

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Peranan sistem informasi didunia IT saat ini sangat penting terutama bagi perusahaan. Teknologi hardware dan software komputer baru, terus bermunculan. Peminat dibidang Sistem Informasi pun terus bertambah. Permasalahannya, teori dan aplikasi komputer sangat luas, menyebar keseluruh bidang kehidupan manusia, dan berkembang sangat pesat. Sehingga harus diikuti terus oleh peminatnya agar tidak tertinggal jauh.

Sistem merupakan sekumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

##### **II.1.1. Pengertian Sistem**

Kata sistem berasal dari bahasa Yunani yaitu "*Systema*" yang berarti kesatuan. Sistem adalah sekumpulan komponen yang saling berhubungan yang harus bekerja bersama-sama untuk menghasilkan suatu kesatuan metode, prosedur teknik yang digabungkan dan diatur sedemikian rupa sehingga menjadi satu kesatuan yang berfungsi untuk mencapai tujuan.

Menurut Anatol Rapoto, Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain ( Asbon Hendra, S.Kom: 2011 : 155 ).

Secara umum karakteristik sistem yang baik adalah :

1. Komponen (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, Bekerja sama membentuk satu kesatuan.

2. Batas Sistem (*Boundry*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu diluar batas sistem yang memengaruhi operasi dari suatu sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari susbsistem yang satu ke subsistem lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan kedalam sistem.

(Asbon Hendra, S.Kom: 2011 : 159).

### **II.1.2. Pengertian Informasi**

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat (Asbon Hendra, S.Kom : 2011 : 167).

Agar informasi yang kita sajikan lebih bermanfaat maka terlebih dahulu dibuat aliran informasi yang lebih jelas dan lengkap. Berkaitan dengan penyediaan informasi bagi manajemen dalam mengambil suatu keputusan, yang diperoleh harus berkualitas, maka kualitas dari informasi tergantung pada :

1. Akurat : Akurat berarti bahwa informasi harus bebas dari kesalahan - kesalahan dan tidak bisa (menyesatkan) dan jelas mencerminkan maksudnya.
2. Tepat Waktu : Tepat waktu berarti bahwa informasi yang datang pada penerima tidak boleh terlambat.
3. Relevansi : Relevansi berarti bahwa informasi benar-benar berguna bagi suatu tindakan keputusan yang dilakukan oleh seseorang.

### **II.1.3. Pengertian Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Asbon Hendra, S.Kom : 2011 : 169).

Suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya (Asbon Hendra, S.Kom : 2011 : 168).

## II.2 Sistem Informasi Geografis

GIS (*Geographic Information System*) atau Sistem Informasi Berbasis Pemetaan dan Geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang berkait erat dengan sistem pemetaan dan analisis terhadap segala sesuatu serta peristiwa-peristiwa yang terjadi di muka bumi.

SIG merupakan suatu sistem yang mengorganisir perangkat keras (*hardware*), perangkat lunak (*software*) dan data, serta dapat mendayagunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh yang berkaitan dengan aspek keruangan.

Data yang diolah pada SIG adalah data geospasial (data spasial dan data non spasial). Data spasial adalah data yang berhubungan dengan kondisi geografi misalnya sungai, wilayah administrasi, gedung, jalan raya dan sebagainya. Sedangkan data non-spasial adalah selain data spasial yaitu berupa text atau angka biasanya disebut atribut.

### II.2.1. Subsistem SIG

Sistem Informasi Geografis dapat diuraikan menjadi beberapa subsistem sebagai berikut:

#### 1. *Data Input*

Subsistem ini merupakan tahap persiapan dan pengumpulan data spasial, dan *attribute* dari berbagai sumber. Dalam tahap ini juga dilakukan konversi data analog ke format digital yang sesuai.

### 2. *Data Management*

Subsistem ini mengorganisasikan baik data spasial maupun *attribute* ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, di *update*, dan di *edit*.

### 3. *Data Manipulations*

Subsistem ini adalah penyesuaian terhadap data masukkan untuk proses lebih lanjut, misalnya: penyamaan skala, pengubahan system proyeksi, generalisasi dan sebagainya.

### 4. *Data Query*

Subsistem ini adalah penelusuran data menggunakan lebih dari satu layer, berfungsi untuk memberikan informasi untuk *analysis*, dan memperoleh data yang diinginkan.

### 5. *Data Analysis*

Subsistem ini adalah kemampuan untuk *analysis* data spasial untuk memperoleh informasi baru. Dengan pembuatan model scenario “*What if?*”. Salah satu fasilitas analisis yang banyak digunakan adalah analisis tumpang susun peta (*overlay*).

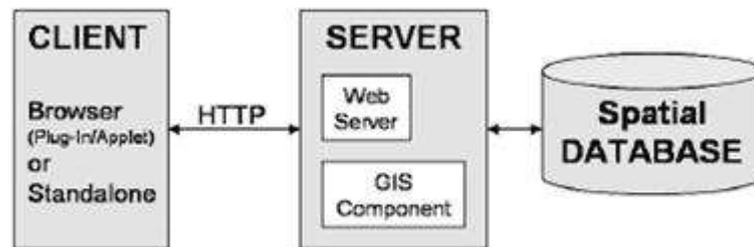
### 6. *Data Visualisasi*

Subsistem ini adalah penyajian hasil berupa informasi baru atau basis data yang ada baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti dalam bentuk : peta, table, grafik, dan lain-lain.

## II.2.2. **Arsitektur SIG**

Untuk dapat melakukan komunikasi dengan komponen yang berbeda-beda di lingkungan web maka dibutuhkan sebuah *web server*. Karena standart dari geo

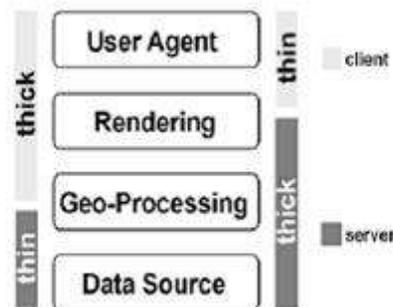
data berbeda beda dan sangat spesifik maka pengembangan arsitektur system mengikuti arsitektur '*Client Server*'.



**Gambar II.1. Arsitektur WEB GIS**

Gambar diatas menunjukkan arsitektur minimum sebuah system Web GIS. Aplikasi berada disisi client yang berkomunikasi dengan Server sebagai penyedia data melalui web Protokol seperti HTTP (*Hyper Text Transfer Protocol*). Aplikasi seperti ini bisa dikembangkan dengan *web browser* (Mozilla Firefox, Opera, Internet Explorer, dll). Untuk menampilkan dan berinteraksi dengan data GIS, sebuah browser membutuhkan *Pug-In* atau *Java Applet* atau bahkan keduanya. Web Server bertanggung jawab terhadap proses permintaan dari *client* dan mengirimkan tanggapan terhadap respon tersebut. Dalam arsitektur web, sebuah web server juga mengatur komunikasi dengan *server side* GIS Komponen. *Server side* GIS Komponen bertanggung jawab terhadap koneksi kepada database spasial seperti menterjemahkan *query* kedalam SQL dan membuat representasi yang diteruskan ke server. Dalam kenyataannya *Side Server* GIS Komponen berupa software *libraries* yang menawarkan layanan khusus untuk analisis spasial pada data. Selain komponen hal lain yang juga sangat penting adalah aspek fungsional yang terletak di sisi *client* atau di *server*. Gambar

berikut dua pendekatan yang menunjukkan kemungkinan distribusi fungsional pada system *client / server* berdasarkan konsep *pipeline visualization*.



**Gambar II.2. Thin Vs Thick system pada Client Server**

Pendekatan-1 : *Thin Client* : Memfokuskan diri pada sisi server. Hampir semua proses dan analisa data dilakukan berdasarkan *request* disisi *server*. Data hasil pemrosesan dikirimkan ke client dalam format HTML, yang didalamnya terdapat file gambar sehingga dapat dilihat dengan *browser*. Pada pendekatan ini interaksi pengguna terbatas dan tidak fleksibel

Pendekatan-2 : *Thick / Fat Client* : Pemrosesan data dilakukan disisi *client*, data dikirim dari *server* ke *client* dalam bentuk data vector yang disederhanakan. Pemrosesan dan penggambaran kembali dilakukan disisi *client*. Cara ini menjadikan user dapat berinteraksi lebih interaktif dan *fleksibel*.

(<http://dennycharter.wordpress.com/2008/05/08/konsep-dasar-web-gis/>)

### II.2.3. Konsep Model Data Spasial Pada SIG

Data spasial merupakan data yang paling penting dalam SIG. Data spasial ada 2 jenis yaitu data raster dan data vector. yaitu :

a. Data Raster

Model data raster menampilkan, menepatkan dan menyimpan spasial dengan menggunakan struktur matriks atau pixel-pixel yang membentuk grid. Akurasi model data ini sangat bergantung pada resolusi atau ukuran pixelnya (sel grid) di permukaan bumi. Konsep model data ini adalah dengan memberikan nilai yang berbeda untuk tiap-tiap pixel atau grid dari kondisi yang berbeda.

b. Data Vektor

Model data vector yang menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan titik-titik, garis-garis, kurva atau polygon beserta atribut-atributnya. Bentuk dasar representasi data spasial didalam sistem model data vector, didefinisikan oleh sistem koordinat kartesian dua dimensi (x,y).

#### **II.2.4. Peta**

Pada awalnya, data geografis hanya disajikan diatas peta dengan menggunakan symbol, garis dan warna. Elemen-elemen geografis ini dideskripsikan di dalam legenda misalnya: garis hitam tebal untuk jalan utama, garis hitam tipis untuk jalan sekunder dan jalan-jalan yang berikutnya. Peta dapat digunakan untuk berbagai kegiatan mulai dari suatu sederhana sampai ke suatu kegiatan yang sangat kompleks atau multiguna.

Peta adalah penyajian grafis dari seluruh atau sebagian permukaan bumi pada suatu bidang datar dengan menggunakan suatu skala dan sistem proyeksi tertentu. Secara umum peta terbagi beberapa jenis, yaitu:

a. Peta Topografi

Peta topografi memperlihatkan posisi horizontal serta vertical dari unsur alam dan unsur buatan manusia dalam bentuk tertentu. Peta topografi dikenal sebagai peta yang bersifat umum karena unsur-unsur yang disajikan adalah unsur yang terdapat di permukaan bumi sesuai dengan kegunaan dari peta bersangkutan, misalnya: peta kadaster (pendaftaran tanah) menyajikan data mengenai garis kepemilikan tanah bersama dengan sudut dan panjangnya, pemilik dan ukuran persil dan informasi lainnya.

b. Peta Tematik

Peta tematik adalah suatu bentuk peta yang menyajikan unsur-unsur tertentu dari permukaan bumi sesuai dari tema atau topic dari peta yang bersangkutan, misalnya: peta tata guna lahan, peta geologi. Peta tematik umumnya digunakan sebagai data analisis dari beberapa unsur permukaan bumi didalam pengambilan keputusan..

Dasar utama dalam pembuatan peta adalah pengadaan sistem koordiant yang dapat menghubungkan antara satu titik dengan titik lainnya. Sistem koodinat geografis adalah suatu sistem koordinat titik di permukaan bumi dimana posisinya ditentukan oleh dua perpotongan dua buah garis lengkung yaitu:

1. Garis Meridian

Garis meridian adalah ellips terbesar (karena titik pusatnya behimpitan dengan pusat bumi) dipermukaan bumi yang melalui kutub-kutub bumi.

## 2. Garis Paralel

Garis paralel adalah lingkaran di permukaan bumi yang bisang lingkarannya memotong tegak lurus sumbu putar bumi.

Besarnya titik perpotongan meridian dan pararel ditentukan dengan:

### 1. Lintang

Pengertian lintang pada suatu titik adalah panjang busur yang diukur pada meridian dihitung dari ekuator sampai ke paralel yang melalui titik tersebut.

### 2. Bujur

Pengertian bujur suatu titik adalah panjang busur yang diukur pada suatu garis paralel antara meridian pengamatan dengan Meridian Nol (Meridian Greenwich).

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

UML (*Unified Modelling Language*) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Perancangan perangkat lunak (program-program aplikasi) sesungguhnya merupakan salah satu pekerjaan yang paling banyak menuntut keahlian dan keterampilan manusia yaitu kemampuan dalam hal analisis dan perancangan, kemampuan teknis pemrograman, serta kemampuan manajerial (pengelolaan).

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras,

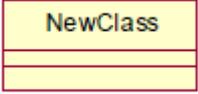
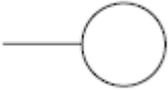
sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

UML (*Unified Modelling Language*) merupakan keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya untuk sistem yang dibangun dengan menggunakan pemrograman berorientasi obyek.

Adapun simbol-simbol notasi UML (*Unified Modelling Language*) di tujukkan pada table II.1.

**Tabel II.1 Simbol-simbol UML (*Unified Modelling Language*)**

Simbol	Arti	Fungsi
	Notasi Actor	Actor menggambarkan segala pengguna software aplikasi (user).
	Notasi Use Case	Use case menjelaskan urutan kegiatan yang dilakukan actor dan system untuk mencapai suatu tujuan tertentu.

	<p>Notasi Class</p>	<p>Class merupakan pembentuk utama dari system berorientasi obyek, karena class menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama.</p>
	<p>Notasi Interface</p>	<p>Interface merupakan kumpulan operasi tanpa implementasi dari suatu Class.</p>
	<p>Notasi Interaction</p>	<p>Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.</p>
	<p>Notasi Note</p>	<p>Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model.</p>
	<p>Notasi Dependency</p>	<p>Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.</p>

	<p>Notasi Asociation</p>	<p>Association menggambarkan navigasi antar class (navigation), berapa banyak obyek lain yang bias berhubungan dengan satu obyek (multiplicity antar class) dan apakah suatu class menjadi bagian dari class lainnya (aggregation).</p>
	<p>Notasi Generalization</p>	<p>Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik.</p>
	<p>Notasi Realization</p>	<p>Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.</p>

Sumber: *Pemodelan visual dengan UML (Munawar)*

### II.3.1. Pengertian Database

Database (basis data) merupakan kumpulan data yang saling berhubungan, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan suatu komponen penting dalam sistem informasi, karena merupakan basis yang menyediakan informasi bagi pemakai.

Elemen-elemen data di suatu berkas (*file*) *database* harus digunakan untuk pembuatan suatu output. Demikian juga dengan input yang direkam di *database*,

*file-file database* harus mempunyai elemen-elemen untuk menampung input yang dimasukkan. Struktur dari suatu file database adalah seperti pada Gambar II.5 :



**Gambar II.3. Hirarki Data dalam Database**

*Sumber : " Perancangan dan Pengembangan Sistem Informasi (Budi Sutdejo ; 2002 : 18) "*

Hirarki / tingkatan data dalam database yaitu:

1. Database : Suatu database menggambarkan data yang saling berhubungan antara satu dengan yang lainnya.
2. File : Suatu file menggambarkan satu kesatuan data yang sejenis, dimana kumpulan dari file membentuk suatu database.
3. Record : Suatu record menggambarkan suatu unit data individu yang tertentu dimana kumpulan dari record membentuk suatu field.
4. Field : Suatu field menggambarkan atribut dari record yang menunjukkan item dari data seperti nama, alamat, dimana kumpulan dari field membentuk suatu record.
5. Byte : Merupakan atribut dari field yang berupa huruf yang membentuk nilai dari sebuah field.
6. Bit : Bagian terkecil dari data secara keseluruhan yaitu berupa karakter ASCII (*American Standard Code Form Information Interchange*) nol atau satu yang merupakan komponen pembentuk byte.

## II.4 Bahasa Pemrograman PHP

PHP merupakan bahasa pemrograman web yang bersifat server - side HTML = embedded scripting, di mana scriptnya menyatu dengan HTML dan berada disisi server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di server tetapi disertakan HTML biasa. PHP dikenal sebagai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis seperti ASP (Active Server Pages) dan JSP (Java Server Pages).

PHP pertama kali dibuat oleh Rasmus Lerdroft, seorang programmer C. Semula PHP digunakannya untuk menghitung jumlah pengunjung di dalam webnya. Kemudian ia mengeluarkan Personal Home Page Tools versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan script PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah mampu mengakses database dan dapat terintegrasi dengan HTML. Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang softwarena.

Versi terbaru yaitu PHP 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya Zend Engine yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP3 scripting engine. Yang lainnya adalah build in HTTP session, tidak lagi menggunakan library tambahan seperti pada PHP3. Tujuan dari bahasa scripting

ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada *web-browser*, tetapi prosesnya secara keseluruhan dijalankan *web-server*.

(<http://rastindo.info/informatika/pengetahuan/51-pengertian-php.html>).

## **II.5. MapServer**

### **II.5.1. MapServer**

MapServer merupakan suatu aplikasi server yang memungkinkan suatu halaman web dapat memuat suatu peta yang mengandung informasi seperti Sistem Informasi Geografis. MapServer dibangun berdasarkan atas kebutuhan untuk membangun aplikasi SIG diatas yang berbasis web. Dengan menggunakan MapServer diharapkan suatu peta dapat memberikan informasi secara lengkap dan tepat, karena sistem yang berbasis web akan lebih mudah dan cepat diakses oleh pengguna dari berbagai tempat hanya dengan menggunakan komputer, web-browser dan jaringan internet.

MapServer adalah salah satu program aplikasi CGI (*Common Gateway Interface*) yang dapat digunakan untuk membangun aplikasi Sistem Informasi Geografis berbasis web (*WebGIS*). MapServer menyediakan beberapa fitur yang diperlukan untuk pengolahan suatu peta, misalnya memperbesar atau memperkecil ukuran skala peta, pengolahan informasi yang berkaitan dengan suatu lokasi dan lainnya. MapServer membantu penyajian suatu peta yang lebih interaktif, dimana pengguna dapat mengakses informasi geografis yang lengkap hanya dengan menggunakan komputer, *web-browser* dan jaringan internet.

Dalam penyusunan skripsi ini, penulis merancang suatu perangkat lunak yang akan dapat memberikan informasi peta jalur trayek angkutan umum kota Medan. Aplikasi yang dirancang akan menggunakan teknologi SIG yaitu MapServer khususnya paket MS4W yang bekerja pada sistem operasi Windows.

MapServer merupakan aplikasi *freeware* dan *open source* yang memungkinkan pemakai menampilkan data spasial (peta) di web. Aplikasi ini pertama kali dikembangkan di Universitas Minnessota, Amerika Serikat untuk proyek ForNet (sebuah proyek untuk manajemen sumber daya alam) yang disponsori NASA (*National Aeronautics and Space Administration*). Dukungan NASA dilanjutkan dengan dikembangkannya proyek TerraSIP untuk manajemen data lahan. Saat ini, karena sifatnya yang terbuka (*open source*), pengembangan MapServer dilakukan oleh pengembang diberbagai Negara.

Pada bentuk paling dasar MapServer berupa sebuah program CGI (*Common Gateway Interface*). Program tersebut akan dieksekusi di web server dan 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.

### **II.5.2. Arsitektur MapServer**

Interaksi antara klien dengan server berdasar scenario *request* dan respon. Web *browser* di sisi klien mengirim *request* ke server web. Karena server web tidak memiliki kemampuan pemrosesan peta, maka *request* berkaitan dengan pemrosesan peta akan diteruskan oleh server web ke server aplikasi dan

MapServer. Hasil pemrosesan akan dikembalikan lagi melalui server web, terbungkus dalam bentuk file HTML atau applet.

Arsitektur aplikasi pemetaan di web dibagi menjadi dua pendekatan sebagai berikut :

a. Pendekatan *Thin Client*

Pendekatan ini memfokuskan diri pada sisi server. Hampir semua proses dan analisis data dilakukan berdasarkan permintaan (*request*) di sisi server. Data hasil pemrosesan kemudian dikirimkan ke klien dalam format standar HTML, yang di dalamnya terdapat file gambar dalam format standar (misalnya GIF, PNG atau JPG) sehingga dapat dilihat menggunakan sembarang web *browser*. Kelemahan utama pendekatan ini menyangkut keterbatasan pilihan interaksi dengan pengguna yang kurang fleksibel.

b. Pendekatan *Thick Client*

Pada pendekatan ini, pemrosesan data dilakukan di sisi klien menggunakan beberapa teknologi seperti control ActiveX atau applet. Kontrol ActiveX atau applet akan dijalankan di klien untuk memungkinkan *web browser* dengan kemampuan standar. Dengan adanya pemrosesan di klien, maka transfer data antara klien dengan web server akan berkurang.

MapServer menggunakan pendekatan *thin client*. Semua pemrosesan dilakukan di sisi server. Informasi peta dikirimkan ke *web browser* di sisi klien dalam bentuk file gambar (JPG, PNG, GIF atau TIFF). Untungnya , saat ini kelemahan pendekatan *thin client* dalam hal interaksi dengan pengguna sudah

jauh berkurang dengan adanya *framework* aplikasi seperti *Chameleon* atau *CartoWeb*.

### II.5.3. Fitur-fitur Pada MapServer.

MapServer mempunyai fitur-fitur sebagai berikut :

- a. Menampilkan data spasial dalam format vector seperti : *Shapefile* (ESRI), *ArcSDE* (ESRI), *PostGIS* dan berbagai format vector lain menggunakan *Library OGR*.
- b. Menampilkan data spasial dalam format raster seperti *TIFF/GeoTIFF*, *EPPL7* dan berbagai format data raster lainnya dengan menggunakan *library GDAL*.
- c. Dapat dikembangkan (*customizable*), dengan tampilan keluaran yang dapat diatur menggunakan file-file template.
- d. Dapat melakukan seleksi objek berdasarkan nilai, berdasarkan titik area, atau berdasarkan sebuah objek spasial tertentu.
- e. Mendukung *rendering* karakter berupa *font true type*.
- f. Mendukung penggunaan data raster maupun data vector yang *di-tiled* (dibagi-bagi menjadi sub bagian yang lebih kecil sehingga proses untuk mengambil dan menampilkan gambar dapat dipercepat).
- g. Dapat menggambarkan elemen peta secara otomatis : skala grafis, peta indeks dan legenda peta.
- h. Dapat menggambarkan peta tematik yang dibangun menggunakan ekspresi logika atau ekspresimen regular.

- i. Dapat menampilkan label dari objek spasial, dengan label dapat diatur sedemikian rupa sehingga tidak saling tumpang tindih.
- j. Konfigurasi dapat diatur secara *on the fly* (dapat disetting dalam keadaan Online) melalui parameter yang ditentukan pada URL.
- k. Dapat menangani beragam system proyeksi secara *on the fly*.

Saat ini, selain dapat mengakses MapServer sebagai program CGI, MapServer juga dapat diakses sebagai modul MapScript, melalui bahasa *script* : PHP, Perl, Python, atau java. Akses fungsi-fungsi MapServer melalui *script* akan lebih memudahkan pengembangan aplikasi.

#### **II.5.4. Pengetahuan dasar MapServer**

Dalam pengembangan aplikasi berbasis MapServer, diperlukan beberapa pengetahuan dasar sebagai berikut :

- a. Pengetahuan tentang peta digital antara lain meliputi skala format, bentuk, koordinat dan system proyeksi. Pengetahuan dalam bidang ini sangat penting karena peta digital memang merupakan data utama yang dikelola oleh MapServer.
- b. Pengetahuan tentang system operasi dan server web pada tempat dimana MapServer akan dipasang.
- c. Pengetahuan tentang struktur dan cara penanganan file berformat HTML. Informasi yang dihasilkan MapServer akan dikirm ke klien (*Web Browser*) dalam format HTML.
- d. Pengetahuan tentang pemrograman di web. Misal dengan menggunakan menggunakan PHP di sisi server atau Javascript di sisi klien. Aplikasi

MapServer umumnya bersifat dinamis dan interaktif, sehingga hampir dipastikan pengguna perlu melakukan penyesuaian.

- e. Pengetahuan tentang basis data, karena data spasial hampir tidak pernah lepas dari informasi lain dalam basis data.

### II.5.5. Komponen Pembentuk MapServer

Perkembangan MapServer sebagai sebuah aplikasi *open source*, banyak memanfaatkan aplikasi lain yang juga bersifat *open source*, sedapat mungkin menggunakan aplikasi yang sudah tersedia jika memang memenuhi kebutuhan, untuk menghemat sumber daya dan waktu pengembangan. Pembahasan komponen MapServer terdiri dari empat komponen yaitu :

- a. Komponen untuk akses data spasial.

Komponen ini bertugas untuk menangani baca/tulis data spasial, baik yang tersimpan sebagai file maupun yang tersimpan pada DBMS (*Database Management System*).

- b. Komponen untuk penggambaran peta.

MapServer akan mengirimkan tampilan peta berupa gambar. Pemakai dapat memilih apa format data gambar yang akan digunakan, dan dari beberapa komponen tersebut dapat membentuk gambar peta yang dihasilkan oleh MapServer.

- c. Komponen untuk menangani proyeksi peta.

Digunakan MapServer untuk keperluan menangani system proyeksi peta.

- d. Komponen pendukung.

Misalnya software *editor* seperti *macromedia dreamweaver 8*, dll.

### II.5.6. Struktur File Map

MapServer menggunakan file \*.MAP (file dengan akhiran .map, misalnya jawa.map) sebagai file konfigurasi peta. File ini akan berisi komponen tampilan peta seperti definisi *layer*, definisi proyeksi peta, pengaturan legenda, skala dan sebagainya. Secara umum, file \*.map memiliki beberapa karakteristik sebagai berikut :

- a. Berupa file teks.
- b. Tidak *case sensitive* (tidak membedakan antara karakter yang ditulis dengan huruf besar atau huruf kecil), sebagai contoh : kata “LAYER”, “layer” maupun “Layer” memiliki arti yang sama pada file \*.map. Hal ini tidak berlaku bagi penamaan atribut, misalnya nama *field* pada sebuah Shapefile (file \*.shp). Nama *field* harus dituliskan persisi seperti yang ditulis pada sumbernya. Meskipun tidak *case sensitive*, sebaiknya perlu ditentukan aturan penggunaan huruf besar atau kecil untuk menjaga konsistensi. Pada umumnya digunakan huruf besar untuk menuliskan isi file \*.map.

Teks yang mengandung karakter bukan alfanumerik (huruf dan angka), harus berada di dalam tanda petik, misalnya : “/opt/webgis/map” (karena karakter ‘/’ bukan karakter alfanumerik). Meskipun keharusan ini hanya berlaku untuk teks yang mengandung karakter bukan alfanumerik, sebaiknya secara konsisten menggunakan tanda petik untuk setiap variable teks.

## II.6. MySQL

### II.6.1. MySQL

MySQL merupakan aplikasi *database* yang termasuk paling populer dalam lingkungan Linux, kepopuleran ini ditunjang karena performansi *query* dari databasenya yang sangat cepat dan jarang bermasalah. Saat ini, MySQL telah tersedia juga dalam lingkungan Windows, MySQL di lingkungan Windows diletakan pada direktori *c:\mysql\bin* adalah direktori yang berisi daftar modul *executable* dari MySQL.

MySQL merupakan database yang dikembangkan dari bahasa SQL (Structure Query Language). SQL sendiri merupakan bahasa yang terstruktur yang digunakan untuk interaksi antara script program dengan database server dalam hal pengolahan data. Dengan SQL, kita dapat membuat tabel yang nantinya akan diisi dengan data, memanipulasi data ( misalnya menambah data, menghapus data dan memperbaharui data ), serta membuat suatu perhitungan dengan berdasarkan data yang ditemukan. *MySQL* adalah program relational database management system (RDMS) yang bersifat free dan open source. *MySQL* sangat cepat, gratis, mudah dikonfigurasi, mudah dipelajari dan tersedia source codenya. Itulah beberapa kelebihan utama dari *MySQL* dibanding RDMS komersial yang ada.

*MySQL* menggunakan *SQL* (Structured Query Language) yang merupakan standar global dalam management database. *SQL* adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti *Oracle*, *Postgre SQL*, *SQL Server*, dan lain-lain. Sebagai sebuah

program penghasil *database*, *MySQL* tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interface*). *MySQL* dapat didukung oleh hampir semua program aplikasi baik yang *open source* seperti *PHP* maupun yang tidak, yang ada pada *platform windows* seperti *Visual Basic*, *Delphi*, dan lainnya, Popularitas sebagai aplikasi *web* dikarenakan kedekatannya dengan popularitas *PHP*, sehingga seringkali disebut sebagai *Dynamic Duo*.