# BAB II

# LANDASAN TEORI

## 2.1. Kajian Pustaka

### 2.1.1. Perencanaan Strategis

Perencanaan strategis dari berbagai aspek pengertian yaitu penentuan kegiatan pada waktu yang akan datang, suatu proses, falsafah dan struktur.[1]

Perencanaan strategis adalah proses pemilihan tujuan tujuan organisasi, penetuan strategi, kebijakan dan program program strategi yang diperlukan untuk tujuan tersebut dan penetapan metode yang di perlukan untuk menjamin strategi dan kebijakan telah diimplementasikan.[1]

### 2.1.2. Jaringan Internet

Jaringan Internet Merupakan rangkaian hubungan jaringan computer yang dapat diakses secara umum di seluruh sedunia, yang mengirimkan data dala bentuk paket data berdasarkan standart internasional protocol (IP).Lebih dalam lagi internet adalah kumpulan jaringan dari jaringanjaringan computer dunia yang terdiri dari jutaan unit-unit kecil.[2]

### 2.1.3. Metode SCRUM

Scrum merupakan sebuah kerangka kerja dimana pihak-pihak dapat mencari jalan keluar dari permasalahan yang kompleks dan pada saat yang bersamaan membuat produk yang memiliki nilai setinggi mungkin secara produktif dan kreatif (Schwaber, et al, 2011).[3]

### 2.1.4. Flowmap

Setiap *flowmap* memiliki beberapa pengertian yang akan dijelaskan sebagai berikut :

Definisi *flowmap* menurut Ladjamudin bin Al-Bahra adalah sebagai berikut :

*"Flowmap* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowmap* merupakan cara penyajian dari suatu algoritma*"*

Bagian alir terdiri dari lima macam, yaitu :

- Bagan alir sistem (*systems flowmap*)
- Bagan alir dokumen (*document flowmap*)

Bagan alir dokumen atau disebut juga bagan alir formulir atau *paperwork flowmap* merupakan bagan alir yang menunjukan arus dari laporan dan formulir dan termasuk tembusan-tembusannya.[4]

### 2.1.5. UML (*Unified Modeling Language)*

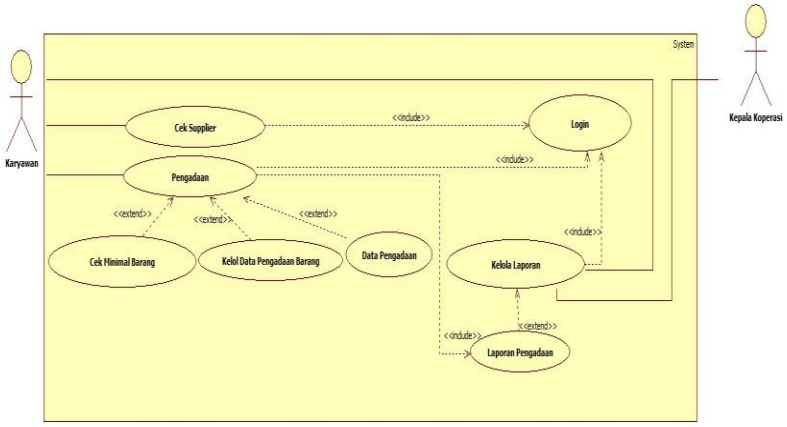
UML *(Unified Modelling Language*) adalah sebuah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan , menspesifikasikan dan membangun perangkat lunak. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.[5]

#### 2.1.5.1 *Usecase Modelling* (*Usecase*)

*Use-case modeling* menurut Bentley (2004, p270) disebutkan yaitu "*the process of modeling a system;s function in terms of business events, who initiated the events, and how the system responds to those events*". Yang dapat diartikan yaitu bahwa pemodelan *usecase* merupakan suatu proses dari permodelan suatu fungsi sistem dalam hubungannya dengan kejadian-kejadian bisnis, yang menginisiasikan kejadian-kejadian, dan bagaimana sistem menjawab dari kejadian-kejadian itu.

Menurut Bentley (2004, p271), *use-case diagram* disebutkan yaitu "*a diagram that depicts the interaction between the system and external systems and users. In other words, it graphically describes who will use the system and in what ways the user expects to interact with the system*". Jadi *use-case diagram* merupakan

sebuah diagram yang menggambarkan interaksi antara sistem dan sistem-sistem eksternal dan pengguna-pengguna. Dengan kata lain, ini secara jelas menerangkan siapa yang menggunakan sistem dan dengan cara bagaimana pengguna mengharapkan untuk berinteraksi dengan sistem.[5]

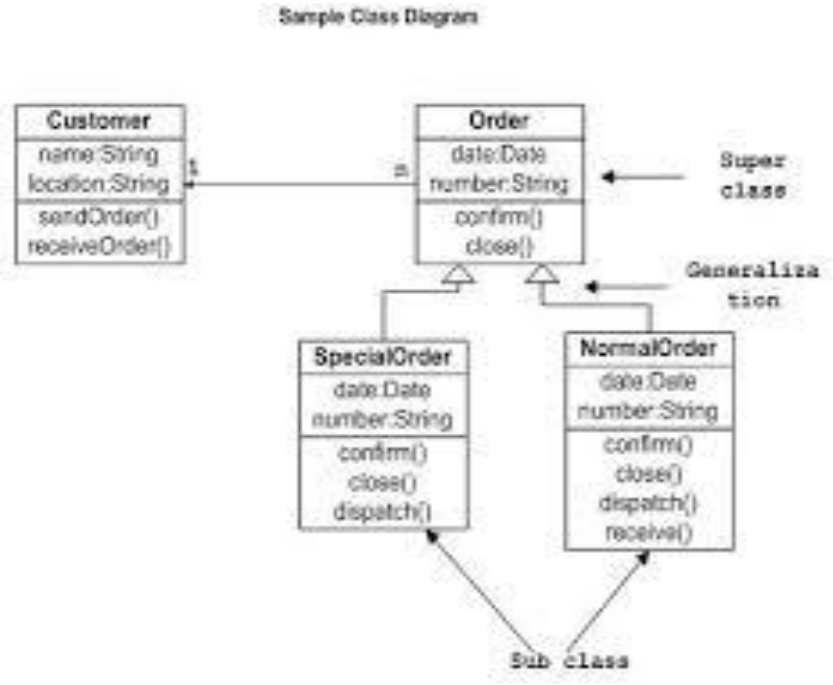



*Gambar 2.1 Usecase*

### 2.1.5.2 *Class Diagram*

*Class diagram* digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka. Biasanya, dibuat beberapa *class diagram* untuk sistem tunggal. Beberapa diagram akan menampilkan *subset* dari kelas-kelas dan relasinya. Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun.

*Class diagram* adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum kode ditulis, dan membantu untuk memastikan bahwa sistem adalah desain terbaik.[5]



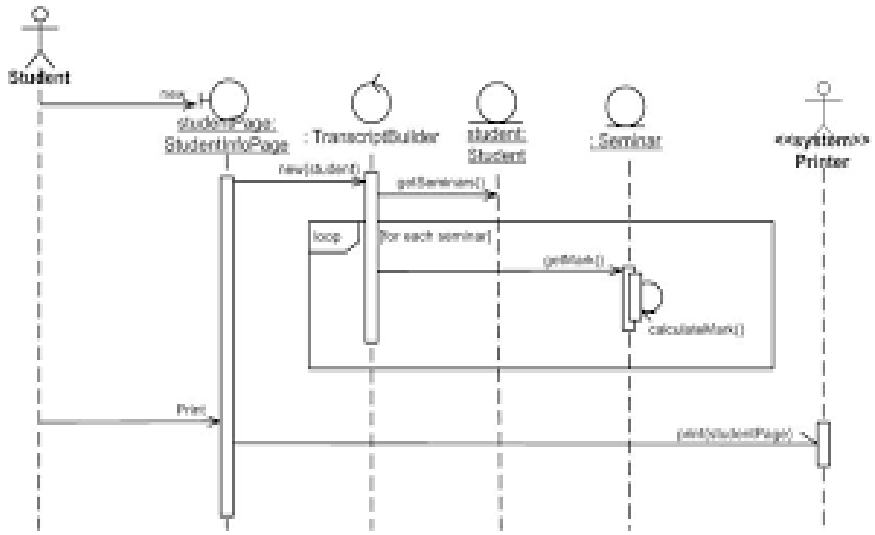

*Gambar 2.2 Class Diagram*

**2.1.5.3 *Sequence Diagram***

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Tujuan penggunaan *sequence diagram* :

1. Mengkomunikasikan requirement kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model *design*.
2. Merupakan diagram yang paling cocok untuk mengembangkan model deskripsi *usecase* menjadi spesifikasi *design*.[5]



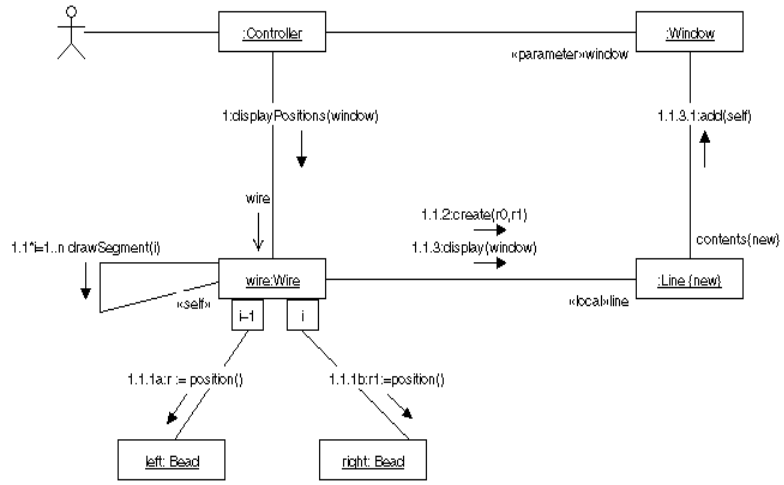

*Gambar 2.3 Sequence Diagram*

**2.1.5.4 *Collaboration Diagram***

*Collaboration Diagram* yaitu menggambarkan interaksi antar objek seperti *sequence diagram* tetapi lebih menekankan pada peran masing masing objek dan bukan pada waktu penyampaian *message*. setiap *message* memiliki *sequence number* dimana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

Tools yang digunakan pada *collaboration Diagram* :

1. *Object* : *object* merupakan *instance* dari sebuah *class* dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah kotak dengan nama *object* didalamnya dan diawali dengan sebuah titik koma.
2. *Actor* : *Actor* juga dapat berkomunikasi dengan *object*, maka dari itu *actor* juga dapat diurutkan sebagai kolom. Simbol *actor* pada *collaboration diagram* sama pada *actor usecase diagram*.
3. *Message* : *message* digambarkan dengan anak panah yang mengarah antar *object* dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar *object*.[5]

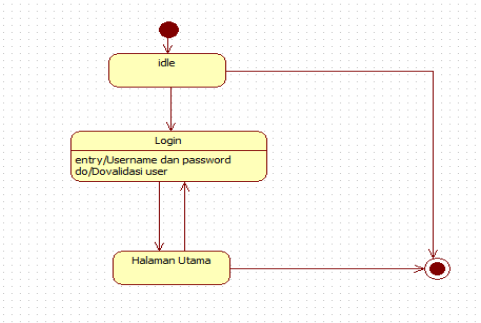



*Gambar 2.4 Collaboration Diagram*

### 2.1.5.5 *Statechart Diagram*

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

*State* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antara *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan di tuliskan dalam ukuran siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir di gambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.[5]

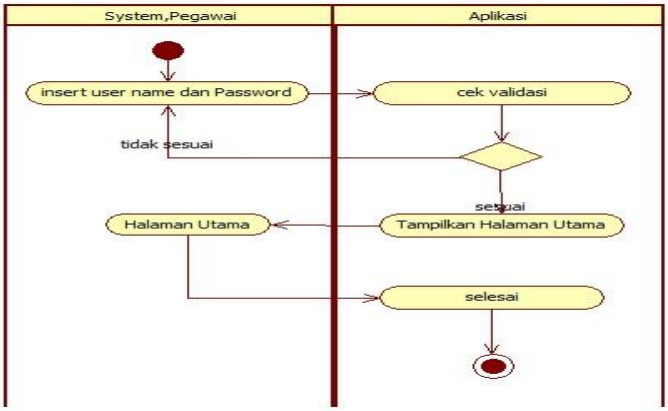



*Gambar 2.5 Statechart Diagram*

### 2.1.5.6 *Activity Diagram*

*Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

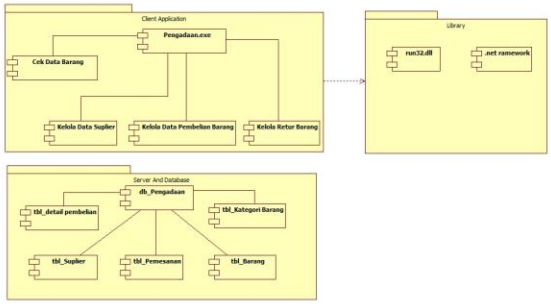
*Activity diagram* merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.[5]




*Gambar 2.6 Activity Diagram*

### 2.1.5.7 *Component Diagram*

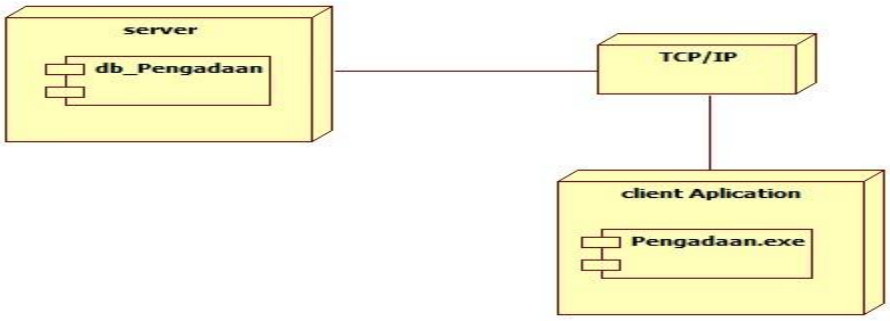
*Component diagram* menggambarkan struktur dan hubungan antara komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak merupakan modul yang berisi *code* baik berisi *source code* maupun *binary code*, baik *library* maupun *executable* baik yang muncul pada *complie time, link time* maupun *run time*.[5]




*Gambar 2.7 Component Diagram*

### 2.1.5.8 *Deployment Diagram*

*Deployment Diagram* menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem dimana komponen akan terletak (pada mesin,server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal hal lain yang bersifat fisikal.[5]




*Gambar 2.8 Deployment Diagram*

### 2.1.6. SQL Server

SQL adalah sebuah konsep pengoprasian basis data terutama untuk proses seleksi, pemasukan,pengubahan dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis. SQL juga merupakan bahasa yang digunakan untuk berkomunikasi dengan *database*. Menurut ANSI *(American National Institute)* Bahasa ini adalah sebuah standar untuk RDBMS *(Relational Database Management System).*[6]

### 2.1.7. PHP

PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman web *server-side* yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru (update). Semua script PHP dieksekusi pada server dimana script tersebut dijalankan.[7]

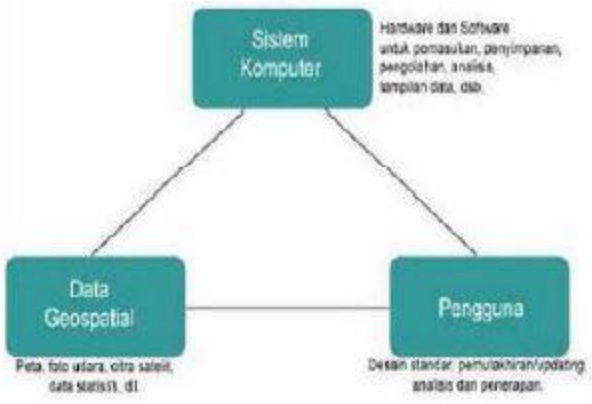
### 2.1.8. Sistem Informasi Geografis

Pada dasarnya, istilah Sistem Informasi Geografis (SIG) merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi dan geografis. Dengan melihat unsur-unsur tersebut, maka jelas SIG merupakan salah satu sistem informasi yang menekankan pada unsur "informasi geografis". SIG terdiri dari data spasial dan aspasial.[8]

Sistem Informasi Geografi (SIG) 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. Teknologi SIG mengintegrasikan operasi pengolahan data berbasis database yang biasa digunakan saat ini, seperti pengambilan data berdasarkan kebutuhan, serta analisis statistik dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya.

Dari definisi yang ada, diambil satu buah definisi yang dapat mewakili SIG secara umum yaitu sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisa dan menghasilkan data bereferensi geografi atau data geospatial, untuk mendukung pengambilan keputusan dalam

perencanaan dan pengolahan seperti penggunaan lahan, sumber daya alam, lingkungan transportasi, perencanaan fasilitas kota, dan pelayanan umum lainnya.




Gambar 2.9 Komponen Kunci SIG

Data yang diolah pada SIG ada 2 macam yaitu data geospasial (data spasial dan data nonspasial).[9]

### 2.1.9. Data Spasial

Data spasial adalah data yang berhubungan dengan kondisi geografi misalnya sungai, wilayah administrasi, gedung, jalan raya dan sebagainya. Seperti yang telah diterangkan pada gambar diatas, data spasial didapatkan dari peta, foto udara, citra satelit, data statistik dan lainlain. Hingga saat ini secara umum persepsi manusia mengenai bentuk representasi entity spasial adalah konsep raster dan vector.[9]

#### 2.1.9.1 Data Raster

Data Raster didefinisikan sebagai data sel pada grid yang mempunyai letak koordinat. Data raster biasanya menggunakan file dengan ekstensi *.tiff dan *.jpg dengan tambahan file georeferensi seperti *.wld dan sebagainya.[10]

### 2.1.9.2 Data Vektor

Pada Model Vektor digunakan untuk merepresentasikan tipe data diskrit seperti jalan, bangunan dan lain lain. Data Vektor memiliki 3 bentuk data penyajian yaitu titik, garis dan area.[10]

#### 2.1.9.2.1 Titik

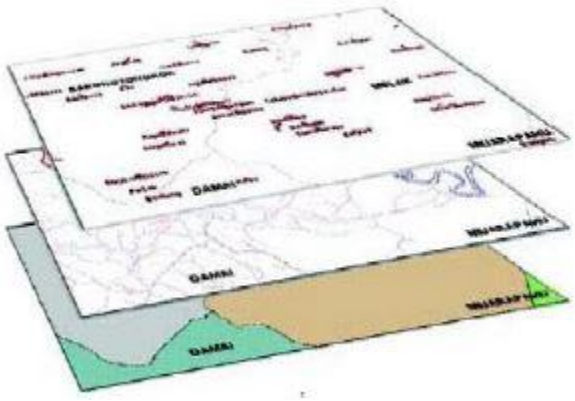
Sebuah dimensi objek yang spesifik yang menunjukkan lokasi geografi melalui sekumpulan sistem koordinat. Dengan istilah lainnya adalah point.[10]

#### 2.1.9.2.2 Garis

Sebuah dimensi objek yang menghubungkan beberapa titik membentuk *Polyline.*[10]

#### 2.1.9.2.3 Area

Sebuah Objek 2 dimensi yang merupakan sebuah lokasi pada permukaan bumi. Istilah lainnya adalah *Polygon.* Spasial data yang di tampilkan akan membentuk lapisan paisan yang disebut dengan layer yang saling menumpuk.[10]



Gambar 2.8 Bentuk Layer Peta

### 2.1.10. Data Non-Spasial

Nonspasial adalah selain data spasial yaitu data yang berupa text atau angka. Biasanya disebut dengan atribut. Data non-spasial ini akan menerangkan data spasial atau sebagai dasar untuk menggambarkan data spasial. Dari data non-spasial ini nantinya dapat dibentuk data spasial. Misalnya jika ingin menggambarkan peta

penyebaran penduduk maka diperlukan data jumlah penduduk dari masing-masing daerah (data non-spasial), dari data tersebut nantinya kita dapat menggambarkan pola penyeberan penduduk untuk masing – masing daerah.[9]

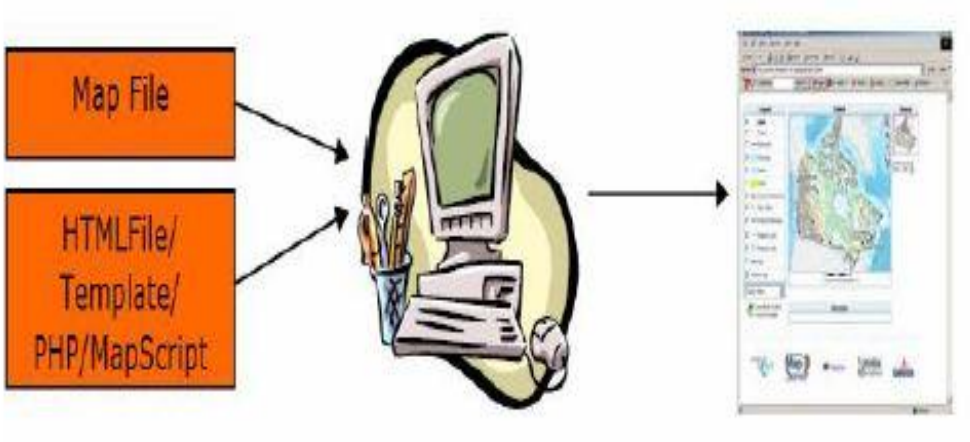
### 2.1.11. Mapserver

Merupakan webserver berbasis data spasial yang memungkinkan dalam pembacaan spasial data yang berbentuk shapefile (*.shp) dengan menyediakan pembacaan bahasa pemrograman mapscript dan AJAX (Asynchronous Javascript And XML).

Dalam perkembangannya, mapserver dikembangkan pihak-pihak ketiga dengan memberikan framework opensource untuk penunjang pengerjaan layout interaksi pada peta GIS (*Geography Information System*).[11]

Pada Bentuk paling dasar MapServer berupa sebuah program CGI(Common Gateway Interface) dimana program tersebut akan di eksekusi di web server dan berdasarkan beberapa parameter tertentu (terutam konfigurasi dalam bentuk file MAP) akan menghasilkan data yang kemudian akan dikirim ke web browser baik dalam bentuk gambar peta ataupun bentuk lain.[12]

Perhatikan alur menampilkan peta di dalam aplikasi mapserver berikut ini.




Gambar 3.0 Proses Penyajian Peta Mapserver

### 2.1.12. OpenLayers

Metode penampilan peta pada geoserver menggunakan suatu teknik yang dinamakan OpenLayers. OpenLayers menggunakan bahasa pemograman Javascript API (Aime, 2008). API adalah Application Programming Interface. API merupakan suatu mekanisme yang memungkinkan programmer menggunakan data dari aplikasi lain (lintasberita.com, 2008). Suatu aplikasi dapat mensharing data dengan aplikasi lain melalui API ( www. Programmableweb.com ,2008 ).

OpenLayers merupakan aplikasi dari AJAX, sehingga membuat tampilan peta menjadi lebih interaktif dan menarik. Pemakaian OpenLayers dilakukan dengan memanggil script ini ke dalam peta dasar yang telah disediakan. Peta yang akan ditampilkan pada halaman web diletakkan pada suatu tag khusus di halaman HTML. Meletakkan peta pada halaman web umumnya dilakukan dengan menggunakan div tag untuk menampung peta dan mereferensikan element ini ke Document Object Model (DOM) ( Aime, 2008).

Basic map pada halaman HTML merupakan peta kosong. Basic map ini digunakan untuk menampung peta yang dipanggil dengan menggunakan OpenLayers. Proses memunculkan peta di dalam basic map pada halaman HTML, dilakukan dengan mendefenisikan layer peta yang akan ditampilkan. Pendefenisian layer peta ini dilakukan dengan memanggil fungsi dari OpenLayers.

Di dalam browser, OpenLayers yang merupakan javascript API berfungsi untuk mengendalikan event-event atau kejadian yang terjadi di browser. Misalnya, ketika user melakukan drag pada objek peta atau ketika user mengklik peta, atau interaksi yang terjadi didalam objek DOM baik interaksi dari mouse maupun keyboard. Respon-respon dari event-event tersebut diatur oleh openLayers. (http://openlayers.org/dev/doc/examples, 2008).[13]