

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi Geografis

2.1.1 Pengertian Sistem Informasi Geografis (SIG)

Menurut Prasetyo (2003) SIG adalah suatu sistem yang meng-*capture*, mengecek, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang secara spasial (keruangan) mereferensikan kepada kondisi bumi (Prasetyo, 2003).

Menurut Mohd dkk, (2010). Sistem informasi geografis adalah sebuah sistem untuk pemetaan dan menganalisa obyek yang ada di bumi yang dilakukan dengan cara mengumpulkan, menyimpan dan mengelola beberapa tipe data dengan komponen data spasial. Data SIG biasanya menyimpan lebih dari satu *layer*. Ini adalah aspek dasar dari SIG, dan bekerja dengan *layer* dari informasi geografi yang secara umum disebut sebagai *data integration*. Teknologi SIG sudah terintegrasi dengan *database* yang *powerful* dengan tampilan yang unik dibandingkan dengan peta biasa.

Pada dasarnya sistem ini memanfaatkan *hardware*, *software*, *user* dan keefektifan dalam mengumpulkan, menyimpan, menganalisis dan menampilkan informasi yang berhubungan dari suatu daerah yang ada di bumi.

Definisi SIG selalu berubah karena SIG merupakan bidang kajian ilmu dan teknologi yang relatif masih baru. Dibawah ini pengertian lain dari SIG :

SIG merupakan suatu sistem yang mengorganisir perangkat keras (*hardware*), perangkat lunak (*software*), dan data, serta dapat mendaya-gunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh informasi yang berkaitan dengan aspek keruangan (Husein, 2003).

Dari penggabungan dari beberapa pengertian di atas dapat diambil kesimpulan bahwa SIG adalah suatu sistem yang terdiri dari *hardware*, *software*, dan data yang saling bekerjasama sehingga dapat digunakan untuk meng-*capture*, menganalisa, mengintegrasikan dan menampilkan data yang berkaitan dengan aspek keruangan.

Alasan SIG dibutuhkan adalah karena penanganan data spasial sangat sulit terutama karena peta dan data statistik cepat kadaluarsa sehingga tidak ada pelayanan penyediaan data dan informasi yang diberikan menjadi tidak akurat.

Berikut adalah dua keistimewaan analisa melalui *Geographical Information System (GIS)* (Husein, 2003) :

1. *Analisa Proximity*

Analisa Proximity merupakan suatu geografi yang berbasis pada jarak antar *layer*. Dalam analisis *proximity* SIG menggunakan proses yang disebut dengan *buffering* (membangun lapisan pendukung sekitar *layer* dalam jarak tertentu untuk menentukan dekatnya hubungan antara sifat bagian yang ada).

2. *Analisa overlay*

Proses integrasi data dari lapisan-lapisan *layer* yang berbeda disebut dengan *overlay*. Secara analisa membutuhkan lebih dari satu *layer* yang akan ditumpang susun secara fisik agar bisa dianalisa secara visual (Husein, 2003).

2.1.2 Kemampuan SIG (Prasetyo, 2003)

1. Memetakan Letak

Data realita di permukaan bumi akan dipetakan ke dalam beberapa *layer* dengan setiap *layernya* merupakan representasi kumpulan benda (*feature*) yang mempunyai kesamaan, contohnya *layer* jalan, *layer* bangunan, dan *layer customer*. *Layer-layer* ini kemudian disatukan dengan disesuaikan urutannya. Setiap data pada setiap *layer* dapat dicari, seperti halnya melakukan *query* terhadap *database*, untuk kemudian dilihat letaknya dalam keseluruhan peta.

2. Memetakan Kuantitas

Orang sering memetakan kuantitas, yaitu sesuatu yang berhubungan dengan jumlah, seperti dimana yang paling banyak atau dimana yang paling sedikit. Dengan melihat penyebaran kuantitas tersebut dapat dicari tempat-tempat yang sesuai dengan kriteria yang diinginkan dan digunakan untuk pengambilan keputusan, ataupun juga untuk mencari hubungan dari masing-

masing tempat tersebut. Pemetaan ini akan lebih memudahkan pengamatan terhadap data statistik dibanding *database* biasa.

3. Memetakan Kerapatan

Sewaktu orang melihat konsentrasi dari penyebaran lokasi dari fitur-fitur, di wilayah yang mengandung banyak *feature* mungkin akan mendapat kesulitan untuk melihat wilayah mana yang mempunyai konsentrasi lebih tinggi dari wilayah lainnya. Peta kerapatan dapat mengubah bentuk konsentrasi kedalam unit-unit yang lebih mudah untuk dipahami dan seragam, misalnya membagi dalam kotak-kotak selebar 10 km², dengan menggunakan perbedaan warna untuk menandai tiap-tiap kelas kerapatan.

4. Memetakan Perubahan

Dengan memasukkan variabel waktu, SIG dapat dibuat untuk peta historikal. Histori ini dapat digunakan untuk memprediksi keadaan yang akan datang dan dapat pula digunakan untuk evaluasi kebijaksanaan. Pemetaan jalur yang dilalui badai, dapat digunakan untuk memprediksi kemana nantinya arah badai tersebut. Seorang manajer pemasaran dapat melihat perbandingan peta penjualan sebelum dan sesudah dilakukannya promosi untuk melihat efektivitas dari promosinya.

5. Memetakan Apa yang Ada di Dalam dan di Luar Suatu Daerah.

SIG digunakan juga untuk memonitor apa yang terjadi dan keputusan apa yang akan diambil dengan memetakan apa yang ada pada suatu area dan apa yang ada di luar area.

2.1.3 Aplikasi Sistem Informasi Geografis

Pada sebuah aplikasi SIG, terdapat beberapa fasilitas yang merupakan standar untuk melengkapi peta yang tampil pada layar monitor. antara lain :

1. Legenda

Legenda (*legend*) adalah keterangan tentang obyek-obyek yang ada di peta, seperti warna hijau adalah hutan, garis merah adalah jalan, simbol buku adalah universitas, dan sebagainya.

2. Skala

Skala adalah keterangan perbandingan ukuran di layar dengan ukuran sebenarnya.

3. *zoom in / out*

Peta di layar dapat diperbesar dengan *zoom in* dan diperkecil dengan *zoom out*.

4. Pan

Dengan fasilitas pan peta dapat digeser-geser untuk melihat daerah yang dikehendaki.

5. *Searching*

Fasilitas ini digunakan untuk mencari dimana letak suatu *feature*. Bisa dilakukan dengan meng-inputkan nama atau keterangan dari *feature* tersebut.

6. Pengukuran

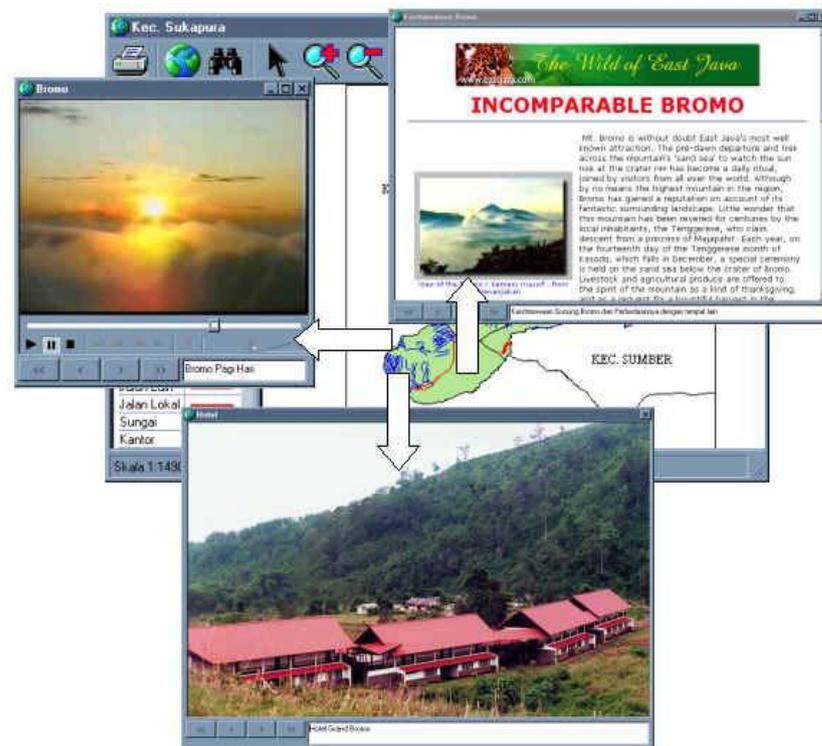
Fasilitas ini dapat mengukur jarak antar titik, jarak rute, atau luas suatu wilayah secara interaktif.

7. Informasi

Setiap *feature* dilengkapi dengan informasi yang dapat dilihat jika *feature* tersebut diklik. Misal pada suatu SIG jaringan jalan, jika diklik pada suatu ruas jalan akan memunculkan data nama jalan tersebut, tipe jalan, desa-desa yang menjadi ujung jalan, dan jalan-jalan lain yang berhubungan dengan jalan itu.

8. Link

Selain informasi dari *database*, SIG memungkinkan pula menghubungkan data *feature* pada peta dengan data dalam bentuk lain seperti gambar, video, ataupun web. Pada Gambar 2.1 adalah contoh *link* dari peta tentang probolinggo yang jika diklik pada bagian Gunung Bromo akan memunculkan video, gambar-gambar, dan web tentang gunung Bromo (Prasetyo, 2003).



Gambar 2.1 Peta yang di-Link Dengan Foto, Video, dan Web

2.1.4 Komponen Sistem Informasi Geografis

1. *Hardware*

Terdiri dari beberapa komponen,

- a. CPU (Central Processing Unit)
- b. *Memory* (Utama dan Tambahan)
- c. *Storage* (alat penyimpanan data dan informasi)
- d. Alat tambahan (*Peripherals*)

Alat masukan (*Input Devices*) : *keyboard*, *mouse*, *digitizers*, pemindai (*scanner*), kamera digital, *workstation* fotogrametris digital.

Alat keluaran (*Output Devices*) : monitor berwarna, *printer*, *plotter* berwarna, perekam film, dan lain-lain.

2. *Software*

Perangkat lunak yang dimaksud adalah yang mempunyai fungsi: pemasukan data, manipulasi data, penyimpanan data, analisis data, dan penayangan informasi geografis.

Beberapa persyaratan yang harus dipenuhi dari Software SIG :

- a. Merupakan *Database Management System* (DBMS).
- b. Memiliki fasilitas Pemasukan dan Manipulasi data Geografi.
- c. Memiliki fasilitas untuk *Query*, Analisis, dan Visualisasi.
- d. Memiliki Kemampuan *Graphical User Interface* (GUI) yang dapat menyajikan hasil (penayangan dan *Printout*) informasi berbasis geografis dan memudahkan untuk akses terhadap seluruh fasilitas yang ada.

3. Data dan Informasi

Data yang dapat diolah dalam sistem informasi geografis merupakan fakta-fakta di permukaan bumi yang memiliki referensi keruangan baik referensi secara relatif maupun referensi secara absolut dan disajikan dalam sebuah peta.

1. Referensi Relatif

Berarti suatu data yang memiliki referensi geografis. Data ini dapat digunakan jika sudah dikaitkan dengan data yang memiliki referensi geografis. Misalnya adalah data jumlah penduduk per kabupaten dikaitkan dengan data administrasi kabupaten.

2. Referensi absolut

Berarti suatu data yang memiliki referensi geografis (sudah memiliki koordinat tertentu di permukaan bumi). Misalnya adalah data titik-titik yang diperoleh dengan menggunakan GPS (*Global Positioning System*).

3. Sumber Daya Manusia

Sumber daya manusia yang terlatih merupakan sebagai komponen terakhir dari SIG. Perannya adalah sebagai pengoperasi perangkat keras dan perangkat lunak, serta menangani data geografis dengan kedua perangkat tersebut (Riyanto dkk, 2009).

2.2 Perangkat Lunak SIG

Pada saat ini, sudah banyak tersedia (baik yang komersial maupun yang *open-source tool* atau perangkat lunak yang berfungsi baik hanya sebagai “*map server*” didalam (arsitektur) aplikasi web maupun sekedar pendukung dalam menampilkan

unsur-unsur spasial yang terdapat didalam ‘*view*’ ke dalam aplikasi browser internet. Perangkat lunak tersebut dengan masing-masing karakter yang khas diantaranya adalah : ArcIMS, MapServer, MapExtreeme, MapObjects IMS, ArcViews IMS, Demis, Autodesk MapGuide, GeoMedia, GeoMedia WebMap Publisher, Alov Map, CartoWeb, BeyondGeo, ReefBase, Kmap, WorldKit, Stylus Studio XML Mapping Tools, Quest, Geoid, GeoNetwork *opensource*, dan masih banyak lagi (Prahasta, 2007).

2.2.1 MapServer

2.2.1.1 Tentang MapServer

MapServer (<http://mapserver.gis.umn.edu>) merupakan aplikasi *freeware* dan *open source* yang memungkinkan kita menampilkan data spasial (peta) di web. Aplikasi ini pertama kali dikembangkan di Universitas Minesotta, Amerika Serikat untuk proyek *ForNet* (sebuah proyek untuk manajemen sumber daya alam) yang disponsori NASA (*National Aeronautics and Space Administration*). *Support* NASA dilanjutkan dengan dikembangkan proyek *TerraSIP* untuk manajemen data lahan. Saat ini, karena sifatnya yang terbuka (*open source*), pengembangan MapServer dilakukan oleh pengembang dari berbagai negara.

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

untuk menangani beragam proyeksi peta. Pada bentuk paling dasar, MapServer berupa sebuah program CGI (*Common Gateway Interface*). Program tersebut akan dieksekusi di *web server*, dan berdasarkan beberapa parameter tertentu (terutama konfigurasi dalam bentuk file *.MAP) akan menghasilkan data yang kemudian akan dikirim ke web browser, baik dalam bentuk gambar peta ataupun bentuk lain. MapServer mempunyai fitur-fitur sebagai berikut:

1. Menampilkan data spasial dalam format vektor seperti: *Shapefile* (ESRI), *ArcSDE* (ESRI), *PostGIS* dan berbagai format data vektor lain dengan menggunakan *library OGR*.
2. Menampilkan data spasial dalam format raster seperti: *TIFF/GeoTIFF*, *EPPL7* dan berbagai format data raster lain dengan menggunakan *library GDAL*.
3. Menggunakan *quadtree* dalam *indexing* data spasial, sehingga operasi-operasi spasial dapat dilakukan dengan cepat.
4. Dapat dikembangkan (*customizable*), dengan keluaran yang dapat diatur menggunakan file-file *template*.
5. Dapat melakukan seleksi obyek berdasar nilai, berdasar titik, area, atau berdasar sebuah obyek spasial tertentu.
6. Mendukung *rendering* karakter berupa *font TrueType*.
7. Mendukung penggunaan data raster maupun vektor yang di-*tiled* (dibagi-bagi menjadi sub bagian yang lebih kecil sehingga proses untuk mengambil dan menampilkan gambar dapat dipercepat).
8. Dapat menggambarkan elemen peta secara otomatis: skala grafis, peta indeks dan legenda peta.

9. Menggunakan skala dalam penggambaran obyek spasial.
10. Dapat menggambarkan peta tematik yang dibangun menggunakan ekspresi logik maupun ekspresi reguler.
11. Dapat menampilkan *label* dari obyek spasial, dengan *label* dapat diatur sedemikian rupa sehingga tidak saling tumpang tindih.
12. Konfigurasi dapat diatur secara *on the fly* melalui parameter yang ditentukan URL.
13. Dapat menangani beragam sistem proyeksi secara *on the fly*.

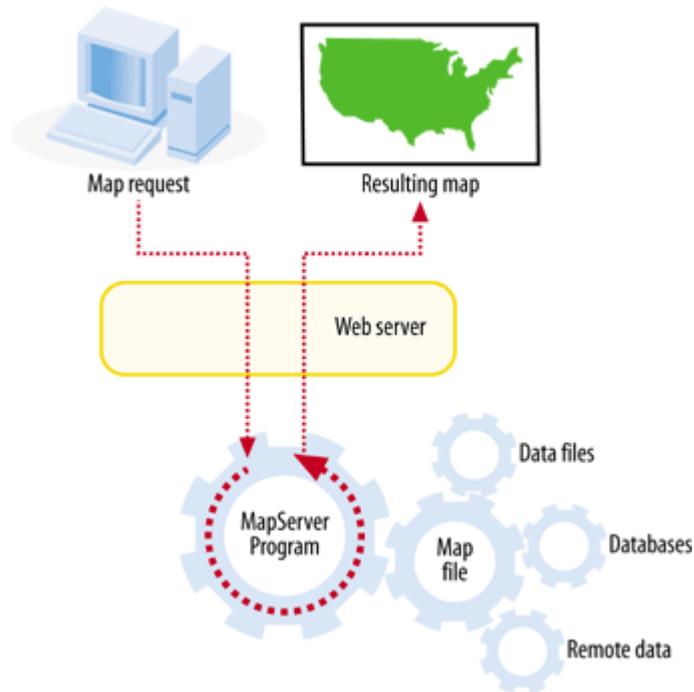
(Nryadin, 2005).

2.2.1.2 MS4W – MapServer for Windows

MS4W adalah suatu paket perangkat lunak yang sangat memudahkan para pengguna di dalam meng-*install* (atau melakukan *set-up*) MapServer pada *platform* sistem operasi Windows. Tujuan utama pembuatan paket ini adalah untuk memudahkan semua (tingkatan) pengguna, secepatnya (terhindar dari segala detil yang rumit), di dalam mempersiapkan lingkungan Ms. Windows (Prahasta, 2007).

2.2.1.3 Cara Kerja MapServer

MapServer bekerja secara berdampingan dengan aplikasi *web server*. *Web server* menerima *request* peta kemudian mengirimkan kembali ke MapServer untuk di buat. MapServer *generate request* image peta dan mengirimkan ke *web server*, kemudian *web server* akan mentransmisikan kembali untuk ditampilkan kepada *user*. Gambar 2.1 menunjukkan diagram operasi standar dari MapServer.



Gambar 2.2 Diagram Operasi Standar MapServer

Fungsi utama dari MapServer adalah melakukan pembacaan data dari banyak sumber dan menemukannya dalam *layer-layer* secara bersamaan menjadi file gambar. Salah satu *layer*-nya bisa saja berupa citra satelit. Sebuah contoh untuk tampilan *overlapping* yang baik dapat dilihat pada Gambar 2.2. dapat dilihat sebuah citra satelit (didapat dari *remote server*) garis-garis jalan, lokasi perkotaan, dan *label-label* kota yang di-*generated* secara dinamis oleh MapServer. Proses penggambaran peta (*rendering*) muncul setiap kali permintaan terhadap peta baru yang dilakukan oleh MapServer termasuk ketika *user* melakukan level *zoom* terhadap tampilan peta.



Gambar 2.3 Peta Dengan Berbagai Informasi *Layer* (Mitchell, 2005).

2.2.1.4 MapServer *Executable*

Secara sederhana MapServer menjalankan *executable* aplikasi CGI pada *web server* yang secara teknis merupakan prose *stateless*. *Stateless* adalah merupakan sebuah proses *request* dan dilanjutkan dengan proses *stop running*. Aplikasi CGI menerima *request* dari *web server*, kemudian memprosesnya dan memberikan sebuah respon atau data ke *web server*. CGI bekerja dengan sangat sederhana dimana tidak diperlukan sebuah pemrograman untuk melakukannya. Hanya perlu mengedit berdasarkan *text base* konfigurasi file, membuat halaman web, dan menyimpannya dan menempatkannya untuk dikerjakan oleh *web server*. MapServer CGI *executable* bekerja sebagai perantara antara file peta dengan

program *web server* yang *request* peta. *Request* dikirimkan dalam bentuk CGI parameter dari *web server* menuju MapServer. Gambar yang dibuat oleh MapServer selanjutnya memberikan *feed back* ke *web server* yang kemudian ditampilkan dalam browser (Mitchell, 2005).

2.2.2 ArcView

ArcView merupakan salah satu perangkat lunak sistem informasi geografi yang dikeluarkan oleh ESRI (*Environmental Systems Research Institute*). ArcView memiliki kemampuan melakukan visualisasi data, eksplorasi data, menjawab *query* (baik *database* spasial maupun non spasial), menganalisis data secara geografis, dan sebagainya (Riyanto dkk, 2009).

2.3 PHP/Mapscript

Mapscript adalah antar muka pemrograman Mapserver. PHP/Mapscript memungkinkan kita melakukan akses terhadap mapscript API (*Application Programming Interface*) dari lingkungan PHP, dengan menggunakan kelas PHP.

PHP/Mapscript tersedia sebagai sebuah modul PHP, dalam bentuk file DLL (*Dynamic Linked Library*) pada *platform* Windows, atau dalam bentuk *shared object* pada *platform* Linux. Modul PHP/Mapscript biasanya dipasang bersamaan dengan instalasi Mapserver (Nryadin, 2005).

2.4 *Asset Management*

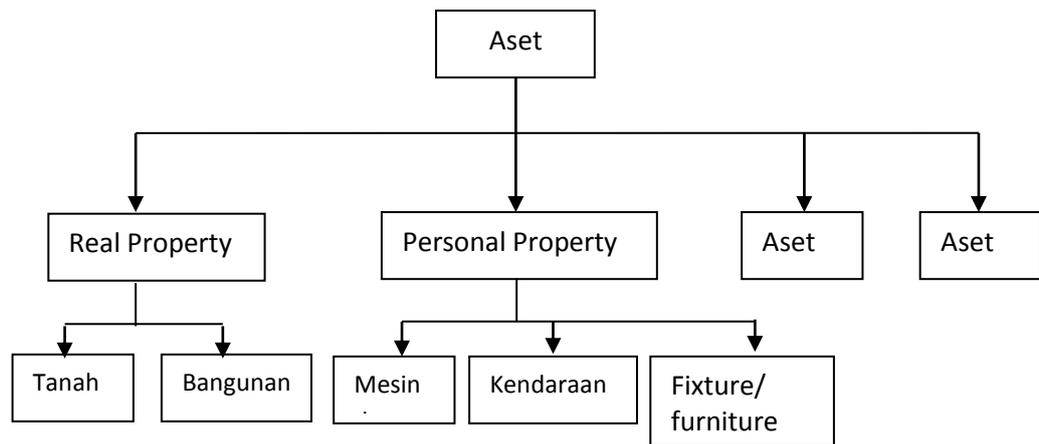
Secara umum manajemen aset adalah sebuah proses pengelolaan segala sesuatu baik berwujud dan tidak berwujud yang memiliki nilai ekonomik dan dapat mendorong tercapainya tujuan dari individu dan organisasi.

Definisi dari *Asset management* menurut Danylo dalam buku Diklat Teknis Substantif Spesialisasi Pengelolaan kekayaan Negara (2007), adalah sebuah metodologi untuk meng-efisiensikan dan mengalokasikan secara tepat sumber daya diantara yang masih valid, masih bisa bersaing dan *objective*. Kaganova dan McKellar mendefinisikan manajemen aset : *Property asset management* dapat didefinisikan sebagai proses mengambil keputusan dan mengimplementasikan hubungannya ke dalam pendapatan, kegunaan, dan pembuangan dari *real property*.

Walaupun manajemen aset dapat dipresentasikan sesuai dengan jenis aset atau konsentrasi kegiatannya tetapi beberapa ahli tidak ingin membuat definisi manajemen aset secara spesifik dengan menyatakan tidak ada definisi yang pasti mengenai manajemen aset (*working defenition*). Meskipun demikian, dari beragam definisi tersebut dapat disimpulkan bahwa manajemen aset mencakup proses mulai dari proses perencanaan sampai dengan penghapusan (*disposal* dan perlunya monitoring terhadap aset-aset tersebut selama umur penggunaannya oleh suatu organisasi atau Kementerian/Lembaga (Diklat Teknis Substantif Spesialisasi Pengelolaan Kekayaan Negara, 2007).

Sistem informasi “Asset Management” adalah sebuah aplikasi pengelolaan aset yang ditujukan untuk perusahaan besar yang memiliki aset dengan jumlah banyak

yang seharusnya memerlukan sebuah divisi sendiri dalam pengelolaan aset tersebut. Aset dapat dikelompokkan berdasarkan dua kategori kelompok utama yaitu Real Property dan Personal property. Gambar 2.3 adalah diagram pembagian kelompok aset.



Gambar 2.4 Diagram Pembagian Kelompok Aset (Dit.TIK UPI, 2008).

2.4.1 Manajemen Aset Tetap

Manajemen aset tetap (Siregar, 2004) merupakan proses akuntansi yang bertujuan untuk melacak status aset perusahaan, termasuk lokasi, kondisi dan karakteristik pembeda lainnya dari setiap aset. Pengelolaan aset tetap ini akan berhasil bila didukung oleh pencatatan perusahaan yang update untuk mencerminkan status aktual dari semua aset tetap dalam hal nilai saat ini. Ada beberapa cara yang dapat digunakan dalam pengelolaan aset tetap, seperti penggunaan teknologi untuk mengidentifikasi aset dari orang lain, melacak gerakan aset tersebut dari satu lokasi ke lokasi lain, dan menghitung depresiasi yang terkait dengan setiap aset tetap.

Salah satu elemen kunci dari manajemen aset tetap adalah selalu mengetahui berapa banyak unit dan posisi aset tetap yang dimiliki perusahaan. Tergantung

pada sifat dari bisnis, aset tetap dapat berpindah tempat, lokasi, atau dipindahkan dari satu departemen ke departemen lain. Sebagai contoh, sebuah perusahaan manufaktur akan melacak jumlah *forklift* yang dimiliki dan dioperasikan sebagai bagian dari program normal produktivitas bisnis. Maka setiap *forklift* akan diberi tanda atau nomor seri. Nomor seri akan digunakan untuk mengidentifikasi aset dalam catatan akuntansi dan juga melacak posisi *forklift* yang digunakan. Penggunaan teknologi modern juga bermanfaat bila aset tetap berpindah tempat, dengan menggunakan *scanner* memungkinkan untuk melacak waktu dan tanggal setiap pergerakan aset tetap, serta menyediakan data penting untuk manajemen.

Berikut adalah Tahapan Manajemen Aset Tetap (Siregar, 2004) :

1. Inventarisasi Aset

Inventarisasi aset terdiri dari dua aspek yaitu

- Aspek fisik (bentuk, luas, volume atau jumlah, jenis, alamat dan lain-lain) dan.
- Aspek yuridis atau legal (status penguasaan, masalah legal yang dimiliki, batas akhir penguasaan dan lain-lain).

Proses kerja yang dilakukan antara lain adalah pendataan, kodefikasi, pengelompokkan, dan administrasi sesuai dengan tujuan manajemen aset.

2. Legal Audit

Dalam rangka inventarisasi aset tetap, juga perlu dilakukan legal audit, yaitu melakukan pengecekan terhadap status penguasaan aset dengan cara mengecek semua sertifikat dan bukti kepemilikan aset, seperti sertifikat tanah, BPKB kendaraan bermotor dan sebagainya. Lingkup kerja yang dilakukan adalah:

- inventarisasi status penggunaan aset, sistem, dan prosedur penguasaan atau pengalihan aset.
- identifikasi dan mencari solusi atas permasalahan legal.
- strategi untuk memecahkan berbagai permasalahan legal yang terkait dengan penguasaan atau pengalihan aset.

Legal audit dapat digunakan untuk mengatasi berbagai permasalahan legal menyangkut status kepemilikan aset, antara lain status hak penguasaan yang lemah, aset dikuasai pihak lain, pemindahtanganan aset yang tidak termonitor, dan lain-lain.

3. Penilaian Aset

Penilaian aset merupakan satu proses kerja untuk melakukan penilaian atas aset yang dikuasai. Biasanya ini dikerjakan oleh konsultan penilai yang independen. Hasil dari nilai aset tersebut akan dapat dimanfaatkan untuk mengetahui nilai kekayaan aset tetap. Penilaian aset tetap dimaksudkan untuk memberikan saldo aset yang lebih relevan bagi pencatatan di laporan keuangan periode terkait dan periode sesuai nilai wajarnya (*fair value*) pada saat penilaian aset.

4. Optimalisasi Aset

Optimalisasi aset merupakan proses kerja dalam manajemen aset yang bertujuan untuk mengoptimalkan (potensi fisik, lokasi, nilai, jumlah atau volume, legal dan ekonomi) yang dimiliki aset tersebut. Dalam tahapan ini, aset yang dikuasai perusahaan diidentifikasi dan dikelompokkan atas aset yang memiliki potensi dan tidak memiliki potensi. Pemanfaatan aset tetap adalah pendayagunaan aset yang dimiliki perusahaan yang sedang atau tidak digunakan untuk tugas pokok dan fungsi atau dapat memberikan nilai tambah atas perusahaan/pemanfaatan dari

aset yang bersangkutan, aset tetap tersebut dapat dimanfaatkan secara optimal oleh pihak ketiga dengan tidak mengubah status kepemilikan. Hasil akhir dari tahapan ini adalah rekomendasi yang berupa sasaran, strategi, dan program untuk mengoptimalkan aset yang dikuasai. Contoh bentuk optimalisasi pemanfaatan aset adalah penyewaan aset, peminjaman, kerjasama pemanfaatan aset tetap, dan bangunan guna serah.

5. Pengawasan dan pengendalian aset

Pengawasan dan pengendalian terhadap pemanfaatan dan pengalihan aset diperlukan agar setiap penanganan terhadap satu aset dapat termonitor jelas, mulai dari lingkup penanganan hingga siapa yang bertanggungjawab mengelola aset tersebut (Siregar, 2004).

2.5 Framework Pmapper

Framework adalah sekumpulan *library* yang diorganisasikan pada sebuah rancangan arsitektur untuk memberikan kecepatan, ketepatan dan kemudahan dan konsistensi didalam pengembangan aplikasi atau pemecahan suatu masalah.

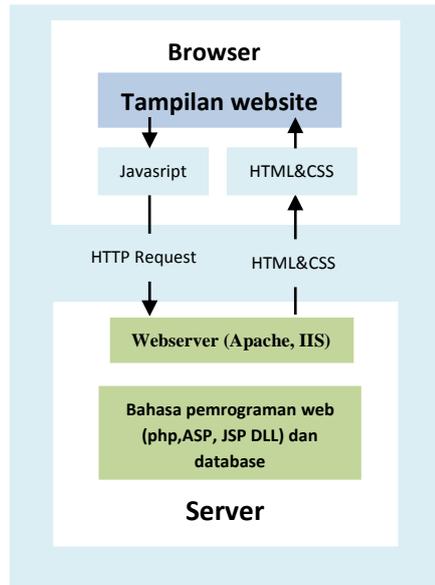
Pmapper adalah sebuah *framework* berbasis MapServer dan PHP/MapScript. Pmapper menyediakan *tools* yang sangat bagus dan siap untuk digunakan dan mempunyai sebuah *plugin* API untuk menambah fungsionalitas nya.

Pmapper dikembangkan oleh Armin Burger, dan didistribusikan dibawah GNU GPL *lisence* :

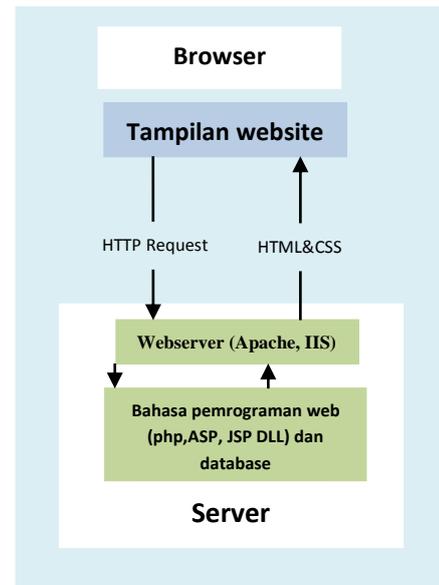
- Adalah aplikasi tanpa bayar.
- Memungkinkan untuk memodifikasi *source code*.
- Dan dapat digunakan dalam berbagai keperluan (Valentini, 2011).

2.6. AJAX

Ajax merupakan suatu teknik/metode pengambilan data dari *server* menggunakan sebuah fungsi dalam javascript, yaitu XMLHttpRequest (XHR). Merupakan obyek yang dirancang untuk memungkinkan permintaan layanan ke *server* HTTP secara asinkron. Kalau boleh ditarik kesimpulan, dengan Ajax, memungkinkan aplikasi web dapat berinteraksi dengan *server* dibelakang *layer*, sehingga tidak mempengaruhi halaman web secara keseluruhan. Artinya, halaman web tidak perlu dipanggil ulang secara keseluruhan hanya untuk mengganti sebahagian dari isi halaman web. Pada model web dengan ajax, pendekatan yang dilakukan adalah mengirimkan data dalam jumlah yang lebih kecil/ke *server* sesuai permintaan dari *user*. Misalnya *user* hanya perlu pencarian data, maka tidak perlu semuanya dikirimkan ke *server* (tidak perlu *header*, logo, *footer*), tapi cukup bagian teks pencarian datanya saja, inilah yang membuat model ajax dapat merespon permintaan *user* dengan lebih cepat dibandingkan model tradisional. Dengan cara ini pula, beban *bandwidth* di *server* dapat lebih dihemat. Pada Gambar 2.3 dapat dilihat model web dengan ajax. Dibandingkan dengan model tradisional Gambar 2.4 akan terkesan model ajax lebih panjang arsitekturnya, ada tambahan lapisan Ajax pada browser (atau sering disebut dengan *Ajax Engine*). Lapisan Ajax inilah yang mengatur komunikasi antara *user* klien dengan *server*. Sepintas penambahan lapisan Ajax pada arsitektur akan terlihat memperlambat proses, karena komunikasi ke *server* tidak lagi dilakukan secara langsung, namun yang dilakukan oleh lapisan Ajax jauh lebih sedikit dibandingkan kalau yang diminta adalah untuk keseluruhan halaman (Hakim, 2011).



Gambar 2.5 Model Ajax (Hakim, 2011)



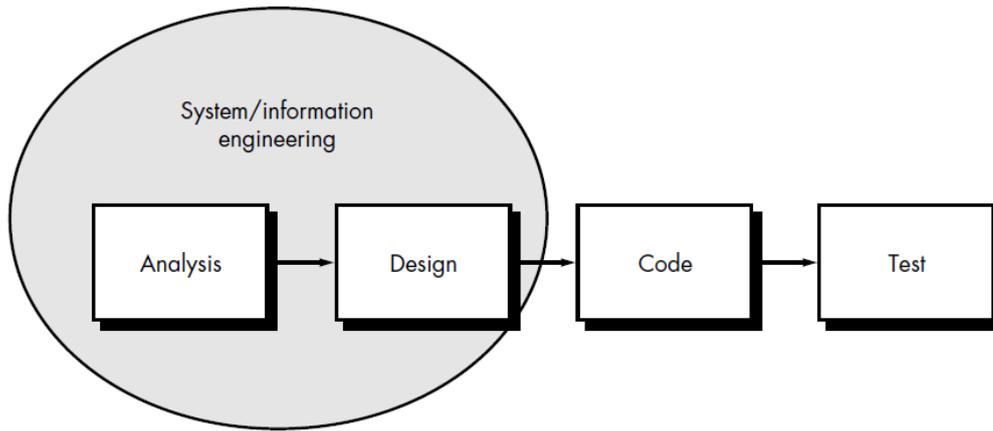
Gambar 2.6 Model Tradisional (Hakim, 2011)

2.7 JQuery

JQuery adalah Javascript *Library*, yaitu kumpulan kode/fungsi Javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode Javascript, termasuk dalam membuat kode Ajax (Hakim, 2011).

2.8 Metode Waterfall

Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan *Classic Life Cycle* atau model *waterfall*. Model ini pertama kali muncul pada tahun 1970 yang diperkenalkan oleh Winston W. Royce. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing* atau *verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Gambar 2.5 menggambarkan tahapan pada model *waterfall* menurut Pressman (2001).



Gambar 2.7. Model *Waterfall* (Pressman, 2001)

Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model *Waterfall* menurut Pressman:

1. *System / Information Engineering and Modeling*

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan sebagainya. Tahap ini sering disebut dengan *Project Definition*. Menurut Franco-Plata dkk, (2012) tahapan ini adalah menentukan masalah-masalah yang akan diselesaikan dengan menggunakan *tools SIG*, menentukan dan membatasi ruang lingkup dan konteks umum dari sistem yang akan dikembangkan.

2. *Software Requirements Analysis*

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang dibuat, maka para *software engineer* (Analis) harus mengerti tentang informasi dari *software*, misalnya fungsi yang dibutuhkan dan *user interface*. Dari dua aktivitas tersebut

(pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.

Menurut Garouani dkk, (2012) Menganalisis *software* harus dilakukan secara hati-hati dan teliti sehingga fungsionalitas dan/atau *business* dari aplikasi dapat tercapai dan sesuai dengan yang diinginkan oleh *user key*. Masalah *Compability* dari *software* SIG pada tahap ini harus sudah terselesaikan.

3. *Design*

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan *software* menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

4. *Coding*

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis dikerjakan oleh programmer.

5. *Testing / Verification*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Menurut Bell dkk., (2006), semua desain peta harus diuji untuk memastikan bahwa peta mengkomunikasikan pesan yang

dimaksud oleh *user*. Penyertaan dalam pengujian harus sebanding dengan konsekuensi dari kesalahan interpretasi (kesalahan dalam menjelaskan isi dalam peta). Pengujian desain dapat berkisar dari yang sederhana melalui pengujian isi peta (seperti navigasi peta) dimaksudkan untuk komunikasi internal saja, untuk pengujian kegunaan yang lebih menyeluruh dengan perwakilan dari *user* diharapkan memiliki distribusi yang luas.

6. *Maintenance*

Pemeliharaan suatu *software* diperlukan termasuk pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu saja melainkan membutuhkan pengembangan atas kekurangan yang ditimbulkan oleh *software* tersebut.. Ketika dijalankan mungkin saja masih ada *errors* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.