BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Konsep Dasar Sistem

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai berikut :

Pengertian Menurut JOG :

"Sistem adalah suatu jaringan kerja dari prosedur- prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu."

Pendekatan yang menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut :

Menurut JOG :

"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu."

Terdapat juga pengertian sistem yang lain yaitu :

Menurut ZUL:

“Sistem adalah himpunan sesuatu “benda” nyata atau abstrak yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, dan saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan untuk mencapai tujuan tertentu secara efisien dan efektif.”

Dari definisi diatas dapat pula disimpulkan bahwa sistem adalah :Kumpulan dari elemen-elemen yang saling berhubungan, berkaitan, berhubungan, berketergantungan dan saling mendukung membentuk satu kesatuan atau organisasi untuk mencapai tujuan dan sasaran tertentu secara efektif dan efisien

Karaktersitik Sistem

Karakteristik sistem menurut Jogiyanto H.M dalam bukunya yang berjudul *Pengenalan Komputer* adalah:

Menurut JOG :

Sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen (components), batas sistem (boundary), lingkungan luar sistem (environment), penghubung (interface), masukan (input), keluaran (output), pengolah (proses), sasaran (objectives),atau tujuan (goal).

Berikut ini penjelasan dari karakteristik sistem diatas:

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerjasama membentuk satu kesatuan. Komponen sistem dapat berupa subsistem atau bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut dengan supra sistem.

b. Batas Sistem

Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem

Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung *(Interface)* Sistem

Merupakan media penghubung antara satu subsistem dengan subsistem lainya.

e. Masukan *(Input)* Sistem

Adalah energi yang dimasukkan kedalam sistem.

f. Keluaran *(Output*) Sistem

Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah Sistem (Proses)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

h. Sasaran (*Objective*s) Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran, bila tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan.

Klasifikasi Sistem

Sistem juga dapat diklasifikasikan dari beberapa sudut pandang, klasifikasi sistem menurut jogiyanto H. M (1988 : 779) diantaranya adalah sebagai berikut :

1. Sistem Abstrak dan Sistem Fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara phisik. Sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem Alamiah dan Sistem Buatan Manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia, sistem ini melibatkan interaksi antara manusia dengan mesin.

3. Sistem Tertentu dan Sistem Tak Tentu

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi antara bagian-bagiannya dapat dideteksi dengan pasti. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Sistem Terbuka

   Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengn lingkungan luarnya.

Konsep Dasar Informasi

Informasi mempunyai peranan yang sangat penting dalam organisasi. Suatu organisasi yang kurang mendapatkan informasi akan luruh,kerdil dan pada akhirnya akan berakhir. Informasi dapat didefinisikan sebagai berikut :

Menurut JOG :

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

Menurut ZUL:

" Informasi adalah data yang sudah diolah, atau dimanipulasi sesuai dengan keperluan tertentu."

Dari definisi-definisi informasi diatas maka dapat diambil kesimpulan bahwa sumber informasi adalah data yang telah diolah yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata.

Kualitas informasi

Kualitas informasi terdiri dari tiga hal, yaitu:

1. Akurat *(accurate)*

   Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan, harus jelas mencerminkan maksudnya.

2. Tepat pada waktunya *(timeliness)*

   Tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat.

3. Relevan *(relevance)*

Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya.

Nilai informasi

Nilai dari informasi *(value of information)* ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkanya. Suatu informasi dikatakan bernilai jika manfaatnya lebih efektif dibandingkan biaya mendapatkannya. Lebih lanjut sebagian besar informasi tidak dapat persis ditaksir keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitasnya.

Definisi Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah darimana informasi tersebut bisa didapatkan?. Informasi dapat diperoleh dari sistem informasi (*information systems*) atau disebut juga dengan *processing systems* atau *information processing systems* atau *information-generating systems*. Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai berikut:

Sistem informasi adalah suatu sistem didalam 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.

2.2 Komponen Sistem Informasi

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technologi block*), blok basis data (*database block*), dan blok kendali (*komputers block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi merupakan “kotak alat” (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima output, menjalankan model,

menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi. Misalnya teknisi adalah operator komputer, pemrogram, operator pengolah kata, spesialis telekomunikasi, analis sistem, penyimpanan data dan lain sebagainya.

5. Blok Basis Data

   Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

6. Blok Kendali

   Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk

meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat.

2.3 Konsep Merancang Basis Data

2.3.1 Definisi

Istilah basis data banyak menimbulkan interpretasi yang berbeda. Anthoni J. Fabbri dan A. Robert Schwab, mendefinisikan basis data sebagai berikut :

Basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data

Menurut George Tsu-der Chou, basis data dapat didefinisikan sebagai berikut :

Basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tatacara yang khusus.

Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. Sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara data opersional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien.

2.3.2 Tahap Perancangan Basis Data

Perancangan basis data merupakan langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Perancangan basis data terdiri atas perancangan basis data secara konseptual, perancangan basis data secara logis, dan perancangan basis data secara fisis.

Beberapa komponen yanng terdapat pada perancangan basis data secara konseptual antara lain:

1. Entitas

   Entitas terkadang disebut tipe entitas atau kelas entitas. Entitas adalah objek yang dapat dibedakan dari objek-objek lainnya.

2. Atribut

   Atribut adalah item data yang menjadi bagian dari suatu entitas. Istilah lain dari attribut adalah properti.

3. Hubungan

   Hubungan adalah asosiasi atau kaitan antara dua entitas.

4. Kekangan

   Kekangan digunakan untuk melindungi integritas data (misalnya, melindungi kesalahan sewaktu pengisian data).

5. Domain

   Domain adalah himpunan yang berlaku bagi suatu atribut. Kekangan domain mendefinisikan nama, tipe, format, panjang, dan nilai masing-masing item data.

6. Integritas Referensial

   Integritas referensial adalah aturan-aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik tabel-tabel yang berbeda dalam suatu basis data relasional untuk menjaga konsistensi data.

2.3.3 Metode Pendekatan Sistem

2.3.3.1 Metode Object Oriented

Object Oriented Design (OOD) menerjemahkan model Object Oriented Analysis (OOA) dari dunia nyata ke dalam dunia model implementasi spesifik yang dapat direalisasikan ke dalam perangkat lunak. Proses OOD dapat digambarkan sebagai piramida yang terdiri dari empat lapisan. Lapisan dasar berfokus pada desain subsistem yang mengimplementasikan fungsi-fungsi sistem mayor; lapisan kelas mengkhususkan keseluruhan arsitektur objek dan hirarki kelas yang diperlukan untuk mengimplementasikan suatu sistem; Lapisan pesan menunjukkan bagaimana kolaborasi antar objek akan direalisasi; dan Lapisan tanggung jawab mengidentifikasikan atribut dan operasi yang menandai masing-masing kelas.

Sasaran OOA adalah mengembangkan sederetan model yang menggambarkan perangkat lunak komputer pada saat perangkat lunak itu bekerja untuk memenuhi serangkaian persyaratan yang ditentukan oleh pelanggan. Analisis terstruktur mengadopsi pandangan input, proses dan output yang berbeda mengenai persyaratan. Data dipertimbangkan secara terpisah dari proses mentransformasi data.

Banyak metode OOA yang dikembangkan. Masing-masing memperkenalkan proses untuk analisis produk atau sistem. Serangkaian model yang berkembang keluar dari proses itu dan notasi yang memungkinkan perekayasa perangkat lunak menciptakan masing-masing model dengan cara yang konsisten. Beberapa metode yang ada diantaranya [Pressman, 688]:

a) Metode Booch

Metode Booch meliputi proses pengembangan mikro maupun prosws pengembangan makro. Tingkat mikro menentukan serangkaian tugas analisis yang diaplikasikan lagi untuk masing-masing langkah pada proses makro. Dengan demikian, pendekatan evolusioner dijaga. Metode Booch didukung oleh berbagai piranti otomatis. Outline singkat dari proses pengembangan mikro metode OOA Booch adalah sebagai berikut :

1. Identifikasi kelas dan objek.
2. Identifikasi semnatik dari kelas dan objek.
3. Identifikasi hubungan di antara kelas dan objek.
4. Lakukan sederetan penyaringan.
5. Implementasikan kelas dan objek (dalam konteks OOS, hal ini mengimplikasikan perlengkapan model analisis)

b) Metode Coad dan Yourdon.

Metode Coad dan Yourdon sering dianggap sebagai salah satu dari metode OOA yang paling mudah dipelajari. Notasi permodelan relatif sederhana dan pedoman untuk mengembangkan model analisis tersebut

jelas. Outline singkat dari proses pengembangan metode OOA Coad dan Yourdon adalah sebagai berikut :

1. Identifikasi objek dengan menggunakan kriteria ‘apa yang dicari’
2. Tentukan struktur generalisasi-spesifikasi
3. Tentukan struktur keseluruhan bagian.
4. Identifikasi subjek (reperesentasi dari komponen subsistem).
5. Tentukan atribut.
6. Tentukan pelayanan.

c) Metode Jacobson.

Metode ini juga disebut OOSE (rekayasa perangkat lunak berorientasi objek) metode Jacobson versi sederhana dari metode objectory sebelumnya, yang juga dikembangkan oleh Jacobson. Metode ini dibedakan dari metode yang lain karena sangat menekankan use case-deskripsi atau skenario yang menggambarkan bagaimana pemakai berinteraksi dengan produk atau sistem. Outline singkat dari proses metode Jacobson :

1. Identifikasi pemakai sistem dan seluruh tanggung jawab mereka.
2. Bangun model persyaratan.
3. Bangun model analisis.

d) Metode Rambough.

Rambough dan rekan-rekannya mengembangkan Object Modelling Technique (OMT) menggunakan desain sistem dan desain tingkat objek. Aktivitas analsis menciptakan tiga model : model objek (reperesentasi objek, kelas, hirarki, dan hubungan), model dinamis (reperesentasi dari objek dan tingkah laku sistem)

dan model fungsional (representasi aliran informasi seperti DFD tingkat tinggi melalui sistem tersebut. Outline singkat dari pengembangan OOA Rambough adalah sebagai berikut :

1. Kembangkan pernyataan ruang kerja lingkup masalah.
2. Bangun model objek.
3. Kembangkan model dinamis.
4. Buat model fungsional untuk sistem tersebut.

Perlu dicatat bahwa kerja mulai menggabungkan metode Booch dan Rambough. Himpunan resultan dari notasi dan heuristik yang disebut juga Unlified Method menggabungkan penekanan data Rambough yang kuat dengan tingkah laku dan tekanan operasional dari Booch.

e) Metode Wirfs-Brock.

Metode Wirfs tidak membuat perbedaan yang jelas antara analisis dan tugas desain, malahan mengusulkan proses kontinu mulai dengan penilaian terhadap spesifikasi pelanggan dan berakhir dengan desain. Outline singkat mengenai tugas yang berhubungan dengan aktivitas Wirfs-Brock adalah sebagai berikut :

1. Evaluasi spesifikasi pelanggan.
2. Gunakan uraian gramatikal untuk mengekstrak keal calon dari spesifikasi tersebut.
3. Kelompokkan kelas dengan tujuan untuk mengidentifikasi superkelas.
4. Tentukan tanggung jawab untuk masing-masing kelas.

5. Identifikasi hubungan antar kelas.
6. Tentukan kolaborasi di antara kelas berdasarkan tangung jawab
7. Bangun representasi hirarki kelas untuk memperlihatkan hubungan pewarisan.
8. Bangunlah grafik kolaborasi untuk sistem.

Meskipun terminologi dan langkah untuk masing-masing metode OOA ini berbeda, keseluruhan proses OOA benar-benar mirip. Untuk melakukan OOA perangkat lunak harus melakukan langkah-langkah generik sebagai berikut :

a) Cari persyaratan pelanggan untuk sistem OOA.
b) Pilih kelas dan objek dengan menggunakan persyaratan dasar sebagai panduan.
c) Identifikasi atribut dan operasi untuk masing objek sistem.
d) Tentukan struktur dan hirarki yang mengorganisasi kelas,
e) Bangun suatu model hubungan objek.
f) Bangun suatu model tingkah laku objek.
g) Kaji model analisis OO terhadap use case skenario.

2.4 Pengertian *Unified Modeling Language (UML)*

Menurut Hans-Erik UML adalah bahasa untuk menspesifkasi, menvisualisasi , membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model., deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak

lainnya (STT[10]). Selain itu menurut Hans-Erik UML adalah bahasa pemodelan yang menggunakan konsep oreintasi *object.* UML dibuat oleh *Grady booch* , *James Rumbaugh*, dan *Ivar Jacobson* di bawah bendera *Rational Software Corp* (STT[10]). UML menyediakan notasi – notasi yang membantu memodelkan sistem dari berbagai perspektif.. UML tidak hanya digunakan dalam pemodelan perangkat lunak , namun hampir dalam semua bidang yang mebutuhkan pemodelan.

2.4.1 Bagian-bagian dari UML

Bagian–bagian utama dari UML adalah *view, diagaram, model element, dan general mechanism.*

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat gratifk, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam UML antarai lain : *use case view, logical viw , component viwm , concurrences view,* dan *deployment view*.

a. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors. Actor* yang berinterakasi dengan sistem dapat berupa user atau sistem lainnya. *View* ini digambarkan dalam *use case diagrams* dan kadang–kadang dengan *activity diagramsView* ini digunakan terutama untuk pelanggan , perancang (*designer*) , pengembang *(developer*), dan penguji sistem (*tester*).

b. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class , object*, dan r*elationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state sequence, collaboration, dan activity diagrams* untuk model dinamisnya, *View* ini digunakan untuk perancang (designer) dan pengembang (developer).

c. Component view

Mendiskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. *View* ini diganbarkan dalam *component view* dan digunakan untk pengembang (developer).

d. Concurrency view

Membagi sistem ke dalam proses dan prosessor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration,* dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

e. Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam deplovment diagrams dan digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

2. Diagram

Diagram berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

a. Use Case Diagram

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskirpsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbo*l namun juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam system.

b. Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara : *associated* (terhubunga satu sama lain), *dependent* (satu *class* tergantung / menggunakan *clas*s yang lain), *speciated* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagrams.*

c. State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *objec*t dari suatu *class* dan keadaan yang menyebabkan *state* berubah.. Kejadian dapat berupa

object lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class,* hanya yang mempunyai jumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

d. Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antar *object* juga interaksi antara *object* sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan digunakan *sequence diagrams*, tapi jika penekannya pada konteks gunakan *collaboration diagrams.*

f. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *user case* atau interaksi.

g. Component Diagram

Menggambarkan struktur fisik kode dari kompen. Komponen dapat berupa *source code*, komponen biner, atau *executeable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga menbuat pemetaan dari logical view ke *component view*.

h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukan hubungan komputer dengan perangkat *(nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

2.4.2 Gambaran dari UML

1. UML sebagai bahasa Pemodelan

UML merupakan bahasa pemodelan yang memiliki pembendaharaan kata dan cara untuk mempresentasikan secara fokus pada konseptual dan fisik dari suatu sistem. Contoh untuk sistem software yang *intensive* membutuhkan bahasa yang menunjukan pandangan yang berbeda dari arsitektur sistem, ini sama seperti menyusun.mengembangkan *software develpoment life cycle*. Dengan UML akan memberitahukan kita bagaimana untuk membuat dan membaca bentuk model yang baik, tetapi UML tidak dapat membertahukan model apa yang akan dibangun dan kapan akan membangun model tersebut. Ini merupakan aturan dalam *software development prosess.*

2. UML sebagai bahasa untuk Menggambar kan Sistem (Visualizing)

UML tidak hanya merupakan simbol grafikal, cukup dengan tiap simbol pada notasi UML merupakan penetapan semantik yang baik. Dengan cara ini, satu pengembang dapat menulis model UML dan pengembang lain atau perangkat yang sama lainnya dapat mengartikan bahwa model tersebut tidak ambigu. Hal ini

akan mengurangi *error* yang terjadi karena perbedaan bahasa daalm komuniksi model konseptual dengan model lainnya.

3. **UML sebagai bahasa untuk Menspesifikasi Sistem (Specifying)**

Maksudnya membangun model yang sesuai, tidak ambigu dan lengkap. Pada faktanya UML menunjukan semua spesifikasi keputusan analisis, desain dan implementasi yang penting yang harus dibuat pada saat pengembangan dan penyebaran dari sistem software intensif.

4. **UML sebagai bahasa untuk Membangun Sistem (Constructing)**

UML bukan bahasa pemograman visual, tetapi model UML dapat dikoneksikan secara langsung pada bahasa pemograman visual.

5. **UML sebagai bahasa untuk Pengdokumentasian Sistem (Documenting)**

Maksudnya UML menunjukan dokumentasi dari arsitektur sistem dan detail dari semua UML, hanya memberikan bahasa untuk memperlihatkan permintaan dan untuk tes UML, menyediakan bahasa untuk memodelkan aktivitas dari perencanaan project dan manajemen pelepasan (*release management*).

2.4.3 TUJUANPENGGUNAAN UML

Tujuan Pengunaan UML

1. Memodelkan susatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep beorientasi *object*
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

2.5 KOMPONEN-KOMPONEN YANG TERLIBAT DALAM USE CASE DIAGRAM

. Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram* , namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor* dimana *actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sisstem, hanya menerima infromasi dari sistem atau keduannya menerima dan memberi informasi pada sistem, *actor* hanya berinterkasi dengan use case tetapi tidak memiliki kontrol atas *use case*. *Acttor* digambarkan dengan *stick man.*

Ada beberapa kemungkinan yang menyebabkan *actor* tersebut terkait dengan sistem antara lain :

a. Yang berkepentingan terhadap sistem dimana adanya arus informasi baik yang diterimanya maupun yang dia inputkan ke sistem
b. Orang ataupun pihak yang akan mengelola sistem tersebut
c. *External resource* yang digunkan oleh sistem
d. Sistem lain yang berinteraksi dengan sistem yang akan dibuat

2. Use Case

Use Case adalah gambaran fungsionalitas dari suatu sistem , sehingga *costumer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan di bangun. *Use case* diagram adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih

dititikberatkan pada fungsionalitas yang ada pada sistem , bukan berdasarkan alur atau urutan kejadian. Cara menentukan Use Case dalam suatu sistem :

a. Pola perilaku perangkat lunak aplikasi
b. Gambaran tugas dari sebuah *actor*
c. Sistem atau “benda” yang memberikan sesuatu yang berniali kepada *actor*
d. Apa yang dikerjakan oleh suatu perangkat lunak (bukan bagaimana cara mengerjakannya).

3. Relasi dalam Use Case

Ada beberapa relasi yang terdapat pada *use case diagarm* :

a. *Association* , menghubungkan link antar element
b. *Generaliization* , disebut juga *inheritance* (pewarisan), sebuah element pdapat merupakan spesialiasi dari elemen lainnya.
c. *Dependency*, sebuah element bergantung dalam beberapa cara elemen lainnya.
d. *Aggregation*, bentuk assosiation dimana sebuah elemen berisi elemen lainnya

4. Use Case Diagram

Adalah gambaran grapichal dari beberapa atau semua actorm use case dan interaksi diantaranya yang memperkenalkan suatu sistem

2.6 CLASS DIAGRAM

2.6.1 Definsi Object dan Class

Object adalah gambaran dari entity, baik dunia nyata atau konsep dengan batasan – batasan dan pengertian yang tepat. *Objec*t bisa mewakili sesuatu yang nyata seperti komputer. Setiap *object* dalam sistem memiliki tiga karakteristik, yaitu *State* (stataus), *Behaviour* (sifat) dan *Identity* (identitas).

Cara mengidentifikasi *object* :

1. Pengelompokan berdasarkan kata / frase benda pada skenario
2. Berdasarkan daftar kategori object , antara lain :
    a. object fisik, contoh : pesawat telepon
    b. spesifikasi /rancangan/deskirpsi, contoh : deskripsi pesawat
    c. tempat , contoh : gudang
    d. transaksi , contoh : penjualan
    e. butir yang terlibat pada transaksi , contoh : barang jualan
    f. peran, contoh : pelanggan
    g. wadah , contoh : pesawat terbang
    h. piranti , contoh PABX
    i. kata benda abstrak, contoh : kecanduan
    j. kejadian, contoh: pendaratan
    k. aturan atau kebijakan, contoh : aturan diskon
    l. catalog atau rujukan , contoh : daftar pelanggan

Class adalah deskripsi sekelompok *objec*t dari *property* (atribut), sifat (operasi) relasi antar *object* dan sematik yang umum. *Class* merupakan *template*

untuk membent5uk *objex*t. Setiap *objec*t merupakan contoh dari beberapa *class* dan *object* tidak dapt menjadi contoh lebih dari *satu class*. Penamaan *class* menggunakan kata benda tunggal yang merupakan abtraksi yang terbaik. Pada UML , *class* digambarkan dengan segi empat yang dibagi. Bagian atas merupakan nama dari *class*. Bagian yang tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (operasi).

2.6.2 Interaction Diagram

1. Use Case Relization

Fungsionalitas *use case* direpresentasaikan dengan aliran peristiwa–peristiwa. Skenario digunakan untuk menggambarkan bagaimana u*se case–use case* direalisasikan sebagai interaksi antara *object–object. Use case realization* menggambarkan bagaimana realisasi dari setiap *use case* yang ada pada *use case model*. Untuk menggambarkan bagaimana realasasi dari suatu *use case* dapat menggunakan beberapa diagram, diantaranya adalah *Class Diagram owned by Use Case Realization* serta *Interaction Diagram.*

2. Interaction Diagram

Merupakan model yang menjelaskan bagaimana sejumlah *object* bekerja sama dalam beberapa kelakuan. *Interaction Diagram* menerangkan kelakuan dari suatu *use case.* Diagram ini menggambarkan sejumlah *object* dari pesan yang dijalankan antar *object* dengan *use case.* Ketika kita memberikan pesan, aksi yang dihasilkan adalah sebuah pernyataan tereksekusi yang membentuk abstraksi dari prosedur komputasi. Dalam UML, kita dapat memodelkan beberapa jenis aksi , yaitu :

1. Call : memanggil operasi yang ada pada *object*, *object* mungkin mengirim ke dirinya sendiri , menghasilkan pemanggilan lokal dari operasi
2. Return : mengembalikan nilai dari caller
3. Send : mengirimkan sinyal ke object
4. Create : membuat sebua object
5. Destroy : mematikan *object*, *object* mungkin saja mematikan diri.

Interaction diagram digunakan ketika kita ingin melihat kelakuan dari beberapa *object* dalam *use care* tunggal. Diagram ini baik saat menunjukan kolaborasi diantara *object-object*, namun kurang baik dalam mendefinisikan *behavior*. Ada dua macam *Interaction Diagram* yaitu : *Sequence Diagra*m dan *Collaboration Diagram.*

3. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara sejumlah *object* dalam urutan waktu. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antar *object* juga interaksi antar *object* yang terjadi pada titik tertentu dalam ekseskusi sistem. Dalam UML, *object* pada *diagram sequence* digambarkan dengan segi empat yang berisi nama dari *object* yang digarisbawahi. Pada *object* terdapat 3 cara untuk menamainya yaitu : nama *object* , nama *object* dan *class* serta nama *class.* Dalam *diagram sequence* setiap *object* hanya memiliki garis yang digambarkan garis putus – putus ke bawah. Pesan antar *object*

digambarkan dengan anak panah dari objext yang mengirimkan pesan ke *object* yang menerima pesan.

4. Collaboration Diagram

Collaboration Diagram merupakan cara alterantif untuk menggambarkan skenario dari sistem. Diagram ini menggambarkan interaksi *object* yang diatur *object* sekelilingnya dan hubungan antara setiap *object* dengan object yang lainnya.

Collaboration diagram berisi :

- *Objec*t yang digambarkan dengan segiempat
- Hubungan antara *object* yang digambarkan dengan garis penghubung
- Pesan yang digambarkan dengan teks danb panah dari object yang mengirim pesan ke penerima pesan

5. Perbedaan Sequence Diagram dengan Collaboration Diagram

Sequence Diagram memberikan cara untuk melihat skenario dari sistem berdasarkan waktu (apa yang terjadi pertama kali , apa yang terjadi selanjutnya). User akan lebih mudah membaca dan mengerti tipe diagram ini. Karenanya , sangat berguna pada fase analisis awal. Sedangkan *Collaboration Diagram* cenderung untuk memberikan gambaran besar dari skenario selama kolaborasi disusun dari object sekellilingnya dan hubungan antar *object* yang satu dengan lainnya. Diagram ini akan nampak digunakan pada pengembangan tahap desain , ketika kita merancang implemantasi dari hitungan.

2.7 State Transiton Diagram dan Activity Diagram

2.7.1 State Transition Diagram

Use case dan skenario menyediakan cara untuk menggambarkan kelakuan sistem yakni interaksi antara *object – object* di dalam sistem. Kadang – kadang diperlukan untuk melihat kelakuan di dalam *object*. *State transition diagram* menunjukan *state – state* dari *object* tunggal , *event – event* atau pesan yang menyebabkan transisi dari satu *state* ke *state* yang lain dan *action* yang merupakan hasil dari perubahan sebuat *state. State transition diagram* tidak akan dibuat untuk setiap *class* di *state,. State transition diagram* hanya dibuat untuk *class* yang berkelakuan dinamis. *Interaction diagram* dapat dipelajari untuk menentukan *dynamic object* di sistem, yaitu *object* yang menerima dan mengirim beberapa pesan. *State tranistion* dioagram juga sangat berguna untuk meneliti kelakuan dari sebuah kumpulan *whole class* dan *control class.*

2.7.2 States

State adalah sebuah kondisi selama kehidupan sebuah *object* ketika *object* memenuhi beberapa kondisi, melakukan beberapa *action* atau menunggu sebuah *event*. *State* dari sebuah *object* dapat di karakteristikan oleh nilai dari satu atau lebih atribut – atribut dari *class. State- state* dari sebuah *object* ditentukan dengan pengujian / pemeriksaan atribut = atribut dan hubungan – hubungan dari *objec*t.

2.7.3 State Transitions

State Transition merepresentasikan sebuah perubahan dari **state** awal ke sebuah state berikutnya (yang mungkin dapat sama dengan *state* awal.) sebuah *actio*n dapat menyertai sebah *state transition*. Ada dua cara untuk membuat

transisi sebuah *state- ototmatis* dan *tidak otomtis*. *State transition* yang otomatis terjadi ketikca *activity* dari *state* awal telah lengkatp – tidak ada *event* yang terasosiasi dengan *statae transition* yang belum bernama.. *State transition* yang tidak otomatis disebabkan oleh sebuah *event* ternama (salah satu dari *obje*ct atau dari luar sistem). Kedua tipe dari *state transition* dipertimbangkan untuk membuat waktu nol dan tidak dapat di interupsi. Sebuah *state transition* di representasikan oleh sebuah panah yang meunjukan dari *state* awal ke *state* berikutnya.

2.7.4 Special State

Ada dua *state* khusus yang ditambahkan di *state transisiton diagram*. Pertama adalah *start state*. Masing – masing diagram harus mempunya satu dan hanya satu *start- state* ketika *object* mulai dibuat.*State* khusus berikutnya ada *stop state.*

2.7.5 State Transition Details

Sebuah *State transition* dapat mempunya sebuat *action* dan / atau sebuah kondisi penjaga (*guard condition*) yang terasosiasi dengannya, dan mungkin juga memunculkan sebuah *even*t. Sebuah *actio*n adalah kelakuan yang terjadi ketika *state transition* terjadi. Sebuah *even*t adalah pesan yang dikirm ke *object* lain di sistem. Kondisi penjaga adalah ekspersi *boolean* dari nilai atribut – atribut yang mengijinkan sebuah *state transition* hanya jika kondisinya benar. Kedua *action* dari pengaja adalah kelakuan dari object dan secara tipikal menjadi oeprasi. Seringkali operasi – operasi ini adalah tersendiri- hal itu , merka digunkana hanya oleh *object* dirinya sendiri.

2.7.6 State Details

Action – action yang mengiringi seluruh *state transition* ke sebuah state mungkin ditempatkan sebagai sebuah *entry action* dalam *state.* Demikian juga, *action – action* yang mengiringi sebluruh *state transition* keluar dari sebuah *state* mungkin ditempatkan sebagai sebuah aksi keluar dalam state. Kelakuan yang terjadi dalam *state* disebut *activity.* Sebuah *activity* memulai ketika *state* dimasukan dan salah satu dari melengkapi atau diinterupsi oleh sebuah *state transition* yang keluar.

2.8 Activity Diagram

Activity Diagram memodelkan *work flow* proses bisnis dan urutan akitfitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status.

Elemen – elemen activity diagram :

1. *Status start*(Mulai) dan *end* (akhir)
2. Aktifitas yang merepresentasikan sebuah langkah dalam *workflow*
3. *Transition* menunjukan terjadinya perubahan status aktifitas *(Transition show what sate follows another)*
4. Keputusan yang menunjukan alternatif dalam *workflow*..
5. *Synchronization bars* yang meunjukan subflow parallel. *Syncronization bars* dapat digunakan untuk menunjukan *concurent thread* pada *workflow* proses bisnis

6. *Swimlanes* yang merepresentasikan *role* bisnis yang bertanggung jawab pada aktifitas yang berjalan.

2.9 SISTEM INFORMASI GEOGRAFIS

Informasi permukaan bumi telah berabad-abad disajikan dalam bentuk peta. Peta yang mulai dibuat dari kulit hewan, sampai peta yang dibuat dari kertas, semuanya menyajikan data geografis dalam bentuk gambar-gambar maupun coretan-coretan. Peta-peta umum (*general purpose)* menggambarkan topografi suatu daerah ataupun bats-batas (administrative) suatu wilayah atau Negara. Sedangkan peta-peta tematik (*thematic)* secara khusus menampilkan distribusi keruangan (*spasial distribution)* kenampakan-kenampakan seperti geologi, goemorfologi, tanah, vegetasi, atau sumber daya alam.

Apa yang tersaji pada sebuah peta, tidak lain adalah data atau informasi tentang permukaan bumi. Namun demikian, suatu peta juga dapat menggambarkan distribusi social ekonomi suatu masyarakat, seperti misalnya peta desa tertinggal, peta kependudukan, dan sebagainya. Dengan demikian, dapat dikatakan bahwa peta memuat atau mengandung data yang mengacu bumi (*geo-referenced data).* Yang diacu tidak lainadalah posisinya yaitu system koordinat bumi, baik yang menggunakan system bujur/lintang, atau system UTM (*Universal Transfer Mercator).*

Era komputerisasi telah membuka wawasan dan pardigma baru dalam proses pengambilan keputusan dan penyebaran informasi. Data yang mempresentasikan

“dunia byata” dapat disimpan dan diproses sedemikian rupa ehingga dapat disajikan dalam bentuk yang lebih sederhana dan sesuai dengan kebutuhan.

Teknologi komputer yang mampu menangani basis data (*database)* dan menampilkan suatu gambar(grafik), merupakan salah satu alternative yang dipilih untuk menyajikan suatu peta. Walaupun demikian, system informasi geografis tidak boleh hanya dipandang sebagai pemindahan peta konvensional (tradisional) ke benuk peta digital. Sebab dengan kemampuannya memanipulasi data, computer dengan system informasi goegrafisnya dapat menghasilkan suatu informasi berharga yang lain yang diperoleh dari hsil analisis yang diprogramkan padanya.

Sejak pertengahan 1970-an, telah dikembangkan system-sitem yang secara khusus dibuat untuk menangani masalah-masalah informasi yang bereferensi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini mencakup :

1. Pengorganisasian data dan informasi.
2. Menempatkan informasi pada lokasi tertentu.
3. Melakukan komputasi, memberikan ilustrasi keterhubungan satu sama lainnya (koneksi) beserta analisa-analisa spasial lainnya.

Sebutan umum untuk system yang menangani masalah-masalah di atas adalah Sistem Informasi Geografis (SIG). Dalam beberapa literatur SIG dipandang sebagai hasil dari kombinasi antara system computer untuk bidang kartografi atau Sistem komputer untuk bidang perancangan dengan teknologi basis data.

Kebutuhan data keruangan dan analisisnya tidak hanya dibutuhkan oleh ilmuwan kebumian saja. Para perencana kekotan juga memerlukan infoemasi tentang distribusi penggunaan lahan. Insinyur sipil juga membutuhkan informasi permukaan bumi untuk merencanakan dan membangun jalan, jembatanh, atau bendungan. Pemerintah daerah dalam membuat rencana pengembangan wilayah dan menentukan kebijakannya juga memerlukan informasi keruangan seperti misalnya distribusi perumahan, daerah industri, daerah perdagangan, sarana transportasi, fasilitas umum, dan sebagainya.

Dari pembahasan singkat di ats, dapatlah ditunjukkan bahwa informasi yang diperoleh dari bumi (*real world)* tersebut, diambil, diolah, dan akhirnya dihasilkan informasi yang berguna bagi para pengguna. Gambar di bawah ini menunjukkan proses perencanaan, yaitu proses informasi yang dimulai dan berakhir pada bumi (*real world).*

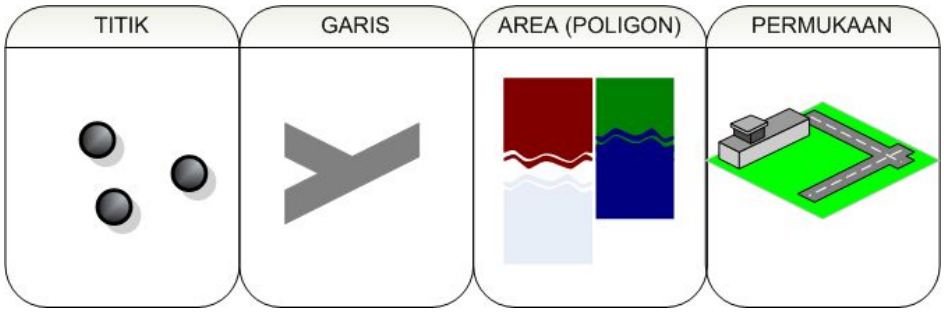
2.9.1 DEFINISI SISTEM INFORMASI GEOGRAFIS

Sebagian besar defiisi yang iberikan di dalam berbagai pustaka masih bersifat umum, belum lengkap, tidak presisi dan bersifat elastik hingga seringkali agak sulit untuk membedakan yang dengan system-sistem informasi yang masih “serumpun” (sebagai contoh, seperti yang telah disinggung di atas adalah sistem informasi karografi, yang berbasiskan komputer dan sistem-sistem komputer untuk bidang perancangan yang telah dilengkapi dengan berbagai kemampuan analisis spasial).

Sistem informasi geografis sebenarnya akronim dari :

1. Sistem adalah kumpulan elemen-elemen yang saling integrasi dalam lingkungan yang dinamis untuk mecapai tujuan tertentu.
2. Informasi berasal dari pengolahansejumlah data.
3. Geografi digunakan karena berdasarkan pada geografsi atau spasial. Objek ini mengarah pada spesifikasi lokasi dalam suatu IspaceI yang ditampilkan ke dalam suatu peta.

Data Spasial berupa titik, garis, poligon (2-D), permukaan (3-D).



Gambar2.1 Data spasial (sumber Purwadhi, 1997)

Format Titik

a. Koordinat tunggal

b. Tanpa panjang

c. Tanpa luasan

Contoh:

a. lokasi kecelakaan

b. Letak pohon

Format Garis

a. koordinat titik awal dan akhir

b. mempunyai panjang tanpa luasan

Contoh:

a. jalan, sungai

b. utility

Format Poligon

koordinat dengan titik awal dan akhir sama mempunyai panjang dan luasan

Contoh:

a. tanah persil

b. bangunan

Format Permukaan

a. area dengan koordinat vertical

b. area dengan ketinggian

Contoh:

a. peta slope

b. bangunan bertingkat

Melihat akronim di atas dapat diambil kesimpulan bahwa Sistem Informasi Geografis adalah sistem komputer yang digunakan untuk memasukkan, menyimpan, mmeriksa, mengintegrasikan, memanipulasi, menganalisa, dan menampilakn data yang berhubungan dengan posisi-posisi permukaan bumi (tampilan geografinya).

Sistem informasi geografis diharapkan mampu memberikan kemudahan-kemudahan yang diinginkan, seperti :

1. Penanganan data geospasial menjadi lebih baik dalam format baku.
2. Revisi dan peutakhiran data menjadi lebih mudah.

3. Data geospasial dan informasi menjadi lebih mudah dicari, dianalisa, dan direpresentasikan.
4. Menjadi produk yang mempunyai nilai tambah.
5. Kemampuan menukar data geospasial.
6. Penghematan waktu dan biaya.
7. Keputusan yang diambil menjadi lebih baik.

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

Berikut ada dua keistimewaan analisa melalui SIG, yaitu :

1. Analisa Proximity

   Merupakan suatu analisa geografi yang berbasis pada jarak antar layer. Sistem informasi geografis menggunakan proses yang disebut *buffering* (membangun lapisan pendukung struktur layer dalam jarak tertentu untuk menentukan dekatnhya 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 1 layer yang akan ditumpang susun secara fisik agar bias dianalisa secara visual.

Definisi SIG selalu berkembang, bertambah, dan bervariasi. Hal ini terliohat dari banyakanya definisi yang telah beredar. Selain itu, SIG juga merupakan suatu bidangilmu dan teknologi yang relative baru, digunakan oleh berbagai bidang disiplin ilmu dan berkembang dengan cepat. Berikut ini

merupakan sebagian kecil dari definisi-definisi SIG yang telah beredar diberbagai pustaka.

1. Definisi GIS (Rhind, 1988):

   GIS is a computer system for collecting, checking, integrating and information related to the surface of the earth.

2. Definisi GIS yang dianggap lebih memadai (Marble & Peuquet, 1983) and (Parker, 1988; Ozemoy *et al.*, 1981; Burrough, 1986):

   GIS deals with space-time data and often but not necessarily, employs computer

   hardware and software.

3. Purwadhi, 1994:

   a) 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.

   b) SIG merupakan manajemen data spasial dan non-spasial yang berbasis komputer dengan tiga karakteristik dasar, yaitu: (i) mempunyai fenomena aktual (variabel data non-lokasi) yang berhubungan dengan topik permasalahan di lokasi bersangkutan; (ii) merupakan suatu kejadian di suatu lokasi; dan (iii) mempunyai dimensi waktu.

2.9.2 SUBSISTEM SISTEM INFORMASI GEOGRAFIS

Dari definisi-definisi di atas, maka sistem informasi geografis dapat diuraikan menjadi beberapa subsistem berikut :

a. Data input : subsistem ini bertugas utnuk mengumpulkan dan mempersiapkan data spasial dan atribut dari sumber. Subsistem ini pula yang bertanggung jawab dalam mengkonversi atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan oleh SIG.
b. Data output : subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti table, grafik, peta, dan lain-lain.
c. Data management : subsistem ini mengorganisasikan baik data spasial maupun ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil,di*update,*dan d*iedit.*
d. Data manipulation dan analisis : subsistem ini menetukan informasi-informasi yang dapat dihasilkan oleh SIG. selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

2.9.3 SUMBER DATA SISTEM INFORMASI GEOGRAFIS

Sistem Informasi Geografis memerlukan data masukan agar dapat berfungsi dan memberikan informaasi lain hasil analisisnya. Data masukan tersebut diperoleh dari tiga sumber, yaitu : (a) lapangan, (b) peta, (c) citra penginderaan jauh.

a) **Data lapangan.** Data ini diperoleh langsung dari pengukuran lapangan secara langsung, seperti misalnya pH tanah, salinitas air, curah hujan suatu wilayah, dan sebagainya.

b) **Data peta.** Informasi yang telah terekam pada kertas atau film, dikonversikan ke dalam bentuk digital. Misalnya, peta geologi, peta tanah, dan sebagainya. Apabila data sudah terekam dalam bentuk peta, tidak lagi diperlukan data lapangan, kecuali untuk pengecekan kebenarannya.

c) **Data penginderaan jauh.** Citra penginderaan jauh yang berupa foto udara atau radar dapat diinterpresentasi terlebih dahulu sebelum dikonversi ke dalam bentuk didgital. Sedangkan citra yang diperoleh dari satelit yang sudah dalam bentuk digital dapat langsung digunakan setelah diadakan koreksi seperlunya.

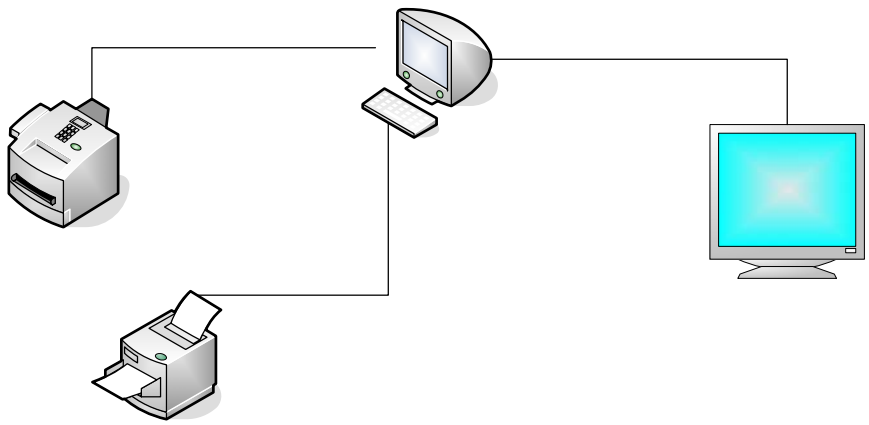
Ketiga sumber tersebut saling mendukung satu sama lain. Data lapangan dapat digunakan unuk membuat peta fisis, sedangkan data penginderaan jauh juga memerlukan data lapangan untuk lebih memastikan kebenaran data terbeut. Jadi ketiga sumber ini saling berkaitan, melengkapi, dan mendukung, sehingga tidak boleh ada yang diabaikan.

2.9.4 KOMPONEN SISTEM INFORMASI GEOGRAFIS

SIG merupakan sistem yang kompleks, yang biasanya terintegrasi dengan lingkungan sistem-sistem computer yang lain di tingkat fungsional dan jaringan SIG terdiri dari beberapa komponen berikut :

1. Perangkat keras

Pada saat ini Sistem Informasi Geografis tersedia untuk berbagai *platform* perangkat keras mulai dari PC *desktop, workstation,* hingga *multiuser host* yang dapat digunakan oleh banyak orang secara bersamaan dalam jaringan komputer yang luas, berkemampuan tinggi, memiliki ruang penyimpanan (*harddisk*) yang besar dan mempunyai kapasitas memori (RAM) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat secara ketat terhadap karakteristik-karakteristik fisik keras ini sehingga keterbatasan memori pada PC pun dapat diatasi. Adapun perangkat kerasyang sering digunakan untuk SIG adalah komputer (PC, *mouse, printer, plotter, dan scanner*.

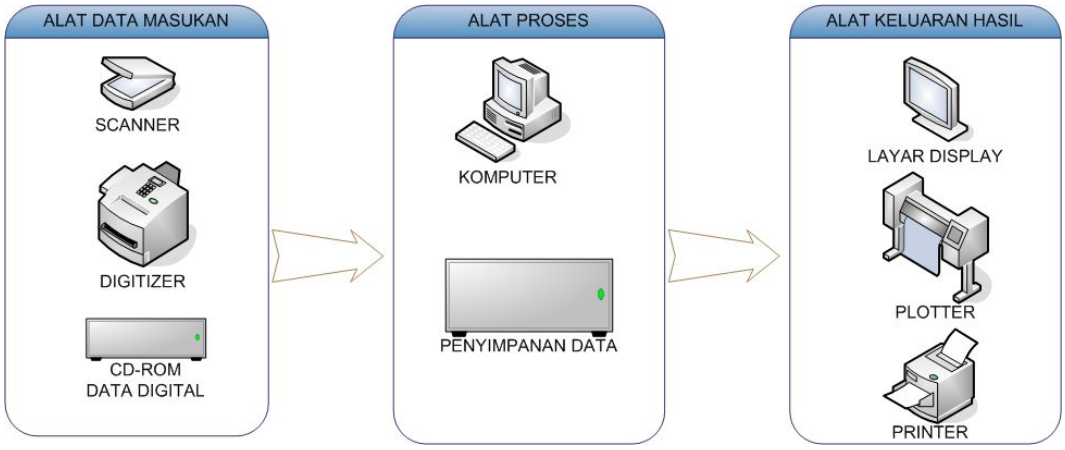


Gambar2.2 General Hardware Setup for a Microcomputer-based GIS

2. Perangkat lunak

Bila dipandang dari sisi lain, SIG juga merupakan perangkat lunak yang tersusun secara modular dimana basis data memegang peranan kunci. Setiap subsistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri

dari beberapa pemodul, hingga tidak mengherankan bila ada perangkat SIG yang terdiri dari ratusan modul program (*.exe) yang masing-masing dieksekusi sendiri.



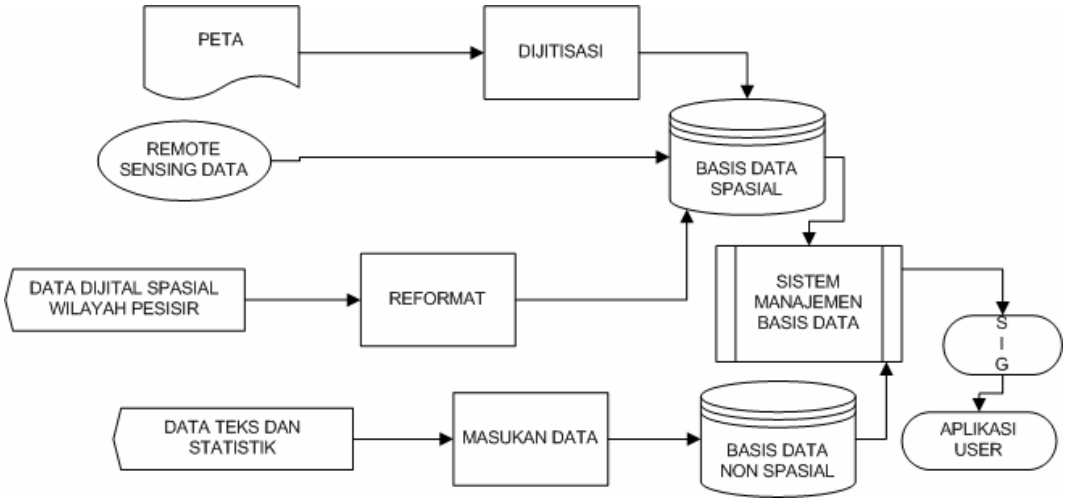
Gambar2.3 Komponen perangkat keras GIS

3. Data dan informasi Geografis.

SIG dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara tidak langsung dengan meng*importnya* dari perangkat-perangkat lunak SIG yang lain maupun secara langsung mendigitasi data spasialnya dari peta dan memasukkan data atributnya dari table-tabel dan laporan dengan menggunakan *keyboard*.

Syarat pengorganisasian data:

Volum kecil dengan klasifikasi data yang baik; Penyajian yang akurat; Mudah dan cepat dalam pencarian kembali (data retrieval) dan penggabungan (proses komposit).



Gambar2.4 Pengorganisasian data

Type Data :

a. Data lokasi:

Koordinat lokasi

Nama lokasi

Lokasi topologi (letak relatif:sebelah kiri danau A,sebelah kanan pertokoan B)

b. Data non-lokasi:

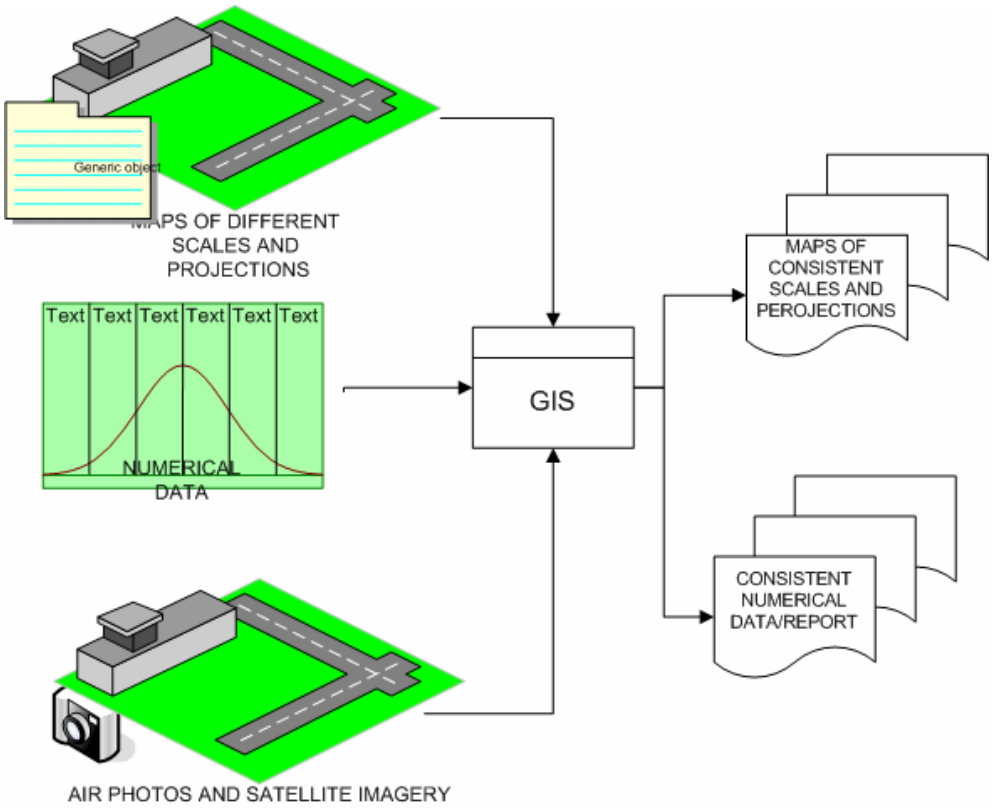
Curah hujan

Jumlah panen padi

Terdiri dari variabel (tanah), kelas (alluvial), nilai luas (10 ha), jenis (pasir)

c. Data dimensi waktu (temporal):

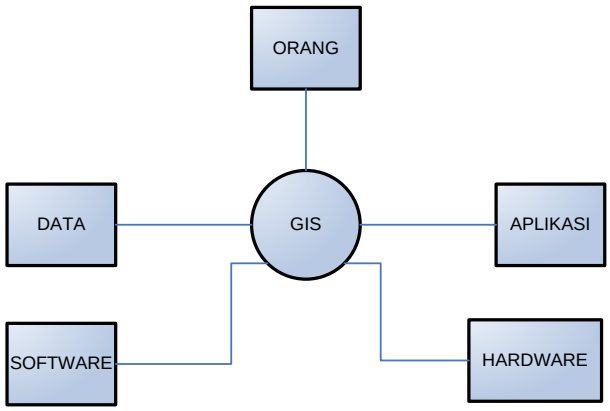
Data non-lokasi di lokasi bersangkutan dapat berubah dengan waktu (misal: data curah hujan bulan Desember akan berbeda dengan bulan Juli)



Gambar2.5 Masukan dan Keluaran Basis Data SIG

4. Manajemen.

Suatu proyek SIG akan berhasil jika di-*manage* dengan baik dan dikerjakan oleh orang-orang yang memiliki keahlian yang tepat pada semua tingkatan.



Gambar2.6 Komponen SIG (John E. Harmon, Steven J. Anderson. 2003)

2.9.5 APLIKASI-APLIKASI SISTEM INFORMASI GEOGRAFIS

Berikut adalah contoh-contoh aplikasi SIG :

a) Aplikasi SIG dalam bidang Sumber Daya Alam (inventarisasi, manajemen, dan kesesuaian lahan untuk pertanian, perkebunan, kehutanan, perencanaan tata guna lahan, analisis daerah rawan bencana alam, dan sebagainya).
b) Aplikasi SIG bidang **perencanaan** (perencanaan pemukiman transmigrasi, perencanaan tataruang wilayah, perencanaan kota, perencanaan lokasi, dan relokasi industri, pasar, pemukiman, dan sebagainya).
c) Aplikasi SIG bidang **kependudukan atau demografi** (penyusunan data pohon, penyediaan informasi kependudukan atau sensus dan social ekonomi, sistem informasi untuk pemilihan umum, dan sebagainya).
d) Aplikasi bidang **lingkungan** berikut pemantauannya (perencanaan sungai, danau, laut, evaluasi pengendapan Lumpur atau sediment baik di sector danau, sungai, atau pantai, pemodelan pencemaran udara, limbah berbahaya, dan sebagainya).
e) Aplikasi bidang **pertanahan** (manajemen pertanahan, sistem informasi pertanahan, dan sejenisnya).

2.9.6 KEMAMPUAN SISTEM INFORMASI GEOGRAFIS.

1) Memetakan letak.

Data realita di permukaan bumi akan dipetakan ke dalam beberapa *layer (lapisan)* dengan setiap *layer* 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 basis data, untuk kemudian dilihat letaknya dalam keseluruhan peta. Kemampuan ini memungkinkan seseorang untuk mencari di mana 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 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 jumalh, seperti dimana yang paling banyak atau dari yang paling sedikit. Dengan melihat penyebaran kuantitas tersebut, dapat menacri tempat-tempat yang sesuai dengan criteria 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 statistic dibanding basis data biasa.

3) Memetakan kerapatan (*Densities*).

Sewaktu orang melihat konsentrasi dari penyebaran lokasi dari *feature-feature* diwilayah 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 ke dalam

unit-unit yang lebih mudah untuk dipahami dan beragam. Pemetaan kerapatan sangat berguna untuk data-data yang berjumlah besar seperti sensus atau data statistic daerah.

4) Memetakan perubahan.

Dengan memasukkan variable-variabel waktu, SIG dapat dibuat untuk peta historikal. Histori ini dapat digunakan untuk memprediksi keadaan yang akan datang dan dapat pula digunakan untuk evaluasi kebijakan.

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 di luar area.

2.9.7 FASILITAS SISTEM INFORMASI GEOGRAFIS.

Pada sebuah aplikasi SIG, terdapat beberapa fasilitas yang merupakan standar untuk melengkapi peta ynag tampil di layer monitor antara lain ;

a. Legend

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

b. Skala

Skala adalah keterangan perbandingan ukuran di layar dengan ukuran sebenarnya.

1) Zoom in : peta di layar dapat diperbesar dengan zoom in dan dapat diperkecil dengan zoom out.
2) Pan : dengan fasilitas pan peta dapat digeser-geser untuk melihat daerah yang dikehendaki.
3) *Searching* : fasilitas ini digunakan untuk mencari di mana letak suatu *feature*. Bias dilakukan dengan meng-*input*-kan nama atau keterangan dari *feature* tersebut.
4) Pengukuran : fasilitas ini dapat mengukur jarak antara titik, jarak rute, atau luas suatu wilayah secara interaktif.
5) Informasi : setiap *feature* dilengkapi dengan informasi yang dapat dilihat jika *feature* tersebut di-*klik*.

2.9.8 MANFAAT SISTEM INFORMASI GEOGRAFIS

Dengan SIG memudahkan dalam melihat fenomena kebumian dengan perspektif yang lebih baik.SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. Dengan tersedianya komputer dengan kecepatan dan kapasitas ruang penyimpanan besar seperti saat ini, SIG akan mampu memproses data dengan cepat dan akurat dan menampilkannya. SIG juga mengakomodasi dinamika data, pemutakhiran data yang akan menjadi lebih mudah. Ambil contoh seperti kondisi Aceh yang

beberapa waktu lalu dihempas tsunami yang mengakibatkan korban dan kerugian yang banyak sekali. Bayangkan dengan citra satelit yang beresolusi tinggi dapat dilihat kondisi suatu lokasi dipermukaan bumi secara akurat. Kemudian hasil survey rekan-rekan yang bekerja dilapangan dapat langsung di masukkan dalam database spasial yang telah ada sebelumnya untuk mengetahui lokasi rawan dan butuh segera ditangani. Informasi tersebut kemudian bisa di *upload* ke internet dan tersebarlah informasi ke penjuru dunia. Gambar dibaw ah ini merupakan salah satu citra Ikonos yang merekam sebelum dan sesudah kejadian tsunami di Aceh. Integrasi data citra satelit seperti yang tampak pada gambar tersebut dengan data-data yang lain menggunakan SIG akan menghasilkan informasi baru yang benar-benar sangat membantu sekali. Seperti saat ini dalam proses pemulihan kembali Aceh, rekan-rekan yang berkarya dibidang SIG sedang melakukan penyusunan data spasial kembali



Gambar2.7 Contoh data SIG

2.9.9 KARAKTERISTIK SISTEM INFORMASI GEOGRAFIS

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 system dan decision support system 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.

2.10 PETA

2.10.1 PENGERTIAN PETA.

Pemetaan adalah proses menyajikan informasi muka bumi yang berupa fakta (dunia nyata, baik bentuk permukaan bumi, maupun sumber daya alam lainnya).

Peta merupakan sarana informasi yang menggambarkan dunia nyata (permukaan bumi) dalam bentuk penyajian grafis melalui smbol-simbol.

2.10.2 JENIS PETA.

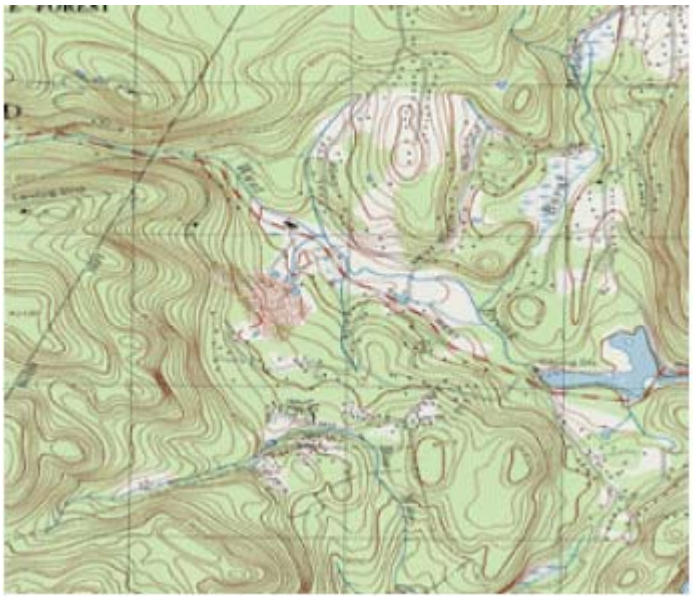
Peta dijeniskan berdasarkan :

a) Berdasarkan jenisnya, terdiri dari :

1) Peta Hidrologi ; informasi tentang kedalaman dan keadaan dasar laut serta informasi lainnya yang diperlukan untuk navigsi pelayaran.
2) Peta Geologi : informasi tentang geologi suatu daerah.
3) Peta Kadaster : memuat informasi tentang kepemilikan tanah.
4) Peta Irigasi : informasi tentang jalur irigasi pada suatu daerah.
5) Peta Jalan : informasi tentang jalur jalan pada suatu daerah.
6) Peta Kota : informasi jaringan transportasi, *drainage*, sarana kota, dan lain-lain.
7) Peta Teknis : memuat informasi umum tentang keadaan permukaan bumi yang mencakup kawasan tidak luas.
8) Peta Topografi : memuat informasi umum tentang keadaan permukaan bumi beserta infromasi ketinggiannya menggunakan garis kontur.

Berasal dari bahasa yunani, *topos* yang berarti tempat dan *graphi* yang berarti menggambar. Peta topografi memetakan tempat-tempat dipermukaan bumi yang berketinggian sama dari permukaan laut menjadi bentuk garis-garis kontur, dengan satu garis kontur mewakili satu ketinggian.

Walaupun peta topografi memetakan tiap interval ketinggian tertentu, namun disertakan pula berbagai keterangan pula yang akan membantu untuk mengetahui secara ljauh mengenai daerah permukaan bumi yang terpetakan tersebut keterangan-keterangan itu disebut *legenda peta.*



Gambar2.8 Contoh peta topografi

Legenda peta antara lain berisi tentang :

a. Judul Peta

Judul peta ada dibagian tengah atas. judul peta menyatakan lokasi yang ditunjukkan oleh peta yang bersangkutan, sehingga lokasi yang berbeda akan mempunyai judul yang berbeda pula

b. Nomor Peta

Nomor peta biasanya dicantumkan diselah kanan atas peta. Selain sebagai nomor regisrtasi dari badan pembuat, nomor peta juga berguna sebagai petunjuk jika kita memerlukan peta daerah lain disekitar suatu daerah yang terpetakan. Biasanya di bagian bawah disertakan pula lembar derajat yang mencantumkan nomor-nomor peta yang ada disekeliling peta tersebut.

c. Koordinat Peta

Koordinat adalah kedudukan suatu titik pada peta. Koordinat ditentukan dengan menggunakan sistem sumbu, yaitu garis-garis yang saling berpotongan tegak lurus. Sistem koordinat yang resmi dipakai ada dua, yaitu :

1. Koordinat Geografis

Sumbu yang digunakan adalah garis bujur (bujur barat dan bujur timur) yang tegak lurus terhadap katulistiwa, dan garis lintang (lintang utara dan lintang selatan) yang sejajar dengan katulistiwa. Koodinat geografis dinyatakan dalam satuan derajat, menit, dan detik.

2. Koordinat Grid

Dalam koordinat grid, kedudukan suatu titik dinyatakan dalam ukuran jarak terhadap suatu titik acuan. Untuk wilayah Indonesia, titik acuan nol terdapat disebelah barat Jakarta (60 derajat LU, 68 derajat BT). Garis vertikal diberi nomor urut dari selatan ke utara, sedangkan garis horizontal diberi nomor urut dari barat ke timur.

Sistem koordinat mengenal penomoran dengan 6 angka, 8 angka dan 10 angka. Untuk daerah yang luas dipakai penomoran 6 angka, untuk daerah yang lebih sempit digunakan penomoran 8 angka dan 10 angka (biasanya 10 angka dihasilkan oleh GPS).

d. Kontur

Kontur adalah garis khayal yang menghubungkan titik-titik yang berketinggian sama dari permukaan laut, sifat-sifat garis kontur adalah :

1. Satu garis kontur mewakili satu ketinggian tertentu.

2. Garis kontur berharga lebih rendah mengelilingi garis kontur yang lebih tinggi.

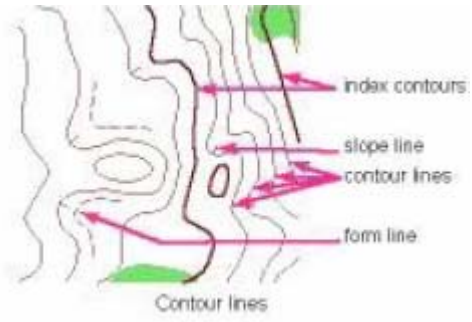
3. Garis kontur tidak berpotongan dan tidak bercabang.

4. Interval kontur biasanya 1/2000 kali skala peta.

5. Rangkaian garis kontur yang rapat menandakan permukaan bumi yang curam/terjal, sebaliknya yang renggang menandakan permukaan bumi yang landai.

6. Rangkaian garis kontur yang berbentuk huruf "U" menandakan punggungan gunung.

7. Rangkaian garis kontur yang berbentuk huruf "V" terbalik menandakan suatu lembah/jurang.



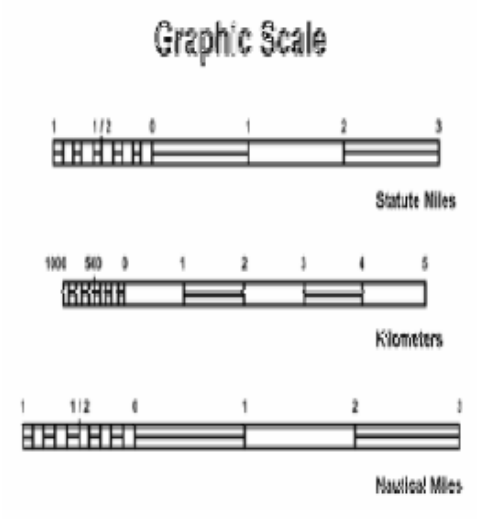
Gambar2.9 Garis kontur

e. Skala Peta

Skala peta adalah perbandingan antara jarak pada peta dengan jarak horizontal di lapangan. Ada dua macam cara penulisan skala, yaitu :

1. Skala angka, contoh : 1:25.000 berarti 1 cm jarak dipeta = 25.000 cm (250 m) jarak horizontal di medan sebenarnya.

2. Skala garis, contoh: berarti tiap bagian sepanjang blok garis mewakili 1 km jarak horizontal.

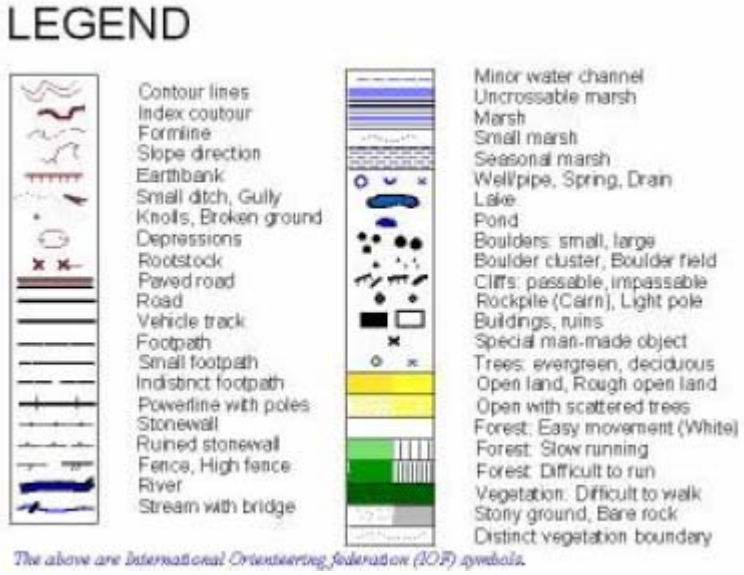


Gambar2.10 Skala angka peta

f. Legenda Peta

Legenda peta biasanya disertakan pada bagian bawah peta. Legenda ini memuat simbol-simbol yang dipakai pada peta tersebut, yang penting diketahui : triangulasi, jalan setapak, jalan raya, sungai, pemukiman, ladang, sawah, hutan dan lainnya. Di Indonesia, peta yang umumnya digunakan adalah peta keluaran Direktorat Geologi Bandung, kemudian peta dari Jawatan Topologi, atau yang sering disebut peta AMS (American Map Service) dibuat oleh Amerika dan rata-rata dikeluarkan pada tahun 1960. Peta AMS biasanya berskala 1:50.000 dengan interval kontur (jarak antar kontur) 25 m. Selain itu ada peta keluaran Bakosurtanal (Badan Koordinasi Survey dan Pemetaan Nasional) yang lebih baru, dengan skala 1:50.000 atau 1:25.000 (dengan interval kontur 12,5m).

Peta keluaran Bakosurtanal biasanya berwarna.



Gambar2.11 Legenda peta

g. Tahun Peta

Peta topografi juga memuat keterangan tentang tahun pembuatan peta tersebut, semakin baru tahun pembuatannya, maka data yang disajikan semakin akurat.

h. Arah Peta

Yang perlu diperhatikan adalah arah Utara Peta. Cara paling mudah adalah dengan memperhatikan arah huruf-huruf tulisan yang ada pada peta. Arah atas tulisan adalah Arah Utara Peta.Pada bagian bawah peta biasanya juga terdapat petunjuk arah utara yaitu

1. Utara sebenarnya/True North : yaitu utara yang mengarah pada kutub utara bumi.

2. Utara Magnetis/Magnetic North : yaitu utara yang ditunjuk oleh jarum magnetis kompas, dan letaknya tidak tepat di kutub utara bumi.

3. Utara Peta/Map North : yaitu arah utara yang terdapat pada peta.

b) Berdasarkan skala.

1) Peta Skala Besar : skala peta 1:100.000 atau lebih.
2) Peta Skala Sedang : skala peta 1:10.000 – 1:100.000.
3) Peta Skala kecil : kuarng dari 1:10.000.

Skala menunjukkan ketelitian dan kelengkapan informasi yang tersaji. Semakin besar skalanya maka semakin teliti dan semakin lengkap peta yang disajikan.

c) Berdasarkan Penurunan dan Penggunaan.

1) Peta Dasar : digunakan untuk membuat peta turunan dan perencanaan umum maupun pengembangan suatu wilayah.
2) Peta Tematik : digunakan berdasarkan peta dasar dan memuat peta-peta tertentu.

2.11 Borland Delphi 7.0

Delphi merupakan salah satu bahasa pemrograman tingkat tinggi berbasis Windows. Delphi digolongkan ke dalam bahasa pemrograman visual yang menitik beratkan pada pemrograman berorientasi objek (*object oriented programming*). Membuat program menggunakan Delphi tidaklah terlalu sulit, karena bahasa pemrograman ini dikembangkan menggunakan bahasa pascal. Bagi yang sudah terbiasa dengan bahasa pemrograman pascal, akan merasa lebih mudah dalam mempelajari Delphi. Karena bentuk penulisan *listing* programnya tidak memiliki perbedaan yang terlalu jauh. Dalam perkembangannya, Delphi

memiliki beberapa objek-objek yang canggih, dalam bentuk kontrol program yang dikelompokkan ke dalam *toolbox* yang biasa disebut dengan *component palette*.

Apabila dibandingkan dengan bahasa pemrograman lain yang berbasis visual, Delphi memiliki kelebihan yang terletak pada ketersediaan berbagai macam kontrol program yang lebih banyak dan lebih canggih. Walaupun sedikit mirip dengan kontrol program yang dimiliki Visual Basic, Delphi memiliki keunggulan dalam penulisan *listing* program yang lenih canggih dan serba otomatis.

Menggunakan bahasa pemrograman Delphi kita dapat menghasilkan program-program yang canggih, dimulai dari pemrograman multimedia, grafis sampai dengan pemrograman basis data yang menggunakan jaringan. Bahkan dengan Delphi kita juga bisa membuat program yang dapat mengakses data dari internet.

1. Operasi-operasi database :
    a) Membuat file database
    b) Memberi index
    c) Meng-entri data
    d) Meng-edit data
    e) Meng-delete data
    f) Men-sortir data
    g) Men-cari data
    h) Mem-filter data
    i) Me-relasi-kan tabel

j) Membuat report

2. Jenis aplikasi database :

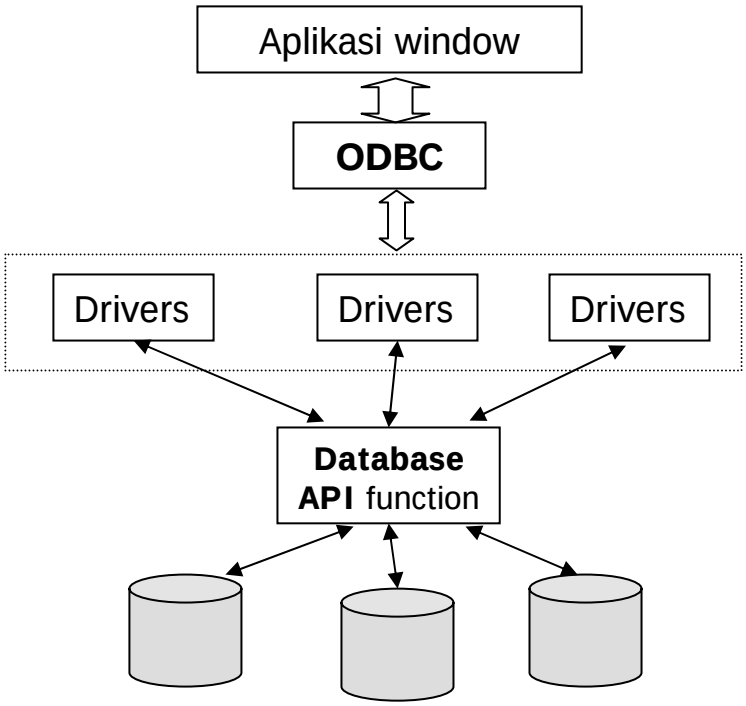
a) Stand Alone,

b) Client/server

c) Distributed

3. Jenis file database :

Dbf, Paradox(db), Access (mdb), Interbase, MySQL, SQL Server,Sybase, Oracle, dan lain-lain.



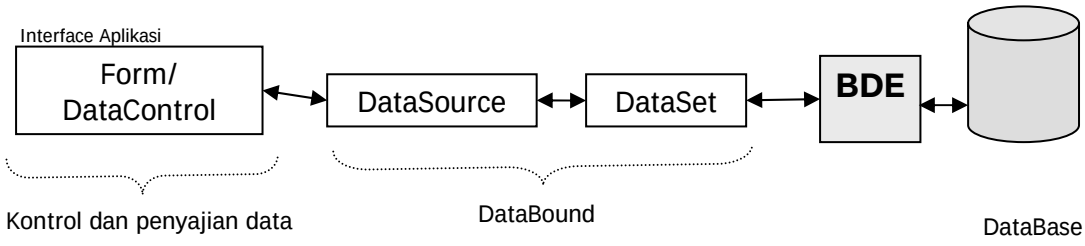
Gambar2.12 File database dengan jenis yang berbeda

Komponen Aplikasi database DELPHI

1. Delphi memiliki sistem (Engine) sendiri untuk mengelola sistem Database yaitu : BDE (Borland DataBase Engine), sehingga setiap komputer yang

menjalankan aplikasi database Delphi umumnya membutuhkan keberadaan BDE pada sistemnya. Tetapi Delphi tetap memberi kesempatan jika kita ingin membuat aplikasi database tanpa melibatkan BDE.

2. Selain BDE, Delphi juga memiliki beberapa komponen khusus yang harus digunakan untuk membuat aplikasi database Delphi. Komponen database tersebut ada yang berfungsi sebagai pengait file database ke aplikasi (*Data bound*) dan ada yang berfungsi untuk meng-akses dan menyajikan data tersebut
3. Yang termasuk sebagai pengait data (Data bound) yaitu : Table, Query, DataSource,…
4. Yang termasuk sebagai pengakses dan penyaji data yaitu : DBGrid, DBEdit, DBLabel, DBComboBox,DBRadioGroup,DBImage, DBNavigator, …
5. Komponen-komponen database Delphi umumnya terletak di tab DataAccess dan Data Controls pada Componen Palette
6. Hubungan antara : file database, BDE, komponen dan Form dapat diamati melalui bagan berikut :



Gambar2.13 Hubungan antara : file database, BDE, komponen dan Form

2.12 MapObjects 2.2

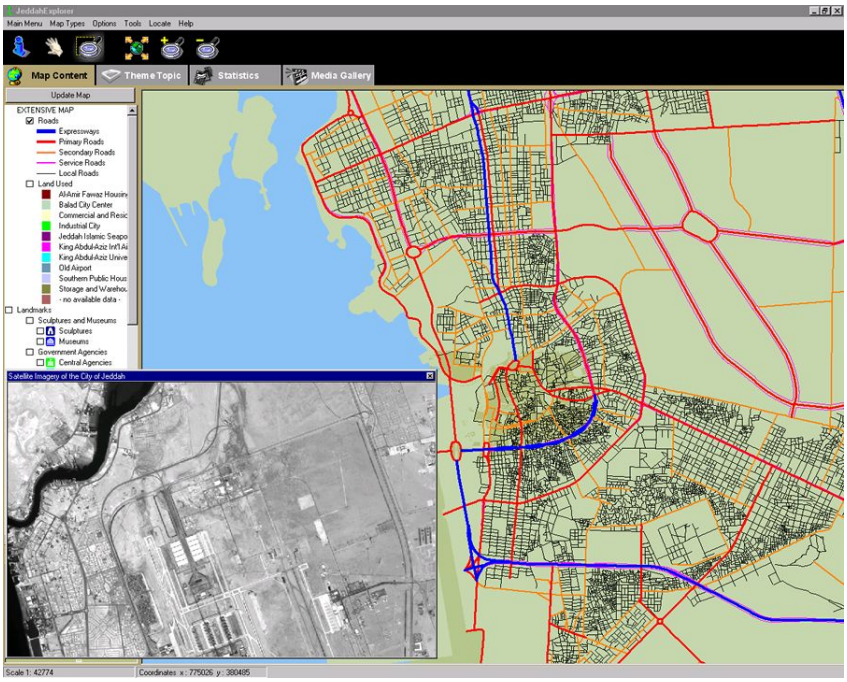
MapObject adalah sebuah software canggih yang mempunyai komponen pendukung Sistem Informasi Geografis dan juga pemetaan. Para penggunanya dapat menggunakan Map Objects untuk membuat aplikasi yang berkaitan dengan pemetaan yang dinamis dan juga kemampuan aplikasi berbasis Sistem Informasi Geografis. Map Objects menyediakan semua keperluan yang dibutuhkan untuk solusi yang terperinci dalam membuat aplikasi pemetaan dan Sistem Informasi Geografis.

Dengan Map Objects kita dapat :

1. menambah komponen pemetaan untuk menambah apliaksi yang telah ada.
2. membangun aplikasi untuk menampilkan data tetapi tetap ringan.
3. membuat pemetaan dan program Sistem Informasi Geografis yang memenuhi tugas-tugas yang spesifik dan berbagai keperluannya.
4. mengembangklan aplikasi basis query yang sederhana yang memudahkan dalam kemampuan akses data yang dibuat dengan solusi Sistem Informasi Geografis yang berpengalaman.

FITUR-FITUR PENTING MAP OBJECTS

1. **On-the-Fly-projection :** pengguna dapat mengkombinasikan data dari berbagai project ke dalam sebuah project yang umum untuk menampilkan dan analisis. Dan juga, berbagai layer peta dapat di*export* ke dalam sebuah proyeksi. Untuk mengoptimasikan kemampuan kita dalam membuat berbagai jenis proyeksi di dalam jangka wakti yang minim, MapObjects sekarang dilengkapi dengan ESRI ArcInfo proyeksi komponen .
2. **Kemampuan pendukung data yang tidak parallel** : MapObjects menawarkan kemampuan untuk mendukung bagi berbagai jenis sumber data seperti :
    - **a)** Format Sistem Informasi Geografis yang standar ArcInfo, ESRI Shapefile, ESRI GRID.
    - **b)** Design pertolongan untuk komputer : format (CAD) (DGN, DXF, dan DWG), file CAD, dan file-file AutoCAD 2000 DWG.
    - **c)** Akses ke database eksternal melalui ActiveX Data Objects (ADO), Data Access Objects (DAO), dan Open Database Connectivity (ODBC).
    - **d)** Berbagai katalog gambar, bahkan berbagai jenis format gambar seperti : GeoTIFF, TIFF, JPEG, GIF, ERDAS, and MrSID
    - **e)** ESRI ArcView StreetMap untuk geocoding
    - **f)** Database-database ArcSDE (ESRI's Spatial Database Engine)



Gambar.2.14 Kemampuan pendukung data yang tidak parallel

3. Kemajuan dalam penaganan data

   MapObjects menyediakan filter spasial dan atribut yang lengkap untuk mengoptimalkan aplikasinya. Sebagai contoh pilihan, pengguna dapat mengakses aplikasi ArcSDE dengan tepat dari aplikasi MapObjects. MapObjects juga mendukung aplikasi gambar transparan, tampilan, dan juga rotasi dari layer data vector dan raster

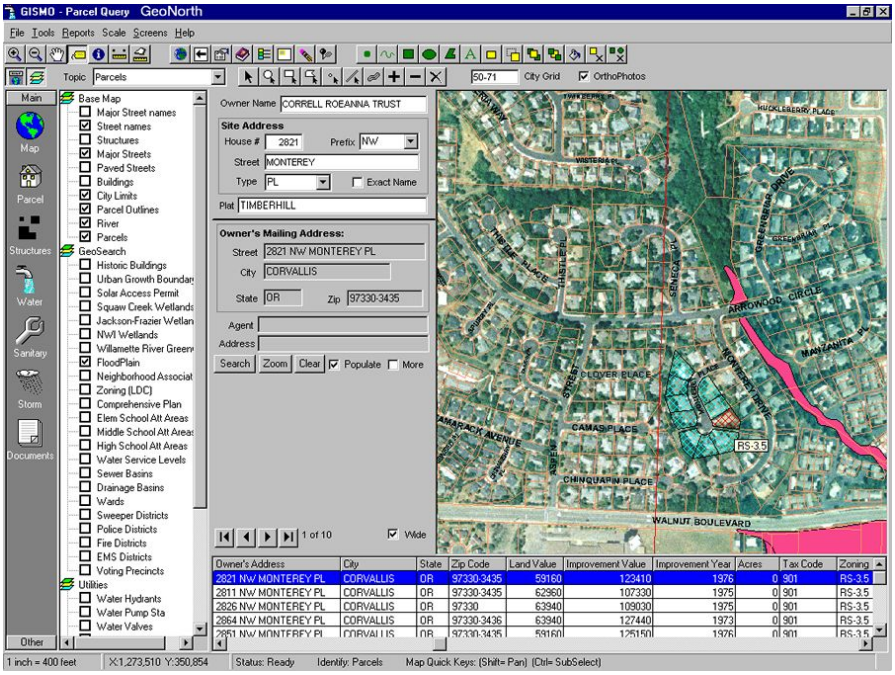
4. Kemampuan **State-of-the-art geocoding**

   MapObjects memberi kemampuan pada pengguna untuk dapat melakukan pencocokan alamat yang cepat dan akurat termasuk pengalamatan internasional dan proses penolakan. Dengan kemampuan *StreetMap*, pengguna dapat membaca, menampilkan, dan *geocoding* sebuah alamat jalan dari database *StreetMap* yang dikompresi.

Dengan catatan : untuk membaca database *StreetMap* yang dikompresi, pengguna harus mempunyai ijin.

5. Mendukung **Web-based mapping**

   Pengguna dapat menempatkan peta dinamik, peta yang sesuai di internet menggunakan teknologi ArcIMS.



Gambar 2.15 teknologi ArcIMS

6. Diperlengkapi manajemen GPS.

   MapObjects mendukung pencarian titik secara dinamik, garis, polygon, lingkaran, dan elips, mempermudah untuk mengolah aktifitas GPS.

7. Fungsi-fungsi Geometrik.

   MapObjects includes a robust spatial geometry library for unions, intersections, and buffers. MapObjects termasuk sebuah penyimpanan geometri spasial untuk *union, intersection*.

8. Manfaat **Run-time deployment**

   Saat pengguna mulai dapat membangun aplikasi, penyebaran menjadi faktor yang genting. Perlengkapan persebaran *run-time* MapObjects membantu pengguna untuk menyalurkan aplikasinya dengan mudah dan efisien.

9. **Kemampuan penerjemah**

   MapObjects mendukung kemampuan menerjemah untuk layer-layer ArcSDE, penerjemah ini dapat berkoneksi kepada semua versi ArcSDE dan pengguna dapat menampilkannya. Bahkan, pengguna dapat mengidentifikasi dan memilih versi-versi dasar pada nama versinya.

10. Pengontrol yang sangat membantu.

    MapObjects mempunyai sebuah legenda dan pengendali ukuran skala, termasuk *source code*, didesain untuk membuatnya lebih pengguna bagi pengguna untuk mengembangkan aplikasinya. Fitur ini didasarkan pada control yang sama yang digunakan pada aplikasi ArcExplorer, data *browser* ESRI GIS yang disalurkan bebas biaya. MapObjects mengandung koleksi dari keperluan yang sama digunakan pada ArcExplorer,.

11. Lebih diperlengkapi dengan pemetaan dan SIG perlengkapan komponennya

    Komponen ini membuat pengguna dapat menampilkan, mengubah dengan *query*, dan menganalisis data peta dinamik.

12. Mengubah fitur menggunakan metode tematik

    Pengguna dapat menambah nilai peta, memecah kelas-kelas, menambah symbol, peta kepadatan dengan symbol titik, dan grafik pada aplikasinya. MapObjects sekarang juga mendukung kemampuan, symbol grafik yang umum digunakan.

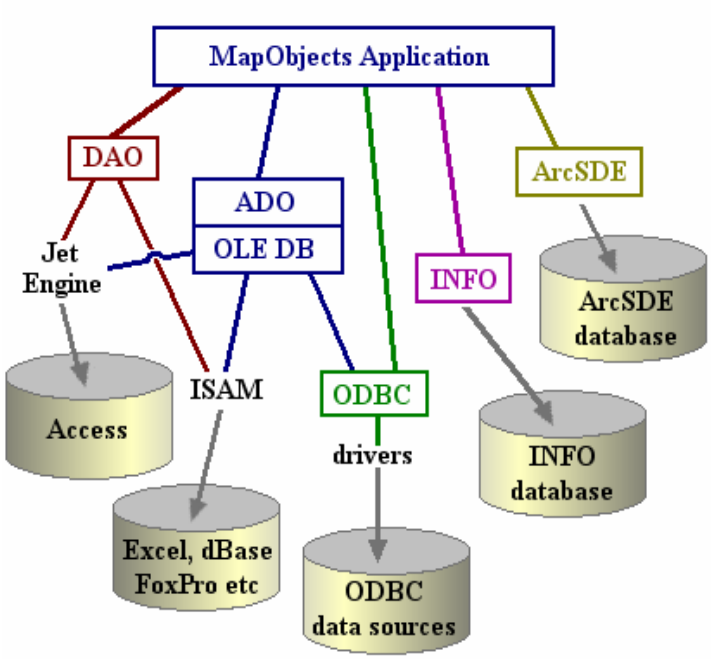
13. Pemilihan fitur atribut dan *query*

    Query-query dapat dilakukan menggunakan ekspresi SQL yang standar.

MapObjects mendukung aplikasi yang dibuat dengan cara mengakses data dari berbagai data base dan sumber data yang lain. Data dikembangkan dalam objek tabel MapObjects, yang kemudian dapat digunakan dalam berbagai cara. Pengunaan umum dari sebuah table yang baru adalah untuk menggabungkan data tersebut dengan fitur geografis, dengan menggunakan sbuah relasi. Table tersebut juga digunakan dalam *geocoding*, aksi *rendering* dan area-area fungsi dari MapObjects.

Data dapat diakses melalui arsitektur sumber data *Microsoft's OLE DB* menggunakan data objek *Active X*, atau Microsoft Jet database engine databases (.MDB), menggunakan Data Access Objects (DAO) 3.5. MapObejcts juga mendukung database-database ArcSDE, ODBC-based, database ISAM, dan

database INFO. Diagram di bawah ini rute secara konseptual digunakan untuk mengakses data oleh table MapObjects.



Gambar2.16 pengaksesan data dari berbagai macam databse oleh MapObjects

2.13 Shapefile ESRI

Format data Shapefile ESRI menjelaskan geometrid an atribut dari referensi fitur-fitur di dalam beberapa file dengan spesifik ekstensi file yang disimpan dalam folder yang sama dalam media penyimpanan. File-file ini terdiri dari sebuah induk utama file, dan sebuah table dBASE. Induk utama file adalah sebuah pengaksesan secara langsung, file panjang *record variabel* dimana setiap *record* menjelaskan sebuah bentuk dengan sebuah daftar dari tepi-tepi bentuk tersebut. Dalam file index, setiap record mengandung *offset* dari record file utama yang berhubungan dari awal hingga file utama. Table dBase mengandung atribut-

atribut fitur dengan 1 record setiap fitur. Hubungan satu ke satu antara geometrid an atribut didasarkan pada nomor record. Record-record atributnya dalam file dBase disimpan dalam aturan yang sama dengan record-record pada file utama. File utama, file index, dan file dBase mempunyai prefik dan sufik yang sama.

Shp : file yang menyimpan fitur-fitur geometri

Shx :file yang menyimpan index dari fitur geometri

Dbf : file dBase yang menyimpan informasi atribut dari fitur-fitur geometri.

Operator-operator tertentu melaksanakan pada Shapefile dalam pembuatan file lain dengan prefik yang sama dengan komponene Shapefile. Sebagai contoh termasuk *geocoding*, dimana hasil metode BuidIndices dalam sebuah file baru dengan sufik, '.gcd' dan indeks spasial dimana metode BuildIndex membuat 2 file baru, dengan ekstensi '.sbn' dan '.sbx'. file-file ini dibuat dalam lokasi yang sama dengan Shapefile tapi tidak penting dan dapat diabaikan dengan aman saat meng-*copy* data. Bagaimanapun, sangat disarankan untuk menjaga file ekstra ini untuk mencegah pembuatan kembali file-file tersebut. Pada umumnya, aplikasi-aplikasi seharusnya tidak menganggap keberadaan dari file-file ini dan bila penting dapat membuat file-file ini dengan sendirinya.

Spesifikasi Shapefile mendukung tipe-tipe dari bentuk-bentuk seperti : titik, *Polylines, polygon, multipoints*, dan *multipatches*. Bahkan, spesifikasi mendukung bagi bentuk-bentuk koordinat 3 dimensi dan penyimpanan dari nilai-nilai yang terukut. Sebagai gantinya, saat membaca sebuah tipe bentuk *multipatch* dari sebuah *Geodataset*, MapOnjects akan mengkonversinya ke dalam sebuah

objek *polygon* dengan setiap dari bagian-bagiannya mewakili sebuah bagian dari bentuk *multipatch*. Hasil objek *polygon* bereaksi seperti objek *polygon* yang lainnya dalam MapObject tapi tidak dapat diubah.

2.14 PENGUJIAN DAN IMPLEMENTASI

2.14.1 Pengujian.

Ada dua macam pengujian perangkat lunak yang dapat dilakukan, yaitu metode *white box* dan *black box*. *White box testing* dilakukan di awal program, sedangkan *black box testing* baru dilakukan pada tahap berikutnya. Pengujian dilakukan untuk melakukan penilaian tentang keberhasilan program dan menetukan apakah program sudah sesuai dengan keinginan *user* atau belum.

White Box Testing

White Box Testing yaitu metode desain *test case* yang menggunakan struktur kontrol desain procedural untuk memperoleh *test case*. *Test case* dapat diperoleh dengan :

a. Menjamin bahwa semua *independent path* pada suatu modul telah digunakan minimal satu kali.
b. Menggunakan semua keputusan logis pada sisi *true* dan *false*.
c. Mengeksekusi semua loop dalam batasannya dan pada batas operasionalnya.
d. Menggunakan struktur data internal untuk menjamin validitasnya

Pengujian ini harus memenuhi kriteria sebagai berikut :

a) Mengurangi pelaksanaan *test case* untuk mencapai hasil pengujian yang dingiinkan.

b) *Test case* akan menunjukkan ketidaksesuaian (ketidaksinkronan) beberapa kesalahan, tetapi kurang menunjukkan detil kesalahan.

Software yang dibangun tidak selalu sempurna. Terkadang ada kesalahan yang baru akan diketahui saat pengujian atau bahkan saat implementasi. Berikut ini kriteria *software* yang ”cacat” :

a. Kesalahan logika dan asumsi yang salah berbanding terbalik dengan probabilitas jalur program yang akan dieksekusi. Kecenderungan kesalahan ini terjadi pada`tahap desain dan implementasi fungsi, kondisi atau kontrol yang berada di luar pikiran. Kesalahan ini terjadi jika pemrosesan yang khusus cenderung diabaikan.
b. Selalu ada keyakinan bahwa *logical path* tidak akan dieksekusi pada basis regular. Kesalahan ini terjadi karena adanya kesalahan asumsi tentang aliran data control.
c. Kesalahan tipologis yang merupakan kesalahan yang acak atau random. Perpindahan dari bahasa satu ke bahasa pemrograman lain menyebabkan timbulnya kesalahan sintak.

Path testing adalah metode pengujian yang memungkinkan desainer *test case* mengukur kompleksitas logika desain prosedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur ekseskusi. Metode ini menjamin bahwa setiap statement akan dilalui minimal sekali dalam proses pengujian. Tujuannya adalah meyakinkan bahwa himpunan *test case* akan menguji setiap *path* pada suatu program paling sedikit satu kali. Titik awal *path testing* adalah suatu program *flow graph* yang menunjukkan node-node yang

menyatakan keputusan program, misalnya : kondisi if-then-else dan busur yang menyatakan alur kontrol. Statement yang memiliki kondisi ada pada node-node dalam *flow graph*. *Path testing* menggambarkan alur kontrol di mana setiap cabang ditunjukkan oleh *path* yang terpisah sedangkan loop ditunjukkan oleh arrows looping dan kembali ke loop kondisi node.

Black Box Testing

Black Box testing merupakan pendekatan pengujian dimana program dianggap sebagai suatu ’*black box*’ (’kotak hitam’). Program *test case* berbasiskan spesifikasi *test planning* dapat dimulai sejak awal proses pengembangan sistem.

Black box testing berfokus pada persyaratan fungsional perangkat lunak. Metode ini memungkinkan *software developer* untuk mendapatkan serangkaian kondisi input yang mempergunakan semua persyaratan fungsional program. *Black box testing* bukan alternatif *whit box testing*, namun merupakan pelengkap kesalahan jika dibandingkan metode *white box testing.*

Pengujian *Black box* berusaha menemukan kesalahan :

a) Fungsi-fungsi yang tidak benar atau hilang.
b) Kesalahan *interface*.
c) Kesalahan dalam struktur data atau akses database eksternal.
d) Kesalahan kinerja.
e) Inisialisasi dan kesalahan terminasi.

Black box testing cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *Black box* memperhatikan struktur kontrol, sehingga perhatian

berfokus pada domain informasi. *Black box testing* dirancang untuk dapat menjawab pertanyaan berikut :

a. Bagaimana validitas fungsional diuji ?
b. Kelas input apa yang akan membuat *test case* menjaid baik?
c. Apakah sistem sensitive terhadap nilai input tertentu?
d. Bagaimana batasan daru suatu data diisolasi?
e. Bagaimana kecepatan dan volume data yang dapat ditolerir oleh sistem ?
f. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

2.14.2 Implementasi Sistem

Menurut WHI

"Konstruksi sistem adalah pengembangan, instalasi dan pengujian komponen-komponen sistem"

Penekannya terletak pada blok pembangunan sistem informasi dari perspektif pembangunan sistem. Pemicu fase kontruksi sistem adalah persetujuan spesifikasi dengan desain fisik yang merupakan hasil dari fase desain. Dengan komponen sistem untuk desain tersebut. Akhirnya kita akan membangun sistem fungsional. Sistem fungsional dapat diimplementasikan atau dikirim sebagai sistem operasional.

Tujuan fase kontruksi adalah untuk membangun dan menguji sebuah sistem fungsional yuang memenuhi persyaratan bisnis dan desain dan untuk mengimplementasikan *interface* antara sistem baru dengan sistem produksi yang telah ada.pemrograman biasanya dikenal sebagai aspek utama dari fase kontruksi.

Akan tetapi kecenderungan solusi sistem yang melibatkan perolehan dan pembelian paket perangkat lunak, maka implementasi dan integrasi komponen-komponen perangkat lunak menjadi suatu aspek visibel yang berimbang dan umum pada fase kontruksi.

Setiap perubahan sistem akan memerlukan transisi dari sistem lama ke sistem yang baru. Implementasi akan membantu mengatasi masalah *start up* yang terjadi dari membawa sistem produksi kepada operasi.

Hasil dari fase implementasi adalah sistem operasional yang akan masuk ke tahap *operation and support* dari siklus hidupnya. Fase ini mempertimbangkan blok pembangun yang sama seperti waktu konstruksi. Tugas fase implementasi adalah sebagai berikut :

a) Menguji sistem. Setelah perangkat lunak dan program diinstall dan diuji maka harus diadakan tes final terhadap sistem. Analisis sistem memfasilitasi penyelesaian tugas ini dan mengkomunikasikan berbagai hal dan masalah dengan anggota tim proyek. Pemilik dan pengguna sistem memiliki wewenang apakah sistem berjalan baik atau tidak. Pembangun sistem memberikan spesialisasi. Sistem tes mungkin akan menghasilkan modifikasi yang diperlukan oleh program. Iterasi ini akan dilanjutkan sampai pengujian sistem dianggap berhasil.
b) Menyiapkan rencana konversi. Dengan menggunakan spesifikasi desain untuk sistem baru, analisis membuat rencana detil konversi. Rencana ini mengidentifikasi database yang harus diinstal, pelatihan pengguna akhir dan mengkonversi sistem lama ke sistem yang baru.

c) Menginstall database. Pembangun sistem memegang peran uatama dalam aktivitas ini. Programmer aplikasi akan membuat program aplikasi untuk mengekstrak data dari database yang sudah ada dan mempopulasikan database baru.

d) Melatih para pengguna. Sistem diperuntukkan bagi pengguna. Mereka harus dilatih untuk mengurangi kemungkinan terjadinya *human error*.

e) Beralih ke sistem baru. Pada konversi dari sistem lama ke sistem yang baru, manajer proyek mengawasi tahap tersebut. Pemilik sistem memberikan umpan b alik berkenaan dengan sistem baru. Analisis, desain, dan pembangun sistem akan menilai umpan balik tentang penggunaan aktual dri sistem tersebut. Input kunci dari aktivitas ini adalah rencana konversi yang telah dibuat pada fase impementasi sebelumnya. Hasil utamanya adalah sistem operasional yang diletakkan dalam produksi dalam bisnis tersebut.