

BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Untuk membuat tampilan yang menarik memang tidak mudah dilakukan. Seorang perancang tampilan selain harus mempunyai jiwa seni yang memadai, ia juga harus mengerti selera pengguna secara umum. Hal lain yang perlu disadari oleh seorang perancang tampilan adalah bahwa ia harus bisa meyakinkan pemrogramnya bahwa apa yang ia bayangkan dapat diwujudkan dengan peranti bantu yang tersedia (Insap Santoso ; 2009 : 185).

Perancangan merupakan proses pengolahan hasil analisis perangkat lunak menjadi rencana pengembangan perangkat lunak dan batasan-batasan perangkat lunak atau masalah yang mungkin dihadapi dalam pengembangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan arsitektur, perancangan modul, dan perancangan antarmuka. (Y. Yohakim Marwanta ; 2010 : 5).

Bagi perancang antarmuka, hal yang sangat penting untuk ia perhatikan adalah mendokumentasikan semua pekerjaan yang dilakukan. Dokumentasi rancangan dapat dikerjakan atau dilakukan dengan beberapa cara :

1. Membuat sketsa pada kertas
2. Menggunakan peranti purwarupa GUI
3. Menuliskan keterangan yang menjelaskan tentang kaitan antara jendela.
4. Menggunakan peranti bantu CASE (*Computer Aided Software Engineering*).

Cara kedua dan keempat tidak selalu dapat diterapkan, karena peranti tersebut biasanya harus dibeli dan seringkali cukup mahal. Cara ini kebanyakan diterapkan pada pembuatan antarmuka grafis untuk suatu jenis pekerjaan berskala besar.

II.1.1. Cara Pendekatan

Sebuah program aplikasi pastilah ditujukan kepada pengguna, yang utama, bukan perancangan program aplikasi tersebut. Program aplikasi pada dasarnya dapat dikelompokkan dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula dan program aplikasi yang akan digunakan oleh pengguna umum, yang juga sering dikenal dengan sebutan public software. Karena perbedaan pada calon pengguna, maka perancang program antarmuka perlu memperhatikan hal ini (Insap Santoso ; 2009 : 186).

Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi yang serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna maupun ragam antarmuka yang akan digunakan. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan, yakni pendekatan yang disebut dengan pendekatan perancangan berpusat ke pengguna (*user centered design approach*). Cara pendekatan ini berbeda pendekatan perancangan oleh pengguna (*user design approach*).

Pendekatan perancangan berpusat ke pengguna adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna di sini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancangan antarmuka sedang menggambar wajah antarmuka yang mereka inginkan. Dengan kata lain, perancangan dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya. Sementara perancangan menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna, seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan (Insap Santoso ; 2009 : 187).

Pada perancangan oleh pengguna, pengguna sendirilah yang merancang wajah antarmuka yang diinginkan. Di satu sisi, cara ini akan mempercepat proses pengimplementasian modul antarmuka. Tetapi di sisi yang lain, hal ini justru sangat memberatkan pemrogram karena apa yang diinginkan pengguna belum tentu dapat diimplementasikan dengan mudah, atau bahkan tidak dapat dikerjakan dengan menggunakan peranti bantu yang ada.

Perancang program aplikasi yang dimasukkan dalam kelompok kedua, atau *public software*, perlu menganggap bahwa program aplikasi tersebut akan digunakan oleh pengguna dengan berbagai tingkat kepandaian dan karakteristik yang sangat beragam. Di satu sisi keadaan ini dapat ia gunakan untuk memaksa pengguna menggunakan antarmuka yang ia buat, tetapi pada sisi lain pemaksaan itu akan berakibat bahwa program aplikasinya menjadi tidak banyak

penggunanya. Satu kunci penting dalam pembuatan modul antarmuka untuk program-program aplikasi pada kelompok ini adalah dengan melakukan customization. Dengan customization pengguna dapat menggunakan program aplikasi dengan wajah antarmuka yang sesuai dengan selera masing-masing pengguna.

Salah satu contoh dari adanya kemampuan yang dimiliki oleh sebuah program aplikasi atau sistem operasi yang dapat disesuaikan dengan karakteristik pengguna adalah pengaturan desktop pada OS X versi 10.5 favoritnya, sehingga pengguna dapat mengubahnya sesuai keinginan justru akan membuat mata pengguna itu sakit, dikarenakan mata harus melakukan akomodasi maksimum terus menerus untuk menyesuaikan dengan warna tampilan yang ada.

Selain cara pendekatan yang dijelaskan di atas, Anda yang terbiasa menulis program-program aplikasi mungkin mempunyai cara khusus untuk berhadapan dengan pengguna. Tetapi perlu Anda ingat bahwa apapun cara yang Anda gunakan, Anda tetap harus mempunyai pedoman bahwa pada akhirnya program itu bukan untuk Anda sendiri, tetapi akan digunakan oleh orang lain. Dengan kata lain, jangan pernah mengabaikan pendapat (calon) pengguna program aplikasi Anda. (Insap Santoso ; 2009 : 188).

II.1.2. Prinsip Dan Petunjuk Perancangan

Antarmuka pengguna secara alamiah terbagi menjadi empat komponen model pengguna, bahasa perintah, umpan balik, dan penampilan informasi. Model pengguna merupakan dasar dari tiga komponen yang lain (Insap Santoso ; 2009 : 188).

Model mental pengguna merupakan model konseptual yang dimiliki oleh pengguna ketika ia menggunakan sebuah sistem atau program aplikasi. Model ini memungkinkan seorang pengguna untuk mengembangkan pemahaman mendasar tentang bagian yang dikerjakan oleh program, bahkan oleh pengguna yang sama sekali tidak mengetahui teknologi komputer. Dengan pertolongan model itu pengguna dapat mengantisipasi pengaruh suatu tindakan yang dilakukan dan dapat memilih strategi yang cocok untuk mengoperasikan program tersebut. Model pengguna dapat berupa suatu simulasi tentang keadaan yang sebenarnya dalam dunia nyata, sehingga ia tidak perlu mengembangkannya sendiri dari awal.

Setelah pengguna mengetahui dan memahami model yang diinginkan, dia memerlukan peranti untuk memanipulasi model itu. Peranti pemanipulasian model ini sering disebut dengan bahasa perintah (command language), yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasionalkan. (Insap Santoso ; 2009 : 189).

Komponen ketiga adalah umpan balik. Umpan balik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa bentuk umpan balik terutama ditujukan kepada pengguna yang belum berpengalaman dalam menjalankan program sebuah aplikasi. Umpan balik dapat digunakan untuk

member keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut.

Komponen keempat adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini perancang harus menampilkan pesan-pesan tersebut seefektif mungkin sehingga mudah dipahami oleh pengguna. Setelah memahami beberapa prinsip dalam perancangan antarmuka pengguna. Pada bagian berikut ini akan diberikan petunjuk singkat tentang perancangan antarmuka yang akan Anda lakukan sebagai seorang perancang tampilan.

II.1.3. Urutan Perancangan

Perancangan dialog, seperti halnya perancangan sistem yang lain, harus dikerjakan secara atas ke bawah. Proses perancangannya dapat dikerjakan secara bertahap sampai rancangan yang diinginkan terbentuk, yaitu sebagai berikut (Insap Santoso ; 2009 : 190).

1. Pemilihan ragam dialog

Untuk suatu tugas tertentu, pilihlah ragam dialog yang menurut perkiraan cocok untuk tugas tersebut. Ragam dialog dapat dipilih dari sejumlah ragam dialog yang telah dijelaskan pada bab-bab sebelumnya. Pemilihan ragam dialog dipengaruhi oleh karakteristik populasi pengguna, tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut. Ragam dialog yang terpilih dapat berupa sebuah ragam tunggal, atau sekumpulan ragam dialog yang satu sama lain saling mendukung.

2. Perancangan Struktur Dialog

Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna sebaiknya banyak dilibatkan, sehingga pengguna langsung mendapatkan umpan balik dari diskusi yang terjadi. Pada tahap ini suatu purwarupa dialog seringkali dibuat untuk memberik gambaran yang lebih jelas kepada calon pengguna.

3. Perancangan format pesan

Pada tahap ini tata letak tampilan dan keterangan tekstual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengefektifkan pengguna tombol.

II.2. Sistem

Kata sistem mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok, yaitu : (Kusrini, dkk ; 2010 : 5).

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang dalam hal ini sistem itu didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul

bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.

2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi di dalam sistem. Prosedur (procedure) didefinisikan oleh Richard F. Neushl sebagai urutan operasi kerja (tulis menulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian di dalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri-sendiri, tetapi sebaliknya, saling berhubungan hingga membentuk satu kesatuan sehingga tujuan sistem itu dapat tercapai.

II.2.1. Karakter Sistem

Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu, antara lain : (Kusrini, dkk ; 2010 : 6).

1. Komponen sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi membentuk suatu komponen sistem atau bagian-bagian dari sistem.

2. Batasan sistem (*boundary*)

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. Subsistem

Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

4. Lingkungan Luar Sistem (*Environment*)

Suatu sistem yang ada di luar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*interface*)

Media penghubung antara suatu subsistem dengan subsistem lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

6. Masukan Sistem (*input*)

Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energy yang dimasukkan supaya sistem tersebut dapat berinteraksi.

7. Keluaran sistem (*output*)

Hasil energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

8. Pengolahan Sistem (*process*)

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

9. Sasaran Sistem (*object*)

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan. (Kusrini, dkk ; 2010 : 7).

II.2.2. Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan menjadi seperti berikut : (Kusrini, dkk ; 2010 : 7).

1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah suatu sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik adalah sistem yang ada secara fisik.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam sedangkan sistem buatan adalah sistem yang dirancang oleh manusia.

3. Sistem tertentu dan sistem tak tentu

Sistem tertentu adalah suatu sistem yang operasinya dapat diprediksi secara tepat sedangkan sistem tak tertentu adalah sistem dengan perilaku ke depan yang tidak dapat diprediksi.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak terpengaruh oleh lingkungan luar atau otomatis, sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh oleh lingkungan luar.

II.3. Pengertian Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan

informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar disbanding biaya untuk mendapatkannya. (Kusrini, dkk ; 2010 : 7).

II.4. Kualitas Informasi

Informasi yang berkualitas memiliki 2 kriteria, yaitu : (Kusrini, dkk ;2010:8).

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bisa ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya (*timelines*)

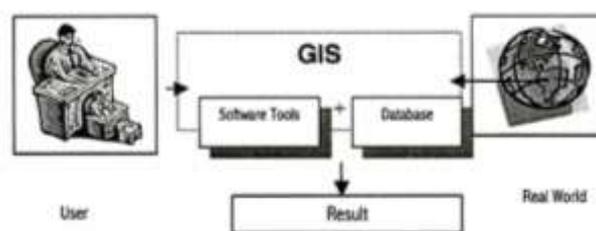
Informasi yang datang pada penerima tidak boleh terlambat. Di dalam pengambilan keputusan, informasi yang sudah usung tidak lagi bernilai. Bila informasi datang terlambat sehingag pengambilan keputusan terlambat dilakukan, hal ini dapat berakibat fatal bagi perusahaan.

3. Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. Di samping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (value of information) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih besar disbanding biaya untuk mendapatkannya.

II.5. Sistem Informasi Geografis (SIG)

Sistem informasi geografis (SIG) atau Geographic Information System (GIS) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis. Hal ini dikarenakan teknologi SIG banyak mendasarkan pada teknologi digital ini sebagai alat analisis.



Gambar II.1. Pola Keterkaitan GIS

(Sumber : Sistem Informasi Geografis Menggunakan ARC VIEW GIS, Eko Budiyanto, 2010)

Seperti tergambar dari namanya, SIG merupakan sebuah sistem yang saling berangkaian satu dengan yang lain. Akronim GIS terkadang dipakai sebagai istilah untuk geographical information science atau geospatial information studies yang merupakan ilmu studi atau pekerjaan yang berhubungan dengan Geographic Information System. Dalam artian sederhana sistem informasi geografis dapat disimpulkan sebagai gabungan kartografi, analisis statistic dan teknologi sistem basis data (*database*). (Edy Irwansyah ; 2013 : 1).

Pengertian sistem informasi geografis menurut beberapa ahli :

1. Burrough, 1986

Kumpulan alat yang *powerful* untuk mengumpulkan, menyimpan, menampilkan dan mentranformasikan data spasial dari dunia nyata (*real world*).

2. Aronoff, 1986

Segala jenis prosedur manual maupun berbasis komputer untuk menyimpan dan memanipulasi data bereferensi geografis.

3. ESRI, 2004

Sebuah sistem untuk mengatur, menganalisa dan menampilkan informasi geografis.

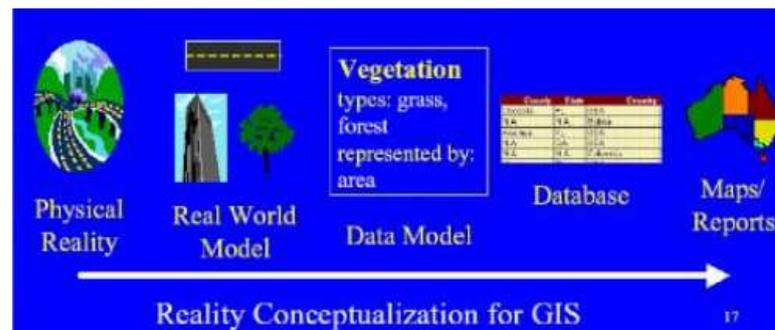
Sehingga dapat dirangkum konsep sebuah sistem informasi geografis adalah sebagai berikut :

1. Informasi geografis adalah informasi mengenai tempat dipermukaan bumi.
2. Teknologi informasi geografis meliputi Global Positioning System (GPS), remote sensing dan Sistem Informasi Geografis.
3. Sistem Informasi Geografis adalah sistem komputer dan piranti lunak (*software*).
4. Sistem Informasi Geografis digunakan untuk berbagai macam variasi aplikasi.
5. Sains Informasi Geografis merupakan ilmu sains yang melatarbelakangi teknologi sistem informasi geografis.

SIG tidak lepas dari data spasial, yang merupakan sebuah data yang mengacu pada posisi, obyek dan hubungan di antaranya dalam ruang bumi. Data spasial merupakan salah satu item dari informasi di mana di dalamnya terdapat informasi mengenai bumi termasuk permukaan bumi, di bawah permukaan bumi, perairan, kelautan dan bawah atmosfer. (Edy Irwansyah ; 2013 : 1).

II.6. Konsep *Real World*

Konsep *real world* merupakan sebuah cara bagaimana SIG mengubah realitas fisik sebuah dunia menggunakan model menjadi sebuah sistem informasi geografis yang dapat disimpan, dimanipulasi, diproses dan dipresentasikan.



Gambar II.2. Tampilan Konsep Real World

(Sumber : Sistem Informasi Geografis Prinsip Dasar dan Pengembangan Aplikasi , Edy Irwansyah ; 2013).

Konsep *Real World* memiliki beberapa tahapan sebagai berikut :

1. *Physical Reality*

Merupakan tahapan di mana menganalisa dunia nyata yang akan dibuat menjadi SIG.

2. *Real World Model*

Tahapan mengubah obyek-obyek yang ada di dunia nyata menjadi model.

3. *Data Model*

Tahapan yang mengubah model-model obyek dunia nyata menjadi sebuah tipe data.

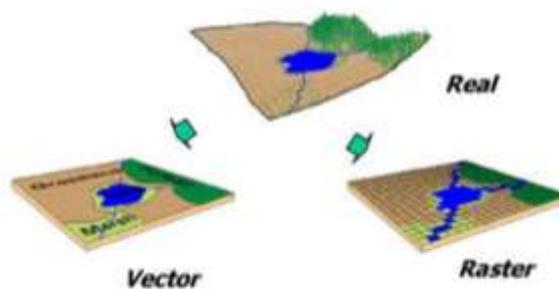
4. *Database*

Menyimpan keseluruhan data model ke dalam sistem basis data.

5. *Maps/Reports*

Merupakan hasil akhir dunia nyata yang telah dikonversi menjadi sebuah sistem informasi geografis.

SIG merepresentasikan real world dengan data spasial yang terbagi atas dua model data yaitu model dan raster dan model data vector. Keduanya memiliki karakteristik yang berbeda, selain itu dalam pemanfaatannya tergantung dari masukan data dan hasil akhir yang akan dihasilkan.

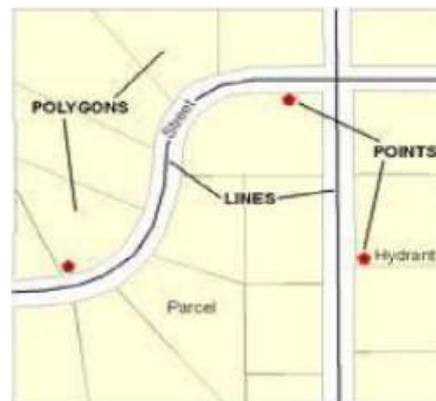


Gambar II.3. Tampilan data Raster dan data Vektor

(Sumber : Sistem Informasi Geografis Prinsip Dasar dan Pengembangan Aplikasi , Edy Irwansyah ; 2013).

1. Vektor

Dalam data *vector* bumi direpresentasikan sebagai suatu mosaik yang terdiri atas garis (*arcline*), polygon (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik/point (node yang mempunyai label), dan nodes (merupakan titik potongan antara dua buah garis).



Gambar II.4. Contoh data *vector*

(Sumber : Sistem Informasi Geografis Prinsip Dasar dan Pengembangan Aplikasi , Edy Irwansyah ; 2013).

Model data *vector* merupakan model data yang paling banyak digunakan, model ini berbasiskan pada titik (*points*) dengan nilai koordinat (x,y) untuk membangun obyek spasialnya. Obyek yang dibangun terbagi menjadi tiga bagian lagi yaitu berupa titik (*point*), garis(*line*), dan area (*polygon*).

a) Titik (*point*)

Titik merupakan representasi grafis yang paling sederhana pada suatu obyek titik tidak mempunyai dimensi tetapi dapat ditampilkan dalam bentuk symbol baik pada peta maupun dalam layar monitor. Contoh lokasi fasilitasi kesehatan, lokasi fasilitas kesehatan.

b) Garis (*line*)

Garis merupakan bentuk linear yang menghubungkan dua atau lebih titik dan merepresentasikan obyek dalam satu dimensi. Area (*polygon*)

Polygon merupakan representasi obyek dalam dua dimensi. Contoh Danau, Persil Tahan.

2. Data Raster

Data raster atau disebut juga dengan sel grid adalah data yang dihasilkan dari sistem penginderaan jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan pixel (picture element). Pada data raster, resolusi (definisi visual) tergantung pada ukuran *pixel*-nya.

Dengan kata lain, resolusi pixel menggambarkan ukuran sebenarnya di permukaan bumi yang diwakili oleh setiap pixel pada citra. Semakin kecil ukuran permukaan bumi yang direpresentasikan oleh satu sel, semakin tinggi resolusinya. Data raster sangat baik untuk merepresentasikan batas-batas yang berubah secara gradual, seperti jenis tanah, kelembaban tanah, vegetasi, suhu tanah dan sebagainya.



Gambar II.5. Contoh Data Raster

(Sumber : Sistem Informasi Geografis Prinsip Dasar dan Pengembangan Aplikasi, Edy Irwansyah ; 2013)

Keterbatasan utama dari data raster adalah besarnya ukuran file, semakin tinggi resolusi grid-nya semakin besar pula ukuran filenya. Keuntungan utama dari format data vector adalah ketepatan dalam merepresentasikan fitur titi, batasan dan garis lurus.

II.7. Sistem Pengelolaan Data Spasial

Pengelolaan data spasial merupakan hal yang penting dalam pengelolaan lingkungan. Pengelolaan yang tidak benar dapat menimbulkan berbagai dampak

yang merugikan. Bencana dalam skala besar dan kecil merupakan contoh dari system pengelolaan data spasial yang tidak terencana dan terorganisir dengan baik.

Banyak pihak terkait dengan masalah ini. Pengelolaan lahan selalu memanfaatkan berbagai data, baik data spasial terestris maupun data penginderaan jauh. Pengelolaan data banyak dilakukan oleh lembaga-lembaga seperti BAPPEDA dan lembaga swadaya masyarakat lainnya. Beberapa lembaga secara khusus mengelola data-data spasial untuk tujuan-tujuan tertentu, seperti bakosurtanal yang mengelola berbagai data spasial untuk tujuan evaluasi, survey, dan pemetaan.

Pengelolaan lingkungan banyak memanfaatkan berbagai teknologi baik dalam penyediaan, penyimpanan, pengolahan atau penyajian data. Pemanfaatan teknologi ini dimaksudkan untuk peningkatan akurasi dan efektivitas system pengelolaan itu sendiri. Teknologi yang banyak digunakan dalam hal ini adalah teknologi yang terkait dengan system informasi geografis.

II.8. Google Map

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *Google Maps* dalam web yang telah kita buat atau pada blog kita yang berbayar

maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript*. (Rusdianto ; 2011 : 1).

Cara membuat *Google Maps* untuk ditampilkan pada suatu web atau blog yaitu :

1. Hanya dengan membutuhkan pengetahuan mengenai **HTML** serta *JavaScript*,
2. Koneksi Internet yang sangat stabil.
3. Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. (Yusro; 2013 : 1).

Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia.

Pengertian *Google Maps* adalah layanan gratis peta dan pemetaan digital yang bisa dimanfaatkan untuk mengamati peta dunia melalui browser web. (Menurut istilah Google Help)

Google map atau *google earth* adalah sebuah fasilitas dari google untuk para pengguna internet, layanan dari google ini sendiri gratis alias tanpa ada pungutan biaya satu rupiah pun. Kecuali, biaya akses internet. Nah dengan adanya google map ini, kita bisa melihat seakan dunia itu sempit. Mungkin ada di antara anda yang sering bertualang ketempat-tempat yang belum pernah dikunjungi

sebelumnya, lalu anda tersesat karena tidak mengetahui rute jalan yang akan ditempuh, maka anda dapat menggunakan akses internet anda untuk melihat rute jalan melaalui aplikasi *Google Map*. Saktinya, dan dapat mengetahui rute jalan ditempat yang akan anda tuju, walaupun anda belum pernah secara langsung ketempat yang anda tuju.

Google Maps memanfaatkan *teknologi digital imaging*.seperti foto Satelit sehingga kamu bisa melihat bagaimana *landscape* planet bumi apabila dilihat dari luar angkasa.

Google map atau anda juga bisa menyebutnya *google earth* adalah layanan gratis yang hanya menunjukkan nama jalan, tidak bisa untuk nama gunung, nama sungai atau yang lainnya, namun *google map* ini lebih lengkap dari yang lainnya. Dengan aplikasi *zooming* maka anda dapat melihat lebih jelas wilayah yang belum anda kunjungi, yakni hanya menggunakan layanan *google map* maka dunia seakan menjadi sempit. Fasilitas yang disediakan *google* dalam layanan *google map* nya ini dapat dibilang hampir sempurna, bagaimana tidak anda disuguhkan layanan cetak peta dalam aplikasi *google earth*.

II.8.1 Kekurangan *Google Maps*

Kekurangan dari *Google Maps* adalah sebagai berikut :

1. SketchUp atau Building Maker

Tahukah Anda jika pengguna juga bisa ciptakan model 3D via *Google Maps* atau *Google Earth*? Ini berarti Anda tidak hanya sekadar sebagai pengguna. *Google* berikan kesempatan edukasi membuat model 3D via *SketchUp* atau *Building Maker*.

2. Kontribusi

Google telah petakan hampir seluruh wilayah di dunia. Pun juga, mereka memotret jutaan foto lokasi dan jalan raya. Namun, Google tak mampu berikan keterangan di semua lokasi. Pengguna bisa berkontribusi berikan saran, koreksi, diskripsi, serta *additional* foto lokasi.

3. Bisa digunakan *offline*

Google Maps hadir di pelbagai perangkat *mobile* contohnya *Android*. Namun tahukah Anda jika aplikasi ini tetap bisa digunakan meski dalam kondisi luring? *Ya*, walaupun tak terkoneksi dengan internet, Anda tetap bisa bernavigasi via *Google Maps*.

4. Tidak sempurna

Meski telah memetakan hampir seluruh wilayah di dunia, *Google Maps* tetap memiliki kekurangan. Misalnya saja soal lokasi yang tak bisa dijangkau. Korea Utara, adalah salah satu wilayah yang tidak bisa disentuh *Google Maps*.

5. Lalu-lintas

Di beberapa lokasi tertentu, *Google Maps* mampu tampilkan data lalu-lintas terkini secara *real-time*. Misalkan di beberapa lokasi di Britania Raya.

6. Lokasi yang tak mungkin

Jika dibilang tak mungkin dijamah, memang benar. Namun *Google Maps* terbukti mampu ‘mengunjunginya’. Sebut saja [Gunung Everest](#) atau ‘Kota Hantu’ di Fukushima. Bahkan *Google Maps* juga telah sambangi gua bawah tanah; gua Akiyoshi-do, Jepang, dan beberapa gua di Hutan Hujan Amazon.

7. Simulasi penerbangan

Ingin *berlagak* sebagai pilot, sila coba simulasi penerbangan di *Google Earth*. Di opsi *tool* pilih *Enter Flight Simulator*. Kemudian tinggal pilih jenis pesawat dan lokasiya.

8. *Google Earth*

Google Earth adalah bagian dari *Google Maps*. Layanan ini miliki kelebihan yang tidak dimiliki *Google Maps*. Dulu menggunakan *Google Earth* harus mengunduh aplikasinya terlebih dahulu. Kini, *Google Earth* bisa langsung dinikmati di situs *Google Maps* (*web*). Namun, pengguna harus mengunduh terlebih dahulu *plugin* *Google Earth* untuk menjalankan peta digital interaktif ini.

II.9. Pengertian PHP

PHP adalah akronim dari hypertext preprocessor, yaitu suatu bahasa pemrograman berbasis kode-kode (script) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web browser menjadi kode HTML. Kode PHP mempunyai ciri-ciri khusus, yaitu : (Diar Puji Oktavian ; 2010 : 31)

- a. Hanya dapat dijalankan menggunakan web server, misalnya apache

- b. Kode PHP diletakkan dan dijalankan di web server
- c. Kode PHP dapat digunakan untuk mengakses database, seperti mysql, postgresql, oracle
- d. Merupakan software yang bersifat open source
- e. Gratis untuk didownload dan digunakan
- f. Memiliki sifat multiplatform, artinya dapat dijalankan menggunakan system operasi apapun, seperti linux, unix windows

Dengan menggunakan PHP selain memberikan keuntungan seperti pada beberapa poin di atas, juga didukung oleh banyak komunitas. Hal ini yang membuat PHP terus berkembang. Selain itu, dapat belajar lebih banyak lagi tentang tips dan trik penggunaannya dari berbagai komunitas, lembaga pendidikan, ataupun melalui media internet.

Sebelum mempelajari PHP, sebaiknya lakukan beberapa persiapan terlebih dahulu, seperti menyediakan web server dan program bantu editor PHP. Dengan editor khusus, maka dapat dengan mudah mendeteksi jika terjadi kesalahan penulisan perintah.

PHP singkatan dari PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman web server side yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (server side HTML embedded scripting). PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan

informasi yang diterima client selalu yang terbaru atau up to date. Semua script PHP dieksekusi pada server di mana script tersebut dijalankan. (Anhar ; 2010 : 3).

II.9.1. Alasan Mempelajari PHP

Beberapa alasan bagi yang mempelajari PHP adalah karena :

1. Kesederhanaan.

User yang baru belajar pemrograman, alasan ini pasti merupakan alasan utama untuk mulai belajar PHP. Karena kesederhanaan tersebut, maka menjadi merasa mudah untuk belajar PHP. User yang sedikit tahu atau bahkan sama sekali tidak mengerti tentang pemrograman PHP bisa dengan cepat belajar dan mencoba membuat aplikasi web PHP. Selain itu, PHP memiliki banyak sekali fungsi built in untuk menangani kebutuhan standar pembuatan aplikasi web. Dengan adanya fungsi-fungsi tersebut, maka tentu saja proses belajar PHP terutama dalam pengembangan aplikasi akan jauh lebih mudah karena semua sudah tersedia.

2. Dalam sisi pemahaman.

PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

3. PHP adalah bahasa open source yang dapat digunakan di berbagai sistem operasi seperti linux, unix, macintosh, dan windows. PHP dapat dijalankan secara *runtime* melalui console serta dapat menjalankan perintah-perintah system. Open source artinya *code-code* PHP terbuka untuk umum dan tidak harus membayar biaya pembelian atas keaslian license yang biasanya cukup mahal.

4. Web Server yang mendukung PHP dapat ditemukan dimana-mana, mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
5. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti support langsung ke berbagai macam database yang populer, misalnya Oracle, PostgreSQL, MySQL dan lain-lain. (Anhar ; 2010 : 4).

II.9.2. PHP dan Internet

Sebenarnya saat berinternet menggunakan browser seperti Mozilla, Internet Explorer, Opera, dan Safari, halaman yang muncul di depan layar komputer adalah halaman yang diperoleh dari proses pemanggilan dengan menuliskan alamat sesuai nama *file* yang terdapat pada *web browser*. (Anhar ; 2010 : 4).



Gambar II.6. Mengetikkan Alamat URL

(Sumber : Sistem Informasi Geografis Menggunakan ARC VIEW GIS, Eko Budiyanto, 2010)

II.10. MySQL

MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (Database Management System) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, PostgreSQL, dan lain-lain.

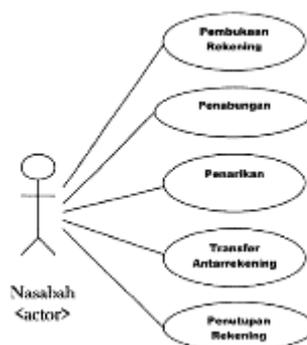
MySQL merupakan DBMS yang multithread, multi user yang bersifat gratis yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan Swedia, yaitu MySQL AB. MySQL AB memegang hak cipta kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larson, dan Michael Monty Widenius. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau open source sehingga bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung atau support secara gratis. Pemrograman PHP juga sangat mendukung / support dengan database MySQL sehingga apabila mempelajarinya dengan sungguh-sungguh dapat mengaplikasikan PHP dan MySQL dalam membuat aplikasi website maupun dalam membuat website. (Anhar ; 2010 : 21).

II.11. UML

Pemodelan perangkat lunak bekerja dengan cara yang cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan, ia biasanya membuat denah-denang yang menggambarkan bentuk jadi dari bangunan. Sebagai seorang perancang sistem perangkat lunak juga bertindak dengan cara yang serupa, hanya saja yang dirancang bukan bangunan, melainkan sistem perangkat lunak. Menggambarkan komponen sistem perangkat lunak dalam bentuk-bentuk geometri tertentu dalam aplikasi (*Adi Nugroho; 2009:6*).

II.11.1. Diagram Use Case

Membuat *use case* diagram yang komprehensif merupakan hal yang sangat penting dilakukan pada tahap analisis. Dengan menggunakan *use case* diagram, akan mendapatkan banyak informasi yang sangat penting yang berkaitan dengan aturan-aturan bisnis yang coba ditangkap. Dalam hal ini, setiap objek yang berinteraksi dengan sistem perangkat lunak misalnya, orang, suatu perangkat keras, sistem lain, dan sebagainya merupakan *actor* untuk sistem perangkat lunak, sementara *use case* merupakan deskripsi lengkap tentang bagaimana sistem perangkat lunak berperilaku untuk para *actornya*. Dengan demikian, *use case* diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para *actor* dengan sistem perangkat lunak yang sedang kita kembangkan. Untuk lebih jelas dapat dilihat pada Gambar II.4.



Gambar II.7. Diagram Use Case
(Sumber : Adi Nugroho; 2009 : 8)

Actor pada dasarnya ditentukan berdasarkan perannya (role) pada program /aplikasi yang sedang kita kembangkan, bukan sebagai objek-objek secara

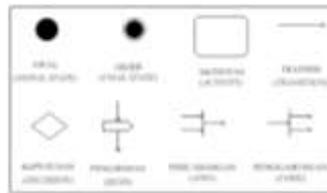
mandiri. Sebagai contoh, jika mengambil kasus ATM (Anjungan Tunai Mandiri), seseorang (objek tunggal) mungkin bisa dikelompokkan sebagai actor Karyawan Bank serta Nasabah jika orang tersebut merupakan karyawan bank yang bersangkutan sekaligus sebagai nasabah karena memiliki tabungan di bank tersebut. Sementara itu, Adi, Ana Geuis, dan beberapa orang lainnya dapat dikelompokkan menjadi actor nasabah jika mereka semua masing-masing memiliki tabungan di bank tersebut. Dalam hal ini, kita akan coba mengambil contoh *actor* nasabah untuk menentukan *use casenya*.

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek, karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*. UML juga memberikan standar penulisan sebuah sistem blueprint, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. UML terdiri dari beberapa diagram, yaitu *use case diagram*, *class diagram*, *state diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, dan *deployment diagram*. (MHD Imam Alfarisyi, dkk ; 2011 : 55).

II.11.2. Diagram Activity

Activity diagram menggambarkan urutan aktifitas yang digunakan untuk menjelaskan aktifitas dari sebuah operasi. Pada *activity diagram* terdapat keadaan aksi yang berisi spesifikasi dari aktifitas tertentu. Diagram ini berisi, pilihan

keputusan dan kondisi serta spesifikasi message yang dikirim atau diterima sebagai gambaran dari aksi.



Gambar II.8. Diagram Activity
(Sumber : library.upnvj.ac.id ; 2011)

II.11.3. Sequence

Diagram interaksi yang menekankan pada waktu pengiriman *message*. *Sequence* diagram menunjukkan sekumpulan objek dan pengiriman serta penerimaan *message* antar objek. Objek yang umumnya memiliki nama atau instansiasi dari *class*, tapi dapat pula merupakan turunan dari *things* lain, seperti *collaboration*, *component* dan *node*. Diagram ini digunakan untuk mengilustrasikan *dynamic view* dari sistem.

II.11.4. State

Sebuah *state* diagram menggambarkan keadaan mesin, transisi, *event* dan *activity*. Diagram ini adalah pelengkap khusus untuk mendeskripsikan sebuah *class* yang menggambarkan *state* dari objek dari *class* dan *event* yang menyebabkan *state* berubah. *Event* tersebut dapat berasal dari objek yang mengirimkan suatu *message* atau dari kondisi yang terpenuhi.

II.12. Konsep Basis Data

Database merupakan kumpulan file yang saling berhubungan. Akan tetapi *database* tidak hanya kumpulan file. *Record* di dalam tiap file harus dapat dihubungkan dengan *record* di dalam file lain. (Toni Kurniawan ; 2011 : 40)

Dalam manajemen *database* relational terdapat komponen utama dalam konsep database :

1. *Field* adalah unit terkecil data yang disimpan dalam database. Unit terkecil data yang disimpan dalam database.
 - a. *Primary key* yaitu field yang unik dan mengidentifikasi satu record.
Contoh *customer number* dan *order number*.
 - b. *Secondary key* yaitu field yang mengidentifikasi sebuah record atau bagian dari beberapa record yang terkait.
 - c. *Foreign key* yaitu field yang menunjuk beberapa record pada file lain.
Contoh *Order Record* berisi *foreign key customer number*.
 - d. *Descriptive field* yaitu non key field.
2. *Record* adalah kumpulan *field* yang diatur dalam format yang predetermined (telah ditentukan).
 - a. *Fixed length record structures*

Sebagian besar teknologi database memaksakan struktur *record fixed length*, dalam artian setiap *instance record* mempunyai *field* yang sama, jumlah field yang sama, dan ukuran logika yang sama. Akan tetapi beberapa sistem *database* akan mengkompresi *field-field* dan

nilai-nilai yang tidak digunakan untuk menghemat ruang penyimpanan disk.

b. *Variable length record structured*

Memperoleh *record-record* pada *file* yang sama memiliki *length* yang berbeda.

3. *Field* dan Tabel

File adalah kumpulan semua kejadian dari struktur record yang ditentukan.

Tipe-tipe dari file yaitu :

- a. File induk / master adalah *file* penting dalam sistem dan akan tetap ada selama siklus hidup sistem informasi berputar.
- b. File transaksi adalah file yang digunakan untuk merekam data dari suatu transaksi yang terjadi.
- c. File laporan adalah file yang berisi sistem informasi yang akan ditampilkan.
- d. File sejarah adalah file yang berisi data masa lalu yang sudah tidak aktif lagi.
- e. *File pelindung* adalah salinan dari *file-file* yang masih aktif di *database* pada saat tertentu yang digunakan bila *file database* rusak.
- f. File kerja adalah suatu proses program secara sementara karena memori komputer tidak mencukupi.