

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Geografis

Menurut (Endah Dharmaputri) Era komputerisasi telah membuka wawasan dan paradigma baru dalam proses pengambilan keputusan dan penyebaran informasi. Data yang merepresentasikan “dunia nyata” dapat disimpan dan diproses sedemikian rupa sehingga dapat disajikan dalam bentuk-bentuk yang lebih sederhana dan sesuai kebutuhan. Sejak pertengahan tahun 1970-an, telah dikembangkan sistem-sistem yang secara khusus dibuat untuk menangani masalah informasi yang bereferensi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini mencakup :

1. Pengorganisasian data dan informasi.
2. Penempatan informasi pada lokasi tertentu
3. Melakukan komputasi memberikan ilustrasi keterhubungan informasi, beserta analisa spesial lainnya.

II.1.1. Subsistem SIG

Sistem Informasi Geografis dapat diuraikan menjadi beberapa subsistem yaitu:

1. *Data Input*

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spesial dan atribut dari berbagai sumber. Subsistem ini pula yang

bertanggung jawab dalam mengkonversi atau mentransformasikan format-format data aslinya ke dalam format-format yang digunakan oleh SIG.

2. *Data Output*

Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data seperti tabel grafik, peta dan lain-lain.

3. Manajemen Data

Subsistem ini mengorganisasikan baik data spasial maupun atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, diperbaharui dan diperbaiki.

4. Analisis dan Manipulasi Data

Sub sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

II.1.2. Komponen Sistem Informasi Geografis

Menurut (Endah Dharmaputri) Secara umum, Sistem Informasi Geografis bekerja berdasarkan integrasi komponen, yaitu: *Hardware*, *Software*, Data Manusia dan Metode, kelima komponen tersebut dapat dijelaskan sebagai berikut:

1. *Hardware*

Sistem Informasi Geografis memerlukan spesifikasi komponen *hardware* yang sedikit lebih tinggi dibanding spesifikasi komponen sistem informasi lainnya. Hal tersebut disebabkan karena data-data yang

digunakan dalam SIG, penyimpanannya membutuhkan ruang yang besar dan dalam proses analisisnya membutuhkan *memory* yang besar dan *processor* yang cepat. Beberapa *Hardware* yang sering digunakan dalam Sistem Informasi Geografis adalah *Personal Computer (PC)*, *Mouse*, *Digitizer*, *Printer*, *Plotter* dan *Scanner*.

2. *Software*

Sebuah *Software* SIG haruslah menyediakan fungsi dan *tool* yang mampu melakukan penyimpanan data, analisis dan menampilkan informasi geografis. Dengan demikian elemen yang harus terdapat dalam komponen *software* SIG adalah:

- a. *Tools* untuk melakukan *input* dan transformasi data geografis
- b. Sistem Manajemen Basis Data .
- c. *Tools* yang mendukung *query* geografis, analisis dan visualisasi.
- d. *Geographical User Interface (GUI)* untuk memudahkan akses pada *tool* geografi.

3. Data

Hal merupakan komponen penting dalam SIG adalah data. Secara fundamental, SIG bekerja dengan 2 tipe model data geografis, yaitu model data *vector* dan model data *raster*. Dalam model data *vector*, informasi posisi point, garis dan polygon disimpan dalam bentuk koordinat x,y. Bentuk garis seperti jalan dan sungai dideskripsikan sebagai kumpulan dari koordinat-koordinat point. Bentuk *polygon*, seperti daerah penjualan disimpan sebagai pengulangan koordinat yang tertutup. Data raster terdiri

dari sekumpulan *grid* atau sel seperti peta hasil *scanning* maupun gambar atau *image* . Masing-masing *grid* memiliki nilai tertentu yang bergantung pada bagaimana *image* tersebut digambarkan.

4. Manusia

Komponen manusia memegang peranan yang sangat menentukan, karena tanpa manusia maka sistem tersebut tidak dapat diaplikasikan dengan baik. Jadi manusia menjadi komponen yang mengendalikan suatu sistem sehingga menghasilkan suatu analisa yang dibutuhkan.

5. Metode

SIG yang baik memiliki keserasian antara rencana desain yang baik dan aturan dunia nyata, dimana metode, model dan implementasi akan berbeda untuk setiap permasalahan.

II.1.3. *Quantum GIS*

Menurut (Endah Dharmaputri) *Quantum GIS* adalah aplikasi SIG gratis yang mencakup pemetaan, analisis spesial dan beberapa fitur *Desktop GIS* lainnya. Aplikasi ini sama dengan paket aplikasi GIS komersial namun aplikasi ini di distribusikan secara gratis dibawah lisensi GNU, *Quantum GIS* mendukung format data *vector*, *raster* dan *database (post GIS dan Oracle)*. *Quantum GIS* juga dapat deprogram ulang untuk mengerjakan tugas yang berbeda atau lebih spesifik.

Aplikasi ini juga merupakan suatu aplikasi *multi-flatform* yang dapat dijalankan pada sistem operasi yang berbeda termasuk *MacOS X*, *Linux*, *Unix*, *Windows XP*.

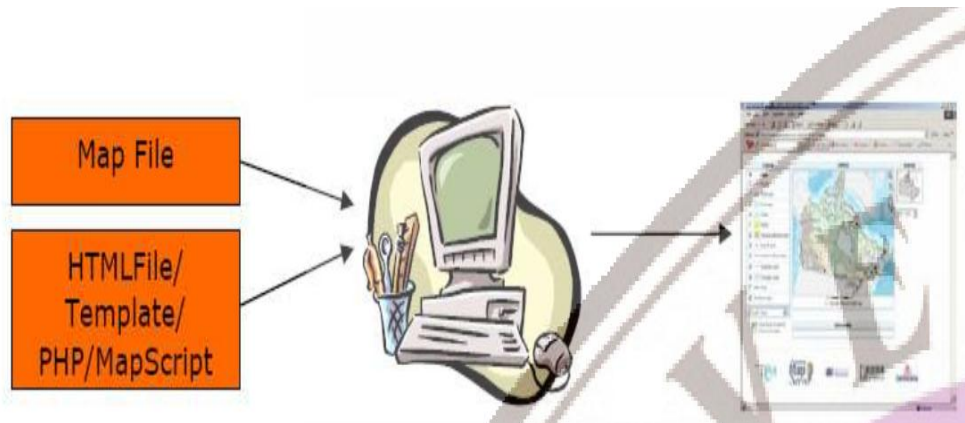
II.1.4. Mapserver

Menurut (Endah Dharmaputri) *Software* digunakan dalam perancangan SIG ini adalah *MapServer*. MS4W (*MapServer for Windows*) adalah paket instalasi *MapServer* untuk *platform Windows*. Dimana *MapServer* (<http://mapserver.gis.unm.edu>) merupakan aplikasi *freeware* dan *Open source* untuk dapat menampilkan SIG di *web*. MS4W dilengkapi dengan berbagai modul tambahan (optional) yang mempermudah kita membangun dan mengadministrasi system WebGIS.

Saat ini, selain dapat mengakses *MapServer* sebagai program CGI, *MapServer* juga dapat diakses sebagai modul *MapScript*, melalui berbagai bahasa pemrograman, seperti *PHP*, *Perl*, *Python*, *Java* dan lain sebagainya. Akses fungsi-fungsi *MapServer* melalui skrip akan lebih memudahkan pengembangan aplikasi WebGIS.

Untuk menjalankan dan menampilkan peta yang dihasilkan oleh *MapServer*, diperlukan dua *file* yaitu *Map File* dan *HTML File*. *Map File* berisikan konfigurasi penyajian peta. yang ditulis dalam bahasa dan sintaks tersendiri. Informasi ini kemudian diolah dan disajikan oleh program *MapServer*. Sedangkan *file HTML* digunakan untuk melakukan format penyajian hasil (peta). Gambar II.4 menyajikan proses penyajian peta. *File HTML* dapat

berupa *HTML* biasa atau *template* yang disisipi sintaks *MapServer* atau *file HTML* yang disisipi *PHP/ MapScript*.



Gambar II.1. Proses Penyajian Peta oleh MapServer
(Sumber: Endah Dharmaputeri; 2010 :5)

Sebelum membuat aplikasi *WebGIS* menggunakan *MapServer*, hal yang harus diperhatikan adalah asitektur penyimpanan *file mapServer* dan data *SIG*. Secara umum ada tiga kategori data yang dimiliki yaitu:

- a. *File MapServer Map file* dan *PHP/MapScript*.
- b. *File HTML* dan gambar/grafis *File web* dan gambar yang disertakan.
- c. *Data vector* dan citra (*raster*) yang digunakan.

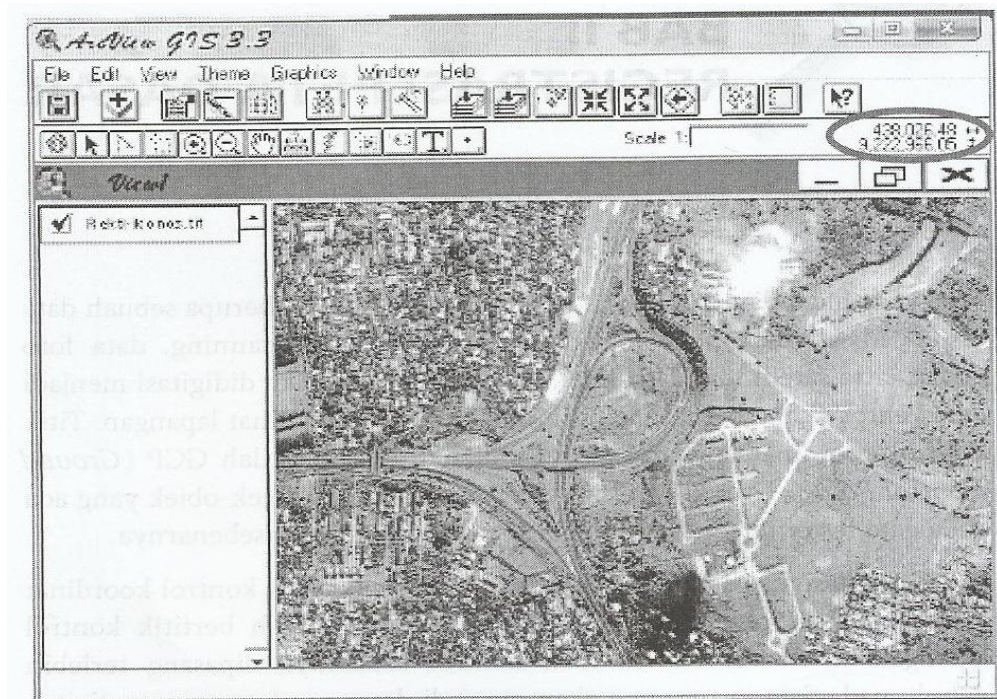
II.2. Registrasi peta dasar

Menurut (Eko Budiyanto;2010:7) Peta yang akan dijadikan dasar pembuatan peta dapat berupa sebuah data *raster*. Misalnya , data peta aalog (tercetak) yang di-*scanning*, dta foto udara , citra satelit, dan lain-lain. Data *raster* yang akan digitasi menjadi data *vector* sering belum memiliki nilai *control* koordinat lapangan. Titik control koordinat lapangan sering dikenal dengan istilah

GCP (*Ground Control Point*). Titik koordinat akan “ meletakkan “ objek-objek yang ada pada data *raster* tersebut pada koordianat lapangan yang sebenarnya.

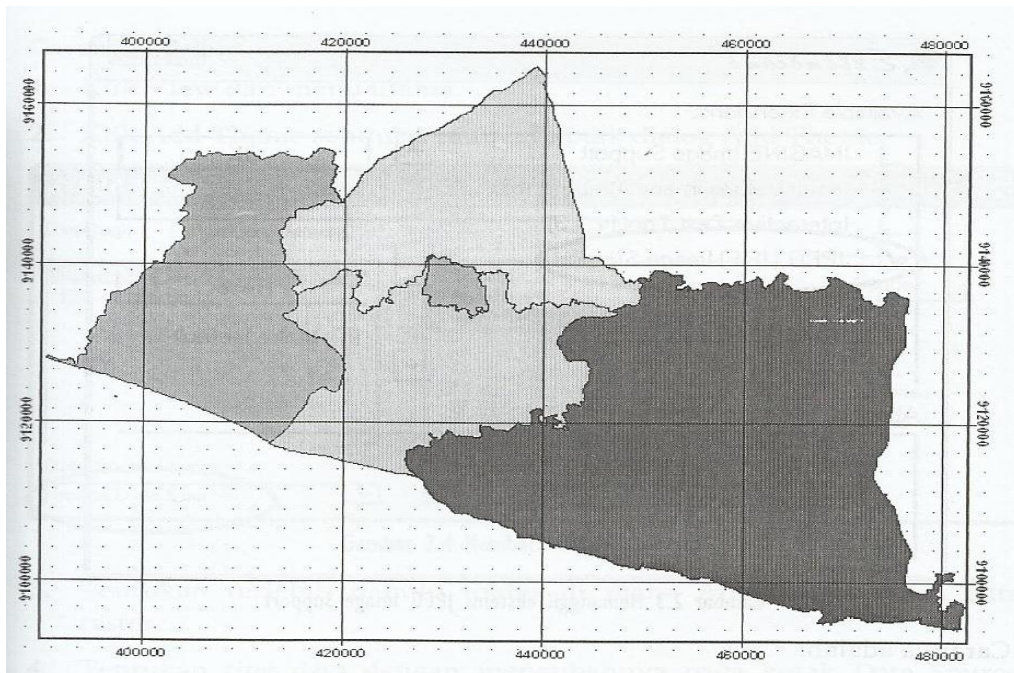
Proses digitasi atas data raster yang memiliki titik control titik koordinat lapangan akan menghasilkan data digitasi yang sudah bertitik *control* lapangan. Oleh karena itu titik kontrol sebaiknya dipasang terlebih dahulu pada data *raster* yang akan menjadi dasar pemnbuatan peta digital. Dalam beberapa perangkat lunak GIS lain, titik *control* dapat dipasang setelah data terdigitasi melalui proses transformasi.

Pemasangan titik *control* koordinat lapangan pada data *raster* disebut sebagai *registrasi data raster*. Proses registrasi ini menyebabkan data *raster* itu memiliki koordinat lapangan. Objek-objek yang termuat dalam data raster tersebut dapat dilihat posisi geografis nya dengan mengarahkan *pointer* pada objek. Koordinat akan ditunjukkan oleh *Arcview GIS* di sebelah kanan baris atas *toolbar*. Koordinat yang ditunjukkan dapat berupa angka koordinat *Universal Transverse Mercator* (UTM) maupun koordinat dengan nilai *decimal degrees*. Koordinat yang akan ditunjukkan sangat tergantung pada angka koordinat yang dipasang pada data *raster* dalam proses registrasi. Dalam proses selanjutnya angka koordinat tersebut akan di proyeksikan ke dalam angka koordinat dengan *system* proyeksi yang lain. Misalnya yang semula berkoordinat UTM kemudian di proyeksikan menjadi nilai *decimal degrees*.



Gambar II.2. Data citra satelit resolusi tinggi
(*Sumber: Eko Budiyanto; 2010 : 8*)

Gambar di atas adalah contoh tampilan data citra satelit yang telah diregistrasi. Koordinat *pointer* dapat dilihat pada nilai koordinat yang berada di sebelah kanan atas jendela *view*. Citra tersebut berada pada *zone 49* Lintang Selatan (nilai ini ditentukan saat registrasi data *raster*). Pada Contoh tertulis nilai 438.026,48 dan 922.966,05. Angka tersebut adalah angka koordinat dengan sistem proyeksi UTM yang menunjukkan koordinat lapangan dari letak ujung *pointer*. Ketika ujung *pointer* bergerak dari satu titik ke titik lain, angka koordinat itu akan terus berubah sesuai dengan letak *pointer*.



Gambar II.3. Peta sebagai sumber reproduksi peta
 (Sumber : *Eko Budiyanto; 2010 : 9*)

Gambar II.1 merupakan contoh lain dari sumber pembuatan peta. Selain menggunakan citra satelit dapat pula menggunakan peta yang disiam ulang. Proses seperti ini sering digunakan dalam perbaikan peta. Dalam pemetaan digital, setelah titik koordinat lapangan dipasang pada data *raster* tersebut dapat digunakan sebagai dasar digitasi atau digunakan dalam pembuatan *vector*.

II.3. Basis Data

Basis data menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah itu data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data

disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan kedalam basisdata, dimodifikasi, dan dihapus.

Kemudian, Siberschatz, dkk, (2002) mendefinisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. System manajemen basis data (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data. Tujuan utama sistem manajemen basis data adalah menyediakan cara menyimpan dan mengambil informasi basis data secara mudah dan efisien. (Janner Simarmata, Iman Paryudi; 2010 : 1).

II.4. Model – model Data

Struktur yang mendasari suatu basis data adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data *semantic*, dan batasan konsistensi. Untuk mengilustrasikan konsep model data, berikut disajikan dua model data, yaitu *entity relationship* model dan *relational model*. Keduanya menyediakan cara mendeskripsikan rancangan basis data pada tingkat logis.

II.4.1. Entity Relationship Model




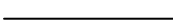
Entity relationship (ER) data model di dasarka pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek . Entitas adalah sesuatu objek dalam dunia nyata yang

dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

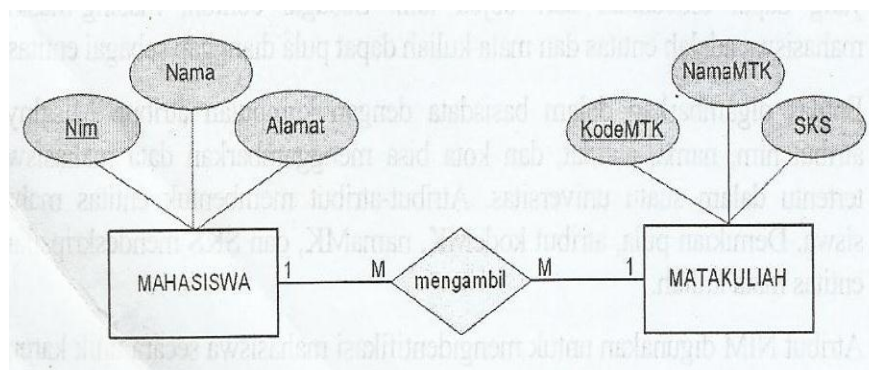
Entitas digambarkan dalam basisdata dengan kumpulan atribut. Misalnya atribut nim, nama, alamat, dan kota bias menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terdapat 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 diambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relation ship set*). (Janner Simarmata, Iman Paryudi; 2010 : 60). Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut :

Tabel II.1 Atribut Entity Relationship (ER)

	Persegi panjang mewakili kumpulan entitas
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Masing-masing komponen diberi nama entitas atau relasi yang diwakilinya. Sebagai ilustrasi, bayangkan anda mengambil bagian *system* basis data universitas yang terdiri atas mahasiswa dan mata kuliah. Gambar II.4 menunjukkan diagram ER dari contoh. Diagram menunjukkan bahwa ada dua kumpulan entitas yaitu mahasiswa dan mata kuliah, dan bahwa relasi mengambil mahasiswa dan mata kuliah.



Gambar II.4. Diagram ER
(Sumber: Jannerr Simarmata, Iman Paryudi, 2010: 60)

II.4.2. Membuat ERD

Entity Relationship Diagram adalah alat pemodelan data utama dan akan membantu dan mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien.

a. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau abstrak di mana kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai pembayaran,

kampus, dan buku. Contoh suatu entitas disebut instansi, misalnya pegawai Adi, pembayaran Joko dan lain sebagainya.

b. Relasi (*Relationship*)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya, mahasiswa bisa mengambil banyak mata kuliah.

c. Atribut (*Attribute*)

Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah property, elemen data, dan *field*. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya, nomor pegawai adalah kunci utama untuk pegawai. (Janner Simarmata, Iman Paryudi; 2010 : 67).

II.5. Pengenalan dan Instalasi PHP

Dunia *internet* semakin berkembang, terutama dalam penggunaannya dalam bidang media komunikasi dan informasi, baik yang sifat *intern* dan umum. Yang dimaksud dengan Informasi *intern* adalah data yang disimpan dalam *server* yang hanya dapat diakses oleh pihak-pihak tertentu. Misalnya, data perusahaan hanya dapat diakses oleh anak perusahaan atau kantor cabang dari sebuah perusahaan yang ada diluar daerah.

Data umum boleh diakses oleh semua pihak. Untuk membuat *website* yang dinamis dan mudah diupdate setiap saat dari *browser*, dibutuhkan sebuah program yang mampu mengolah data dari komputer *client* atau computer *server* itu sendiri sehingga mudah dan nyaman disajikan di *browser*. Salah satu program yang dapat dijalankan di *server* dan cukup andal 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 aktif interaktif dan dinamis. Data yang dikirim oleh pengunjung *website* / komputer *client* akan diaolah dan disimpan pada *database* web *server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP ini, *file* harus diupload ke dalam *server*. Upload adalah proses mentrasfer data atau *file* dari komputer *client* ke dalam *webservice*.

Pemrograman yang berjalan pada *server* banyak sekali. Setiap program mempunyai kelebihan dan kekurangan. Saat ini banyak *website* yang menggunakan program PHP sebagai dasar pengolahan data. Beberapa keunggulan yang dimiliki PROGRAM PHP adalah:

- 1) PHP memiliki tingkat akses yang lebih cepat.
- 2) PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi *internet*.
- 3) PHP memiliki tingkat keamanan yang tinggi.
- 4) PHP mampu berjalan di beberapa *server* yang ada, misalnya *Apache*, *Microsoft IIS*, *PWS*, *AOLserver*, *phhttpd*, *fhhttpd*, dan *Xitami*.

- 5) PHP mampu berjalan di Linux sebagai *platform* sistem operasi utama bagi PHP, namun juga dapat berjalan di *FreeBSD*, *Unix*, *Solaris*, *Windows*, dan yang lain.
- 6) PHP juga mendukung akses ke beberapa *database* yang sudah ada, baik yang bersifat *free/gratis* ataupun komersial. *Database* itu antara lain MySQL, PostgreSQL, Informix, dan *Microsoft SQL Server*.
- 7) PHP bersifat *free* atau gratis.

II.6. File Koneksi PHP Dengan MYSQL

Setelah membuat *database* beserta tabel, langkah selanjutnya adalah membuat sebuah *file* yang berisi kode program PHP yang digunakan untuk mengkoneksikannya dengan *database* MySQL. *File* tersebut akan digunakan pada semua *file* yang akan mengakses *database* MySQL, yaitu dengan menggunakan fungsi **include** dari *file* tersebut ke *file* induk. Berikut ini adalah struktur kode program PHP dalam *file* **sambung.php**

```
<?
// koneksi dengan database MySQL
// sambung.php
$host="localhost";
$username="yonic";
$password="3266";
$database="madiun";
$sambung=mysql_connect($host,$username,$password);
```

```
mysql_select_db($database);
```

```
?>
```

Dengan dibuatnya *file* ini, maka setiap ada program yang membutuhkan koneksi dengan *database* tidak perlu lagi membuat fungsi koneksi secara berulang-ulang, cukup memanggil kode ini.

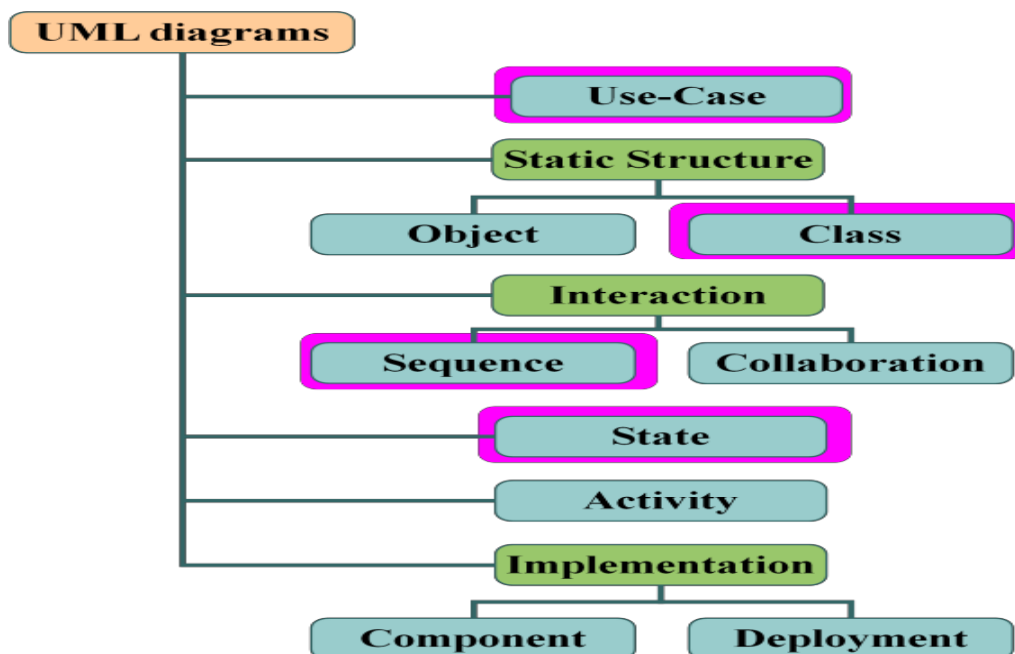
II.7. Unified Modeling Language (UML)

Unified Modelling Language (UML) Menurut (Haviluddin) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara *visual* (Braun, *et. al.* 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, *et. al.* 2004).

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam *fase* sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Secara filosofi UML diilhami oleh konsep yang telah ada yaitu

konsep permodelan *Object Oriented* karena konsep ini menganalogikan system seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. Berikut gambar dari diagram UML di bawah ini:



Gambar II.5. Diagram UML
(Sumber : Haviluddin; 2011 : 2)

II.7.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”*. Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan *visual* yang ekspresif sehingga mereka dapat

mengembangkan dan melakukan pertukaran model data yang bermakna.

2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

II.4.2. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

Struktur diagram dalam UML terdiri atas :

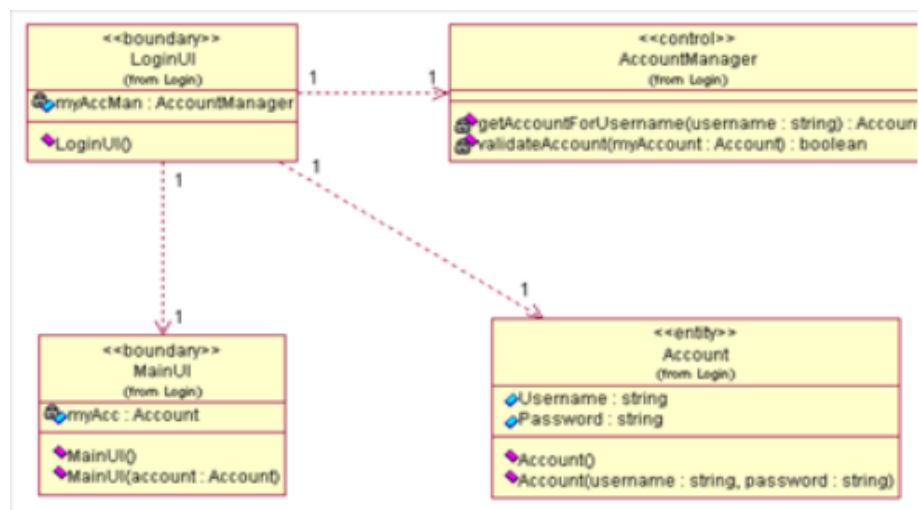
- a. *Class diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.

Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.

Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok :

1. Nama (dan *stereo type*)
2. Atribut
3. Metoda





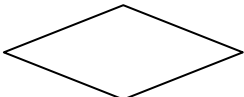

Gambar II.6. Notasi class diagram
(Sumber : Havaluddin; 2011 : 3)

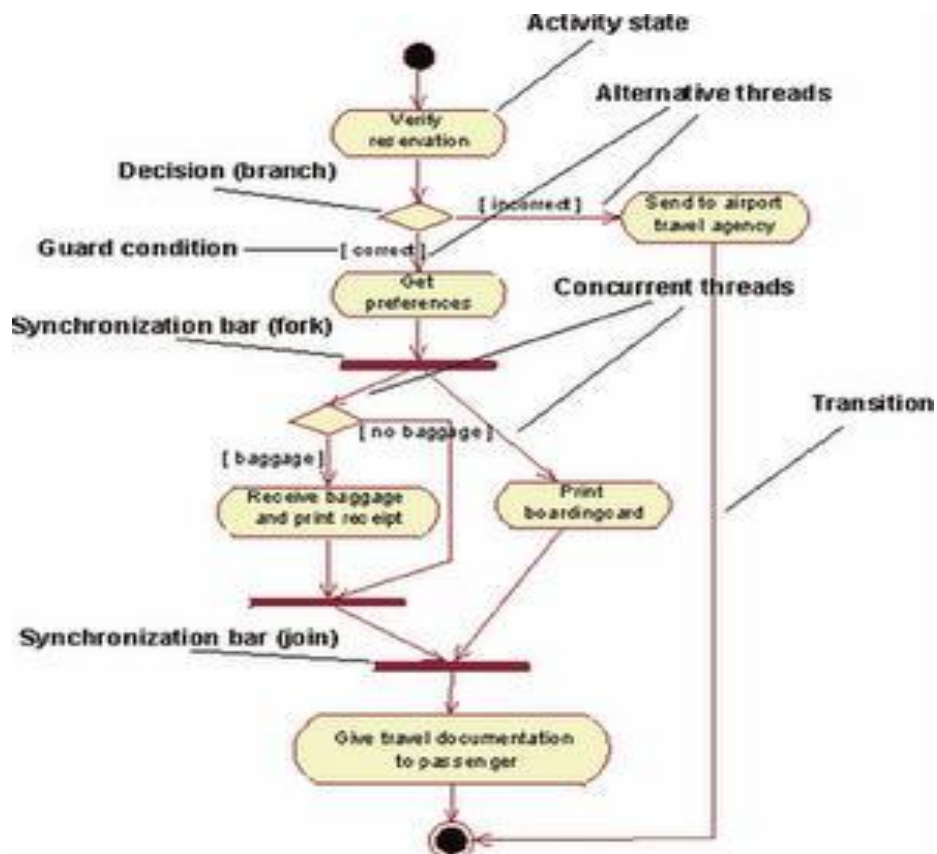
II.7.2. Activity diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*.

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

Tabel II.2 Activity Diagram

	Activity
	Transition
	Decision
	Synchronization Bars

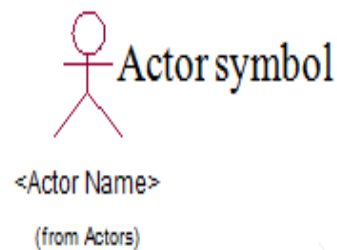


Gambar II.7. Notasi activity diagram
(Sumber : Haviluddin; 2011 : 4)

UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Berikut beberapa notasi dalam UML diantaranya :

1. Actor

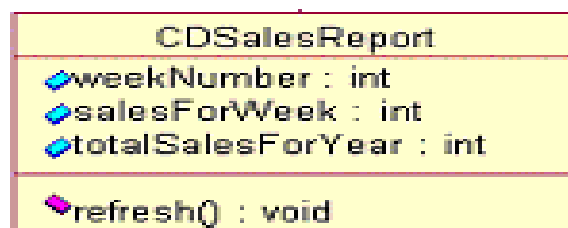
Actor menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.



Gambar II.8. Notasi actor
(Sumber : Haviluddin)

2. Class diagram

Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.

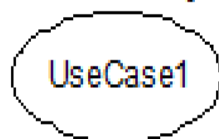


Gambar II.9. Notasi class
(Sumber : Haviluddin; 2011 : 6)

3. *Use Case* dan *use case specification*

Use case adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, *design*, *testing*, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat bahwa *use case* hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem dan tidak untuk menentukan kebutuhan non-fungsional, misalnya: sasaran kinerja, bahasa pemrograman dan lain sebagainya.

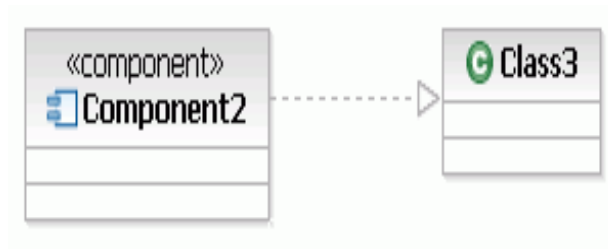
Use-case symbol



Gambar II.10. Notasi *use case*
(Sumber : Havaluddin; 2011 : 6)

4. *Realization*

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.



Gambar II.11. Notasi *realization*
(Sumber : Havaluddin; 2011 : 6)

5. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.



Gambar II.12. Notasi *Interaction*
(Sumber : Havaluddin; 2011 : 6)