BAB II

LANDASAN TEORI

2.1. Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu :

1. Setiap sistem terdiri dari unsur-unsur.
2. Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
3. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
4. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

Pengertian sistem terbagi dua yaitu dilihat dari pendekatan yang menekankan pada prosedur dan pendekatan yang menekankan pada elemen atau komponennya.

Menurut Jogiyanto (2005:1) pendekatan sistem yang menekankan pada prosedur mendefinisikan sistem sebagai jaringan kerja dan prosedur-prosedur

yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan sasaran tertentu.

Adapun pendekatan sistem yang menekankan pada elemen atau komponennya mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Dari kedua pengertian di atas dapat disimpulkan bahwa pengertian sistem adalah kumpulan elemen-elemen atau jaringan kerja dan prosedur-prosedur yang saling berinteraksi satu sama lain untuk mencapai tujuan atau sasaran tertentu.

2.1.1. Karakteristik Sistem

Menurut Jogiyanto (2005:3) suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

a. Komponen-komponen (*Components*)

Suatu sistem terdiri dari kumpulan komponen atau subsistem yang saling berinteraksi atau bekerja sama satu sama lain untuk membentuk satu kesatuan. Setiap komponen atau subsistem tersebut memiliki sifat-sifat dan fungsi tertentu yang mempengaruhi proses sistem secara keseluruhan.

b. Batas Sistem (*Boundary*)

Batas sistem yaitu daerah yang membatasi antara suatu sistem dengan lingkungan luarnya atau dengan sistem yang lainnya. Batas suatu sistem merupakan ruang lingkup (*scope*) dari sistem tersebut. Dan batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

c. Lingkungan Luar Sistem (*Environments*)

Environments dari suatu sistem adalah apapun yang berada diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang bersifat merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lain dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintaenance input* yang digunakan untuk mengoperasikan komputer dan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran (*Output*)

Keluaran merupakan hasil dari energi yang diolah. Keluaran ini dapat berupa keluaran yang berguna atau keluaran yang tidak berguna. Sebagai contoh panas yang dikeluarkan dari sistem komputer merupakan keluaran yang tidak berguna dan merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang berguna dan dibutuhkan.

g. Pengolah (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan yang dibutuhkan oleh manajemen.

h. Sasaran Sistem (*Objectives*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.1.2. Klasifikasi Sistem

Menurut Jogiyanto (2005:6) sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya adalah sebagai berikut:

a. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologi. Adapun sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer.

b. Sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Adapun sistem buatan manusia adalah sistem yang dirancang oleh manusia. Misalnya, sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin yang disebut *human machine system.*

c. Sistem tertentu dan sistem tak tertentu.

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Misalnya sistem komputer. Adapun sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

d. Sistem tertutup dan sistem terbuka.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannnya tidak ada sistem yang sifatnya tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan

menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistemnya terbuka dan terpengaruh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik.

2.2. Pengertian Informasi

Kata informasi berasal dari kata Perancis kuno *informacion* (tahun 1387) yang diambil dari bahasa Latin *informationem* yang berarti "garis besar, konsep, ide". Informasi merupakan kata benda dari *informare* yang berarti aktivitas dalam "pengetahuan yang dikomunikasikan".

Menurut Jogiyanto (2005:8) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Menurut Abdul Kadir (2003:31) informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Dari pengertian di atas disimpulkan bahwa informasi adalah hasil dari data yang telah diolah yang berarti bagi penerima atau pengguna dan berguna dalam mengambil keputusan saat ini atau keputusan masa yang akan datang.

Adapun informasi yang dihasilkan dari proses pengolahan data tersebut pada kenyataannya memiliki kualitas yang berbeda. Menurut Jogiyanto (2005:10) kualitas informasi tergantung dari tiga hal yaitu informasi harus akurat (*accurate*), tepat waktu (*timeliness*) dan relevan (*relevance*).

a. Akurat (*Accurate*)

Berarti informasi harus bebas dari kekurangan atau kesalahan-kesalahan. Akurat juga berarti informasi harus jelas mencerminkan keadaan yang sebenarnya.

b. Tepat waktu (*Timeliness*)

Berarti informasi harus selalu ada pada saat dibutuhkan pengguna dan tidak datang terlambat. Karena informasi merupakan landasan di dalam pengambilan keputusan. Jadi bila terlambat, informasi tersebut tidak mempunyai nilai lagi dan dapat berakibat fatal bagi organisasi.

c. Relevan

Informasi untuk tiap orang satu dengan yang lainnya berbeda. Maka informasi yang relevan adalah informasi yang berguna dan sesuai dengan kebutuhan setiap pemakainya.

2.3. Pengertian Sistem Informasi

Berdasarkan kesimpulan mengenai pengertian sistem dan informasi pada subbab sebelumnya, sistem informasi dapat didefinisikan sebagai kumpulan elemen-elemen yang saling berinteraksi satu sama lain untuk mencapai tujuan tertentu yaitu mengolah data menjadi bentuk yang lebih berarti dan berguna bagi penerimanya serta bermanfaat untuk pengambilan keputusan saat ini atau masa yang akan datang. Pengertian singkatnya sistem informasi adalah suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya.

Dalam bukunya, Abdul Kadir (2003:11) mengutip beberapa pendapat para ahli tentang pengertian dari sistem informasi, diantaranya :

Menurut Hall sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai.

Menurut Alter sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Dari ketiga pengertian di atas dapat disimpulkan bahwa yang dimaksud system informasi sesungguhnya tidak harus melibatkan komputer ketika proses mengolah data menjadi sebuah informasi. Adapun Sistem informasi yang melibatkan komputer biasa disebut sistem informasi berbasis komputer (*Computer-Based Information Systems* atau CBIS). Akan tetapi, pada umumnya kata system informasi diartikan sebagai sistem informasi berbasis komputer, meskipun tidak ditambahkan kata berbasis komputer. Begitupun pada pembahasan skripsi ini, sistem informasi yang dimaksud adalah sistem informasi berbasis komputer.

2.4. Web Sistem Informasi GIS

WebGis adalah *a Geographic Information System (GIS) distributed across a computer network to integrate, disseminate, and communicate geographic information visually on the World Wide Web* (Edward, 2000). Menurut Prahasta

(2007), WebGIS adalah aplikasi GIS atau pemetaan dijital yang memanfaatkan jaringan internet sebagai media komunikasi yang berfungsi mendistribusikan, mempublikasikan, mengintegrasikan, mengkomunikasikan dan menyediakan informasi dalam bentuk teks, peta dijital serta menjalankan fungsi–fungsi analisis dan query yang terkait dengan GIS melalui jaringan internet. Sedangkan menurut Setiawan dan Rabbasa, penggunaan data spasial dirasakan semakin diperlukan untuk berbagai keperluan seperti penelitian, pengembangan dan perencanaan wilayah, serta manajemen sumber daya alam. Pengguna data spasial merasakan minimnya informasi mengenai keberadaan dan ketersediaan data spasial yang dibutuhkan. Penyebaran (*diseminasi*) data spasial yang selama ini dilakukan dengan menggunakan media yang telah ada yang meliputi media cetak (peta), cd-rom, dan media penyimpanan lainnya dirasakan kurang mencukupi kebutuhan pengguna. Pengguna diharuskan datang dan melihat langsung data tersebut pada tempatnya (*data provider*). Hal ini mengurangi mobilitas dan kecepatan dalam memperoleh informasi mengenai data tersebut. Karena itu dirasakan perlu adanya WebGIS.

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

a) Pendekatan *Thin Server*

Pendekatan ini memfokuskan diri pada sisi *server*. Hampir semua proses dan analisis data dilakukan berdasarkan *request* di sisi *server*. Data hasil pemrosesan kemudian dikirim ke *client* dalam format standar.

b) Pendekatan *Thick Client*

Pada pendekatan ini, pemrosesan data dilakukan di sisi *clien* menggunakan beberapa teknologi (Nuryadin, 2005)

Secara umum pengembangan dan implementasi WebGIS akan menunjang penyebaran informasi data spasial. Sehingga orang awam pun akan dapat memiliki akses terhadap data dan hasil analisis GIS. Sekarang ada banyak *software* untuk WebGIS diantaranya adalah *Auto desk MapGuide, Integraph GeoMedia WebWap, ESRI ArcIMS.* Namun perangkat lunak WebGIS tersebut berharga sangat mahal. Maka alternatif penggantinya adalah WebGIS yang *open source* atau *royalti free*. Memang aplikasi *open source* merupakan aplikasi yang cukup sulit untuk dipelajari. Namun aplikasi alternatif ini dapat diterapkan tanpa menggunakan biaya yang besar (Nuryadin, 2005).

2.5. Geoserver

GeoServer adalah sebuah perangkat lunak mapserver yang open source ditulis di dalam bahasa pemrograman JAVA yang memungkinkan pengguna untuk berbagi dan mengedit data geospatial. Dirancang untuk interoperabilitas, menyajikan data dari semua sumber data spatial utama dengan menggunakan standar terbuka.

Karena menjadi proyek berbasis komunitas, GeoServer dikembangkan, diuji, dan didukung oleh berbagai kelompok individu dan organisasi dari seluruh dunia. GeoServer adalah implementasi referensi dari Open Geospatial Consortium (OGC) Web Feature Service (WFS) dan Web Coverage Service (WCS) standar,

serta telah memenuhi persyaratan bersertifikat Web Map Service (WMS). GeoServer menitikberatkan pada kemudahan penggunaan dan standar dalam menyajikan data geospatial lewat web.

GeoServer memungkinkan kita untuk menampilkan informasi spatial kepada dunia. Menerapkan Web Map Service (WMS) standar, GeoServer dapat membuat peta dalam berbagai format output seperti gambar/citra atau data geospatial. GeoServer dibangun di atas Geotools, open source Java GIS toolkit.

GeoServer juga sesuai dengan Web Feature Service (WFS) standar, yang memungkinkan berbagi dan pengeditan data yang digunakan untuk membuat peta. Orang lain dapat memasukkan data ke dalam situs web dan aplikasi, membebaskan perijinan sehingga memungkinkan transparansi yang lebih besar.

2.6. Database

Menurut Alex (2007), *database* merupakan kumpulan data yang saling berhubungan. Hubungan antar data dapat ditunjukan dengan adanya *field*/kolom kunci dari tiap *file*/tabel yang ada. Dalam satu *file* atau tabel terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* (umumnya digambarkan sebagai baris data) terdiri dari *field* yang saling berhubungan menunjukan bahwa *field* tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu *record*.

Elemen data/*field* dari pengertian diatas dapat disimpulkan bahwa *database* mempunyai beberapa kriteria penting, yaitu :

1. Bersifat data *oriented* dan bukan program *oriented*.

2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah *database*- nya.
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.

Prinsip utama *database* adalah pengaturan data dengan tujuan utama fleksibelitas dan kecepatan pada saat pengambilan data kembali. Adapun ciri-ciri *database* diantaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran, dan ketepatan.
2. Data dalam jumlah besar.
3. Berbagi pakai (dipakai bersama sama/*sharebility*).
4. Mengurangi bahkan menghilangkan terjadinya duplikasi data

Secara umum *database* dapat dibagi atas 2 jenis yaitu: (1) *database* spasial adalah suatu sistem *database* yang spesifik,mempunyai model data spasial dan *query* spasial, dan (2) database atribut adalah suatu database yang berisi pengenal atribut yang terdapat dalam *field*. *Database* atribut yang berhubungan dengan *database* spasial dapat berupa *database* eksternal yang terdapat di luar sistem.

2.7. Data Spatial

Data spasial merupakan salah satu item dari informasi, dimana didalamnya terdapat informasi mengenai bumi termasuk permukaan bumi, dibawah permukaan bumi, perairan, kelautan dan bawah atmosfir (Rajabidfard dan Williamson, 2000). Dalam perkembangannya data spasial merupakan salah satu

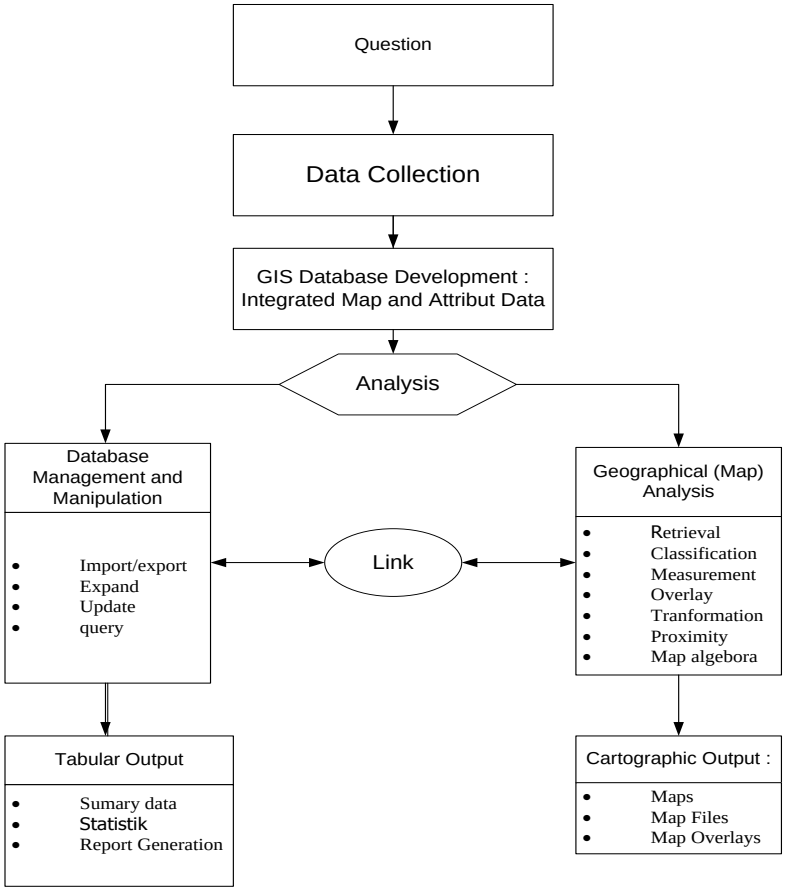
elemen penting sebagai sumber informasi dan dasar analisis di tingkat pemerintahan (bappeda,2007). Data spasial dan informasi turunannya digunakan untuk menentukan posisi dari identifikasi suatu elemen di permukaan bumi (Radjabidfard 2001). Pengetahuan mengenai lokasi dari suatu aktifitas memungkinkan hubungannya dengan aktifiktas lain atau elemen lain dalam daerah yang sama atau lokasi yang berdekatan dan lokasi memungkinkan diperhitungkannya jarak, pembuatan peta, memberikan arahan dalam membuat keputusan spasial yang bersifat kompleks.

Data spasial dapat dihasilkan dari berbagai macam sumber seperti, peta analog, citra satelit, foto udara, data tabular dan data survei. Data Spasial berupa titik, garis, poligon (2-D), permukaan (3-D), terdiri dari informasi posisi geografis.

2.8. Analisis Spasial

Menurut Aronoff (1991), fungsi spasial analisis adalah hal yang membedakan GIS dengan sistem lainnya. Fungsi ini menggunakan data spasial dan atribut yang berada dalam *database* GIS untuk menjawab pertanyaan dunia nyata. Dalam integrasinya dengan kedua jenis data ini yang disebut dengan *integrated analysis of spatial and attribute data*, fungsi *spatial analysis* dikelompokkan atas empat kelompok, yaitu : a) *retrieval, classification and measurement*, b) *overlay*, c) *neighbourhood* dan d) *connectivity or network function*. Masing-masing kelompok terdiri atas beberapa fungsi. Sedangkan Batty dan Longley (2001) membaginya atas enam bagian, yaitu : *queries and reasoning*,

measurement, tranformation, descriptive summaries, opotimation and hypothesis testing. *Retrieval and queries* adalah bagian dari *geography search*. Gambar 2.1 menunjukkan *flow chart* operasi analisis di dalam kerangka GIS (Falbo, Queen dan Blinn, 2002).



Gambar 2.1 : Operasi Analisis GIS (Falbo, Queen dan Blinn, 2002)

2.9. 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).

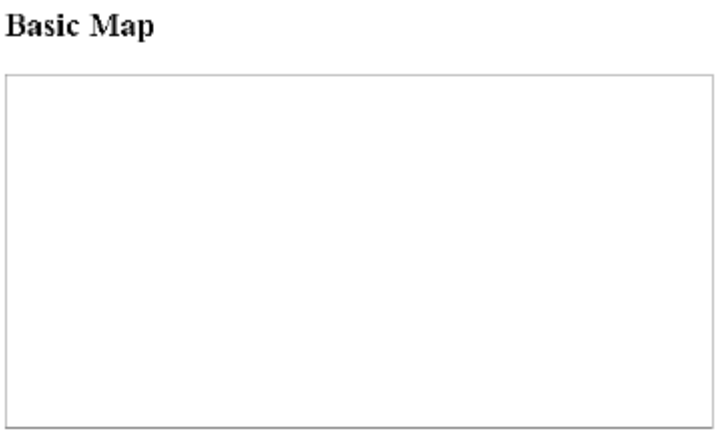
OpenLayers merupakan aplikasi dari AJAX, sehingga membuat tampilan peta menjadi lebih interaktif dan menarik. Pemakaian *OpenLayers* dilakukan dengan memanggil *script* ini ke dalam peta dasar yang telah disediakan. Peta yang akan ditampilkan pada halaman web diletakkan pada suatu *tag* khusus di halaman 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). Contoh kode untuk membuat *basic map* dapat dilihat pada gambar 2.2.

```
<html>
  <head>
    <title>Basic Map</title>
    <style type="text/css">
      #map_container {
        width: 512px;
        height: 256px;
        border: 1px solid gray;
      }
    </style>
    <script src="openlayers/OpenLayers.js">
    </script>
  </head>
  <body>
    <h2>Basic Map</h2>
    <div id="map_container"></div>
    <script>
      // your application code here
    </script>
  </body>
</html>
```

Gambar 2.2 : Pembuatan basic map pada halaman HTML

(http://openlayers.org/dev/doc/examples, 2008)

Basic map pada halaman HTML merupakan peta kosong. *Basic map* ini digunakan untuk menampung peta yang dipanggil dengan menggunakan *OpenLayers* . *Basic map* yang dihasilkan pada halaman HTML dapat dilihat pada gambar 2.3.



Gambar 2.3 : Basic map pada halaman HTML

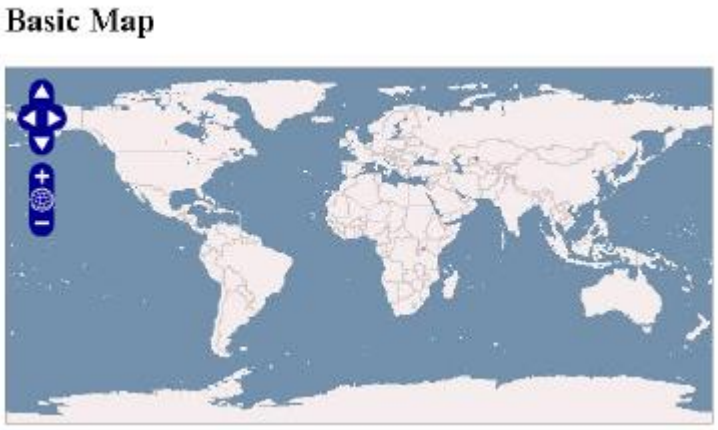
(http://openlayers.org/dev/doc/examples, 2008)

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*. Cara mendefenisian layer ini dapat dilihat dari gambar 2.4 dan hasil pada halaman web HTML dapat dilihat pada gambar 2.5

```
var map = new
        OpenLayers.Map("map_container");
var wms = new OpenLayers.Layer.WMS(
    "OpenLayers WMS",
    "http://labs.metacarta.com/wms/vmap0",
    {layers: "basic"}
);
map.addLayer(wms);
map.zoomToMaxExtent();
```

Gambar 2.4: Kode program untuk mendefenisikan layer

(http://openlayers.org/dev/doc/examples, 2008)



Gambar 2.5 : Peta yang dihasilkan pada halaman HTML

(http://openlayers.org/dev/doc/examples, 2008)

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)

Layer adalah sebuah objek yang ditanamkan pada objek ‘map’ dimana objek layer selalu berhubungan dengan sumber *data spatial* (dalam penelitian ini menggunakan postGIS) sehingga ketika terjadi event drag atau seret pada objek ‘map’ yang membuat gambar map bergeser mengakibatkan objek leyer juga bergeser sesuai dengan posisi koordinatnya. Layer pada map dapat ditambahkan dengan menggunakan fungsi yang telah disediakan oleh openLayers (http://openlayers.org/dev/doc/examples, 2008).

Berikut ini jenis fungsi pada *openLayers* yang digunakan dalam penelitian ini (untuk penggunaan fungsi dalam kode program dapat dilihat pada lampiran tugas akhir ini) :

1. OpenLayers.Map()

Class pada *openLayers* yang berguna untuk menghasilkan peta. Untuk membuat instan dari class ini menggunakan *Javascript* operator *new*. Untuk menampung object dari peta tersebut ke dalam elemen div, digunakan method document.getElementById(). Fungsi Map() adalah sebagai constructor (http://openlayers.org/dev/doc/ examples, 2008)

2. OpenLayers.Bounds()

Class pada openLayers yang berguna untuk menentukan *proyeksi* dan *bounds* peta yang akan digunakan (http://openlayers.org/dev/doc/ examples, 2008). Pada penelitian ini proyeksi peta EPSG : 23847 .

3. OpenLayers.Layer.Vector()

Class pada openLayers yang berfungsi untuk menambahkan objek berupa vector pada peta. Untuk membuat objek ini dibutuhkan input berupa titik

dengan menggunakan fungsi OpenLayers. Geometry. (point.x,point.y), dan sebagai penanda dari objek ini digunakan fungsi OpenLayers. Feature.Vector() yang merupakan class untuk membuat icon pada peta. (http://openlayers.org/dev/doc/ examples, 2008).

4. map.addLayers()

Fungsi untuk menambahkan layer pada peta.

(http://openlayers.org/dev/doc/ examples, 2008).

5. map.addControl()

Fungsi untuk menambahkan *control* peta pada halaman web. (http://openlayers.org/dev/doc/ examples, 2008).

Contoh penggalan kode program untuk mendefenisian layer dengan *OpenLayers* dan menggunakan *geosever* sebagai penghasil peta

```
var bounds = new OpenLayers.Bounds(
    112.28791809082, -1.13623702526093,
    112.512008666992, -0.785094976425171
        );
 var options = {
    controls: [],
    maxExtent: bounds,
    maxResolution: 0.0013716486282647,
    projection: "EPSG:23847",
    units: 'm'
 };
```

```
map = new OpenLayers.Map('map', options);

tiled = new OpenLayers.Layer.WMS(
    "Geoserver layers - Tiled", "http://localhost:8080/geoserver/wms",
    {
        layers: 'padang',
        styles: '',
        height: '550',
        width: '401',
        srs: 'EPSG:23847',
        format: format,
        tiled: 'true',
        tilesOrigin : "112.28791809082,-1.13623702526093"
    },
    {buffer: 0}
);

untiled = new OpenLayers.Layer.WMS(
    "Geoserver layers - Untiled", "http://localhost:8080/geoserver/wms",
    {
        layers: 'padang',
        styles: '',
```

```
            height: '550',
            width: '401',
            srs: 'EPSG:23847',
            format: format
        },
        {singleTile: true, ratio: 1}
    );
    map.addLayers([untiled, tiled]);
```

2.10. PostgreSQL / PostGIS

PostGIS adalah sebuah *database* spasial yang terdapat pada *server* PostgreSQL yang didukung oleh semua fungsi dan objek yang sudah didefenisikan dalam openGIS "*Simple Features for SQL specification*". Dengan menggunakan fungsi spasial yang ada dalam PostGIS kita dapat melakukan analisa spasial dan *query* spasial. Seperti *Oracle Spatial*, *DB2 Spatial*, dan *Server Spatia*l. PostGIS menambahkan kemampuan kepada PostgreSQL untuk dapat melakukan pengolahan data spasial. PostGIS dapat juga dinamakan sebagai *PostgreSQL Spatial*, yang mempunyai kepemilikan terhadap *spatial database extension* (Ramsey, 2000).

PostGIS didesain untuk mengimplementasikan lingkungan SQL92 *with Geometry Types* pada *database* PostgreSQL. PostGIS versi 0.1 di-*release* tahun 2001 dengan dukungan pada tipe data objek spasial dan metoda pengindeksan objek spasial menggunakan GiST. Perkembangan lebih lanjut adalah

implementasi spesifikasi *OpenGIS Consortium* yang lebih baik. Secara singkat fasilitas PostGIS adalah sebagai berikut (Ramsey, 2000):

1. Definisi *Abstract Data Types* untuk objek spasial sesuai spesifikasi *OpenGIS Consortium*
2. Dukungan terhadap format WKT (*Well Known Text*) dan WKB (*Well Known Binary*)
3. Metoda pengindeks-an GIST yang sesuai untuk objek spasial
4. Dukungan akses aplikasi melalui JDBC

Sejak versi 0.5 PostGIS merupakan salah satu sumber data untuk perangkat lunak MapServer (*University of Minnesota).* Sejak saat itu, data PostGIS dapat dilihat secara visual dengan menggunakan MapServer. Versi terakhir menambahkan dukungan transformasi koordinat melalui pustaka PROJ dan dukungan lengkap sistem referensi spasial berdasarkan EPSG (Ramsey, 2000).

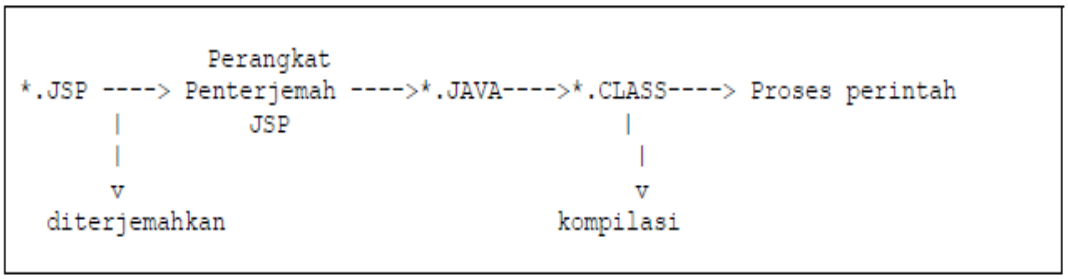
2.11. Java Server Page (JSP)

JSP (*JavaServer Pages*) adalah suatu bahasa pemrograman web (*scripting*)yang bersifat *server side* yang menggabungkan HTML dengan *scripting tag* dan program java. Suatu dokumen JSP ditandai dengan berkas ekstensi **.**jsp. *Java Server Pages* (JSP) merupakan sebuah teknologi *servlet-based* yang digunakan pada *web tier* untuk menghadirkan dinamik dan statik *content*. JSP merupakan *text-based* dan kebanyakan berisi template text HTML yang digabungkan dengan spesifik *tags dynamic content* (Foenadioen, 2007).

JSP dapat berupa gabungan antara baris HTML dan fungsi-fungsi dari JSP itu sendiri. Berbeda dengan *servlet* yang harus dikompilasi oleh USER menjadi *class* sebelum dijalankan, JSP tidak perlu dikompilasi oleh USER tapi SERVER yang akan melakukan tugas tersebut. Pada saat user membuat pertama kali atau melakukan modifikasi halaman dan mengeksekusinya pada web browser akan memakan sedikit waktu sebelum ditampilkan.

(www.IlmuKomputer.com/PengenalanJSP/, 2007).

Daur hidup JSP dapat digambarkan seperti pada gambar 2.6.



Gambar 2.6 : Skema JSP

(www.IlmuKomputer.com/PengenalanJSP/, 2007)

Kelebihan JSP sebagai pendukung pemgembangan aplikasi berbasis web yang handal adalah :

1. JSP merupakan dokumen text seperti HTML, para pengembang menghindari format dan manipulasi yang memungkinkan *String* yang sangat panjang untuk menghasilkan output. *Content HTML* sekarang tidak ditempelkan dengan berbagai macam kode dari Java. Hal ini membuatnya lebih mudah untuk dipelihara.

2. JSP lebih dikenal oleh semua orang dengan pengetahuan dari HTML, hanya dengan mempelajari *markup dynamic*. Hal ini membuatnya mungkin untuk para desainer site untuk membuat template HTML dari sebuah site, dengan para pengembang memprosesnya suatu saat nanti untuk memasukkan *tags* yang menghasilkan *dynamic content*. Hal ini juga memudahkan dalam pengembangan *web page*.

3. JSP memiliki *built-in* yang mendukung untuk penggunaan komponen software yang dapat digunakan kembali (JavaBeans). Hal ini tidak hanya membiarkan para pengembang menghindari kemungkinan menemukan kembali inti / kemudi dari tiap aplikasi, mempunyai software pendukung untuk memisahkan komponen software untuk *menghandle logic promotes separation* dari presentasi dan *business logic.*

4. JSP merupakan bagian solusi dari Java untuk pengembang aplikasi web, merupakan *multi-platform* yang tak terpisahkan dan dapat dijalankan pada berbagai *container servlet* yang *compatible*, dengan mengabaikan vendor atau sistem operasinya.

5. Dalam kaitannya dengan cara kerja JSP, mereka tidak membutuhkan kompilasi dari para pengembang. Kompilasi ini telah ada untuk kita pada kontainer servlet. Modifikasi JSP dideteksi secara otomatis. Hal ini secara relatif membuatnya mudah untuk dibangun.

(www.IlmuKomputer.com/PengenalanJSP, 2007).

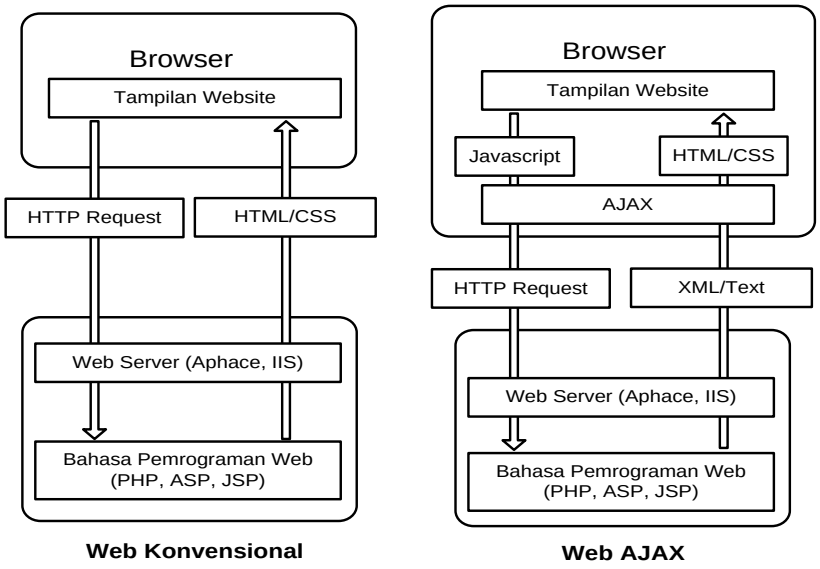
2.12. AJAX (Asynchronous JavaScript And XML)

AJAX merupakan sebuah teknik pemanfaatan *object XMLHttpRequest* dengan *javascript* untuk berkomunikasi dengan server secara *Asynchronous*. Pemanfaatan o*bject XMLHttpRequest* membuat proses berjalan secara *background* atau bekerja dibelakang layar, sementara user dapat tetap berinteraksi dengan halaman web yang ada. AJAX merupakan singkatan dari "*Asynchronous JavaScript And XML*". Istilah AJAX dalam pengembangan web menjadi popular pada beberapa tahun belakangan ini (Heri, 2007).

AJAX bukan merupakan bahasa pemrograman. AJAX hanya merupakan sebuah teknik pemanfaatan *object XMLHttpRequest* dengan *javascript* untuk berkomunikasi dengan server secara *Asynchronous* (Jerry, 2009). Teknik komunikasi *Asynchronous* dengan server sendiri pertama kali dikembangkan oleh microsoft pada tahun 1997. *Object XMLHttpRequest* pertama kali diperkenalkan pada IE5 (circa 1998) dan kemudian dipergunakan secara luas pada Outlook web access. Jesse James Garrett's memberikan istilah AJAX untuk teknik *Asynchronous* ini pada salah satu tulisannya di tahun 2005.

AJAX menggunakan teknologi lama yaitu *javascript* yang melakukan *request* ke server untuk meminta data dalam bentuk Text/XML. Dengan menggunakan konsep ini, *page* HTML bisa membuat koneksi secara asinkron ke server dengan cara mengambil XML atau dokumen text. Selanjutnya XML atau dokumen text ini akan digunakan oleh *javascript* untuk meng-*update* atau memodifikasi *Document Object Model* (DOM) pada *page* HTML. Perbandingan

diagram proses web browser ke server pada web konvensional dan web yang menggunakan AJAX dapat dilihat pada gambar 2.7.

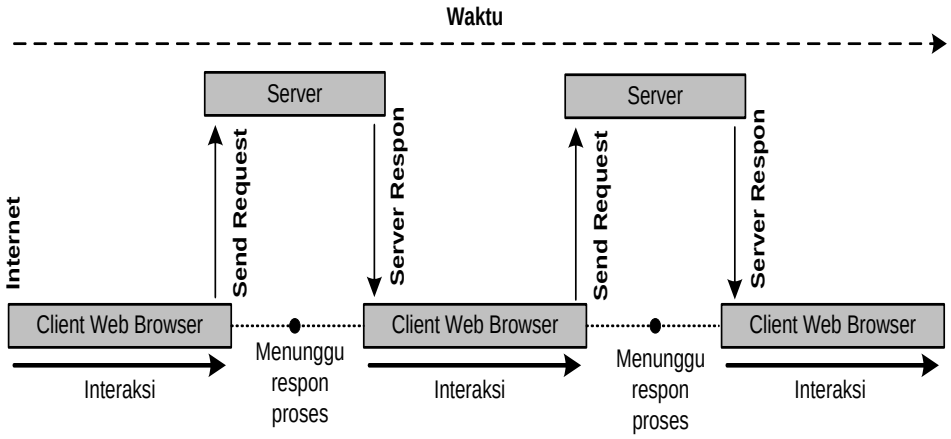


Gambar 2.7 : Perbandingan Web Konvensional dan AJAX

(http://www.dhimasronggobramantyo.com/kategori/6/1, Agustus 2009)

Proses Kerja AJAX

Pada proses pengembangan web dikenal adanya 2 *request* proses, yaitu *Synchronous* dan *Asynchronous*. Kedua proses ini berbeda pada segi pengiriman dan penerimaan respon dari server. Proses *Synchronous* pada pengembangan web tradisoinal dapat dilihat pada gambar 2.8.



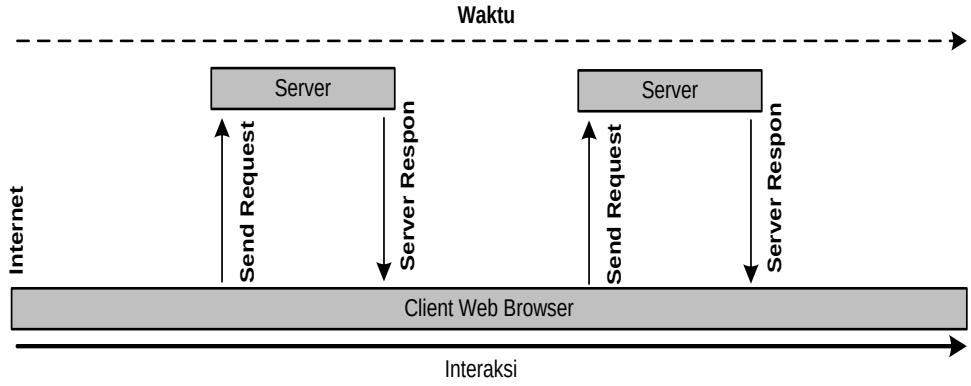
Gambar 2.8 : Proses SYNCHRONOUS

(www.IlmuKomputer.com/Berkenalan dengan AJAX/, 2007)

Saat user berinteraksi dengan sebuah halaman web terdapat banyak pilihan *link* yang dapat ditemukan, dan saat link tersebut di *click* maka halaman web tersebut akan melakukan komunikasi dengan server melalui internet untuk meminta proses ke server (*send Request*). Selama *server* melakukan proses, *user* akan menunggu hasil proses tersebut. Setelah *server* selesai melakukan proses yang diminta maka *server* akan mengembalikan hasil proses yang akan ditampilkan pada *client webbrowser*. Dalam proses *synchronous* ini user melakukan permintaan proses (*request process*) ke *server* dan menunggu hingga *server* mengembalikan hasil proses yang diminta, umumnya hasil proses dari server merupakan sebuah halaman web baru yang ditampilkan pada web *browser user* (Jerry, 2009).

Proses *Asynchronous* dalam pengembangan web menggunakan teknologi AJAX. Pada proses ini *user* masih dapat untuk tetap berinteraksi

dengan halaman web yang ada selama proses *request* dilakukan dan selama *server* belum mengembalikan hasil proses yang diminta. Saat hasil proses telah selesai, kita hanya perlu mengupdate data halaman web yang telah ada. Proses kerja *ASYNCHRONOUS* dapat dilihat pada gambar 2.9.



Gambar 2.9. : Proses ASYNCHRONOUS

(www.IlmuKomputer.com/Berkenalan dengan AJAX/, 2007)

User berinteraksi dengan link-link yang ada untuk meminta proses ke *server*, proses yang diminta akan dikirimkan secara *background* ke *server*, selama *server* belum mengembalikan data hasil proses, user dapat tetap bekerja dengan halaman web yang ada sebelumnya. Setelah *server* selesai melakukan proses, hasil proses tersebut akan dikirmkan kembali kepada *web browser*, saat data yang dikirimkan server telah diterima oleh *webbrowser* user maka data tersebut ditampilkan pada halaman web yang telah ada sebelumnya. Semua proses komunikasi dengan server dilakukan secara *background* atau mungkin bisa dikatakan semua proses dilakukan *Behind the Scene* (Jerry, 2009).

Untuk dapat mengembangkan aplikasi web dengan teknik AJAX ini kita perlu mengkombinasikan beberapa hal berikut:

1. Javascript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML. Sepanjang sejarah internet, bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman yang memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengeksekusian perintah - perintah di sisi user, yang artinya di sisi browser bukan di sisi server web. (Andry, 2003 **).** Javascript digunakan untuk membuat *object XMLHttpRequest* yang digunakan untuk berkomunikasi dengan server secara *asynchronous*.

2. DOM (*Document Object Model*)

Document Object Model (DOM) adalah object model standar untuk HTML dan XML yang bersifat platform independent. Sebuah web browser tidak harus menggunakan DOM untuk menampilkan dokumen HTML. Namun DOM diperlukan oleh *javascript* untuk mengubah tampilan sebuah website secara dinamis. Dengan kata lain, DOM adalah cara *javascript* melihat suatu halaman HTML.(wikipedia.org, 2009)

DOM disini adalah struktrur object dari sebuah document XML/HTML yang berisi tag, attribut, id dan sebagainya. Dengan mengetahui struktur DOM nantinya dapat dilakukan pembacaan data-data tag. Misalnya terdapat document dengan tag HTML berikut:

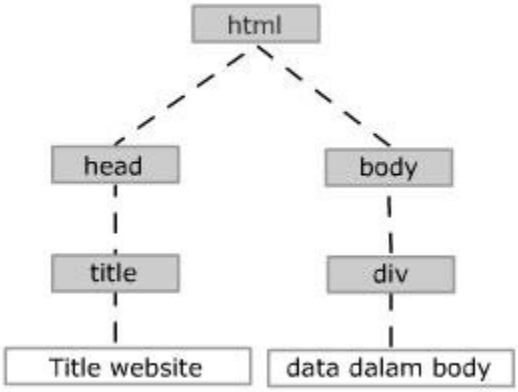
<html>

<head> <title>Title web</title></head>

<body> data dalam body </body>

</html>

Diagram Struktur *Document Object Model* (DOM) pada data tersebut dapat dilihat pada gambar 2.10.



Gambar 2.10 : Diagram Struktur DOM

(www.IlmuKomputer.com/Berkenalan dengan AJAX/, 2007)

3. XML (*eXtensible Markup Language*)

XML merupakan format data yang dikembalikan oleh server. Data XML ini siap dibaca dan ditampilkan untuk mengupdate content pada halaman web. Sebuah data dalam format XML akan memiliki TAG, ATTRIBUT atau juga ID. Untuk membaca data dalam format tersebut dapat mengunakan fasilitas yang ada dalam DOM berikut:

1) getElementsByTagName : digunakan untuk membaca tag berdasarkan NAMA
2) getElementById : digunakan untuk membaca tag berdasarkan ID yang diberikan
3) getAttribute : digunakan untuk membaca tag berdasarkan ATTRIBUTE
4) firstChild : membaca Node array pertama dari data tag yang dibaca
5) lastChild : membaca Node array terakhir dari data tag yang akan dibaca.
6) data : membaca data text dari tag atau node yang diinginkan
7) innerHtml : membaca/menuliskan data kedalam bagian tag.
8) InnerHTML berbeda dengan property data sebelumnya karena innerHTML dapat berupa sebuah struktur data yang ada didalam tag yang ambil dan didalamnya mungkin akan dapat berisi tag - tag baru lagi (www.IlmuKomputer.com/Berkenalan dengan AJAX/, 2007).