dari harga pokok penjualan:

1. Sebagai patokan untuk menentukan harga jual.
2. Untuk mengetahui laba yang diinginkan perusahaan. Apabila harga jual lebih besar dari harga pokok penjualan maka akan diperoleh laba, dan sebaliknya apabila harga jual lebih rendah dari harga pokok penjualan akan diperoleh kerugian.

### 2.2.2. Rumus

Rumus harga pokok penjualan:

$$\text{HPP} = \text{Persediaan awal barang dagangan} + \text{pembelian bersih} - \text{persediaan akhir}$$

$$\text{HPP} = \text{Barang yang tersedia untuk dijual} - \text{persediaan akhir}$$

Gambar 2.1. Rumus HPP (Marshall, 2006).

### 2.2.3. *First In First Out (FIFO)*

Pengertian FIFO adalah barang yang masuk terlebih dahulu baru akan di jual terlebih dahulu.

Contoh perhitungan HPP menggunakan metode FIFO :

Soal :		
Persediaan awal :		
1 januari	20unit @Rp.500,-	Rp.10.000,-
Pembelian :		
10 januari	20unit @Rp.600,-	Rp.12.000,-
15 januari	20unit @Rp.700,-	Rp.14.000,-
20 januari	20unit @Rp.800,-	Rp.16.000,-
Total		Rp.52.000,-
Penjualan :		
13 januari	30unit	
16 januari	20unit	

Gambar 2. 2. Contoh Soal

Jawaban:		
13 januari =	20 x 500 =	Rp.10.000,-
	10 x 600 =	Rp. 6.000,-
16 januari -	10 x 600 =	Rp. 6.000,-
	10 x 700 =	Rp. 7.000,-
Persediaan akhir	10 x 700	= Rp. 7000,-
	20 x 800	= Rp.16.000,-
Total		= Rp.23.000,-
HPP = Rp.52.000 – Rp.23.000 = Rp.29.000,-		

Pada Gambar 2.3, menjawab pertanyaan pada Gambar 2.2. Persediaan awal ditambah pembelian akan menjadi barang yang tersedia untuk dijual, kemudian dikurangkan dengan persediaan akhir agar mendapatkan nilai HPP. Penjualan 13 januari 30 unit diambil dengan metode FIFO dimana barang pertama masuk akan dikeluarkan terlebih dahulu, jadi diambilkan 20 unit pada 1 januari dan 10 unit pada 10 januari. Untuk penjualan 16 januari 20 unit diambil dari 10 unit 10 januari dan 10 unit 15 januari. Maka dari perhitungan diatas akan mendapatkan nilai HPP sebesar Rp.29.000,-

### 2.3 Teknik Pengumpulan Data

Dalam teknik pengumpulan data ini dapat dibagi menjadi 2 macam yaitu data kualitatif dan data kuantitatif. Pada skripsi ini penulis menggunakan teknik pengumpulan data kualitatif. Data kualitatif adalah data yang tidak bisa diukur atau dinilai dengan angka secara langsung (Amirin, 2000). Penelitian kualitatif pada dasarnya merupakan suatu proses penyelidikan yang diambil dari kata-kata dan tindakan. Sedangkan data tertulis, foto dan statistik adalah data tambahan (Moleong, 2007).

#### 2.3.1 Wawancara

Wawancara atau interview dalam pengembangan sistem informasi telah banyak kali dilakukan karena telah diakui sebagai teknik pengumpulan data/fakta (*fact finding*). Teknik wawancara ini adalah salah satu teknik pengumpulan data secara tatap muka langsung, yaitu dimana *interviewer* (pewawancara) secara interaktif melakukan tanya jawab dengan pihak/orang yang diwawancarai. Oleh Kendall, merencanakan wawancara dibagi dalam beberapa tahap yaitu (Kendall & Kendall, 2008) :

- Membaca materi dari latar belakang masalah
- Menentukan/menetapkan tujuan dari wawancara
- Memutuskan siapa yang harus diwawancarai
- Mempersiapkan wawancara dengan baik
- Memutuskan jenis dan struktur pertanyaan

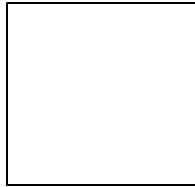
#### 2.4 Data Flow Diagram (DFD)

Data Flow Diagram adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu dengan yang lain dengan alur data. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks daripada data yang dimanipulasi oleh sistem. Dengan kata lainnya, DFD adalah alat pembuatan model yang memberikan penekanan pada fungsi sistem (Parno, 2009). Suatu *data flow diagram* memiliki 4 elemen dasar, yaitu :

- *Data sources and destinations*
- *Data flows*
- *Transformations processes*
- *Data stores.*

#### **2.4.1 Data Sources dan Destinations**

Suatu organisasi atau individu yang mana mengirim atau menerima data dari suatu sistem. Tujuan atau sumbernya merupakan sebuah entity. Lambang dari *Data sources* atau *data destination* dilambangkan dengan gambar persegi. Contoh gambar dapat dilihat pada Gambar 2.1

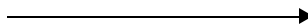


Gambar 2.4 Simbol Data Sources dan Destinations

Sumber : Romney & Steinbart (2009, p. 86)

#### **2.4.2 Data Flows**

Aliran keluar atau masuk data pada sebuah proses dilambangkan dengan kurva atau garis lurus dengan panah disalah satu ujungnya. Contoh gambar terlihat pada Gambar 2.2.

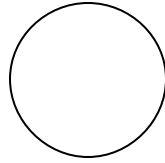


Gambar 2.5 Simbol dari *Data Flows*

Sumber : Romney & Steinbart (2009, p. 86)

#### **2.4.3 Transformation Processes**

Proses yang melakukan transformasi data dari *inputs* ke *outputs*. *Transformations processes* ini sering dilambangkan dengan gambar lingkaran. Contoh gambar seperti terlihat pada Gambar 2.3.

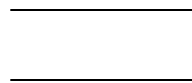


Gambar 2.6 Simbol dari *Transformation Processes*

Sumber : Romney & Steinbart (2009, p. 86)

#### 2.4.4 *Data Stores*

*Data Stores* yang berguna sebagai penyimpanan data ini dilambangkan dengan 2 garis datar atau horizontal. Contoh gambar terlihat pada Gambar 2.4.



Gambar 2.7. Simbol dari *Data Stores*

Sumber : Romney & Steinbart (2009, p. 86)




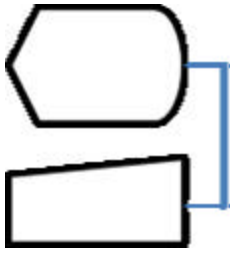




#### 2.5. *Document Flowchart*

*Document flowchart* ini adalah ilustrasi arus dan informasi pada sejumlah *sector* dalam suatu perusahaan. Kegunaan atau fungsi dari *Document flowchart* adalah untuk mengikuti jejak sebuah dokumentasi dari awal sampai pada akhirnya. Gambar dan penjelasan dari *document flowchart* dapat dilihat pada Tabel 2.1 (Romney & Steinbart, 2006) :



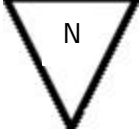

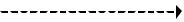

Tabel 2.2 Notasi-notasi pada *Document Flowchart*

Simbol	Nama	Keterangan
	Input / output	Simbol <i>input</i> atau <i>output</i>
	<i>Document</i>	Simbol sebuah dokumen.





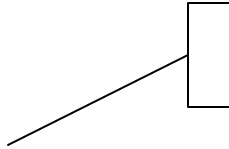
Tabel 2.2 Notasi-notasi pada *Document Flowchart* (sambungan)

	<p><i>Multiple copy of one document</i></p>	<p>Simbol kumpulan rangkap dokumen</p>
	<p><i>Display</i></p>	<p>Simbol informasi ditampilkan melalui sebuah peralatan <i>online output</i> seperti monitor dan lain-lain</p>
  	<p><i>Online keying</i></p>  <p><i>Personal computer</i></p>	<p>Simbol <i>data entry</i> melalui <i>online device</i> seperti PC dan lain-lain.</p>  <p>Simbol <i>display</i> dan <i>online keying</i> adalah digunakan bersamaan untuk melambangkan <i>terminal</i> atau PC.</p>
	<p><i>Computer processing</i></p>	<p>Simbol sebuah bentuk fungsi proses computer</p>
	<p><i>Manual operation</i></p>	<p>Simbol sebuah wujud proses operasi secara <i>manual</i></p>
	<p><i>Auxiliary operation</i></p>	<p>Simbol wujud fungsi proses dengan menggunakan <i>device</i> tetapi bukan komputer</p>
	<p><i>Magnetic disk</i></p>	<p>Simbol penyimpanan data secara permanen dalam sebuah <i>magnetic disk</i></p>

Tabel 2.2 Notasi-notasi pada *Document Flowchart* (sambungan)

 	<p><i>Magnetic file</i></p> <p><i>On-line storage</i></p>	<p>Simbol penyimpanan dalam sebuah <i>magnetic tape</i></p> <p>Simbol penyimpanan data dalam <i>file online</i> sementara pada suatu medium akses langsung seperti disket</p>
	<p><i>File</i></p>	<p>Simbol dokumen-dokumen file pada penyimpanan manual, tulisan yang terdapat pada simbol merupakan cara mengarsip file  N=numerically,  A=alphabetically,  D=by date</p>
	<p><i>Document or processing flow</i></p>	<p>Simbol petunjuk proses atau aliran dokumen</p>
	<p><i>Data/information flow</i></p>	<p>Simbol petunjuk aliran data atau informasi</p>
	<p><i>Communication link</i></p>	<p>Simbol transmisi data dari 1 lokasi ke lokasi lain melalui jalur komunikasi</p>

Tabel 2.2 Notasi-notasi pada *Document Flowchart* (sambungan)

	<i>On-page connector</i>	Simbol penghubung aliran proses pada halaman yang sama
	<i>Off-page connector</i>	Simbol sebuah entri dari, atau keluar menuju pada halaman berbeda.
	<i>Terminal</i>	Simbol sebuah awal, akhir, atau titik perhentian dalam sebuah proses
	<i>Decision</i>	Simbol langkah pengambilan keputusan
	<i>Annotation</i>	Simbol deskripsi komentar tambahan.

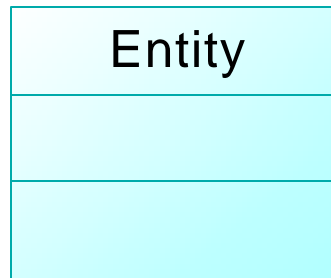
Sumber : Romney & Steinbart (2009, p.93)

## 2.6. *Entity Relationship Diagram (ERD)*

*Entity Relationship* adalah salah satu metode permodelan basis data yang digunakan untuk menghasilkan skema konseptual. Model *Entity Relationship* merupakan alat modeling data yang paling populer dan juga banyak digunakan oleh para perancang database. Tujuan dari permodelan data adalah untuk menyajikan data dan menjadikan data untuk dapat mudah dimengerti, sehingga mempermudah untuk pengaksesan dan perancangan dalam database. Dari tipe konsepnya, data model dapat dibagi menjadi dua bagian kategori yaitu Conceptual Data Model (High Level) dan Physical Data Model (Low Level).

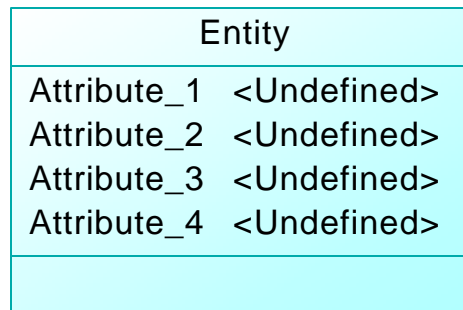
Pada Conceptual Data Model terdapat konsep yang berkaitan dengan pandangan dari pemakai terhadap data, sedangkan dalam Physical Data Model merupakan konsep yang menerangkan secara detail dari bagaimana data tersebut dapat disimpan di dalam komputer. Dengan demikian dari pandangan ini kita dapat menyimpulkan bahwa model *Entity Relationship* digunakan untuk menggambarkan Conceptual Data Model (E-R). ERD memiliki 3 notasi yaitu (Anthony & Lukmanto, 2007) :

- *Entity* merupakan sesuatu tentang yang mana proses bisnis butuhkan sebagai tempat untuk penyimpanan data. Contoh gambar *entity* pada Gambar 2.5.



Gambar 2.8 *Entity*

- *Atribut* merupakan penjelasan dari sifat atau karakteristik pada sebuah *entity*. Contoh gambar terlihat pada Gambar 2.6.



Gambar 2.9 Atribut

- Relasi merupakan asosiasi bisnis yang sedang berjalan yang ada antara satu atau lebih entitas. Hubungan dapat mewakili suatu peristiwa yang menghubungkan entitas atau hanya logis kedekatan yang ada diantara entitas. Contoh gambar terlihat pada Gambar 2.7.



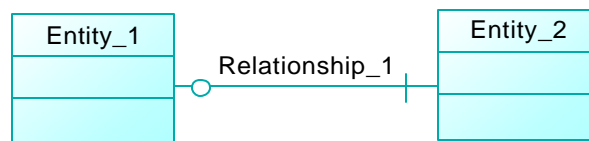
Gambar 2.10 Relasi

## 2.6.1 Jenis-jenis Relasi yang terdapat pada ERD

### 2.6.1.1 Cardinality Ratio Constraint

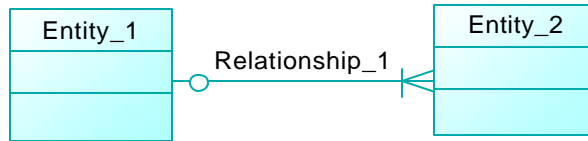
*Cardinality* berfungsi untuk menjelaskan jumlah hubungan/relasi dari *entity-entity* yang berpartisipasi. Terdapat 3 macam CRC yaitu :

- Hubungan 1 : 1 (*One to One Relationship*)  
Relasi ini merupakan relasi dimana tiap elemen pada *entity\_1* berelasi hanya pada 1 elemen pada *entity\_2*, begitu juga sebaliknya. Gambar 2.8 menunjukkan relasi *one to one*.

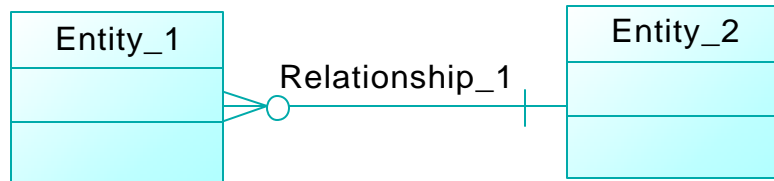


Gambar 2.11. Relasi *One to One*

- Hubungan 1 : M (*One to Many/Many to One Relationship*)  
Relasi ini merupakan relasi dimana tiap elemen pada *entity\_1* berelasi banyak dengan elemen pada *entity\_2*, tetapi pada *entity\_2* hanya berelasi dengan satu elemen pada *entity\_1*, begitu juga sebaliknya.



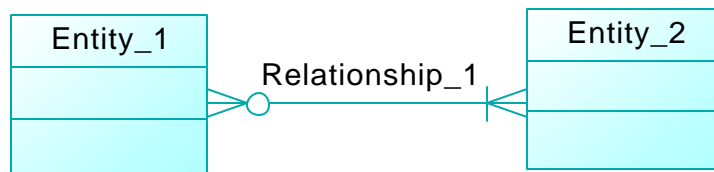
Gambar 2.12. Relasi *One to Many*



Gambar 2.13. Relasi *Many to One*

- Hubungan M : N (*Many to Many Relationship*)

Relasi ini merupakan relasi dimana elemen pada entity\_1 dapat berhubungan dengan banyak elemen pada entity\_2, begitu juga sebaliknya.



Gambar 2.14. Relasi *Many to Many*

### 2.6.1.2 *Participation Constraint*

*Participation* berfungsi untuk menjelaskan keberadaan suatu *entity* yang tergantung dengan *entity* yang lain. Terdapat 2 macam *Participation Constraint* yaitu :

- *Total Participation*

Yaitu keberadaan suatu *entity* yang tergantung pada *entity* lainnya. Setiap *entity* pada set suatu *entity* yang lain diharuskan mempunyai setidaknya 1 relasi. Contoh *Total Participation* dapat dilihat pada Gambar 2.8, Gambar 2.9, Gambar 2.10, Gambar 2.11 pada *entity\_2*.

- *Partial Participation*

Yaitu dimana keberadaan suatu *entity* tidak tergantung pada *entity* lain. Jadi, hubungan setiap *entity* pada set suatu *entity* yang lain tidak mengharuskan memiliki relasi. Contoh *Partial Participation* dapat dilihat pada Gambar 2.8, Gambar 2.9, Gambar 2.10, Gambar 2.11 pada *entity\_1*.

Juga ada yang disebut dengan *Weak Entity* yaitu suatu *entity* yang mungkin memiliki suatu atribut yang bukan miliknya. Dimana keberadaannya tergantung pada *entity* lainnya. Juga *Weak Entity* adalah *entity* yang tidak memiliki *primary key*. Contoh *Weak Entity* dapat dilihat pada Gambar 2.12.



Gambar 2.15. Relasi *Weak Entity*

## 2.7 *PHP Hypertext Preprocessor (PHP)*

*PHP* adalah bahasa pemrograman yang paling banyak dipakai saat ini. *PHP* banyak dipakai untuk membuat web dinamis walaupun tidak tertutup kemungkinan untuk pemakaian lain. Walaupun *PHP* merupakan salah satu bahasa *script* yang terbilang baru dan tersedia secara bebas dan masih memungkinkan untuk dikembangkan lebih lanjut, *PHP* dapat di-integrasikan ke dalam web server, atau dapat berperan sebagai program CGI terpisah (Welling, 2001).

Ada beberapa kelebihan *PHP* dari bahasa pemrograman lain, antara lain :

- Bahasa pemrograman *PHP* adalah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- Web server yang mendukung *PHP* dapat ditemukan dimana-mana, mulai dari apache, IIS, hingga Xitami dengan konfigurasi yang relatif mudah.
- Dalam sisi pengembangannya lebih mudah, karena banyak developer yang siap membantu dalam pengembangan.

- Dalam sisi pemahaman, *PHP* adalah scripting yang paling mudah karena memiliki referensi yang banyak.
- *PHP* adalah bahasa open source yang dapat diberbagai mesin (Linux, Unix, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

Hampir seluruh aplikasi berbasis web dapat dibuat dengan *PHP*, namun fungsi *PHP* yang paling utama adalah untuk menghubungkan database dengan web. Dengan *PHP*, membuat aplikasi web yang terkoneksi dengan database menjadi sangat mudah. Cara penulisan dengan menggunakan *PHP* ada 4 cara yaitu :

- I. `<? echo ("ini adalah script PHP\n"); ?>`
- II. `<?php echo("ini juga\n"); ?>`
- III. `<script language="php">`  
`echo ("tulis pake ini jika html editor Anda tidak mengenali PHP");`  
`</script>`
- IV. `<% echo ("kalau yang ini mirip dengan ASP"); %>`

## 2.8 Database pada *PHPMYAdmin*

### 2.8.1 Database

*Database* adalah kumpulan dari beberapa data dalam jumlah yang banyak, saling berhubungan dan mempunyai arti tertentu. *Database* terdiri dari kumpulan tabel yang berisi baris dan kolom. Tiap baris dan tabel mewakili satu unit data yang disebut dengan *record*. Juga ada kolom dalam tabel tersebut. Kolom itu disebut dengan *field* yang merupakan keterangan dari masing-masing *record*.

Pemrograman *database* sudah banyak sekali dilakukan yaitu dengan menggunakan bahasa yang disebut SQL (*Structured Query Language*). Juga selain SQL, pemrograman *database* meliputi juga objek-objek dari *database*, analisa *query*, dan juga interaksi dengan *Open Database Connectivity* (ODBC).

Ada juga *tool* yang dapat membantu kita untuk dapat membuat database dengan mudah dan cepat dalam melakukan pemrograman dalam *database*. Salah satunya adalah *PHPMYAdmin*. Tool ini merupakan salah satu paket yang disediakan oleh PHPTriad, jadi apabila anda menginstal PHPTriad, maka dengan

otomatis komputer anda juga akan terinstal *PHPMyAdmin*. Setelah *PHPTriad* diinstal, anda harus mengaktifkan dulu *Apache Server* dan *MySQL* server-nya. Selain *PHPTriad*, pilihan lainnya kita dapat menggunakan *XAMPP* (Michael, 2005).

### **2.8.2 MySQL**

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database manajemen system*) atau DBMS yang *multithread*, *multuser*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus lain. *MySQL* dikatakan *multithread* yaitu karena dapat melakukan banyak eksekusi perintah *query* dalam satu permintaan, baik itu menerima dan mengirimkan data. *MySQL* juga dikatakan *multi-user* dalam arti dapat dipergunakan oleh banyak pengguna dalam waktu yang bersamaan sekaligus. Penggunaan *MySQL* yang merupakan sebuah database server sekaligus dapat sebagai *client*, dan dapat berjalan di multi-OS (*Operating System*) juga memiliki keunggulan lain seperti *open source* sehingga penggunaanya tidak perlu membayar *licensi* kepada pembuatnya. Lalu *MySQL* juga dapat mendukung database dengan kapasitas yang besar. Selain itu merupakan *Database Management System* (DBMS) yang mudah digunakan. Bahasa pemrograman yang dapat digunakan untuk mengakses *MySQL* diantaranya yaitu dengan menggunakan C, C++, Perl, PHP, Java, Python (Michael, 2005).

### **2.8.3 PHPMyAdmin**

*PHPMyAdmin* adalah perangkat lunak bebas yang ditulis dalam pemrograman PHP yang digunakan untuk menangani administrasi *MySQL* melalui *World Wide Web* (www).*PHPMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya adalah mengelola basis data, tabel-tabel, *fields*, *relations*, *indeks*, *users*, *permissions*, dan lain-lainnya.

Pada dasarnya, untuk mengelola basis data dengan *MySQL* harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai atau biasanya dikenal dengan *command line* untuk setiap maksud tertentu. Jika seseorang ingin

membuat database, mereka harus mengetikkan baris perintah yang sesuai. Jika seorang menghapus sebuah tabel, ia harus mengetikkan perintah yang sesuai untuk menghapus tabel tersebut. Hal tersebut tentunya sangat menyulitkan sekali, karena seseorang harus hafal dan mengetikkan perintahnya satu-persatu. Saat ini banyak sekali *software* yang dapat dimanfaatkan untuk mengelola basis data dalam *MySQL*, salah satunya adalah dengan menggunakan *PHPMysqlAdmin*. Dengan *PHPMysqlAdmin*, seseorang dapat membuat database, tabel, mengisi data, dan melakukan perintah-perintah lain dengan mudah tanpa harus menghafalkan barisnya.

*PHPMysqlAdmin* bersifat open source yang ditulis dalam PHP yang digunakan untuk menangani administrasi dalam basis data *MySQL* yang dapat diakses melalui *web browser* (*internet explore*, *firefox*, *opera*, dan lain-lain). Fasilitas yang disediakan pada *PHPMysqlAdmin* yaitu dapat membuat dan menghapus database, membuat dan menambah tabel, menghapus, mengedit dan menambah *field*, melakukan berbagai macam perintah SQL, mengatur *key* pada *field*, mengatur *privileges*, mengekspor data ke berbagai format (Michael, 2005).

#### **2.8.4 XAMPP**

XAMPP adalah perangkat lunak bebas/gratis yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program untuk menjalankan fungsinya sebagai server yang berdiri sendiri, dan terdiri atas beberapa program. XAMPP terdiri dari singkatan dari tiap-tiap hurufnya :

- X : Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS.
- A : Apache, merupakan aplikasi web server-nya. Tugas utama dari Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan pada kode PHP yang dituliskan oleh pembuat pada halaman web.
- M : *MySQL*, merupakan aplikasi database servernya.
- P : PHP, bahasa pemrograman web yang dipakai.
- P : Perl, bahasa pemrograman.

XAMPP merupakan aplikasi yang mudah digunakan dan dapat mampu melayani halaman dinamis. Saat ini XAMPP tersedia untuk sistem operasi Microsoft Windows, Linux, Sun Solaris dan Mac OS X (Michael, 2005).