

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Informasi Geografis**

*Geographic Information System* (GIS), merupakan suatu sistem (berbasis komputer) yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena-fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. SIG merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis :

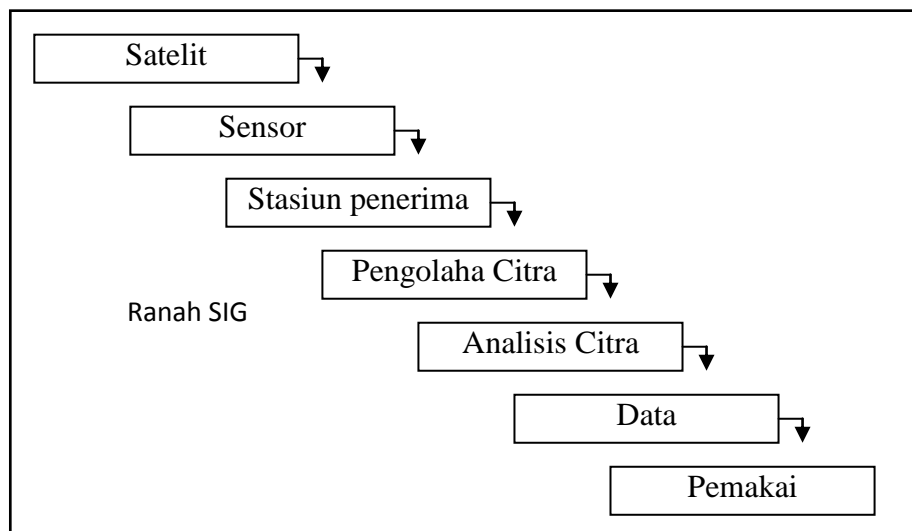
- a. Masukan
- b. Keluaran
- c. Manajemen data (penyimpanan dan pemanggilan data)
- d. Analisis dan manipulasi data (Eddy Prahasta; 2009: 1)

Dalam SIG terdapat berbagai peran dari berbagai unsur, baik manusia sebagai ahli dan sekaligus operator, perangkat alat (lunak / keras) maupun objek permasalahan. SIG adalah sebuah rangkaian sistem yang memanfaatkan teknologi digital untuk melakukan analisis spasial. Sistem ini memanfaatkan perangkat keras dan lunak komputer untuk melakukan pengolahan data seperti:

1. Perolehan dan verifikasi
2. Kompilasi
3. Penyimpanan

4. Pembaruan dan perubahan
5. Manajemen dan pertukaran
6. Manipulasi
7. Penyajian
8. Analisis

Pemanfaatan SIG secara terpadu dalam sistem pengolahan citra digital adalah untuk memperbaiki hasil klasifikasi. Dengan demikian, peranan teknologi SIG dapat diterapkan pada operasionalisasi penginderaan jauh satelit. Pengembangan teknologi penginderaan jauh satelit dapat digambarkan dalam diagram sebagai berikut:

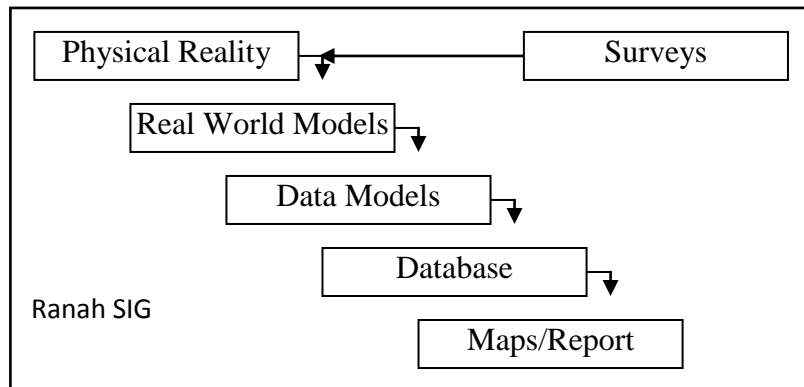


**Gambar II.1 GIS dalam sistem digital satelit**

*Sumber : Eko Budiyanto (2005 : 4)*

Mengingat sumber data sebagian besar berasal dari data penginderaan jauh baik satelit maupun terrestrial terdigitasi, maka teknologi sistem informasi geografi (SIG) erat kaitannya dengan teknologi penginderaan jauh. Namun demikian, penginderaan jauh bukanlah satu-satunya ilmu pendukung bagi sistem

ini. Sumber data lain berasal dari hasil survey terrestrial (uji lapangan) dan data-data sekunder lain seperti sensus, catatan, dan laporan yang terpercaya. Secara diagram hal tersebut dapat digambarkan sebagai berikut:



**Gambar II.2 Sistem kerja SIG**

*Sumber : Eko Budiyanto (2005 : 4)*

Data spasial dari penginderaan jauh dan survey terrestrial tersimpan dalam basisdata yang memanfaatkan teknologi komputer digital untuk pengolahan dan pengambilan keputusannya.

Secara teknis SIG mengorganisasikan dan memanfaatkan data dari peta digital yang tersimpan dalam basisdata. Dalam SIG, dunia nyata dijabarkan dalam peta digital yang menggambarkan posisi dari ruang (*space*) dan klasifikasi, atribut data, dan hubungan antar item data. Kerincian data dalam SIG ditentukan oleh besarnya satuan pemetaan terkecil yang dihimpun dalam basisdata. Dalam bahasa pemetaan kerincian itu tergantung dari skala peta dan dasar acuan geografis yang disebut sebagai peta dasar. Dari dunia nyata diambil tiga hal penting seperti diuraikan di atas, yaitu posisi dan klasifikasi, atribut, serta hubungan antar item tersebut. Ketiga hal tersebut diolah sebagai dasar analisis sistem spasial dalam SIG (Eko Budiyanto; 2005 : 3-5).

### **II.1.1. Media Transmisi**

Media transmisi adalah media yang menghubungkan antara pengirim dan penerima informasi (data), karena jarak yang jauh, maka data terlebih dahulu diubah menjadi kode/isyarat, dan isyarat inilah yang akan dimanipulasi dengan berbagai macam cara untuk diubah kembali menjadi data. Media transmisi digunakan pada beberapa peralatan elektronika untuk menghubungkan antara pengirim dan penerima supaya dapat melakukan pertukaran data. Beberapa alat elektronika, seperti telepon, komputer, televisi, dan radio membutuhkan media transmisi untuk dapat menerima data. Seperti pada pesawat telepon, media transmisi yang digunakan untuk menghubungkan dua buah telepon adalah kabel. Setiap peralatan elektronika memiliki media transmisi yang berbeda-beda dalam pengiriman datanya.

### **II.2. ArcView**

*ArcView* merupakan salah satu perangkat lunak dekstop Sistem Informasi Geografis dan pemetaan yang telah dikembangkan oleh ESRI (*Environmental Systems Research Institute*). Dengan *ArcView*, pengguna dapat memiliki kemampuan-kemampuan untuk melakukan visualisasi, meng-explore, menjawab query (baik basisdata spasial maupun *non-spasial*), menganalisis data secara geografis, dan sebagainya (Eddy Prahasta ; 2009 : 1).

### **II.3. MapServer**

Map server adalah aplikasi *freeware* dan *open source* yang memungkinkan kita menampilkan data spasial (peta) di *web*. Aplikasi ini pertama kali

dikembangkan di Universitas Minnesota, Amerika Serikat untuk proyek *ForNet* (sebuah proyek untuk manajemen sumber daya alam) yang disponsori NASA (*National Aeronautics And Space Administration*). Support NASA dilanjutkan dengan proyek *TerraSIP* untuk manajemen data lahan. Saat ini, karena sifatnya yang terbuka (*open source*), pengembangan MapServer dilakukan oleh pengembangan dari berbagai Negara.

Pengembangan MapServer menggunakan berbagai aplikasi *open source* atau *freeware* seperti Shapelib (<http://shapelib.maptools.org>) untuk baca/tulis format data *Shapefile*, FreeType (<http://www.freetype.org>) untuk merender karakter, GDAL/OGR (<http://www.remotesensing.org/gdal>) untuk baca/tulis berbagai format data *vektor* maupun juga data *raster*, dan Proj.4 (<http://www.remotesensing.org/proj>) untuk menangani beragam proyeksi peta.

Pada bentuk paling dasar, Mapserver berupa sebuah program CGI (*Common Gateway Interface*). Program tersebut akan dieksekusi di *web server*, dan berdasarkan beberapa parameter tertentu (terutama konfigurasi dalam bentuk file \*.MAP) akan menghasilkan data yang kemudian akan dikirim ke *web browser*, baik dalam bentuk gambar peta ataupun bentuk lain (Ruslan Nuryadin; 2005: 3).

#### **II.4. 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.1.

**Tabel II.1 Tipe Diagram UML**

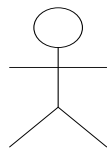
<b>Diagram</b>	<b>Tujuan</b>	<b>Keterangan</b>
Activity	Prilaku prosedural dan paralel	Sudah ada di UML 1
Class	Class, fitur dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara objek. Lebih menekankan kepada link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	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 : Munawar (2005 : 23)*

## II.4.1. 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 beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Berikut notasi actor dalam UML:



**Gambar II.3 : Notasi Actor pada UML**

*Sumber : 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:



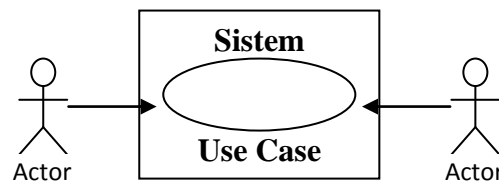
**Gambar II.4 : Notasi Class di UML**

*Sumber : Munawar (2005 : 35)*

### 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 *system* 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.3 :



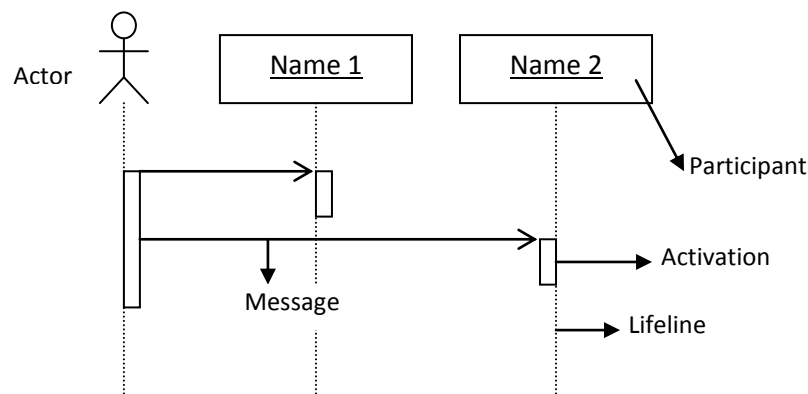
**Gambar II.5 : Notasi *Use Case* pada UML**

*Sumber : Munawar (2005 : 64)*

#### 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.6 : Simbol-simbol yang ada pada *Sequence Diagram***








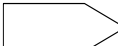
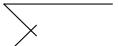

*Sumber : Munawar (2005 : 89)*

## 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.

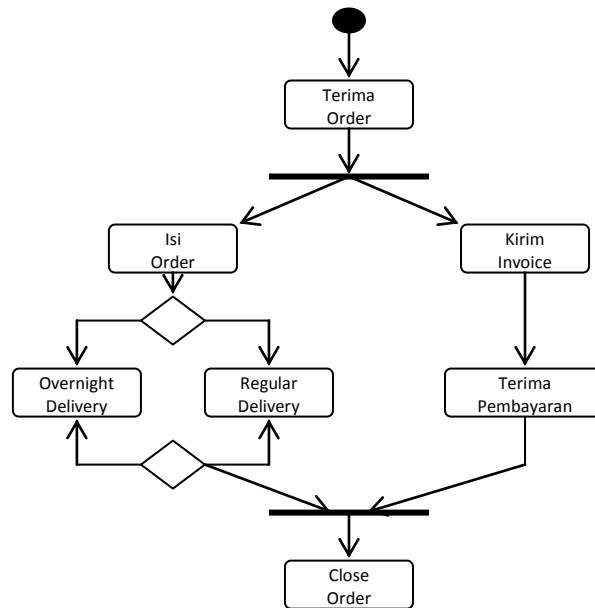
Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan activity diagram.

**Tabel II.2 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 :Munawar (2005 : 109)

Adapun contoh dari Activity Diagram dapat di lihat pada Gambar II.8.



**Gambar II.7 : Contoh Activity Diagram Sederhana**

*Sumber : Munawar (2005 : 111)*

## II.5. 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; 2006: 99).

### II.5.1. 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 ; 2006: 102).

### II.6. 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.3 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 : Jogiyanto (2005: 730)*

## II.7. Entity Relationship Diagram – ERD

### II.7.1. 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.7.2. 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.

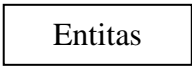
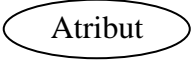


Entitas digambarkan dalam basisdata dengan kumpula atribut. Misalnya atribut nim, nama, alamat dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

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.4 Notasi ERD (Entity Relationship Diagram)**

Simbol	Arti Keterangan
	Persegi panjang mewakili kumpulan entitas
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

*Sumber : Janner Simarmata & Imam Prayudi (2006: 59)*

## **II.8. 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.

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 (Janner Simarmata & Imam Prayudi; 2006: 77).

### **II.9. 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).

### **II.10. 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 datan yang ada didalamnya (Sumber: Bunafit Nugroho; 2009: 13).