BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Perancangan Suatu program aplikasi terdiri dari satu kesatuan system. Terdapat dua kelompok pendekatan di dalam mendefinisikan system, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan pada prosedur mendefinisikan system sebagai berikut:

"Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu" **(Jogiyanto H.M.,2001 : 1).**

Pengertian dari prosedur itu sendiri menurut Richard F. Neuschel adalah suatu urutan-urutan operasi tulis menulis biasanya melibatkan beberapa orang di dalam satu atau lebih departemen yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi yang terjadi.

Lebih lanjut Ferry Fitz Gerald, Andra F. Fitz Gerald dan Warren D. Stalling, Jr mendefinisikan prosedur sebagai berikut:

"Prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerapkan apa (what) yang harus dikerjakan, siapa (who) yang mengerjakan, kapan (when) dikerjakan dan bagaimana (how) mengerjakan" **(Jogiyanto H.M.,2001 : 2)**

Pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan system sebagai berikut :

"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu" **(Jogiyanto H.M., 2001 : 2)**

Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu:

1. Komponen-komponen *(components)*

   Suatu sistem terdiri dari sejumlah komponen yang sering disebut dengan sub sistem yang saling berinteraksi artinya saling bekerja sama membentuk satu kesatuan. Sub sistem itu sendiri yang mempunyai sifa-sifat dari sistem itu sendiri dalam menjalankan suatu fungsi dan mempengaruhi proses sistem secara keseluruhan.

2. Batas sistem *(Boundary)*

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau yang membatasi tersebut dengan lingkungan luarnya sehingga sistem itu dapat dipandang sebagai satu kesatuan. Batasan sistem dapat merupakan ruang lingkup sistem itu sendiri.

3. Lingkungan hasil sistem *(environments)*

   Lingkungan hasil yaitu segala sesuatu yang ada di luar sistem dari batasan sistem yang bisa mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan juga dapat bersifat merugikan sistem.

4. Penghubung sistem *(Interface)*

   Penghubung merupakan media penghubung antara komponen sistem sehingga sumber-sumber data mengalir diantara komponen sistem dan membentuk satu kesatuan.

5. Masukkan sistem *(input)*

   Masukkan yaitu energi yang dimasukkan ke dalam sistem.

6. Keluaran sustem *(output)*

   Keluaran yaitu hasil dari energi yang diolah.

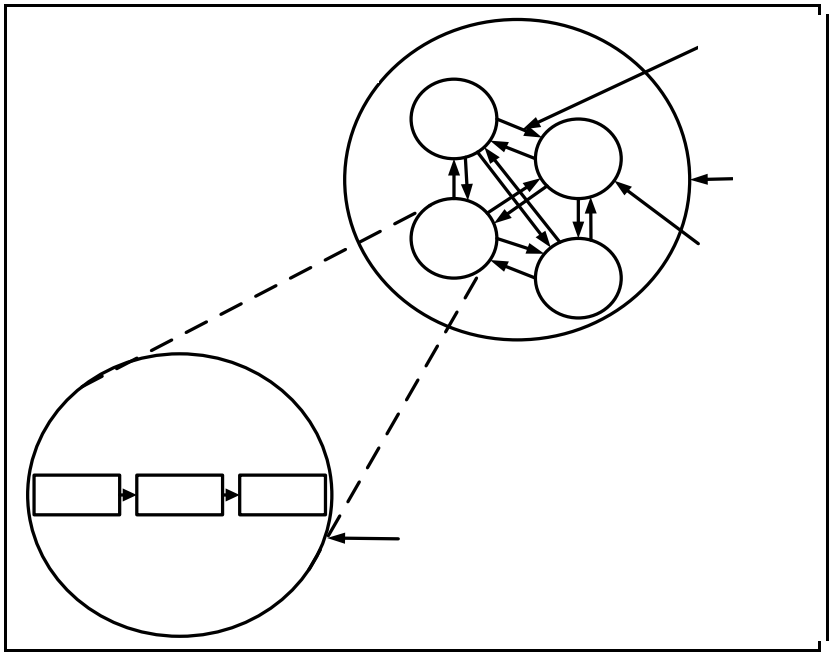
7. Pengolah sistem *(process)*

   Proses yaitu yang akan mengubah masukkan menjadi keluaran.

8. Sasaran sistem *(objective)* atau tujuan *(goal)*

   Sistem berguna jika mempunyai sasaran atau tujuan, dan sistem dikatakan berhasil jika keluaran dihasilkan sistem dapat mencapai sasaran dari sistem itu sendiri.

Karateristik suatu sistem dapat digambarkan seperti berikut:



Gambar 2.1 Karakteristik suatu sistem

(Sumber : JOG [6])

2.1.2 Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

a. Sistem abstrak (*abstract system*) dan sistem fisik (*physical system*)
b. Sistem alamiah (*natural system*) dan sistem buatan (*human made system*)
c. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*)
d. Sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

2.2 Konsep Dasar Informasi

"Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya" **(Jogiyanto H.M., 2001 : 8)**

Sumber informasi adalah data. Data merupakan fakta atau kenyataan yang menggambarkan suatu kejadian-kejadian atau event nyata, kemudian dirumuskan ke dalam sekelompok simbol atau lambang-lambang yang teratur yang menunjukkan kualitas, tindakan atau hal-hal lain.

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi sangat penting di dalam suatu organisasi. Karena informasi dapat berguna bagi suatu organisasi atau seseorang dalam mengambil suatu keputusan menurut JOG [6]. Informasi (*information*) dapat didefinisikan sebagai berikut :

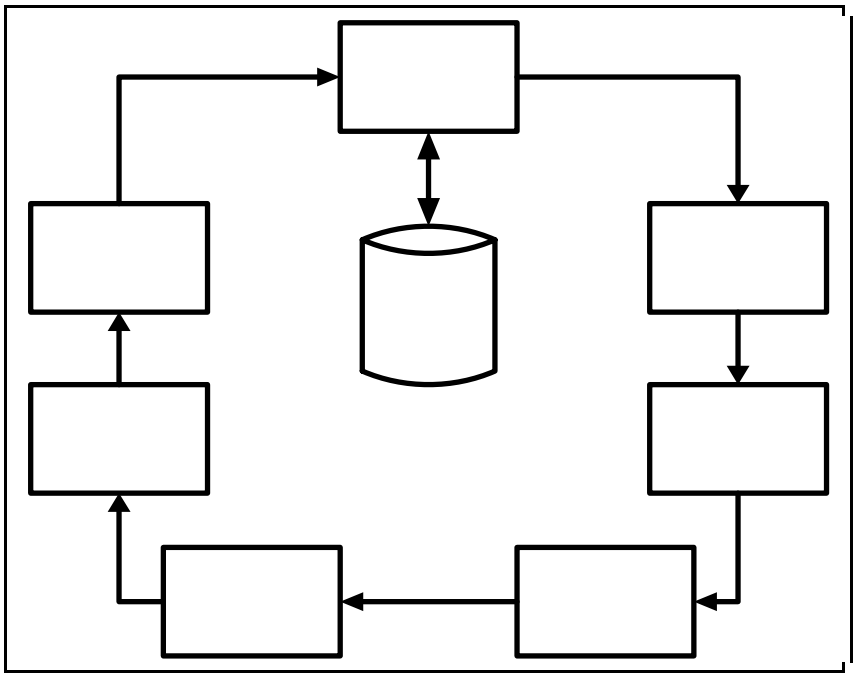
"Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya."

Sedangkan menurut AGS [1] informasi dapat didefinisikan sebagai berikut:

"Informasi adalah suatu kerangka kerja dimana SDM (manusia, komputer) dikoordinasikan untuk mengubah masukan (data) menjadi keluaran (informasi) guna mencapai sasaran perusahaan."

2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*). Siklus ini disebut juga dengan siklus data (*data processing cycles).*



Gambar 2.2 Siklus Informasi

(Sumber : JOG [6])

2.2.2 Kualitas Informasi

Informasi yang baik adalah informasi yang berkualitas. Informasi yang berkualitas ditentukan oleh hal-hal sebagai berikut:

1. Relevan (*Relevance*)

   Informasi tersebut mempunyai manfaat untuk pemakainya dan harus sesuai dengan yang dibutuhkan.

2. Dapat dipercaya (Realibility)

   Informasi yang akan diberikan dapat dipercaya kebenarannya dan mempunyai data-data yang lengkap dan jelas sumber-sumber datanya.

3. Tepat pada waktunya (*Timeliness*)

   Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi, oleh karena itu informasi harus *up to date*.

4. Akurat (*Accurate*)

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan informasi, harus jelas mencerminkan maksudnya. Komponen-komponen data yang akurat adalah sebagai berikut:

   a. *Completeness*, yaitu informasi yang dihasilkan atau yang dibutuhkan memiliki kelengkapan yang baik, karena bila informasi yang dihasilkan sebagian-sebagian tentunya akan mempengaruhi dalam pengambilan keputusan atau menentukan tindakan secara keseluruhan.

b. *Correctness*, yaitu kebenaran informasi dapat dipertanggungjawabkan dan mempunyai bukti-bukti dan fakta yang kuat.

c. *Security* atau keamanan, dalam hal ini informasi yang dikirimkan ke setiap orang yang membutuhkannya perlu pengawasan karena struktur pengecekan dapat memutuskan jika informasi yang sensitif ditujukan kepada pemakai yang tidak sah kepada pihak yang salah.

5. Ekonomis

   biaya pembuatan informasi murah dan memberikan manfaat yang besar bagi pemakai.

2.3 Konsep Dasar Sistem Informasi

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut.

Selain itu sistem informasi dapat didefinisikan sebagai berikut :

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
3. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan

strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan, permasalahannya adalah dimana informasi tersebut didapat. Informasi dapat diperoleh dari sistem informasi. Robert A Leitch dan K. Roscoe Davis mendefinisikan sistem informasi sebagai berikut:

"*Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan."* **(Jogiyanto H.M., 2001 : 8)**

Komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat keras *(hardware)*

   Perangkat keras *(hardware)* terdiri dari komputer

2. Perangkat lunak *(software)*

   Perangkat lunak *(software)* berupa program-program aplikasi yang akan digunakan, yaitu merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

   Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.

4. Prosedur

   Prosedur merupakan dokumentasi prosdedur atau proses sistem, tata cara atau penuntun operasional (aplikasi) dan teknis

5. Manusia

   Manusia adalah pengguna dari sistem informasi.

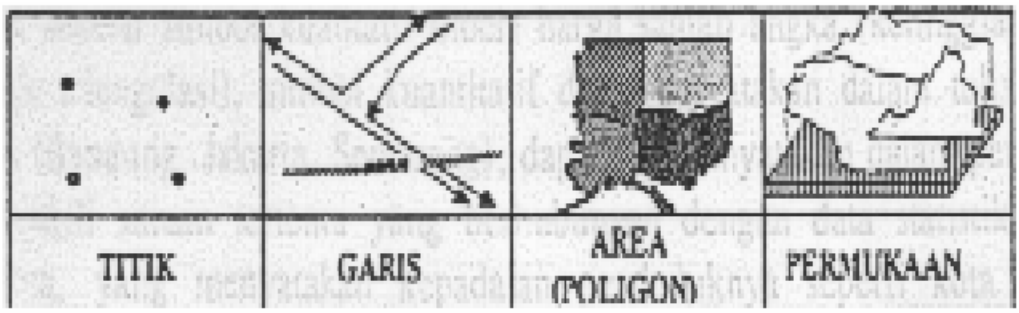
2.4 Konsep Dasar Sistem Informasi Geografis

GIS *(Geographical Information System)* atau dikenal pula dengan SIG (Sistem Informasi Geografis) merupakan sistem infomasi berbasis komputer yang menggabungkan antara unsur peta (geografis) dan informasinya tentang peta tersebut (data atribut) yang dirancang untuk mendapatkan, mengolah, memanipulasi, analisa, memperagakan dan menampilkan data spatial untuk menyelesaikan perencanaan,mengolah dan meneliti permasalahan.

GIS merupakan akronim dari:

1) Geography

Istilah ini digunakan karena GIS dibangun berdasarkan pada ‘geografi’ atau ‘spasial’. Objek ini mengarah pada spesifikasi lokasi dalam suatu space. Objek bisa berupa fisik, budaya atau ekonomi alamiah. Penampakan tersebut ditampilkan pada suatu peta untuk memberikan gambaran yang representatif dari spasial suatu objek sesuai dengan kenyataannya di bumi. Simbol, warna dan gaya garis digunakan untuk mewakili setiap spasial yang berbeda pada peta dua dimensi. Data Spasial berupa titik, garis, poligon (2-D), permukaan (3-D).



Gambar 2.3 Data Spasial

(sumber: Purwadhi, 1997)

a. Format Titik

- Koordinat tunggal
- Tanpa panjang
- Tanpa luasan

Contoh:

- lokasi kecelakaan
- Letak pohon

b. Format Garis

- koordinat titik awal dan akhir
- mempunyai panjang tanpa luasan

Contoh:

- jalan, sungai
- utility

c. Format Poligon

- koordinat dengan titik awal dan akhir sama
- mempunyai panjang dan luasan

Contoh:

- tanah persil
- bangunan

d. Format Permukaan

- area dengan koordinat vertikal
- area dengan ketinggian

Contoh:

- peta slope
- bangunan bertingkat

4 tingkat Model Data Spasial:

1. Gambar kenyataan (*reality*): persis seperti yang kita lihat;
2. Gambar abstrak (*conceptual*);
3. Gambar kejadian tertentu (*logical*): berbentuk diagram atau tabel;
4. Berkas struktur fisik (*physical*): bentuk penyimpanan pada perangkat keras

2) Information

Informasi berasal dari pengolahan sejumlah data. Dalam GIS informasi memiliki volume terbesar. Setiap object geografi memiliki setting data tersendiri karena tidak sepenuhnya data yang ada dapat terwakili dalam peta. Jadi, semua data harus diasosiasikan dengan objek spasial yang dapat membuat peta menjadi intelligent. Ketika data tersebut diasosiasikan dengan permukaan geografis yang

representatif, data tersebut mampu memberikan informasi dengan hanya mengklik mouse pada objek.

Perlu diingat bahwa semua informasi adalah data tapi tidak semua data merupakan informasi.

3) System

Pengertian suatu sistem adalah kumpulan elemen-elemen yang saling berintegrasi dan

berinterdependensi dalam lingkungan yang dinamis untuk mencapai tujuan tertentu.

2.4.1 Definisi Sistem Informasi Geografis

Definisi system informasi geografis selalu berkembang, bertambah, dan bervariasi. Hal ini terlihat dari banyaknya definisi SIG yang telah beredar. Selain itu, SIG juga merupakan suatu bidang kajian ilmu dan teknologi yang relative baru, digunakan oleh berbagai bidang disiplin ilmu, dan berkembang dengan cepat. Berikut merupakan sebagian kecil dari definisi-definisi SIG yang telah beredar di berbagai pustaka"

1. *"SIG adalah sistem yang berbasiskan computer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografi. SIG dirancang untuk mengumpilkan, menyimpan, menganalisis objek-objek data fenomena dimana lokasi geografi merupakan karakteristik yang penting atau krisis untuk dianalisis. Dengan demikian, SIG merupakan system computer yang memiliki empat kemampuan berikut dalam*

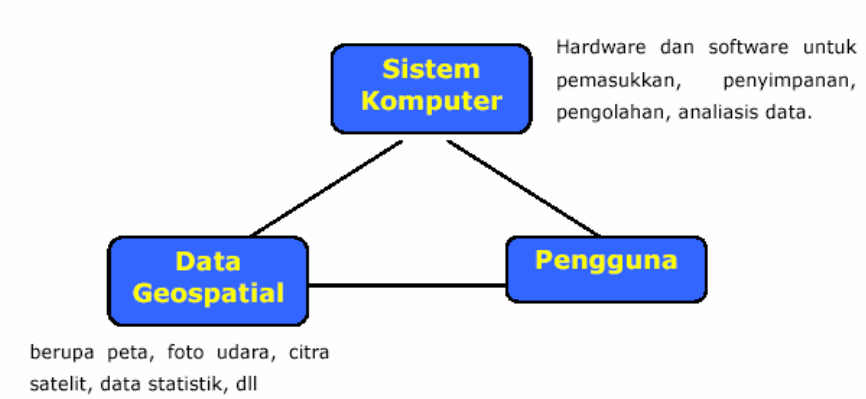
menangani data yang bereferensi geografis : (a) masukkan, (b) manajemen data (penyimpanan dan pemanggilan data), (c) analisis dan manipulasi data, (d) keluaran-keluaran" **(Aronoff, 1989)**

2. *"SIG adalah system yang terdiri dari perangkat keras, perangkat lunak, data, manusia (brainware), organisasi dan lembaga yang digunakan untuk mengumpulkan, menyimpan, menganalisa, dan menyebarkan informasi-informasi mengenai daerah-daerah di permukaan bumi"* **(Chrisman, 1997)**
3. *"SIG adalah system computer yang digunakan untuk memanipulasi data geografi. Sistem ini diimplementasikan dengan perangkat keras dan perangkat lunak computer yang berfungsi untuk: (a) akusisi dan verifikasi data, (b) kompilasi data, (c) penyimpanan data, (d) perubahan dan updating data, (e) manajemen dan pertukaran data, (f) manipulasi data, (g) pemanggilan dan persentasi data, dan (h) analisis data"* **(Bernhardsen, 1992)**
4. *"SIG adalah system komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisa informasi-informasi yang berhubungan dengan permukaan bumi"* **(Demers, 1997)**
5. *"SIG adalah teknologi informasi yang dapat menganalisa, menyimpan, dan menampilkan baik data spasial maupun non-spasial. SIG mengkombinasikan kekuatan perangkat lunak basis data relasional dan paket perangkat lunak CAD"* **(Guo Bo, 2000)**

2.4.2 Komponen GIS

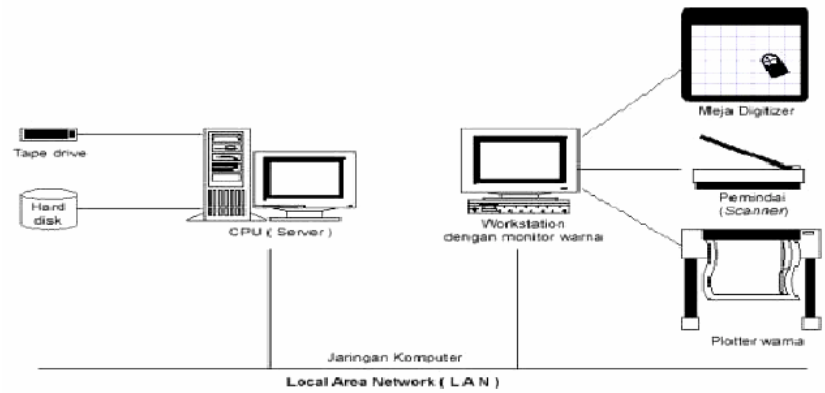
Komponen kunci dalam GIS adalah sistem komputer, data geospatial (data atribut) dan pengguna ,

yang dapat digambarkan sebagai berikut :



Gambar 2.4. Komponen kunci SIG

Sistem komputer GIS terdiri dari perangkat keras *(hardware)*, perangkat lunak *(software)* dan prosedur untuk penyusunan pemasukkan data, pengolahan, analisis, pemodelan ***(modelling),*** dan penayangan data geospatial.



Gambar 2.5 Komponen *Hardware* dalam GIS

Sumber-sumber data geospatial adalah peta digital, foto udara, citra satelit, tabel statistik dan

dokumen lain yang berhubungan. Data geospatial dibedakan menjadi:

1. Data grafis/geometris, mempunyai tiga elemen : titik *(node)*, garis *(arc)* dan luasan *(poligon)* dalam bentuk vector ataupun raster yang mewakili geometri topologi, ukuran, bentuk, posisi dan arah.
2. Data atribut/data tematik

Fungsi pengguna berguna untuk memilih informasi yang diperlukan, membuat standar, update data yang efisien, analisa output untuk hasil yang diinginkan serta merencanakan aplikasi.

2.4.3 Karakteristik SIG

Adapun karakteristik dari system informasi geografis adalah:

1. Merupakan suatu sistem hasil pengembangan perangkat keras dan perangkat lunak untuk tujuan pemetaan, sehingga fakta wilayah dapat disajikan dalam satu sistem berbasis komputer.
2. Melibatkan ahli geografi, informatika dan komputer, serta aplikasi terkait.
3. Masalah dalam pengembangan meliputi: cakupan, kualitas dan standar data, struktur, model dan visualisasi data, koordinasi kelembagaan dan etika, pendidikan, *expert* sistem dan *decision support* sistem serta penerapannya
4. Perbedaannya dengan Sistem Informasi lainnya: data dikaitkan dengan letak geografis, dan terdiri dari data tekstual maupun grafik

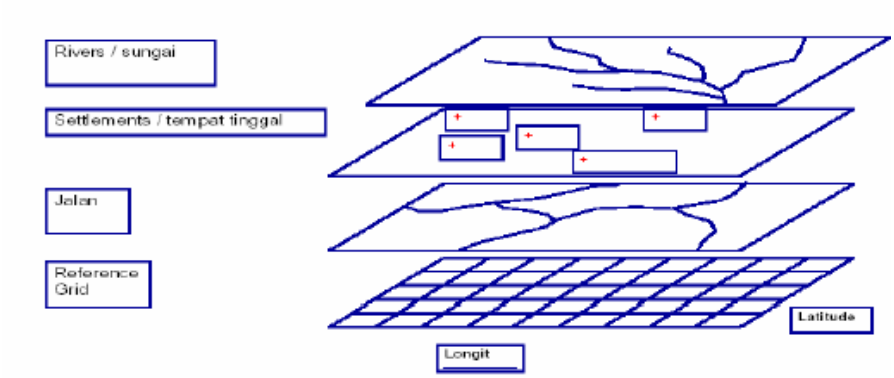
5. Bukan hanya sekedar merupakan pengubahan peta konvensional (tradisional) ke bentuk peta digital untuk kemudian disajikan (dicetak / diperbanyak) kembali
6. Mampu mengumpulkan, menyimpan, mentransformasikan, menampilkan, memanipulasi, memadukan dan menganalisis data spasial dari fenomena geografis suatu wilayah.
7. Mampu menyimpan data dasar yang dibutuhkan untuk penyelesaian suatu masalah. Contoh : penyelesaian masalah perubahan iklim memerlukan informasi dasar seperti curah hujan, suhu, angin, kondisi awan. Data dasar biasanya dikumpulkan secara berkala dalam jangka yang cukup panjang.

2.4.4 Subsistem Utama GIS

GIS terdiri dari empat subsistem utama :

1. **Sub-sistem Masukan**, Perangkat untuk menyediakan data sampai siap dimanfaatkan oleh pengguna; yang berupa peralatan pemetaan terestris, fotogrametri, digitasi, *scanner*, dsb. Pada umumnya *output* dari perangkat tersebut berupa peta, citra dan tayangan gambar lainnya.
2. **Sub-sistem Database**, Digitasi peta dasar pada berbagai wilayah/daerah cakupan dengan berbagai skala telah dan terus dilakukan dalam rangka membangun sistem database spasial yang mudah diperbaharui dan digunakan dengan data literal sebagai komponen utamanya.
3. **Sub-sistem Pengolahan Data**, Pengolahan data baik yang berupa vektor maupun *raster* dapat dilakukan dengan berbagai software seperti

AUTOCAD, ARC/INFO, ERDAS, MAPINFO, ILWIS. Untuk metode vektor biasanya disebut digitasi sedangkan *raster* dikenal dengan metode *overlay*. Salah satu karakteristik *software* GIS adalah adanya sistem *Layer* (pelapisan) dalam menggabungkan beberapa unsur informasi (penduduk, tempat tinggal, jalan, persil tanah, dll). Seperti: *Layer*, *Coverage* (ArcInfo produk ESRI), *Theme* (ArcView produk ESRI), *Layer* (AutoCAD Map produk Autodesk), *Table* (MapInfo produk MapInfo Corp.), dan lain-lainya.



Gambar 2.6 Layer-layer dalam aplikasi GIS

Pada gambar diatas menunjukkan 4 layer yang terdiri atas *grid*, layer jalan, tempat tinggal, dan sungai. Peta akan terlihat berdasarkan layer yang tersusun dimana layer yang paling atas adalah layer yang tampak diatas.

4. **Sub-sistem Penyajian Informasi,** Dilakukan dengan berbagai media agar mudah dimanfaatkan oleh pengguna.

2.4.5 Aplikasi Sistem Informasi Geografis

Pada sebuah aplikasi SIG, terdapat beberapa fasilitas yang merupakan standar untuk melengkapi peta yang tampil di 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 sebagianya.

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-*input*kan 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 meghubungkan data *feature* pada peta dengan data dalam bentuk lain seperti gambar, video, ataupun web.

2.4.6 Kemampuan Sistem Informasi Geografis

Kemampuan inilah yang membedakan SIG dengan Sistem Informasi lainnya yang membuatnya menjadi berguna untuk berbagai kalangan untuk menjelaskan kejadian, merencanakan strategi, dan memprediksi apa yang akan terjadi. Kemampuan SIG antara lain:

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.

   Kemampuan ini memungkinkan seseorang untuk mencari dimana letak suatu daerah, benda, atau lainnya di permukaan bumi. Fungsi ini

dapat digunakan seperti untuk mencari lokasi rumah, mencari rute jalan, mencari tempat-tempat penting dan lainnya yang ada di peta. Orang dapat pula melihat pola-pola yang mungkin akan muncul dengan melihat penyebaran letak-letak *feature*, misalnya sekolah, pelanggan, daerah miskin dan sebagainya.

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 mencari 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.

Sebagai contoh, sebuah perusahaan pakaian anak yang akan menyebarkan brosurnya akan terbantu dengan mengetahui daerah-daerah mana yang punya banyak keluarga dengan anak kecil dan mempunyai pendapatan yang tinggi.

3. **Memetakan** K**erapatan *(Densities)***

Sewaktu orang melihat konsentasi dari penyebaran lokasi dari *feature-feature*, 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, misal membagi dalam kotak-kotak selebar 10 km2, dengan menggunakan perbedaan warna untuk menandai tiap-tiap kelas kerapatan.

Pemetaan kerapatan sangat berguna untuk data-data yang berjumlah besar seperti sensus atau data statistik daerah. Dalam data sensus seperti gambar 3 misalnya, sebuah unit sensus yang mempunyai jumlah keluarga diatas 40 diberi warna hijau, 30-40 hijau muda dan seterusnya. Dengan cara ini orang akan lebih mudah melihat daerah mana yang kepadatan penduduknya tinggi dan mana yang kepadatan penduduknya rendah.

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.

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

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 diluar area. Peta ini digunakan untuk dasar rencana apabila terjadi keadaan darurat. Adakalanya perlu untuk menentukan daerah yang diluar kriteria, misalnya untuk menentukan lokasi pabrik dilakukan di daerah dalam radius lebih dari 1 km.

2.5 Peta

Peta merupakan suatu representasi konvensional (miniatur) dari unsur-unsur (fatures) fisik (alamiah dan buatan manusia) dari sebagian atau keseluruhan permukaan bumi di atas media bidang datar dengan skala tertentu **[Rockville86].**

Adapun persyaratan-persyaratan geometrik yang harus dipenuhi oleh suatu peta sehingga menjadi peta yang ideal adalah:

1. Jarak antara titik-titik yang terletak di atas peta harus sesuai dengan jarak aslinya di permukaan bumi (dengan memperhatikan faktor skala tertentu).
2. Luas suatu unsur yang direpresentasikan di atas peta harus sesuai dengan luas sebenarnya (juga dengan mempertimbangkan skalanya).
3. Sudut atau arah suatu garis yang direpresentasikan di atas peta harus sesuai dengan arah yang sebenarnya (seperti di permukaan bumi).
4. Bentuk suatu unsur yang direpresentasikan di atas peta harus sesuai dengan bentuk yang sebenarnya (juga dengan mempertimbangkan faktor skalanya).

Pada kenyataannya di lapangan merupakan hal yang tidak mungkin menggambarkan sebuah peta yang dapat memenuhi semua kriteria di atas, karena permukaan bumi itu sebenarnya melengkung. Sehingga pada saat melakukan proyeksi dari bentuk permukaan bumi yang melengkung tersebut ke dalam bidang datar (kertas) akan terjadi distorsi.

2.5.1 Jenis Peta

Secara umum peta dapat dibedakan atas:

1. Peta Topografi

   Peta topografi memperlihatkan posisi horizontal serta vertical dari unsur alam dan unsure buatan manusia dalam suatu bentuk tertentu. Peta topografi dikenal sebagai peta yang bersifat umum karena unsure-unsur yang disajikan adalah unsure yang terdapat di permukaan bumi sesuai dengan kegunaan dari peta bersangkutan misalnya: peta kadaster (pendaftaran tanah) menyajikan data mengbenai garis kepemilikan tanah bersama dengan sudut dan panjangnya, pemilik dan ukuran pensil dan informasi lainnya. Peta topografi juga disebut sebagai peta dasar, karena peta topografi dapat digunakan sebagai dasar untuk pembuatan peta-peta lainnya termasuk peta tematik.

2. Peta Tematik

   Peta tematik adalah suatu bentuk peta yang menyajikan unsure-unsur tertentu dari permukaan bumi sesuai dari tema atau topik dari peta yang bersangkutan. Misalnya: Peta tata guna lahan, peta geologi. Peta tematik umumnya digunakan sebagai data analisis dari beberapa unsure permukaan bumi di dalam pengambilan sebuah keputusan. Pada pembuatan peta tematik, peta topografi sebagai dasar sedangkan data tematik yang disajikan adalah hasil survey langsung dan survey tidak langsung.

2.5.2 Proyeksi Peta

Merupakan teknik-teknik yang digunakan untuk menggambarkan sebagian atau keseluruhan permukaan tiga dimensi yang secara kasar berbentuk bola ke permukaan datar dua dimensi dengan distorsi seminimal mungkin. Distorsi dapat dikurangi dengan membagi daerah yang dipetakan menjadi bagian yang tidak terlalu luas dan menggunakan bidang datar. Berikut ini akan dijelaskan proyeksi peta yang sering digunakan terutama proyeksi dalam melakukan proses digitasi :

1. Proyeksi UTM (Universal Transverse Mercator)

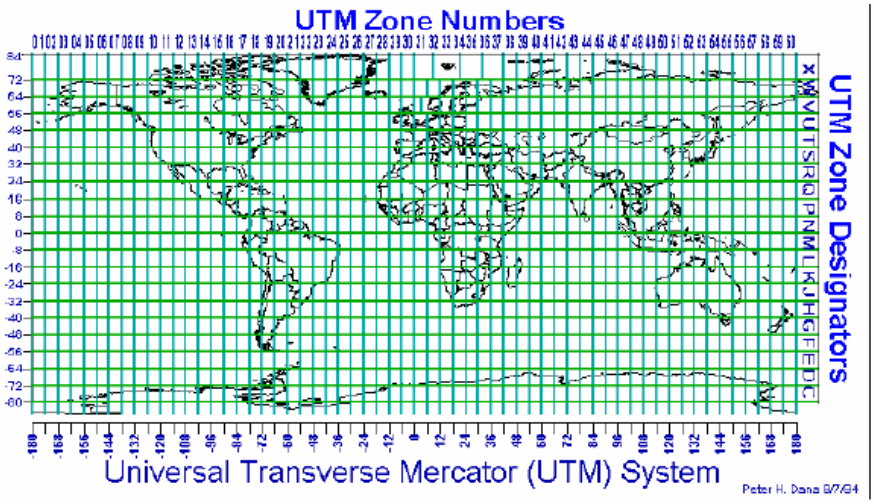
Salah satu proyeksi peta yang terkenal dan sering digunakan adalah UTM. Sebagai ciri hasil proyeksi UTM ini pada sebuah peta, yaitu terdapatnya garis lintang *(Latitude)* dan garis bujur *(Longitude).* Keuntungan peta ini adalah menggunakan sistem koordinat global (seluruh dunia) sehingga apabila kita menggambarkan suatu daerah yang diketahui *Latitude* dan *Longitude*-nya maka apabila kita mau menggabungkan satu peta

dengan peta yang lainnya tidak akan sulit.

Berikut akan dijelaskan mengenai sistem proyeksi ini: Pada sistem proyeksi ini didefinisikan posisi *horizontal* dua dimensi (x,y)utm dengan menggunakan proyeksi silinder, transversal, dan konform yang memotong bumi pada dua meridian standard. Seluruh permukaan bumi, dalam sistem koordinat ini, dibagi menjadi 60 bagian yang disebut sebagai zone UTM. Setiap *zone* ini dibatasi oleh dua meridian sebesar 6° dan memiliki meridian tengah sendiri. Sebagai contoh, *zone* 1 dimulai dari 180° BB hingga 174°BB, *zone* 2 dari 174°BB

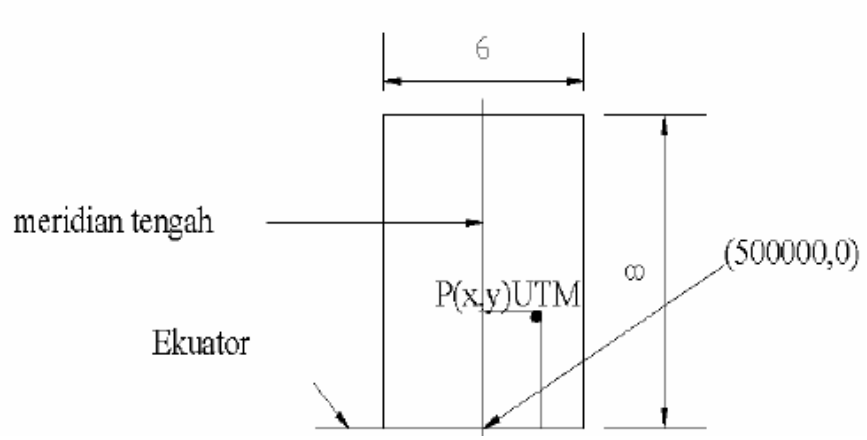
hingga 168°BB, terus ke arah timur hingga *zone* 60 yang dimulai dari 174°BT hingga 180°BT.

Batas lintang di dalam sistem koordinat ini adalah 80° LS hingga 84° LU. Setiap bagian derajat memiliki lebar 8° yang pembagiannya dimulai dari 80° LS ke arah utara. Bagian derajat dari bawah (LS) dinotasikan dimulai dari C,D,E,F, hingga X (tetapi huruf I dan O tidak digunakan). Jadi, bagian derajat 80° LS hingga 72° LS diberi notasi C, 72° LS hingga 64° LS diberi notasi D, 64° LS hingga 56° LS diberi notasi E, dan seterusnya.



Gambar 2.7 Sistem proyeksi UTM

Setiap zone UTM memiliki sistem koordinat sendiri dengan titik nol sejati pada perpotongan antara meridian sentralnya dengan ekuator. Dan, untuk menghindari koordinat negatif, meridian tengah diberi nilai awal absis (x) 500.000 meter. Untuk *zone* yang terletak di bagian selatan ekuator (LS), juga untuk menghindari koordinat negatif, ekuator diberi nilai awal ordinat (y) 10.000.000 meter. Sedangkan untuk *zone* yang terletak di bagian utara ekuator, ekuator tetap memiliki nilai ordinat 0 meter.



Gambar 2.8 salah satu zone UTM

Wilayah Indonesia terbagi dalam 9 *zone* UTM, mulai dari meridian 90° BT hingga meridian 144° BT dengan batas paralel (lintang) 11° LS hingga 6°LU. Dengan demikian, wilayah Indonesia dimulai dari *zone* 46 (meridian sentral 93° BT) hingga zone 54 (meridian sentral 141° BT).

2. Non-Earth

Proyeksi Non-Earth ini merupakan proyeksi yang menggunakan koordinat lokal. Proyeksi ini biasanya digunakan untuk mendigitasi (map info) berupa suatu denah atau peta tersebut bersifat *independen* (hanya terdiri 1 lembar peta tersebut)

Peta ≅ SIG

Peta	**SIG**
• Statis	• Staris & Dinamis
• Proses updating mahal	• Proses updating murah
• Kompleks	• Fleksibel
• Diskrit (lembar per lembar)	• Kontinyu & yang perlu saja
• Analisa & modeling secara langsung tidak mungkin	• Analisa & modeling secara langsung sangat mungkin
• Menurunkan (generate) data perlu interpretasi	• Menurunkan (generate) data tidak perlu interpretasi

2.6 Analisis dan Perancangan Sistem

Perancangan sistem merupakan tahap setelah analisis dan siklus pengembangan sistem, pendefinisian kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi, menggambarkan suatu sistem yang akan dibentuk.

2.6.1 Unified Modelling Langguage (UML)

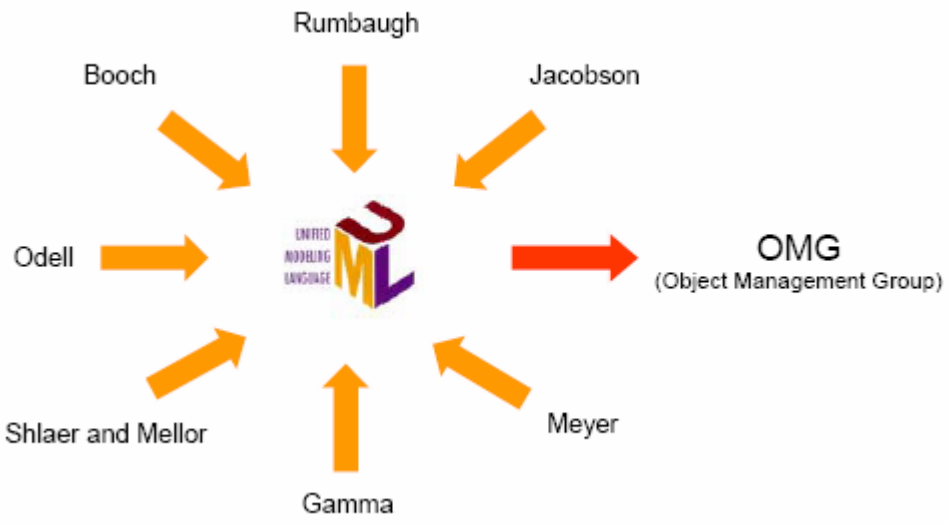
Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

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.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah

ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch [1], metodologi coad [2], metodologi OOSE [3], metodologi OMT [4], metodologi shlaer-mellor [5], metodologi wirfs-brock [6], dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.



Gambar 2.9. Sejarah UML

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh **Object Management Group** (OMG – http://www.omg.org). Tahun 1997 UML versi 1.1 muncul. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999 [7] [8] [9]. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

2.6.2. Konsepsi Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar dibawah.

Major Area	*View*	*Diagrams*	*Main Concepts*
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Gambar 2.10. Konsepsi Dasar UML

Abstraksi konsep dasar UML yang terdiri dari *structural classification, dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams*. *Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagaram tersebut.

Lalu darimana kita mulai ? Untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan:

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

- *use case diagram*
- *class diagram*
- *statechart diagram*
- *activity diagram*
- *sequence diagram*
- *collaboration diagram*
- *component diagram*
- *deployment diagram*

2.6.3. Use Case Diagram

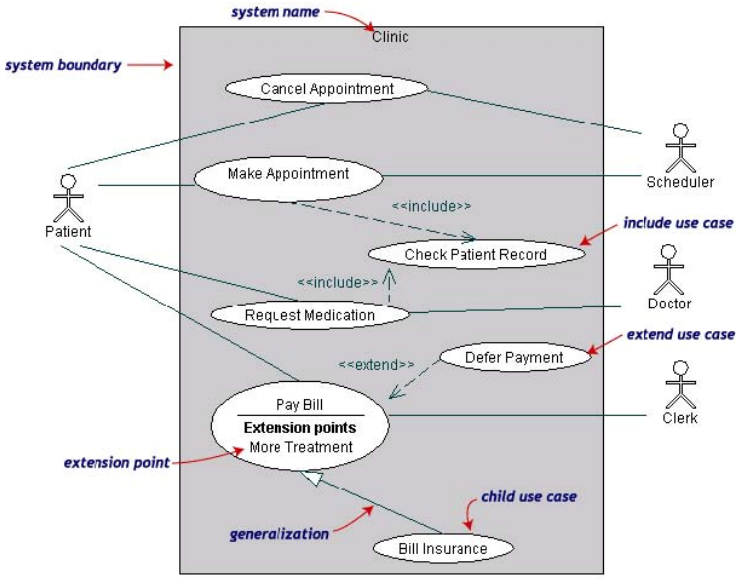
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada system.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Contoh use case diagram :



Gambar 2.11. Gambar contoh Use Case Diagram

2.6.4. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

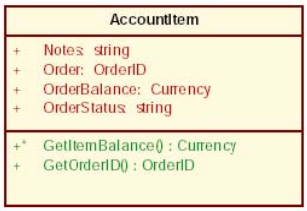
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

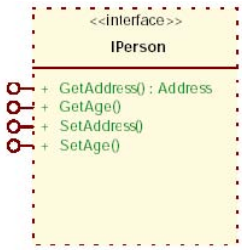
Atribut dan metoda dapat memiliki salah satu sifat berikut :

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja



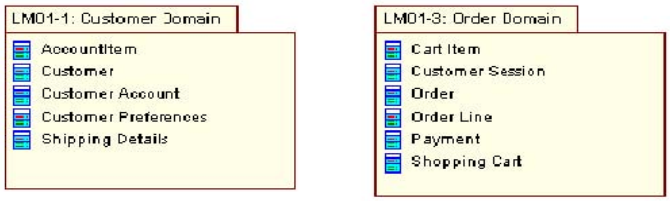
Gambar 2.12. Public

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.



Gambar 2.13. *Class* implementasi dari sebuah interface

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

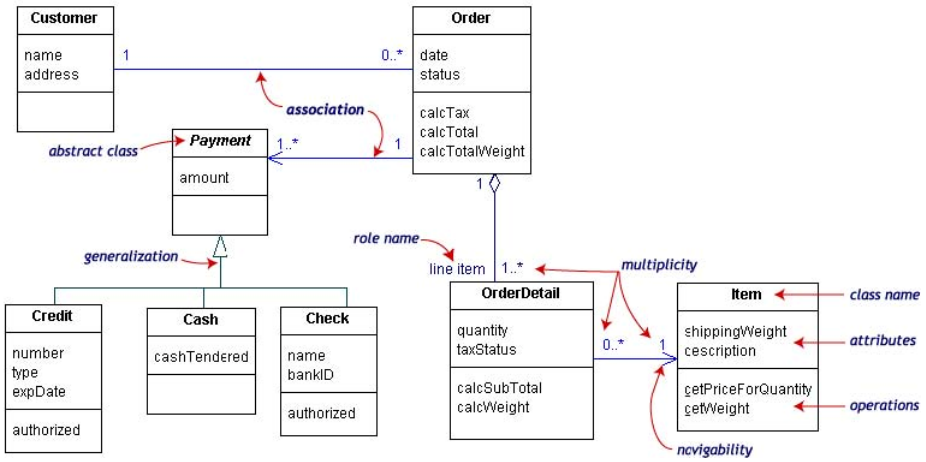


Gambar 2.14. *Class* sebagai *Package*

2.6.5. Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Contoh *class diagram* :



Gambar 2.15. *Class* Diagram

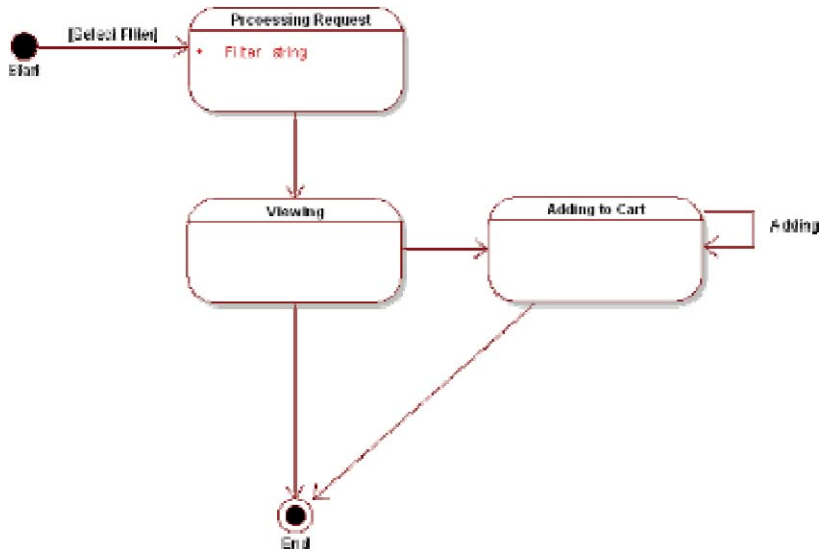
2.6.6. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *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*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

Contoh statechart diagram :



Gambar 2.16. *Statechart* Diagram

2.6.7. Activity Diagram

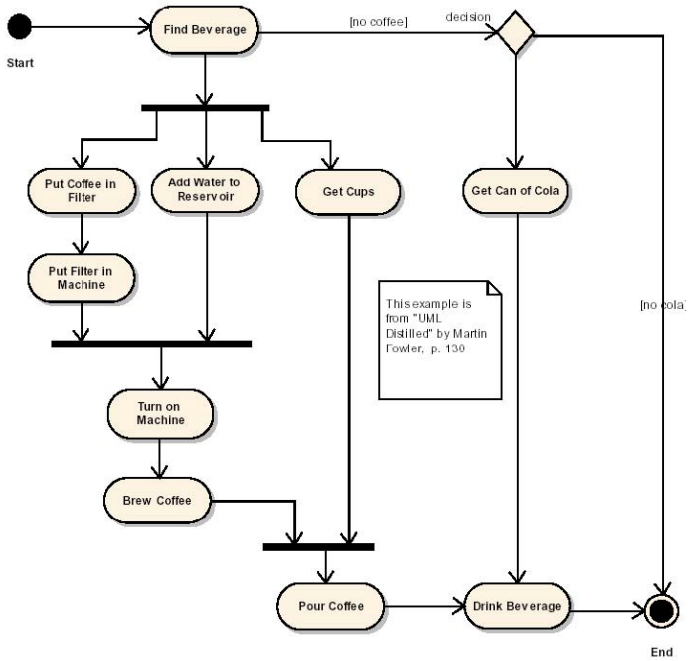
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

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.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Contoh *activity diagram* tanpa *swimlane*:



Gambar 2.17. *activity diagram* tanpa *swimlane*

2.6.8. Sequence Diagram

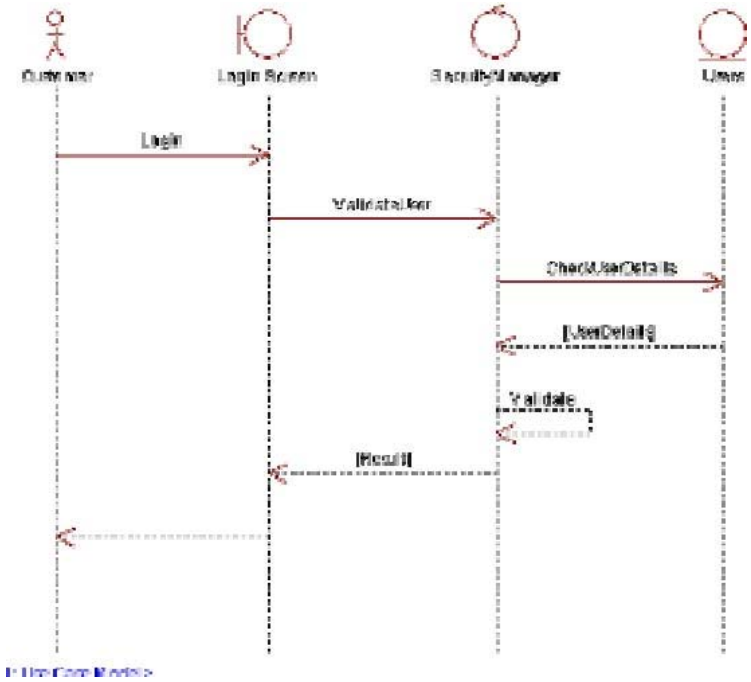
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

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.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

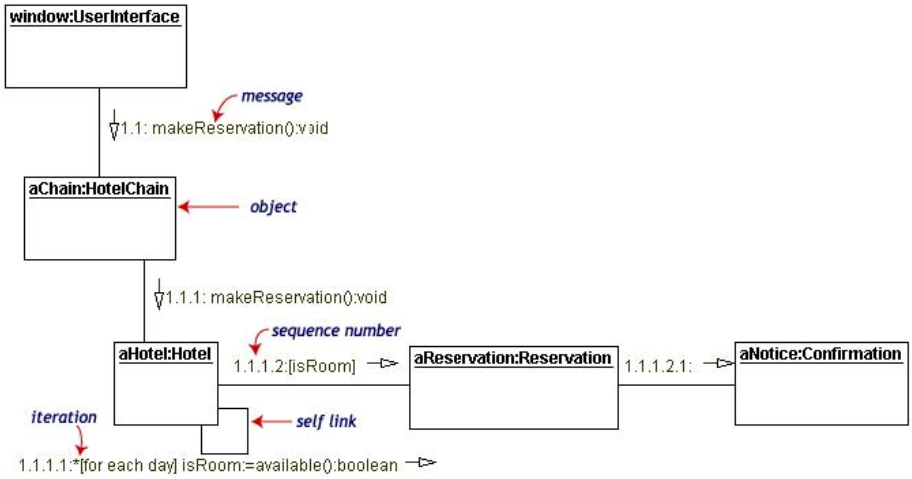
Contoh *sequence diagram* :



Gambar 2.18. *sequence diagram*

2.6.9. Collaboration Diagram

Collaboration diagram juga 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*, di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

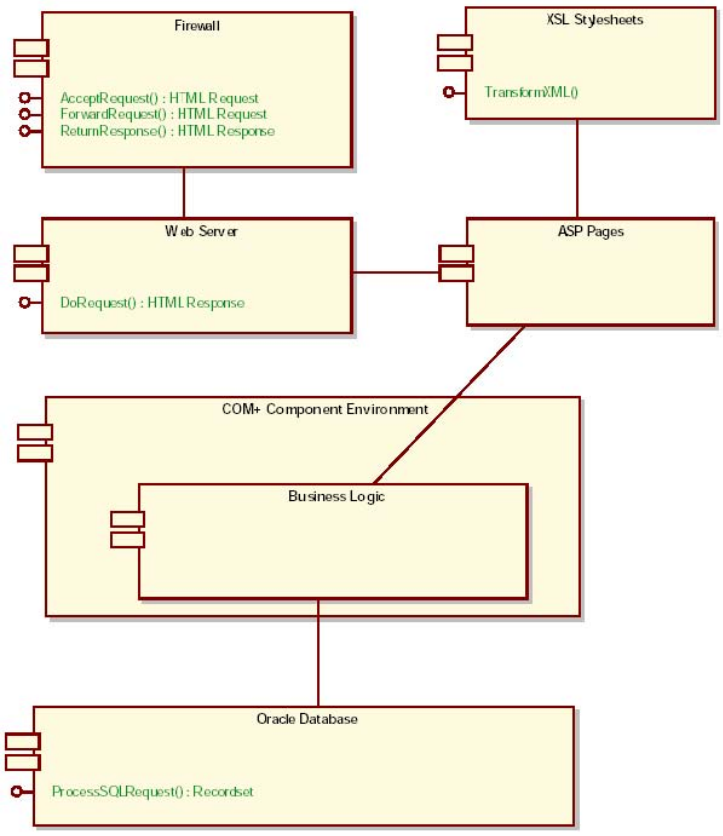


Gambar 2.19. *Collaboration diagram*

2.6.10. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh *component diagram*:

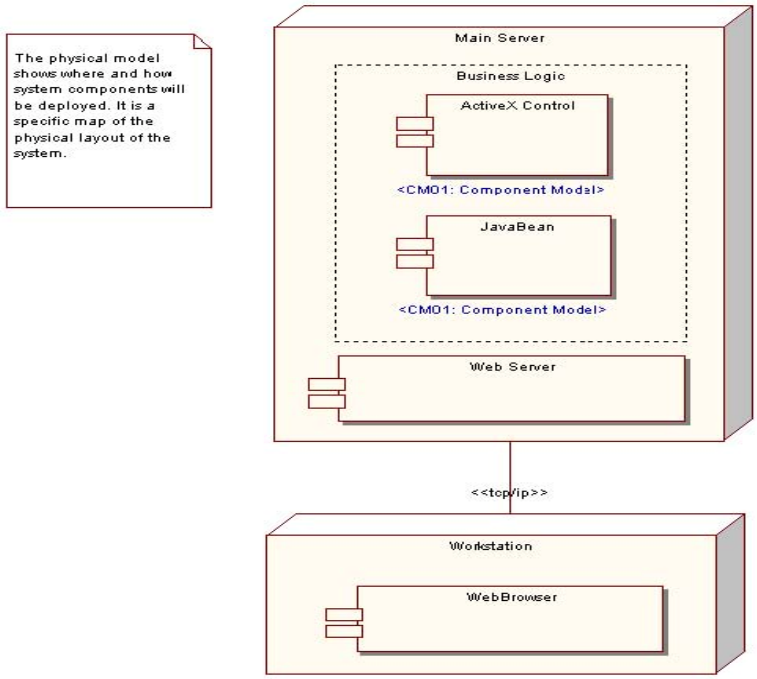


Gambar 2.20. *Component diagram*

2.6.11. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana 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

Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar 2.21. *Deployment/physical diagram*

2.6.12. Langkah – Langkah Penggunaan UML

Berikut ini merupakan tips pengembangan piranti lunak dengan menggunakan UML

1. Buatlah daftar *business process* dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
2. Petakan *use case* untuk tiap *business process* untuk mendefinisikan dengan tepat fungsionalitas yang harus disediakan oleh sistem. Kemudian perhalus use case diagram dan lengkapi dengan *requirement*, *constraints* dan catatan-catatan lain.
3. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.

4. Definisikan *requirement* lain (non-fungsional, *security* dan sebagainya) yang juga harus disediakan oleh sistem.
5. Berdasarkan *use case diagram*, mulailah membuat *activity diagram*.
6. Definisikan objek-objek level atas (*package* atau *domain*) dan buatlah *sequence* dan/atau *collaboration diagram* untuk tiap alir pekerjaan. Jika sebuah *use case* memiliki kemungkinan alir normal dan error, buatlah satu diagram untuk masing-masing alir.
7. Buatlah rancangan *user interface* model yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario *use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domain* dipecah menjadi hirarki *class* lengkap dengan atribut dan metodanya. Akan lebih baik jika untuk setiap *class* dibuat *unit test* untuk menguji fungsionalitas *class* dan interaksi dengan *class* lain.
9. Setelah *class diagram* dibuat, kita dapat melihat kemungkinan pengelompokan *class* menjadi komponen-komponen. Karena itu buatlah component diagram pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen meyakinkan ia berinteraksi dengan baik.
10. Perhalus *deployment diagram* yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan, dan sebagainya. Petakan komponen ke dalam node.
11. Mulailah membangun sistem. Ada dua pendekatan yang dapat digunakan :
    - Pendekatan *use case*, dengan meng-*assign* setiap *use case* kepada tim pengembang tertentu untuk mengembangkan *unit code* yang

lengkap dengan tes.

- Pendekatan komponen, yaitu meng-*assign* setiap komponen kepada tim pengembang tertentu.

12. Lakukan uji modul dan uji integrasi serta perbaiki model berserta *code*nya. Model harus selalu sesuai dengan *code* yang aktual.
13. Piranti lunak siap dirilis.

2.7. Sejarah Internet

Pada awalnya internet merupakan suatu jaringan komputer yang dibentuk oleh departemen pertahanan Amerika di awal tahun enam puluhan, melalui proyek ARPA *(Advance Research Project Agency)* yang disebut ARPANET, dimana mereka mendemonstrasikan bagaimana dengan *hardware* dan *software* komputer yang berbasis UNIX, kita bisa melakukan komunikasi dalam jarak tak terhingga melalui saluran telepon. Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan, dan akhirnya semua standar yang mereka tentukan menjadi cikal bakal untuk pengembangan protokol baru yang sekarang dikenal sebagai TCP/IP *(Tranmission Control Protocol / Internet Protocol)*

ARPANET dibentuk secara khusus oleh empat universitas besar di Amerika, yaitu Stanford Research Institute, University of california at Santa Barbara, University of California at Los Angeles, dan University of Utah, dimana mereka membentuk suatu jaringan terpadu di tahun 1969, dan secara umum ARPANET diperkenalkan pada bulan Oktober 1972. Pada tahun 1981, jumlah

komputer yang tergabung dalam ARPANET hanya 213 komputer, kemudian di awal tahun 1986, bertambah menjadi 2308 komputer, dan 1,3 juta komputer pada tahun 1993. Pada awal tahun delapan puluhan seluruh jaringan yang tergabung dalam ARPANET diubah menjadi TCP/IP, karena proyeknya sendiri sudah dihentikan, dan jaringan ARPANET inilah yang merupakan koneksi utama *(backbone)* dari *internet*.

Proyek percobaan tersebut akhirnya dilanjutkan dan dibiayai oleh NSF *(National Science Foundation),* suatu lembaga pengetahuan seperti LIPI di Indonesia. NSF lalu mengubah nama jaringan ARPANET menjadi NSFNET dimana *backbone*nya memiliki kecepatan tinggi yang dihubungkan ke komputer-komputer yang ada di universitas dan lembaga penelitian terbesar di Amerika. Setelah itu pemerintah Amerika Serikat memberikan ijin ke arah komersial pada awal tahun 1950. jelaslah bahwa awalnya *internet* dikenal sebagai suatu wadah bagi para peneliti untuk saling bertukar informasi kemudian dimanfaatkan oleh perusahaan-perusahaan komersial sebagai sarana bisnis mereka. Sat ini pengguna *internet* tersebar di seluruh dunia dengan jumlah mencapai lebih dari 250 juta orang.

2.7.1 Fasilitas Internet

Saat ini jika orang berbicara tentang *internet*, yang mereka maksud adalah *word wide web (www).* Pada kenyatannya *internet* memiliki banyak bagian yang lain, yaitu:

1. *World wide web* disingkat dengan web adalah bagian yang paling menarik dari *internet*. Melalui *web* kita bisa mengakses informasi yang tidak hanya berupa teks, tetapi juga gambar-gambar, suara-suara, film, dan lain-lain. Untuk mengakses web dibutuhkan *software* yang disebut *browser*. *Browser* yang terpopuler saat ini adalah microsoft Internet Explorer dan Netscape Communicator.
2. *Electronic Mail* disingkat *E-mail* adalah surat elektronik yang dikirim melalui *internet*. Dengan fasilitas ini kita bisa mengirim atau menerima *email* dari dan ke pengguna *internet* di seluruh dunia. Jika dibandingkan dengan pos, fasilitas *email* jauh lebih cepat kita bisa mengirim email ke teman yang ada di luar negeri hanya dalam beberapa menit dan tidak perlu khawatir, surat tersebut akan rusak karena hujan atau hal-hal lainnya. Untuk mengirim atau menerima *e-mail* dibutuhkan program *e-mail* yang terdapat di Microsoft Internet Explorer dan Netscape Communication. Yang perlu dibutuhkan hanya mengetik surat tersebut dan mengirimkannya. Di *internet* juga banyak tersedia *mailing list*, fasilitas yang memungkinkan kita untuk saling berbagi informasi mengenai topik-topik tertentu dengan orang lain. Jika kita bergabung dengan salah satu

mailing list, kita bisa menerima dan mengirim *e-mail* ke semua anggota *mailing list* tersebut.

3. Telnet. Dengan menggunakan telnet kita bisa menggunakan komputer untuk berhubungan dengan komputer lain dan mencari atau mengambil informasi-informasi yang ada di komputer tersebut.
4. *File Transfer Protokol* disingkat FTP. Melalui *software* FTP, anda bisa mengirim data atau file di sebuah komputer ke komputer lain. Proses mengirim file dari sebuah komputer ke komputer anda disebut dengan proses *download*, sedangkan proses mengirim file dari komputer anda ke komputer lain disebut dengan *upload*. Jika anda pernah membuat *homepage*, maka kata *upload* tidak asing lagi.
5. *Gopher* adalah sistem dimana pemakai dapat mengakses informasi dari komputer lain. Beda *gopher* dengan *web* adalah tidak bisa menampilkan gambar, melainkan hanya teks. Oleh sebab itu *gopher* banyak ditingalkan para pemakai *internet* saat ini.
6. *Chat Group* atau *Internet Relay Chat* (IRC) adalah forum dimana pemakai dapat saling berdiskusi atau berbincang-bincang dengan pemakai lain.
7. *News Group* bisa disebut ruang percakapan bagi para anngota yang mempunyai kemungkinan yang sama. Di *internet* tersedia bermacam-macam news group dengan tema yang berbeda-beda. Untuk dapat menikmati fasilitass ini kta harus terkoneksi ke ISP yang menyediakan fsilitas *news group*, karena tidak semua ISP menyediakannya. Adapun fasilitas yang anda gunakan semuanya merupakan bagian dari *internet*.

Melalui fasilitas-fasilitas *internet* itu, kita bisa melakukan banyak hal yang akan menambah pengetahuan dan wawasan kita.

2.7.2 Syarat tergabung ke internet

Untuk berhubungan dengan i*nternet* tentunya harus mempunyai komputer. *Internet* adalah jaringan komputer, oleh karena itu jalan satu-satunya untuk berhubungan dengan *internet* adalah melalui komputer. Selain itu juga diperlukan *harddisk* yang berisi *software* untuk berhubungan dengan *Internet Service Provider* dan internet seperti *web browser*, dan *harddisk* untuk menyimpan informasi-informasi yang anda *download*. Untuk mengakses *internet* tidak cukup hanya dengan komputer saja, tetapi ada alat bantunya seperti:

1. ***Modem***. *Modem* adalah perangkat *hardware* tambahan untuk komputer (baik jenis *card internal* maupun *eksternal* yang terletak di luar komputer). Pada dasarnya *modem* adalah alat yang memungkinkan komputer anda untuk berbicara dengan komputer lain melalui kabel telepon. Kata *modem* berasal dari kata Modulasi Demodulasi yang bisa diartikan sebagai proses perubahan denyut elektronis dari komputer menjadi suara (modulasi) sehingga dapat dikirimkan melalui kabel telepon. Modem penerima akan mengubah nada suara tadi menjadi denyut elektronis kembali untuk diproses selanjutnya oleh komputer (demodulasi).
2. **Telepon Modem** memerlukan telepon untuk melakukan tugasnya. Proses pada saat *modem* terhubung dengan telepon dan anda memulai hubungan *internet service privider* pada dasarnya sama dengan proses anda

menelepon biasa. Jika ada yang menelepon anda pada saat menggunakan modem, maka dia akan menerima nada sibuk dan tidak bisa menghubungi anda, karena *modem* dan telepon berkaitan erat, kecepatan *modem* juga akan berpengaruh terhadap rekening telepon yang harus anda bayar setiap bulan, sehingga *modem* yang cepat akan menghemat uang daripada *modem* yang lambat.

3. ***Software***. *Software* diperlukan untuk berhubungan dengan *internet*. Tanpa *software*nya sesuai, anda tidak bisa mengakses *internet*. Contoh *software* yang diperlukan adalah:

a. Microsoft Internet Explorer dan Netscape Communicator adalah *browser* untuk mengakses *web* yang paling baik dan terpopuler saat ini. Kedua *browser* ini juga sekaligus berfungsi sekaligus berfungsi sebagai FTP, *mail* dan *news group*.
b. Cute FTP, Mozilla dan WSFTP, adalah *software* khusus untuk FTP
c. Untuk *e-mail* anda bisa menggunakan *Internet Mail/ Outlook express* yang merupakan bagian dari Netscape Communicator.
d. MIRC, Yahoo Messenger, atau ICQ adalah program untuk *chating* yang paling popular saat ini. Dengan program ini anda bisa berbincang-bincang dengan orang lain melalui *keyboard*.
e. Internet Service Provider. Untuk bisa bergabung dengan *internet*, anda harus mempunyai akses dengan cara berlangganan ke pentedia jasa *internet* atau yang lebih sering disebut *Internet Service Provider* (ISP). ISP adalah perusahaan yang menawarkan jasa pelayanan kepada anda untuk

berhubungn dengan *internet*. Untuk mengakses, anda perlu cukup menghubungi Internet Service Privider melalui komputer dan modem, selanjutnya *Internet Service Privider* akan mengurus detail-detail yang diperlukan untuk berhubungan dengan internet termasuk biaya SLJJ koneksi tersebut.

2.8. Pengertian PHP

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinam untuk digunakan untuk pemakaian lain. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (Form Interpreted). Pada saat tersebut PHP adalah sekumpulan script yang digunakan untuk mengolah data form dari web.Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah *Personal Home Page/Form Interpreter*. Dengan pelepasan kode sumber ini menjadi *open source*, maka banya programmer yang tertarik untuk ikut mengembangkan PHP.Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter sudah diimplementasikan dalam C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan [6].

Pada tahun 1997, sebuah perusahaan bernama *Zend*, menulis ulang interpreter PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0.Pada pertengahan tahun 1999, *Zend* merilis

interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi.Pata Juni 2004, *Zend* merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman kearah pemrograman berorientasi objek.

PHP merupakan bahasa pemrograman web, penulisan script PHP dapat dijadikan satu dengan kode html dan disimpan dalam satu *file* yang sama ataupun berdiri sendiri dengan nama *file* berekstensi *.php. Ciri khas PHP diawali dengan tanda “<?php“ dan diakhiri dengan “?>”. PHP memiliki fasilitas untuk melakukan koneksi ke *database*. Script program PHP disimpan dan dijalankan di sisi *server* sehingga keamanan data lebih terjamin. Contoh penggunaan script php sebagai berikut :

```
<? php
phpinfo ( );
?>
```

2.9. Pengertian MySQL

MySQL merupakan sebuah sistem manajemen *database open source* yang populer dan gratis untuk *platform UNIX*. Sistem manajemen *database MySQL* menggunakan kumpulan perintah sederhana untuk memasukkan, memanggil,

menghapus, dan memperbarui data, dengan ini kita dapat mengembangkan *database* yang kompleks [6].

Beberapa kemampuan *MySQL* adalah sebagai berikut :

a. *MySQL* bisa diakses dan dimanipulasi dari sejumlah bahasa pemrograman terkenal, diantaranya adalah *C, C++, Java, Perl, Phyton,* dan *PHP*
b. *MySQL* mendukung tipe data yang umum digunakan, termasuk *FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, SET,* dan *ENUM*
c. *MySQL* memungkinkan alokasi *password* pada tiap *server* yang melalui *MySQL* untuk melakukan autentifikasi terenkripsi
d. *MySQL* mendukung berbagai macam metode koneksi, seperti TCP/IP, *socket UNIX*, dan koneksi untuk *Windows* NT/2000
e. *MySQL* bisa diperoleh secara gratis termasuk aplikasi-aplikasi lain yang diperlukan dalam memakai *MySQL*.

MySQL juga merupakan salah satu sistem manajemen *database* yang stabil di pasaran. Ketika *MySQL* diluncurkan pertama kali pada pertengahan 1996, beberapa *bug* dengan cepat dapat diketahui dan diperbaiki. Sekarang *MySQL* sudah menjadi sangat stabil dan mampu penyimpan data yang besar hingga sanggup menampung data lebih dari 4 Giga*byte*s. *Database MySQL* mendukung hampir seluruh bahasa pemrograman, termasuk PHP dan C.

2.9.1 Karakteristik MySQL

Berikut ini digambarkan beberapa karakteristik dari database MySQL [6] :

a. Dapat berintegrasi dengan C, C++, Eiffel, Java, Perl, PHP, Python and Tcl APIs.
b. Dapat bekerja pada beberapa platform OS yang berbeda
c. Memiliki banyak tipe kolom: *signed/unsigned integers* dengan panjang 1, 2, 3, 4, and 8 bytes, tipe: FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, and ENUM.
d. Mendukung penuh operator and fungsi untuk *query* SELECT dan WHERE

   Contoh :

   SELECT CONCAT (first_name," ",last_name)FROM tbl_name
   WHERE income/dependents > 10000 AND age > 30;

e. Fungsi-fungsi SQL diimplementasikan secara penuh pada *class library* yang teroptimalkan sehinggsa proses kerja sistem menjadi cepat, dan biasaanya tidak akan memakan alokasi memori setelah proses inisialisasi query
f. Mendukung penuh pada SQL *query* : GROUP BY dan ORDER BY. mendukung pada group fungsi (COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX() and MIN())
g. Mendukung penuh pada *relational databases*

h. Dapat mensupport ODBC *(Open-DataBase-Connectivity)* untuk Win32 (dengan *source*)
i. Mampu menangani sampai dengan 32 *index* per tablennya, Setiap *index* terdiri atas 1 sampai 16 colum atau bagian-bagian colum. Panjang Maksimal index sampai 500 *bytes*. *Indeks* bisa menggunakan *fields char* atau *varchar*
j. Setiap kolom memiliki nilai default
k. Mampu menempatkan alokasi memory dalam sistem dengan efisien
l. Memiliki fasilitas myisamchk, yang dapat melakukan *checking*, optimasi dan perbaikan dengan sangat cepat
m. Data-data tersimpan dalam suatu *character set* yang terpilih
n. *Aliases* pada table dan kolom dimungkinkan pada standar SQL92
o. Server dapat memberikan pesan error kepada *clients* pada berbagai macam bahasa.

Clients dapat tersambung ke MySQL server dengan menggunakan *TCP/IP Sockets, Unix Sockets (Unixes)*, atau *Named Pipes (NT).*

2.9.2. Kelebihan MySQL

Dengan karakteristik MySQL tersebut diatas maka dapat memberikan kelebihan sebagai berikut [6] :

a. Menghemat waktu proses pengisian data
b. Menghemat waktu proses pengambilan data
c. Proses pengambilan data lebih fleksibel.

d. Data dapat diakses secara bersama oleh lebih dari satu pengguna pada waktu yang bersamaaan.
e. Kemampuan akses data di berbagai tempat dengan menggunakan transmisi elektronik.