

BAB II

LANDASAN TEORI

II.1 Studi Literatur

II.1.1 Sistem

Menurut (Kusrini ; 2009 : 11) Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*).

II.1.2 Data

Menurut (Kusrini ; 2009 : 3) Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian, ambil contoh fakta mengenai biodata mahasiswa yang meliputi nama, alamat, jenis kelamin, agama yang dianut, dan lain-lain. Contoh lain dari fakta mengenai kejadian / transaksi dalam sebuah perusahaan dagang adalah seperti transaksi penjualan yang meliputi waktu transaksi, pelaku transaksinya (pelanggan, kasir), barang yang ditransaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol.

II.1.3. Informasi

Menurut (Kusrini ; 2009 : 4) informasi merupakan hasil olahan data, di mana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk pengambilan keputusan. Informasi juga diartikan sebagai himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu.

Menurut (Kusrini ; 2009 : 4) Suatu informasi berguna bagi Pembuat keputusan karena informasi bisa menurunkan ketidakpastian (meningkatkan pengetahuan) tentang hal yang sedang dipikirkan. Makna dari sebuah informasi tentu berbeda-beda antara seorang dengan lainnya, tergantung pada tingkat kepentingannya, misalnya informasi daftar pelanggan yang potensial akan sangat dibutuhkan oleh bagian marketing di suatu perusahaan guna meningkatkan penjualan produk, tetapi barangkali tidak akan menjadi perhatian dibagian personalia

Kegunaan informasi bagi seseorang juga sangat tergantung pada waktu. Pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan dilain hari, mungkin saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi perbandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut, informasi tersebut menjadi kurang bermakna

II.1.4. Kualitas Informasi

Agar bisa menyediakan keluaran yang berguna untuk membantu manager atau para pengambil keputusan, sebuah sistem informasi harus mampu mengumpulkan data dan mentransformasikan data tersebut kedalam informasi yang memiliki kualitas-kualitas tersebut

Berikut karakteristik informasi yang berkualitas :

1. **Relevan.** Informasi yang disajikan sebaiknya terkait dengan keputusan yang akan diambil oleh pengguna informasi tersebut. Misalnya, seorang manager

yang akan memberikan kredit kepada pelanggan bisa melihat laporan keuangan pelanggan tersebut karena laporan tersebut terkait dengan keputusan yang akan dibuat, yaitu memberikan atau tidak memberikan kredit kepada pelanggan tersebut.

2. **Akurat.** Kecocok antara informasi dengan kejadian-kejadian atau objek-objek yang diwakilinya. Misalnya, laporan inventaris yang tidak akurat menyebutkan bahwa terdapat 15 unit barang yang tersisadi gudang. Kenyataannya, masih ada 51 unit barang di dalam gudang.
3. **Lengkap.** Merupakan derajat sampai seberapa jauh informasi menyertakan kejadian-kejadian atau objek-objek yang berhubungan. Misalnya, penjualan selama satu hari yang seharusnya ada 150 transaksi di laporan hanya tercatat sebanyak 145 transaksi.
4. **Tepat waktu.** Informasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat di gunakan untuk membantu pengambilan keputusan. Misalnya, informasi jadwal ujian seorang mahasiswa disampaikan setelah kegiatan ujian diselenggarakan. Informasi ini menjadi tidak berguna lagi.
5. **Dapat dipahami.** Hal tersebut terkait dengan bahasa dan cara penyajian informasi agar pengguna lebih mudah mengambil keputusan.
6. **Dapat dibandingkan.** Sebuah informasi yang memungkinkan seorang pemakai untuk mengidentifikasi persamaan dan perbedaan antara dua objek

atau kejadian yang mirip. Misalnya, membandingkan laporan pendapatan antara tahun 2006 dan 2007,

II.1.5. Sistem Informasi

Menurut (Kusrini ; 2009 : 11) suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar.

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi

1. Sistem Pemrosesan Transaksi (*Transaction Processing System*) atau TPS
2. Sistem Informasi Manajemen (*Management Information System*) atau MIS
3. Sistem Otomasi Perkantoran (*Office Automation System / OAS*)
4. Sistem Pendukung Keputusan (*Decision Support System*) atau DSS
5. Sistem Informasi Eksekutif (*Executive Information System*) atau EIS
6. Sistem Pendukung Kelompok (*Group Support System*) atau GSS
7. Sistem Pendukung Cerdas (*Intelegent Support System*) atau ISS

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk pendukung manajemen, maka ketiga sistem tersebut sering disebut Sistem Pendukung Manajemen (*management support system*) atau MSS

II.1.6 Pengertian Sistem Informasi

Menurut Tata Sutabri (2005 : 42) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data yang

mendukung fungsi operasional yang bersifat manajerial dengan kegiatan strategi dari satu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto Hartono ; 2005 : 11).

II.2. Sistem Informasi Geografis

II.2.1. Sejarah Sistem Informasi Geografis

Menurut Supriadi (2008 : 1) Sebelum tahun 1940-an analisis geografis melakukan tumpang tindih (*overlay*) beberapa jenis peta pada area tertentu. Namun sejak tahun 1950-an di kembangkan sistem digital untuk melakukan analisis dalam peranan memecahkan permasalahan keruangan. Hingga kini berbagai peranan untuk mengatasi berbagai aspek permasalahan yang berkaitan dengan ruang.

Perkembangan di bidang teknologi komputer telah membawa manfaat yang sangat besar bagi penyebaran informasi. Dengan internet misalnya, kita dapat melihat tempat-tempat yang indah di segala penjuru dunia bila tampilannya memanfaatkan sistem informasi geografi.

II.2.2. Pengertian Sistem Informasi Geografis

Menurut Supriadi (2008 : 1) Sistem Informasi Geografis (SIG) adalah bahagian daripada sistem informasi yang di aplikasikan untuk data geografi atau alat *database* untuk analisis dan pemetaan sesuatu yang terdapat dan terjadi di bumi.

Sistem Informasi Geografis (SIG) merupakan salah satu model sistem informasi yang banyak digunakan untuk membuat berbagai keputusan, perencanaan dan analisis.

SIG memiliki perbedaan pokok dengan sistem informasi lain. Perbedaan ini justru menjadi ciri karakteristiknya. Pada sebuah sistem informasi selain SIG, basis data atributal adalah focus dari pekerjaan sistem, sedangkan SIG mengaitkan data atributal dengan data spasial. SIG member analisis keruangan terhadap data atribut tersebut. SIG menjelaskan dimana, bagaimana dan apa yang akan terjadi secara keruangan yang di wujudkan dalam gambaran peta dengan berbagai penjelasan secara deskriptif, tabular dan grafis.

Dari kemampuannya tersebut, SIG memberi dua jenis model informasi, yaitu dalam bentuk spasial dan deskriptif. Hubungan antara bentuk spasial dan deskriptif di jelaskan secara topologis (Eko Budiyanto ; 2004 : 1).

II.2.3. Perolehan Data Sistem Informasi Geografis

Menurut Eko Budiyanto (2004 : 5) Sistem Informasi Geografis memanfaatkan berbagai macam sumber data baik spasial ataupun atribut dalam

bentuk tabular ataupun deskriptif. Berbagai bentuk data ini digunakan secara bersama-sama ataupun sendiri-sendiri dalam proses analisisnya. Sistem informasi geografis bekerja dengan tipe data *raster* dan data *vector*. Data *raster* secara umum di kenal sebagai *image* atau citra atau gambar. Data *raster* terdiri dari kumpulan piksel yang di wujudkan dalam nilai-nilai spectral. Nilai spectral terendah adalah nilai 0 yang secara visual akan tampak sebagai warna hitam. Nilai spectral tertinggi adalah 255 yang secara visual tampak sebagai warna putih. Data *vector* dalam sistem informasi geografis di kenal beberapa tipe, yaitu tipe titik, tipe garis, dan tipe polygon. Tipe titik (*point*) digunakan untuk menggambarkan fenomena seperti kota, mata air, puncak, gunung, dan sebagainya. Tipe garis (*line*) digunakan untuk menggambarkan fenomena yang berupa garis seperti jalan, rel kereta api, dan sungai. Tipe polygon sering digunakan untuk menggambarkan fenomena berupa wilayah seperti penggunaan lahan, administrasi, penutup lahan, dan sebagainya.

II.3. *MapInfo*

II.3.1. Pengertian *MapInfo*

Menurut Eko Budiyanto (2004 : 19) *MapInfo* adalah *software* pengolah data spasial yang banyak digunakan dalam analisis Sistem Informasi Geografis. *Software* ini memiliki kemampuan seperti *software-software* pengolah data spasial lain seperti *MapInfo* atau *ArcView*. *MapInfo* merupakan *software* pengolah data spasial yang terpadu dengan data tabel. Melalui *software MapInfo* operator dapat

membuat, menampilkan, serta mengadakan perubahan terhadap data spasial atau peta.

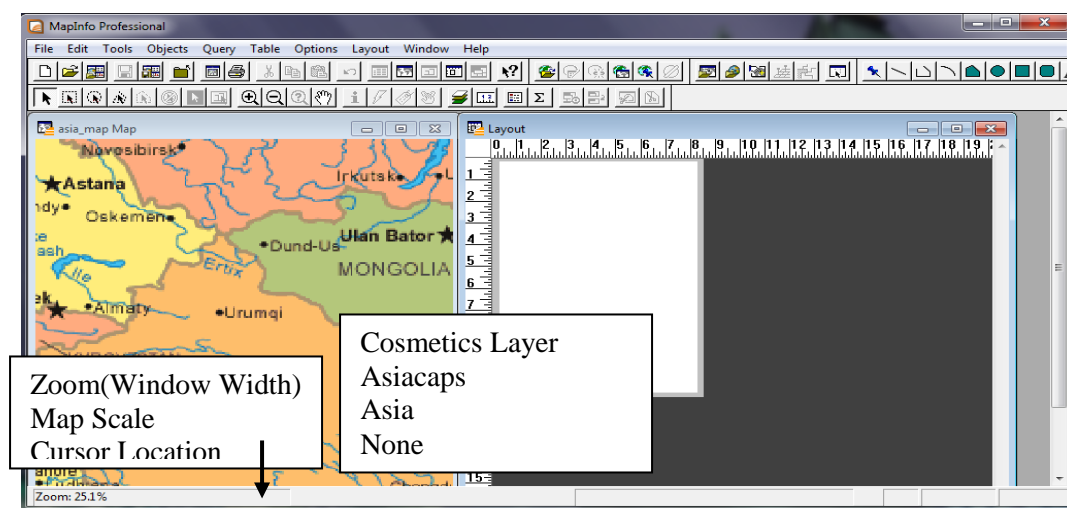
MapInfo bersifat fleksibel dalam penampilan dan perubahan data yang antara lain mencakup :

1. Pembukaan banyak tabel dalam waktu yang bersamaan.
2. Pengendali *property layer* secara individual.
3. Mampu membuat dan memodifikasi peta-peta tematik yang ada.
4. Pencarian informasi terkait dengan data spasial.
5. Sistem kendali proyek peta.

II.3.2. *User interface MapInfo*

Menurut Eddy Prahasta (2006 : 7) *Mapinfo professional* hadir bersama dengan beberapa *user interface* yang di implementasikan dalam berbagai bentuk.

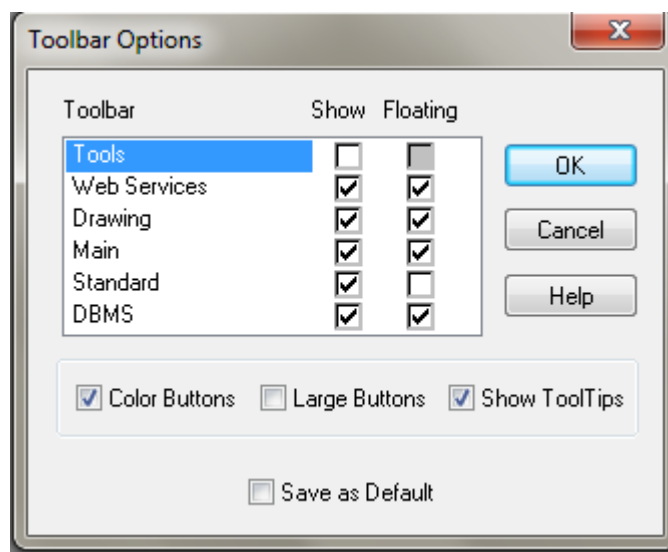
Berikut adalah sebagian di antaranya yang sering di jumpai oleh pengguna.



Gambar II.1. Contoh Tampilan User Interface Mapinfo Profesional

Sumber : Eddy Pahasta (2006 : 7)

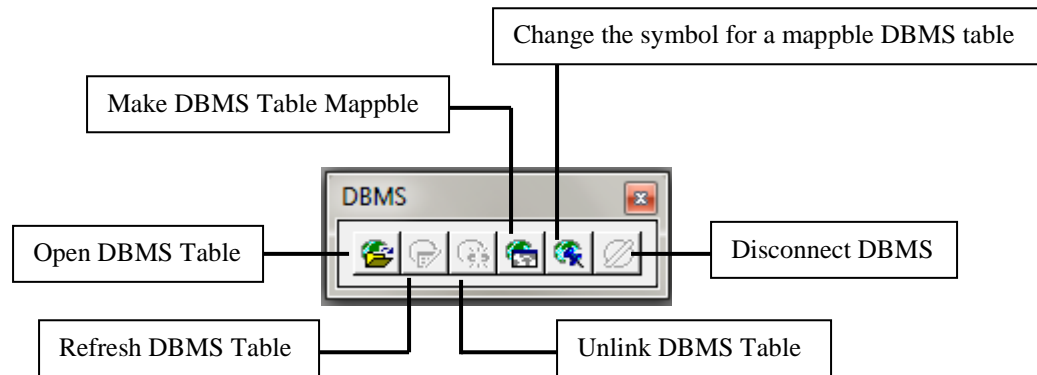
Selain itu, di dalam *MapInfo*, pengguna juga dapat menjumpai sejumlah *user interface* yang di implementasikan dalam bentuk kotak dialog *floating* yang berisi beberapa *tool (icon)* di dalamnya kotak dialog ini dapat di aktifkan dengan cara menggunakan menu “*Option | Toolbars* “ sehingga muncul kotak dialog “*Toolbar Options*”



Gambar II.2. Tampilan Kotak Dialog “Toolbar Options”

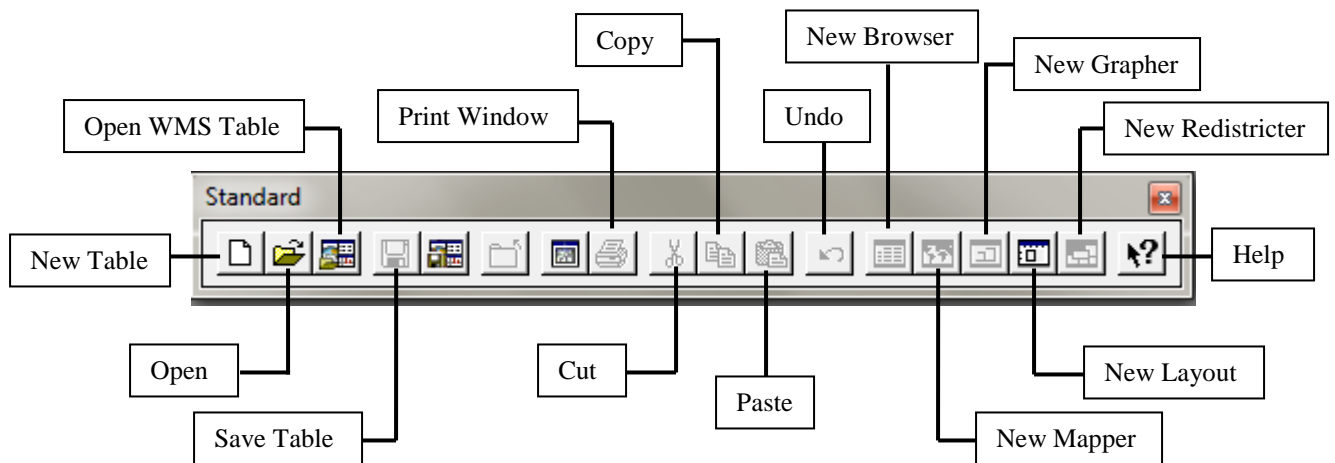
Sumber : Eddy Pahasta (2006 : 8)

Kemudian pengguna dengan menggunakan kotak “*Toolbar Options*” di atas dapat memilih sekaligus beberapa kotak dialog *floating* yang akan di aktifkan pada saat yang sama.



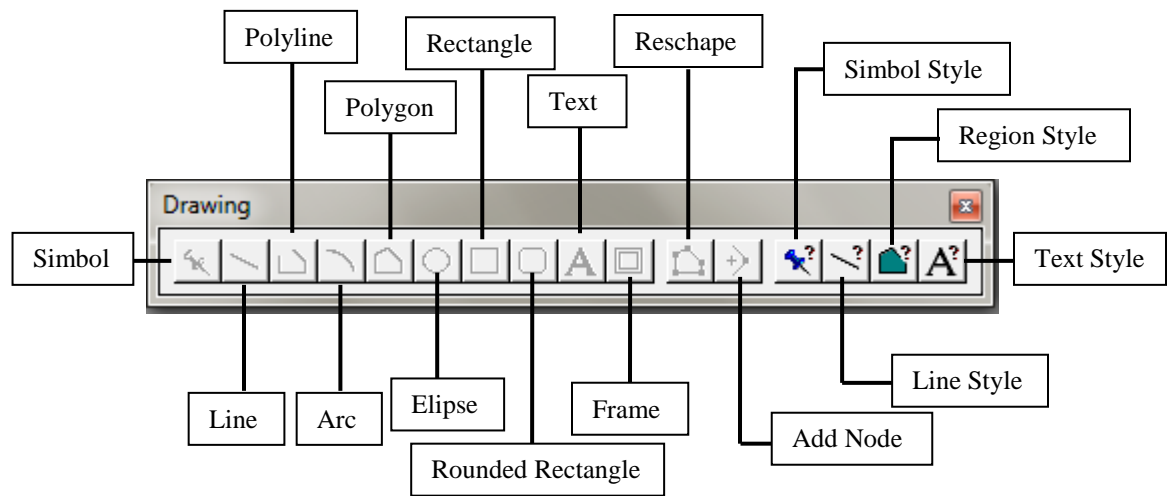
Gambar II.3. Tampilan *Floating Toolbar* “DBMS”

Sumber : Eddy Pahasta (2006 : 9)



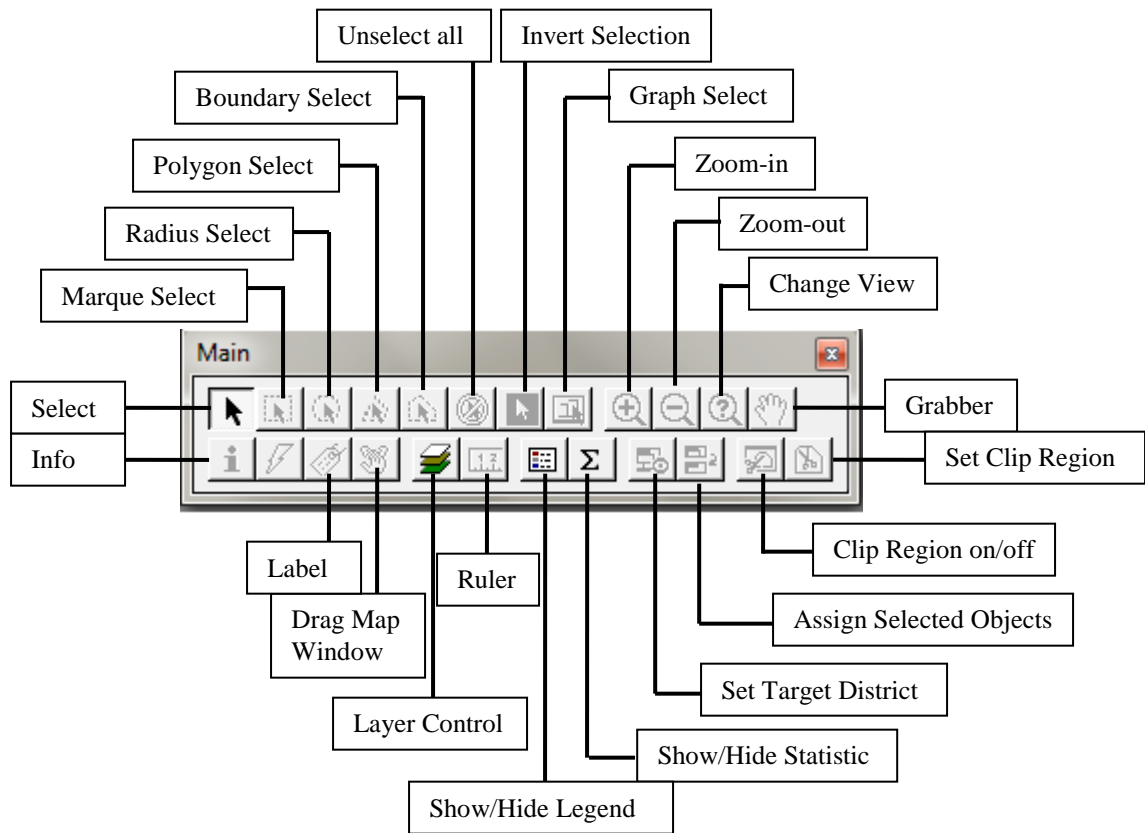
Gambar II.4. Tampilan *Floating Toolbar* “Standar”

Sumber : Eddy Pahasta (2006 : 9)



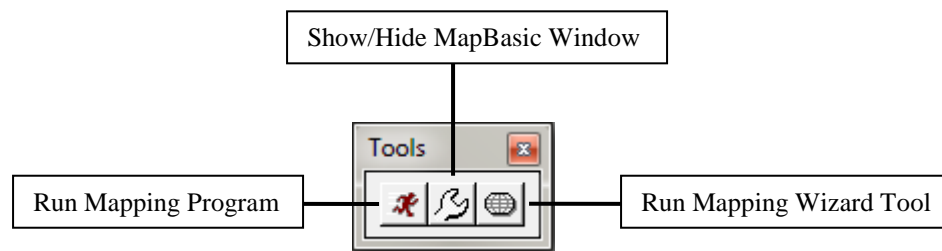
Gambar II.5. Tampilan *Floating Toolbar "Drawing"*

Sumber : Eddy Pahasta (2006 : 9)



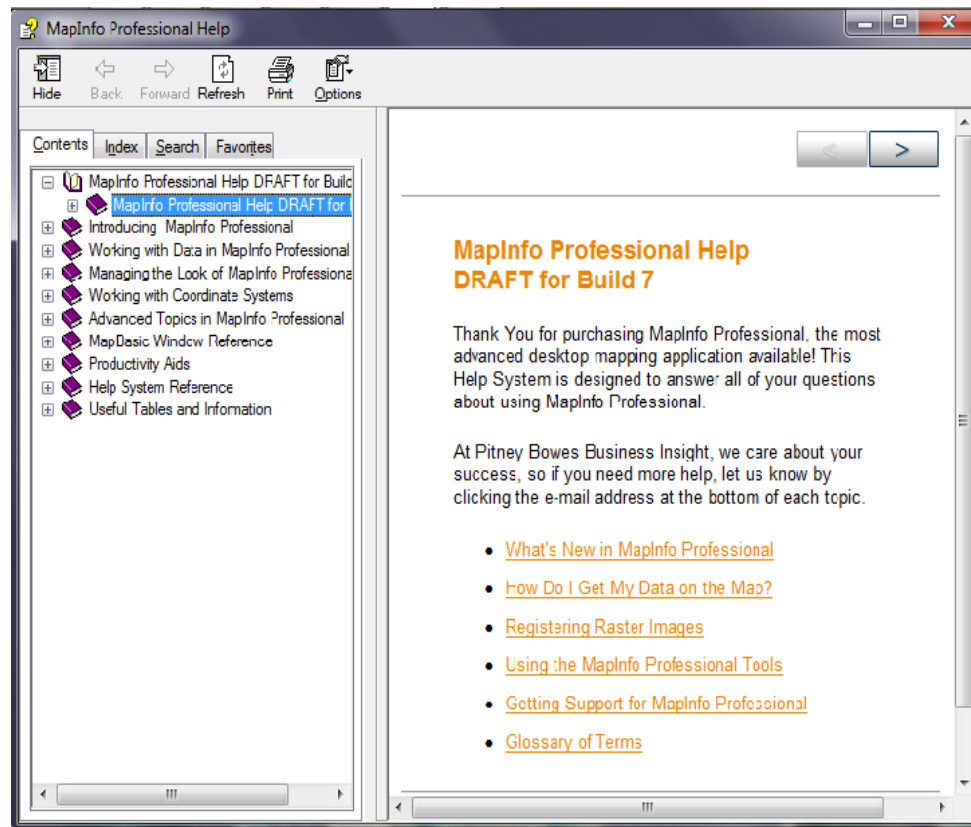
Gambar II.6. Tampilan *Floating Toolbar "Main"*

Sumber : Eddy Pahasta (2006 : 10)



Gambar II.7. Tampilan *Floating Toolbar "Tools"*

Sumber : Eddy Pahasta (2006 : 10)



Gambar II.8. Tampilan Kotak Dialog “Help”

Sumber : Eddy Pahasta (2006 : 11)

II.4. Peta

II.4.1. Konsep Dasar Peta

Menurut Supriadi (2008 : 15) *Output* terpenting dari Sistem Informasi Geografis adalah peta yang menggambarkan jenis, penampakan dan informasi yang beragam. Peta menampilkan rona geografis atau fenomena keruangan lainnya melalui gambar yang menyangkut informasi tentang lokasi dan atributnya. Tujuan dasar pemetaan adalah untuk menyediakan :

1. Penjelasan fenomena geografis.
2. Informasi keruangan dan bukan keruangan.
3. *Feature* peta seperti titik, garis dan area.

II.4.2. Ciri (*Feature*) Peta

Menurut Supriadi (2008 : 15) Informasi lokasi biasanya di tampilkan berupa titik (*point*), garis (*line*), area (*polygon*).

1. Titik (*points*) untuk menggambarkan lokasi tunggal yang terlalu kecil untuk di wujudkan sebagai garis atau area.
2. Garis (*line*) untuk menggambarkan alur, jalur pipa, garis kontur. *Feature* garis (*line*), merupakan sekumpulan hubungan koordinat di tampilkan sebagai garis yang terlalu pipih untuk di tampilkan sebagai sebuah area, seperti jalan atau garis kontur.
3. Area (*polygon*) untuk menggambarkan danau, daerah dan lokasi sensus. *Feature* area (*polygon*) merupakan gambar tertutup yang membatasi daerah yang homogen.

II.4.3. Jenis Peta

Menurut Supriadi (2008 : 16) Peta merupakan gambaran sederhana akan keberadaan di muka bumi. Peta dapat di bagi dalam dua kelompok besar yaitu :

1. Peta Topografi (*topographical map*)

Merupakan peta yang menunjukkan Penampakan alamiah dan buatan manusia di bumi yang di wujudkan oleh kontur, lahan, jalan, kereta api dan

berbagai *feature* penting lainnya. Kondisi muka bumi ini dapat di tayangkan dalam berbagai jenis peta, seperti peta topografi dan peta kontur. Peta topografi menunjukkan penampakan fisik permukaan bumi, seperti jalan, sungai dan bangunan. Sedangkan peta kontur menampakan garis yang menghubungkan titik tertentu yang memiliki kesamaan nilai, misalnya ketinggian tempat dari permukaan laut dan isobar (menunjukkan tekanan udara).

2. Peta Tematik (*Thematic Map*)

Merupakan sumber yang penting dari informasi SIG. Peta tematik merupakan sarana untuk menyampaikan konsep geografis dengan thema tertentu, seperti kepadatan populasi, iklim, jenis tanah, geologi, kesesuaian lahan, pergerakan barang, dan manusia serta penggunaan lahan.

II.5. *UML (Unified Modeling Language)*

Menurut Munawar (2005 : 17) *Unified Modeling Language* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini di sebabkan karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah di mengerti serta di lengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Ada beberapa diagram dalam *Unified Modeling Language*, di antaranya adalah :

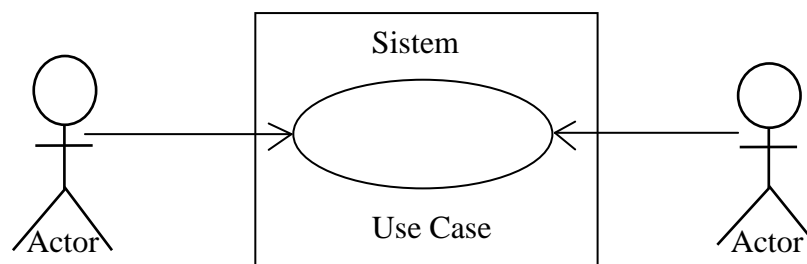
1. *Use Case Diagram*

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem di pakai.

Dalam pembicaraan tentang *Use Case*, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa di mainkan oleh pengguna dalam interaksinya dengan sistem.

Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu aktor, *use case*, dan sistem/sub sistem boundary. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*.

Adapun ilustrasi *actor*, *use case*, dan *boundary* dapat di lihat pada gambar II.9. berikut ini :



Gambar II.9. Use Case Model

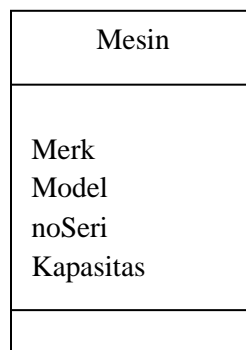
Sumber : Munawar (2005 : 64)

2. *Class Diagram*

Class Diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini di sebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (*operasi*) dan relasi yang sama.

Adapun contoh *class* dan atributnya dapat di lihat pada gambar II.10.

berikut ini :



Gambar II.10. Class dan atributnya

Sumber : Munawar (2005 : 36)

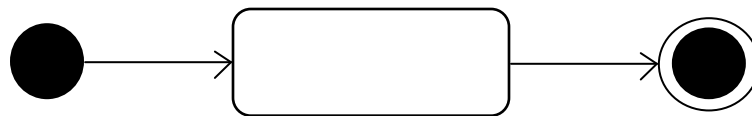
3. *Statechart Diagram*

Interaction diagram dan *statechart* menampilkan dua pandangan yang saling melengkapi tentang perilaku dinamis sebuah sistem. *Interaction* diagram menunjukkan pesan-pesan yang di lewatkan di antara obyek-obyek di dalam sistem selama periode waktu yang pendek. Sedangkan *statechart* diagram menelusuri individu-individu obyek melalui keseluruhan daur hidupnya, menspesifikasikan semua urutan yang mungkin dari pesan-pesan

yang akan di terima obyek tersebut, bersama-sama dengan tanggapan atas pesan-pesan tersebut.

State diagram menyediakan variasi simbol dan sejumlah ide untuk pemodelan. Tipe diagram ini mempunyai potensi untuk menjadi sangat kompleks dalam waktu yang singkat.

Adapun simbol-simbol yang terdapat pada *statechart diagram* dapat di lihat pada gambar II.11. berikut ini :



Gambar II.11. Simbol-simbol Statechart Diagram

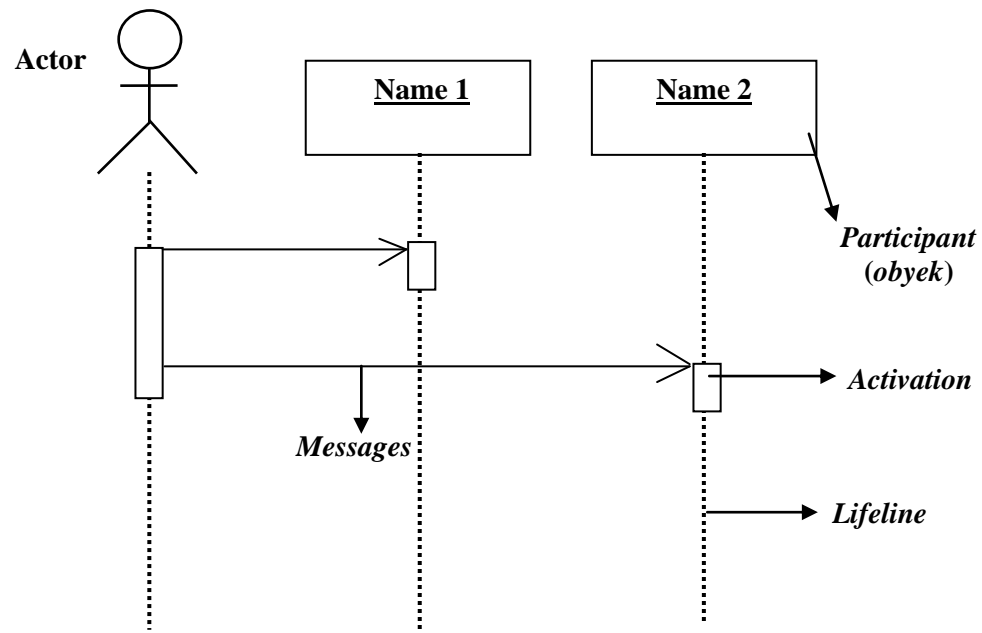
Sumber : Munawar (2005 : 74)

4. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Dengan ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang di letakkan di antara obyek-obyek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang di tuliskan dengan kotak segiempat bernama. *Message* di wakili oleh garis dengan tanda panah dan waktu yang di tunjukkan dengan *progress vertical*.

Adapun simbol-simbol yang terdapat pada *sequence diagram* dapat di lihat pada gambar II.12. berikut ini :



Gambar II.12. Simbol-simbol sequence diagram

Sumber : Munawar (2005 : 89)

5. Collaboration Diagram

Collaboration diagram adalah perluasan dari obyek diagram. (Obyek diagram menunjukkan obyek-obyek dan hubungannya satu dengan yang lain). *Collaboration diagram* menunjukkan *message-message* obyek yang di kirimkan satu sama lain.



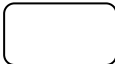
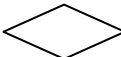






Antara *collaboration diagram* dan *sequence diagram* bisa saling mengisi. Dengan demikian pada *collaboration diagram* kita bisa menambahkan nomor urut pada label sebuah *message* untuk menunjukkan urutan informasi. Titik dua (:) perlu digunakan untuk memisahkan nomor dengan *message*.

6. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku *parallel* sedangkan *flowchart* tidak bisa.

Adapun simbol-simbol yang terdapat pada *Activity diagram* dapat di lihat pada Tabel II.1. berikut ini :

Tabel II.1. Simbol-Simbol Activity Diagram

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork ; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake ; Menunjukkan adanya dekomposisi.
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (Flow Final)

Sumber : Munawar (2005 : 109)

7. *Component Diagram*

Component adalah implementasi *software* dari sebuah *class*. *Class* mewakili abstraksi dari serangkaian *attribute* dan *operation*. Hal terpenting tentang *class* dan *component* adalah sebuah *component* bisa jadi merupakan implementasi lebih dari sebuah *class*.

Dengan memodelkan *component* dan relasinya maka :

1. Klien bisa melihat struktur sistem yang sudah selesai.
2. Pengembang mempunyai struktur untuk panduan kerja.
3. Dokumentator bisa memahami apa yang mereka tulis.
4. Siap untuk digunakan kembali untuk proyek lain.

Adapun Notasi *Component* diagram dapat di lihat pada gambar II.13.

berikut ini :



Gambar II.13. Notasi Component

Sumber : Munawar (2005 : 122)

8. *Deployment Diagram*

Deployment Diagram menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*.

Bagian utama *hardware* / perangkat keras adalah node; yaitu nama umum untuk semua jenis sumber komputasi. Ada 2 tipe node yang mungkin, *Processor* adalah node yang bisa mengeksekusi sebuah *component*, sedangkan *device* tidak. *Device* adalah perangkat keras tipikalnya menjadi *interface* dengan dunia luar.

II.6. *Entity Relationship Diagram (ERD)*

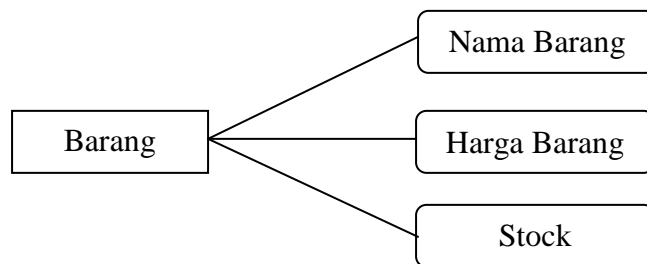
Menurut Kusri S. kom (2007 : 99) ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini *relative* kompleks. Dengan *ERD* kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. dengan *ERD* kita mencoba menjawab pertanyaan. “ Data apa yang kita perlukan ? Bagaimana data yang satu berhubungan dengan yang lain? “

ERD menggunakan sejumlah notasi dan symbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada 3 macam symbol yang digunakan, yaitu :

1. *Entity* adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan di buat. Sebagai contoh adalah barang, pemasok, pekerja dan lain-lain. Seandainya A adalah barang maka A adalah isi dari barang, sedangkan jika B adalah seorang pelanggan maka B adalah isi dari pelanggan. Karena itu

harus di bedakan antara entitas sebagai bentuk umum dari deskripsi tertentu dan isi entitas seperti A dan B dalam contoh di atas. Entitas di gambarkan dalam bentuk persegi empat.

2. *Attribut Entitas* mempunyai elemen yang disebut atribut dan berfungsi mendeskripsikan karakter entitas, misalnya atribut nama barang dari entitas barang. setiap *ERD* bias berisi lebih dari satu atribut. Entitas di gambarkan dalam bentuk elips.



Gambar II.14. Atribut ERD

Sumber : Kusrini S. kom (2007 : 99)

3. Hubungan – *relationship* sebagaimana halnya entitas, hubungan pun harus di bedakan antara hubungan atau bentuk hubungan antara entitas dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara entitas barang dan entitas pelanggan adalah menjual barang, sedangkan isi hubungannya dapat berupa tanggal jual atau yang lainnya. Relationship di gambarkan dalam bentuk intan (*diamonds*).

II.7. PHP (*Hypertext Preprocessor*)

II.7.1. Sejarah Singkat PHP

Menurut mAster.com (2012 : 5) PHP di ciptakan pertama kali Oleh Rasmus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada *Homepage*-nya. Rasmus Lerdorf adalah salah seorang pendukung *open source*. Oleh Karena itu, ia mengeluarkan *Personal Home Page Toll* versi 1.0 secara gratis, kemudian menambah kemampuan *PHP* 1.0 dan meluncurkan *PHP* 2.0.

Pada tahun 1996, *PHP* telah banyak digunakan dalam *website* di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Andi Gutman, Stig Bakken, Shane Caraveo, dan Jim Winstead bekerja sama untuk menyempurnakan *PHP* 2.0. Akhirnya, pada tahun 1998, *PHP* 3.0 di luncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 di keluarkan *PHP* 4.0. Tidak berhenti sampai di sini kemampuan *PHP* terus bertambah, dan sampai saat ini versi terbaru *PHP* yang telah di keluarkan adalah *PHP* 5.0.

II.7.2. Kelebihan PHP

Menurut mAster.com (2012 : 6) *PHP* memiliki banyak kelebihan yang tidak di miliki oleh bahasa script sejenis. *PHP* di fokuskan pada pembuatan script *server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh *CGI* (*Common Gateway Interface*), seperti mengumpulkan data dari *form*, menghasilkan isi

halaman *web dinamis*, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih dari pada kemampuan *CGI (Common Gateway Interface)*.

PHP dapat digunakan pada semua sistem operasi, antara lain *Linux, Unix, Microsoft Windows, Mac OS, RISC OS*. *PHP* juga mendukung banyak *Web Server*, seperti *Apache, Microsoft Internet Information Server, Personal Web Server, Netcape an iPlanet server, oreilly website pro server, audium, Xitami, OmniHTTPd*, dan masih banyak lagi lainnya, bahkan *PHP* dapat bekerja sebagai suatu *CGI processor*.

II.7.3. PHP Programming

Menurut mAster.com (2012 : 7) Sintak program atau *script PHP* di tulis dalam apitan tanda khusus *PHP*. Ada tiga macam pasangan tag *PHP* yang dapat digunakan untuk menandai blok *script PHP*, yakni sebagai berikut :

1. `<?php ... ?>`
2. `<script language="PHP"> ... </script>`
3. `<? ... ?>`

Cara 1 dan 2 merupakan cara yang paling umum digunakan sekalipun cara 3 tampak lebih praktis karena cara 3 tidak selalu di aktifkan pada konfigurasi file php. Ini yang terdapat pada di rektori `c:\apache\php`.

II.8. *MySql Server*

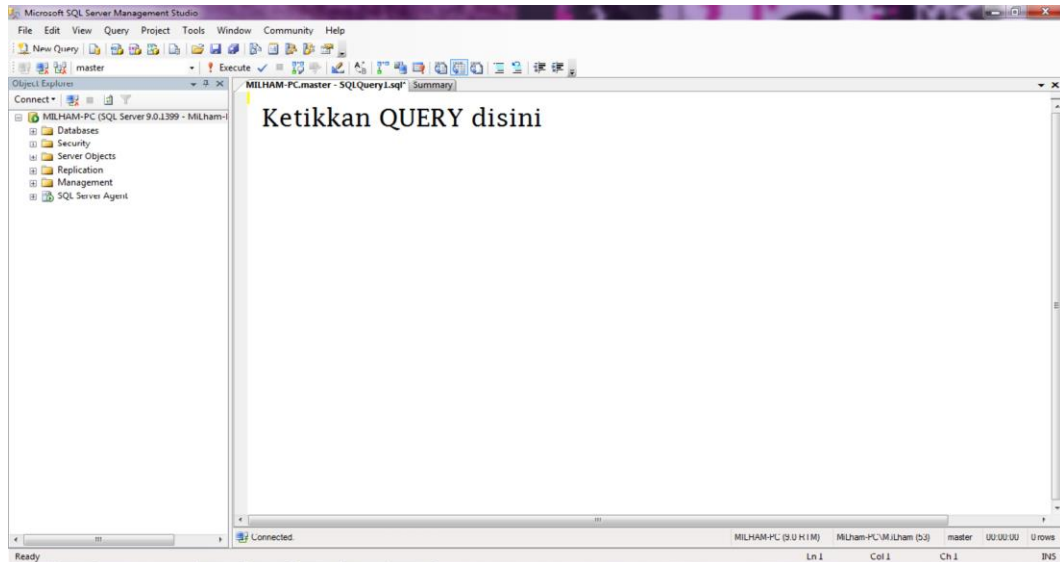
Menurut Emma Utami dan Sukrisno (2008 : 1) SQL (*Structured Query Language*) pada dasarnya adalah bahasa komputer standar yang di tetapkan untuk mengakses dan memanipulasi sistem *database*. Sebuah *database* berisi satu *table* atau lebih dan memiliki nama yang berbeda untuk masing-masing tabel. Masing-masing tabel memiliki satu kolom (*field*) atau lebih dan memiliki baris (*record*). *Query* digunakan untuk mengakses dan mengolah *database*.

SQL terdiri dari 5 (lima) bagian utama, yaitu :

1. Retrieving data: perintah untuk menampilkan data dari *database* (*SELECT*).
2. *Data Definition Language* (*DDL*): bahasa yang digunakan untuk membuat atau menghapus tabel atau *database* itu sendiri (*CREATE, DROP, ALTER*).
3. *Data Manipulation Language* (*DML*): merupakan bahasa untuk memanipulasi/mengubah isi tabel (*INSERT, DELETE, UPDATE*).
4. *Data Control Language* (*DCL*): bahasa yang berhubungan dengan pengendalian akses ke *database* (*GRANT, REVOKE*).
5. *Data Transaction Language* (*DTL*): bahasa yang digunakan untuk mengelola transaksi *database* (*COMMIT, ROLLBACK*).

Untuk menjalankan perintah SQL kita gunakan *Query Analyzer* yang merupakan *tool* yang telah di sediakan oleh *SQL server* 2000.

Berikut adalah tampilan untuk mengetikkan *Query* pada gambar II.16. berikut ini :



Gambar II.15. Tampilan

Untuk mengetikkan Query

Sumber : Emma Utami dan Sukrisno (2008 : 2)