

**PEMBUATAN PROGRAM UNTUK PENYAJIAN INFORMASI
TELAGA DI KABUPATEN GRESIK DENGAN MENGGUNAKAN
MAP OBJECT 2.1 DAN VISUAL BASIC 6.0
(Studi Kasus : Kabupaten Gresik)**



TUGAS AKHIR

**DIAJUKAN UNTUK MEMENUHI PERSYARATAN
DALAM MENCAPI GELAR SARJANA (S-1) TEKNIK GEODESI**

OLEH :

**GALIH JATMIKO
99.25.077**

**JURUSAN TEKNIK GEODESI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI NASIONAL
MALANG
2005**

REPUBLIC OF MALAYSIA
MINISTRY OF DEFENCE
GENERAL INVESTIGATION
(1950-1959)

SILVA SACUT

RECORD NUMBER: 12
IN THE RECORDS OF THE
GENERAL INVESTIGATION

: FILE

ORIENTAL HEAD
1950

RECORD NUMBER: 12
IN THE RECORDS OF THE
GENERAL INVESTIGATION
1950

**PEMBUATAN PROGRAM UNTUK PENYAJIAN INFORMASI
TELAGA DI KABUPATEN GRESIK DENGAN MENGGUNAKAN
MAP OBJECT 2.1 DAN VISUAL BASIC 6.0
(Studi Kasus : Kabupaten Gresik)**

TUGAS AKHIR

**DIAJUKAN UNTUK MEMENUHI PERSYARATAN DALAM MENCAPAI
GELAR SARJANA (S-1) TEKNIK GEODESI**

OLEH :

**GALIH JATMIKO
99.25.077**

Menyetujui :

Dosen Pembimbing I



Ir. Dedy Kurnia Sunaryo, MS.Tis

Dosen Pembimbing II



Ir. Jasmani, M.Kom

Mengetahui :

KETUA JURUSAN TEKNIK GEODESI



Ir. DK. Sunaryo, MS.Tis

Dipertahankan didepan Panitia Penguji Tugas Akhir Jurusan Teknik Geodesi ,
Fakultas Teknik Sipil Dan Perencanaan , Intitut Teknologi Nasional Malang,
dan diterima untuk memenuhi sebagian dari syarat – syarat guna memperoleh
gelar Sarjana (S-1) Teknik Geodesi.

Panitia Ujian Tugas Akhir

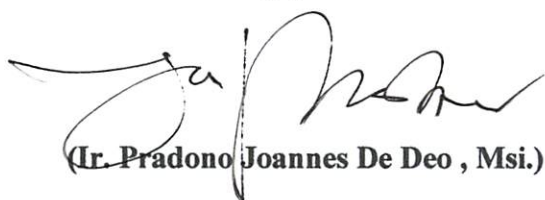

Ketua
(Ir. Agustina Nurul H, MTP)
Dekan FTSP

Sekretaris


(Ir. Dedy Kurnia Sunaryo, MS.Tis)
Ketua Jurusan Teknik Geodesi

Anggota Penguji :

Penguji I


(Ir. Pradono Joannes De Deo , Msi.)

Penguji II


(Ir.Leo Patimena, M.Sc)

Penguji III


Ir. Agus Darpono, MT

Di pertahankan dibagian Panitia Penguji Tugas Akhir Jurusan Teknik Geodesi,
Fakultas Teknik Dan Perencanaan, Institut Teknologi Sepuluh Nopember
dan diterima untuk memenuhi sebagian dari syarat – syarat guna memperoleh
gelar Sarjana (S-1) Teknik Geodesi.

Panitia Ujian Tugas Akhir

sekretaris

Ketua

(Ir. Dedy Kurnia Sumantri, M.Ts)
Ketua Jurusan Teknik Geodesi

(Ir. Agustina Nurul H. MTR)
Dekan FTSP

Anggota Penguji :

Penguji II

Penguji I

(Ir. Leo Paiman, M.Sc)

(Ir. Pradono Johannes De Deo, M.Ts)

Penguji III

Ir. Agus Parbono, MT

Keberhasilan ini Kupersembahkan Untuk: Pa"e Alm Sukiyadi Bu"e, Mbak Copret Beserta Keluarga, adikku, Keluarga koe di Solo, Yang senangtiasan mendukung Doa dan Materi. Dan seorang selalu mamperhatikan aku.

Terimakasihku untuk Teman Yang telah membantu dalam Pembuatan Loporan Tugas Akhir ini: Ambar S.kom, Setyo. ST Dan Susy shogun BE 8324 G Yang Rela menemani daku di kala kepanasan dan Kehujanan

Spesial thanks to : Bouya, Ipunk, Mico, Deni, Hans Dan Semua anak 90A, Hani, Eko, Wiwid, Mamsiong, dan teman di waktu ujian Kompre Yang Lebih dari Sepuluh Kali, semoga Kalian Sukses
.....
.....
.....
.....
.....
.....

GALIHAN

KATA PENGANTAR

Segala puji dan syukur kepada Allah S.w.t. yang dengan Ridho-Nya penulis mampu menyelesaikan Tugas Akhir yang merupakan salah satu syarat untuk mencapai gelar sarjana (S-1) di Jurusan Teknik Geodesi ITN Malang

Penulis mengucapkan terimakasih kepada :

1. Bapak Ir. DK. Sunaryo, MS.Tis. selaku ketua Jurusan Teknik Geodesi ITN Malang dan dosen pembimbing I.
2. Bapak Ir.Jasmani,M.Kom ,selaku dosen pembimbing II.
3. Dosen dan Staf jurusan Teknik Geodesi ITN Malang.
4. Rekan – seperjuangan dan senasib (geodesi 99 oye).
5. Semua sahabat saya di jurusan Teknik Geodesi ITN Malang.
6. Konco- konco Wisma Jejaka, yang lucu- lucu

Semoga Laporan ini bermanfaat bagi mahasiswa geodesi dan pembaca pada umumnya.

Malang , April 2005

DAFTAR ISI

Lembar Judul	i
Halaman Persetujuan	ii
Halaman Pengesahan	iii
Lembar Persembahan	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	vii
Daftar Tabel	viii
Daftar Diagram	ix
BAB I	PENDAHULUAN
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	1
1.3. Tujuan Penelitian	2
1.4 Pendekatan Masalah	2
1.5. Batasan Masalah	2
1.6. Manfaat Penelitian	2
1.7. Sistematika Penulisan	3
BAB II	DASAR TEORI
II.1. Pengertian Telaga	4
II.2 Pengertian Sistim Informasi Geografi (SIG)	5
II.3. Pengertian Basis Data	10
II.4 Digitasi	21
II.5. Topologi	21
II.6. Joint Item	21
II.7. Visual Basic. 6.0	22

II.8. Map Object 2.1	30
II.9. Konsep Pembuatan Program Pada Visual Basic 6.0	32

BAB III PELAKSANAAN PENELITIAN

III.1. Lokasi Penelitian	35
III.2. Data Alat Yang di Gunakan	36
III.1.1. Bahan Penelitian	36
III.1.2. Alat Penelitian	36
III.4. Metodologi Penelitian	38
III.5. Pengertian Bagan Alir Penelitian	40
III.6. Pendesainan Tampilan Informasi Telaga	55
III.7 Pembuatan Basis Data Telaga	55
III.8 Pembuatan Aplikasi Pencar5ian	56

BAB IV HASIL DAN PEMBAHASAN

IV.1. Penyajian Peta	57
IV.2. Penyajian Peta Batas Admonistrasi	57
a. Peta batas Kabupaten Gresik	57
b. Peta batas Kecamatan	58
c. Peta Batas Kelurahan	58
d. Peta Jaringan Jalan	59
e. Peta Lokasi Telaga	60
f. Penyajian Pencarian	62
g. Penyajian Help	63
IV.3. Kelemahan Dan Kelebihan Program	65

30	II.8. Map Object 2.1
32	II.9. Konsep Pembuatan Program Pada Visual Basic 6.0

PELAKSANAAN PENELITIAN **BAB III**

32	III.1. Lokasi Penelitian
36	III.2. Data Alat Yang di Gunakan
36	III.1.1. Bahan Penelitian
36	III.1.2. Alat Penelitian
38	III.4. Metodologi Penelitian
40	III.5. Penelitian Badan Air Penelitian
52	III.6. Pendekatan Tampilan Informasi Telaah
55	III.7. Pembuatan Basis Data Telaah
56	III.8. Pembuatan Aplikasi Penelitian

HASIL DAN PEMBAHASAN **BAB IV**

57	IV.1. Penyajian Peta
57	IV.2. Penyajian Peta Batas Administrasi
57	a. Peta batas Kabupaten Gresik
58	b. Peta batas Kecamatan
59	c. Peta Batas Kelurahan
59	d. Peta Jaring-jalan
60	e. Peta Lokasi Telaah
62	f. Penyajian Peta
63	g. Penyajian Help
65	IV.3. Kesimpulan Dan Kelemban Program

BAB V	PENUTUP	
	V.1. Kesimpulan Struktur Organisasi	66
	V. 2. Saran – saran i	67
	Daftar Pustaka	68
	Lampiran List Program	69

DAFTAR GAMBAR

Gambar II.1. SubSistim SIG.....	7
Gambar II.2. Komponen SIG	9
Gambar II.3. Tahapan Eksternal Basis Data.....	11
Gambar II.4. Tahapan Konsepsil Basis Data	11
Gambar II.5. Tahapan Internal Basis Data.....	12
Gambar II.6. Srtuktur Basis Data Hirarki.....	17
Gambar II.7. Srtuktur Basis Data Network.....	18
Gambar II.8. Srtuktur Basis Data Relational	19
Gambar II.9. Tampilan Layar Visual Basic.....	23
Gambar II.10. Menu Form.....	24
Gambar II.11. Menu ToolBook.....	25
Gambar II.12. Menu ProjeK.....	25
Gambar II.13. Menu Propertis.....	26
Gambar II.14. Code Form	26
Gambar II.15. Map Objek System Architecture	31
Gambar II.16. Tampilan Map	31
Gambar II.17. Desain Menu Form	32
Gambar II.18. Penataan Properti	33
Gambar III.1. Diagram alir Penelitian.....	42
Gambar III.1. Diagram alir Program1	53
Gambar VI.1. Peta Batas Administrasi Kabupaten Gresik.....	57
Gambar IV.2. Peta Batas Kecamatan.....	58

DAFTAR GAMBAR

7 Gambar II.1. Substansi SIG

9 Gambar II.2. Komponen SIG

11 Gambar II.3. Tahapan Eksternal Basis Data

11 Gambar II.4. Tahapan Konseptual Basis Data

12 Gambar II.5. Tahapan Internal Basis Data

17 Gambar II.6. Struktur Basis Data Hirarki

18 Gambar II.7. Struktur Basis Data Network

19 Gambar II.8. Struktur Basis Data Relational

23 Gambar II.9. Tampilan Layer Visual Basic

24 Gambar II.10. Menu Form

25 Gambar II.11. Menu ToolBook

25 Gambar II.12. Menu Project

26 Gambar II.13. Menu Properties

26 Gambar II.14. Code Form

31 Gambar II.15. Map Object System Architecture

31 Gambar II.16. Tampilan Map

32 Gambar II.17. Desain Menu Form

33 Gambar II.18. Penataan Properti

42 Gambar III.1. Diagram alir Penelitian

53 Gambar III.1. Diagram alir Program

57 Gambar VI.1. Peta Batas Administrasi Kabupaten Gresik

58 Gambar IV.2. Peta Batas Kecamatan

Gambar IV.3 Peta Batas Kelurahan.....	59
Gambar IV.4 Peta Jaringan Jalan.....	60
Gambar IV.5. Peta Lokasi Telaga.....	61
Gambar IV.6. Data Atribut Telaga.....	61
Gambar IV.7 Penyajian Form Pencarian	63
Gambar IV.8 Penyajian Form Help	64

BAB I

PENDAHULUAN

1.1. Latar Belakang

Air merupakan sumber daya alam yang mengalir (*flowing resources*) yang keberadaannya senantiasa di batasi oleh waktu (musim), ruang (lokasi) jumlah (kwatintas), dan mutu (kwalitas) mengikuti siklus alami. Keberadaan air dalam musim, lokasi, kuantitas, dan kualitas yang tepat selalu di dambakan oleh masyarakat yang luas, dimanapun mereka berada. Guna menunjang berbagai kebutuhan hidup baik untuk keperluan domestik, perkotaan, industri, pertanian, maupun keperluan lainnya. Pengalaman membuktikan bahwa tidak mungkin suatu wilayah dapat mengembangkan dengan baik tanpa ketersediaan sumber daya alam yang memadai.

Dengan berlakunya undang – undang no. 22 tahun 1999, maka terjadi perubahan mendasar dari sistem administrasi Negara, pelaksanaan otonomi daerah akan bertumpu pada Otonomi Kabupaten atau kota. Dengan adanya otonomi daerah tersebut diharapkan adanya kompetisi yang sehat antar daerah untuk berusaha memajukan daerahnya masing – masing. Kompetisi tersebut berupa usaha – usaha untuk menggali potensi – potensi yang ada. Salah satunya pengembangan potensi sumberdaya air.

Telaga di kabupaten Gresik merupakan salah satu asset daerah yang harus di kembangkan perannya dan manfaatnya, potensi telaga di kabupaten Gresik sebagai sumberdaya air dapat berguna bagi masyarakat sekitarnya terutama untuk pengembangan pariwisata, air bersih, pengembangan budidaya ikan darat dan berbagai keperluan lainnya.

1.2. Identifikasi Masalah

1. Masyarakat belum mengetahui secara pasti lokasi – lokasi Telaga di Kabupaten Gresik
2. Belum adanya Sistem Informasi Telaga yang berbasis data spasial dan data non spasial pada daerah Kabupaten Gresik

SECRET

CONFIDENTIAL

CONFIDENTIAL

The following information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization.

CONFIDENTIAL

The following information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization.

CONFIDENTIAL

The following information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization. This information is being furnished to you for your information only. It is not to be disseminated outside your organization.

(A)

(B)

I.7. Sistematika Penulisan

Adapun sistematika penulisan dalam penyusunan laporan ini meliputi :

Bab I.

Dalam bab ini memuat karakteristik, garis besar atau kerangka berpikir dari penelitian ini hingga pada penyusunan laporan akhir yang meliputi : latar belakang, identifikasi masalah, tujuan penelitian, pendekatan masalah, batasan masalah, manfaat penelitian, sistematika penulisan

Bab II.

Dalam bab ini memuat dasar teori yang mendukung dalam penelitian, yang meliputi : Defenisi Telaga, Pengertian – pengertian penting tentang sumber daya air, Defenisi SIG, Defenisi Basis Data, Defenisi Digitasi, Defenisi Visual Basic, Defenisi Map Object, Defenisi Data Base Management System (DBMS), Defenisi Entitas, Defenisi Topologi, Defenisi Join Item, Defenisi Pembuatan Program.

Bab III.

Dalam bab ini memuat alur proses penelitian, yang meliputi : Diskripsi Lokasi, Persiapan Penelitian, Pengumpulan Data, Menyusun Data Base Spasial, Menyusun Data Base Non spasial, penggabungan data base spasial dan non spasial, pembuatan program, dan menjalankan program aplikasi.

Bab IV

Dalam bab ini memuat hasil penelitian dan pembahasan tentang penyajian informasi Telaga di Kabupaten Gresik.

Bab V.

Dalam bab ini memuat kesimpulan dari penelitian yang dilakukan serta beberapa saran-saran. Dalam penulisan Tugas Akhir (TA) tentang penyajian informasi Telaga di Kabupaten Gresik.

1.3. Tujuan Penelitian

Pelaksanaan penelitian ini memiliki tujuan yaitu :

1. Membuat suatu program yang menyajikan informasi mengenai Telaga yang terletak di kabupaten Gresik dengan memanfaatkan bahasa pemrograman Map Object 2.1 dan Microsoft Visual Basic 6.0.
2. Membangun sistim basis data untuk menyajikan informasi sekitar Telaga di kabupaten Gresik

1.4. Pendekatan Masalah

Pendekatan masalah ini dilakukan dengan cara membuat suatu sistem informasi Telaga yang ada di Kabupaten Gresik dengan didukung teknologi informasi yang berbasis data spasial dan non spasial.

1.5. Batasan Masalah

Dalam masalah ini, penelitian ini yaitu pemanfaatan bahasa pemrograman Microsoft Visual Basic 6.0 dan Map Object. Untuk dapat menyajikan sistim informasi Telaga yang meliputi : lokasi Telaga , Nama Telaga , dan fasilitas pendukungnya di Kabupaten Gresik dengan lebih menarik, dan mudah di operasikan oleh pengguna .

1.6 .Manfaat Penelitian

Manfaat yang di dapat dari penelitian ini adalah :

1. Program ini nantinya dapat menyajikan informasi sekitar Telaga. Baik data spasial dan data non spasial .
2. Memberikan suatu pelayanan informasi Telaga kepada instansi Dinas Pengairan , BAPPEDA , Dinas Pariwisata ,dll dengan lebih Informatif

BAB II

DASAR TEORI

II.1 Pengertian Telaga

Telaga adalah tempat menampung air yang di buat secara alami oleh alam , di mana telaga tersebut mempunyai sumber mata air tersendiri . Fungsi Telaga adalah menampung air yang bertujuan untuk kesejahteraan masyarakat di sekitarnya dan meningkatkan ekonomi regional daerah sekitarnya.

Pengembangan potensi telaga di Kabupaten Gresik di arahkan pada beberapa sektor seperti : Pariwisata, Air minum bersih, kebutuhan sehari – hari , dan budidaya perikanan darat.

(Sudjarwadi , 1994)

Pengertian – pengertian penting

Beberapa pengertian penting di bawah ini yang berhubungan dengan pengolahan sumber daya air menurut Ir. Granita Sutadi, Msc dalam sistim sumberdaya air otonomi daerah III modul I

- Sumberdaya air adalah air dan sumber air termasuk potensi dan kekayaan alam bukan hewani yang terkandung di dalamnya baik yang alamiah maupun yang di usahakan oleh manusia
- Tata pengaturan air (TPA) adalah segala usaha untuk mengatur pembinaan sumber daya air , meliputi kepemilikan , penguasaan , pengolahan , penggunaan , penguasaan dan pengamanan atas air beserta sumber – sumbernya , termasuk kekayaan alam bukan alami yang terkandung di dalamnya guna mencapai manfaat yang sebesar – besarnya
- Perkembangansumber daya air adalah segala usaha mengembangkan pemanfaatan air beserta sumber – sumber dengan perencanaan teknisnya yang teratur dan sesuai guna mencapai manfaat yang sebesar – besarnya bagi masyarakat
- pengolahan sumber daya air adalah segala upaya pengolahan atas air dan sumber – sumber air yang di lakukan guna meningkatkan pelayanan dan menjamin ketersediaan air bagi masyarakat yang mencakupi aspek – aspek perlindungan , pengembangan , dan pengendalian

- daerah pengairan sungai (DPS) adalah satu kesatuan tata air yang terbentuk secara alami niah , dimana aliran air hujan di daerah tersebut akan di serap atau mengalir melalui anak - anak sungai dan sungai utama yang selanjutnya bermuara pada danau atau laut .

II.2. Pengertian Sistem Informasi Geografi (SIG)

II.2.1 Definisi Sistem Informasi Geografi (SIG)

Pada dasarnya istilah Sistem Informasi Geografi (SIG) merupakan gabungan dari tiga unsur pokok, yaitu sistem, informasi dan geografi. Istilah geografi merupakan bagian dari spasial keruangan. Sedangkan informasi geografi mengandung informasi mengenai tempat-tempat yang terletak di permukaan bumi, mengenai posisi dimana suatu obyek terletak dipermukaan bumi dan informasi mengenai keterangan-keterangan (*atribut*) yang terdapat dipermukaan bumi yang posisinya diberikan dan diberitahu.

Dengan memperhatikan pengertian di atas, maka Sistem Informasi Geografi (SIG) merupakan suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan obyek-obyek yang terdapat di permukaan bumi. Adapun definisi yang lainnya adalah sebagai berikut :

1. Sistem Informasi Geografi (SIG) adalah sistem komputer yang digunakan untuk memasukan (*capturing*), menyimpan, memeriksa, mengintegrasikan dan menampilkan data-data yang berhubungan dengan posisi-posisi di permukaan bumi. [Rice20]
2. Sistem Informasi Geografi (SIG) adalah kombinasi perangkat keras dan perangkat lunak komputer yang memungkinkan untuk mengelola (*manage*), menganalisa, memetakan informasi spasial berikut data atributnya (*data deskriptif*) dengan akurasi kartografi. [Basic20]
3. Sistem Informasi Geografi (SIG) adalah sistem yang terdiri dari perangkat keras, perangkat lunak, data, manusia (*brainware*), organisasi dan lembaga yang digunakan untuk mengumpulkan, menyimpan, menganalisa dan menyebarkan informasi-informasi mengenai daerah-daerah di permukaan bumi. [Chrisman97]

4. Sistem Informasi Geografi (SIG) adalah teknologi informasi yang dapat menganalisa, menyimpan dan menampilkan data spasial maupun data non spasial. SIG mengkombinasikan kekuatan perangkat lunak basis data relasional dan paket perangkat lunak CAD. [Guo20]
5. Sistem Informasi Geografi (SIG) adalah suatu fasilitas untuk mempersiapkan, mempresentasikan, dan menginterpretasikan fakta-fakta (kenyataan) yang terdapat dipermukaan bumi (definisi umum). Untuk definisi yang lebih sempit adalah konfigurasi perangkat keras dan perangkat lunak komputer yang secara khusus dirancang untuk proses-proses akuisisi, pengelolaan dan penggunaan kartografi. [Tomlin90]

II.2.2 Subsistem SIG

Jika definisi-definisi di atas diperhatikan maka, Sistem Informasi Geografi (SIG) dapat diuraikan menjadi beberapa subsistem, yaitu :

1. Data input

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini pula yang bertanggungjawab dalam mengkonversi atau mentransformasikan format data-data aslinya kedalam format yang dapat digunakan oleh SIG.

2. Data output

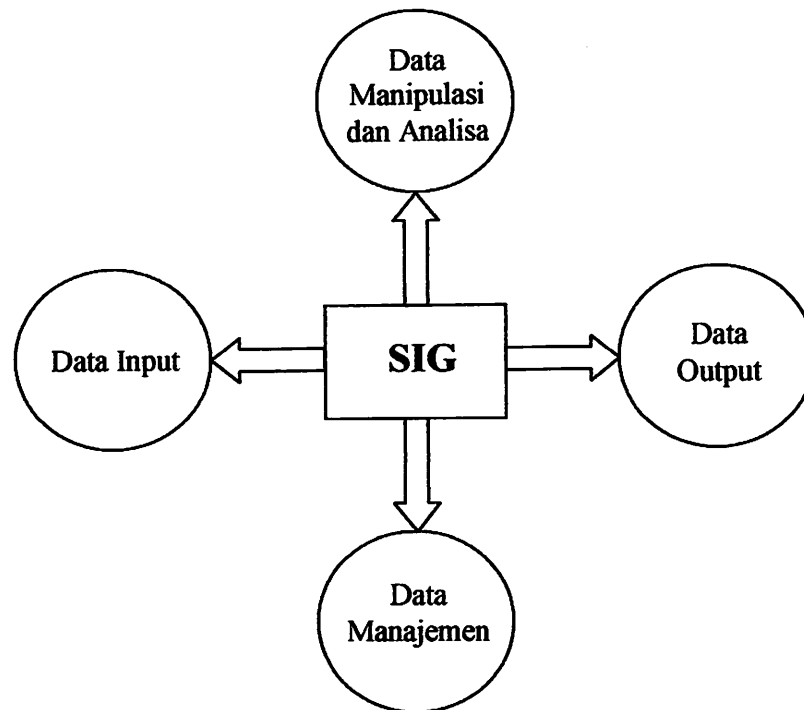
Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian *basisdata* baik dalam bentuk *softcopy* maupun bentuk *hardcopy* seperti tabel, grafik, peta dan lain-lain.

3. Data manajemen

Subsistem ini mengorganisasikan baik data spasial maupun atribut ke dalam sebuah *basisdata* sedemikian rupa sehingga mudah dipanggil, di-*update* dan di-*edit*.

4. Data manipulasi dan analisa

Subsistem ini menampilkan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.



Gambar II.1 Subsistem SIG

II.2.2 Komponen SIG

SIG merupakan sistem kompleks yang biasanya terintegrasi dengan lingkungan sistem-sistem komputer yang lain ditingkat fungsional dan jaringan. Sistem SIG terdiri dari beberapa komponen, yang antara lain sebagai berikut : [Gistut94]

1. Perangkat keras

Pada saat ini SIG tersedia untuk berbagai *platform* perangkat keras mulai dari *PC desktop*, *workstation*, hingga *multiuser* orang secara bersamaan dalam jaringan komputer yang luas, berkemampuan tinggi, memiliki ruang penyimpanan (*hard disk*) yang besar dan mempunyai kapasitas memori RAM yang besar. Walaupun demikian fungsionalitas SIG tidak terikat secara ketat terhadap karakteristik-karakteristik fisik perangkat keras ini sehingga keterbatasan memori pada PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan SIG adalah sebagai berikut :

a CPU (*Central Processor Unit*)

Perangkat ini merupakan bagian dari sistem komputer yang bertindak sebagai tempat untuk pemrosesan semua instruksi-instruksi dan program

(*Processor*). Selain itu, CPU juga mengendalikan seluruh operasi yang ada di dalam lingkungan sistem komputer yang bersangkutan.

b RAM

Perangkat ini digunakan oleh CPU untuk menyimpan (sementara) semua data dan program yang dimasukkan melalui input device baik itu untuk jangka panjang maupun pendek.

c Storage

Perangkat ini merupakan tempat penyimpanan data secara permanen atau semi permanen (*tempory*). Contoh dari perangkat ini adalah *Harddisk*, *Disket*, *CD-ROM* dan pita *magnetis*.

d Input Device

Perangkat ini merupakan peralatan-peralatan yang digunakan untuk memasukan data kedalam SIG. Yang termasuk dalam perangkat ini adalah *keyboard*, *mouse*, *digitizer*, *scanner*, kamera *digital* dan lain sebagainya.

e Output Device

Perangkat ini merupakan peralatan-peralatan yang digunakan untuk merepresentasikan data dan informasi SIG. Yang termasuk perangkat ini adalah layar *monitor*, *printer*, *plotter* dan sebagainya.

f Pheripheral lainnya

Perangkat pelengkap ini merupakan bagian dari sistem komputer SIG yang belum termasuk kedalam perangkat-perangkat yang telah disebutkan di atas. Untuk SIG yang kecil dan sederhana mungkin tidak diperlukan, tetapi untuk SIG dalam skala besar apalagi menggunakan jaringan dan dapat dipresentasikan di jaringan internet (*Web*), maka diperlukan kabel-kabel jaringan, *Modem*, *ISP*, *Router*, *card jaringan/ethernet*, CPU khusus untuk *clients* dan *server* dan sebagainya.

2. Perangkat lunak

Bila dibanding dari sisi lain, SIG merupakan sistem perangkat lunak yang tersusun secara modular dimana *basisdata* memegang peranan penting. Setiap subsistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul. Hingga tidak mengherankan jika ada perangkat

SIG yang terdiri dari ratusan modul program (*.exe) yang masing-masing dapat dieksekusi sendiri. Adapun perangkat lunak yang akan digunakan dalam penelitian ini adalah sebagai berikut :

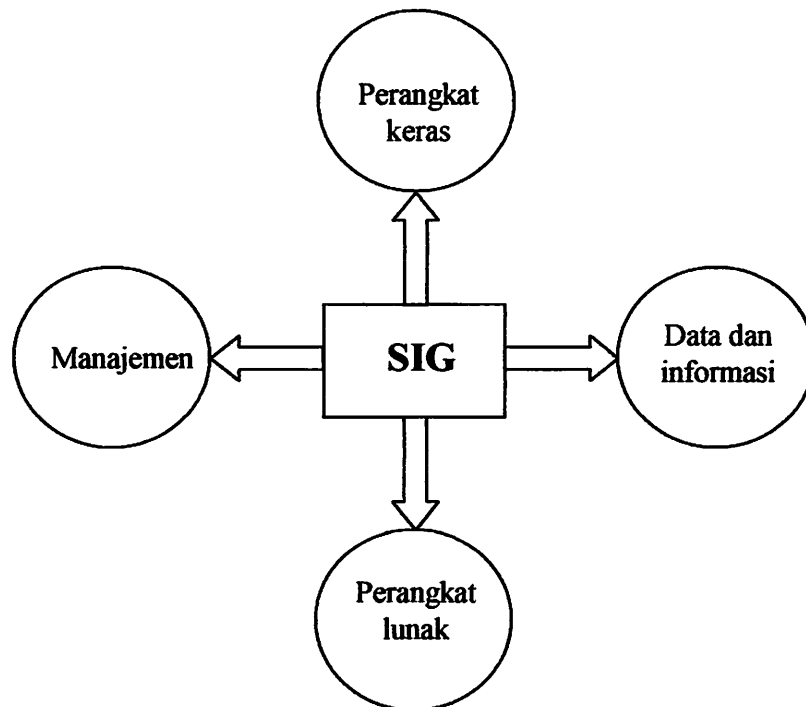
- a Auto Cad 2004
- b Map Object 2.1
- c Visual Basic 6.0
- d Microsoft Access 2000

3. Data dan informasi geografi

SIG dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara tidak langsung dengan cara *import*-nya dari perangkat-perangkat lunak SIG yang lain maupun secara langsung dengan cara mendigitasi data spasialnya dari peta dan memasukan data atributnya dari tabel-tabel dan laporan dengan menggunakan *keyboard*.

4. Manajemen

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



Gambar II.2 Komponen SIG

II.3. Basisdata

II.3.1 Pengertian Basisdata

Basisdata merupakan kumpulan data *non-redundant* yang dapat digunakan bersama (*shared*) oleh sistem-sistem aplikasi yang berbeda. Dengan kata lain, basis data adalah kumpulan data (*file*) *non-redundant* yang saling terkait satu sama lainnya (dinyatakan oleh atribut-atribut kunci dari tabel-tabelnya / struktur data dan relasi-relasi) dalam membentuk bangunan informasi yang penting (*enterprise*). [E.Prahasta, 2001]

Basis data dapat diartikan sebagai kumpulan data tentang suatu benda atau kejadian yang saling berhubungan satu sama lain. Sedangkan data merupakan fakta yang mewakili suatu obyek seperti manusia, hewan, peristiwa, konsep, keadaan dan sebagainya, yang dapat dicatat dan mempunyai arti yang implisit. Data yang dicatat atau direkam dalam bentuk angka, huruf, simbol, gambar, bunyi, atau kombinasinya. [Waljiyanto, 2000].

Sebagai contoh, terdapat daftar nama, nomor telepon, dan alamat orang-orang yang menjadi anggota suatu organisasi. Data tersebut dicatat dalam buku daftar anggota, atau disimpan dalam disket menggunakan komputer personal dan perangkat lunak seperti *Dbase*, *FoxBase*, *MS Acces*, atau *Excel*. Kumpulan data dengan arti yang *implisit* tersebut dinamakan *basisdata*. [Waljiyanto, 2000]

II.3.2 Sistem Manajemen Basisdata

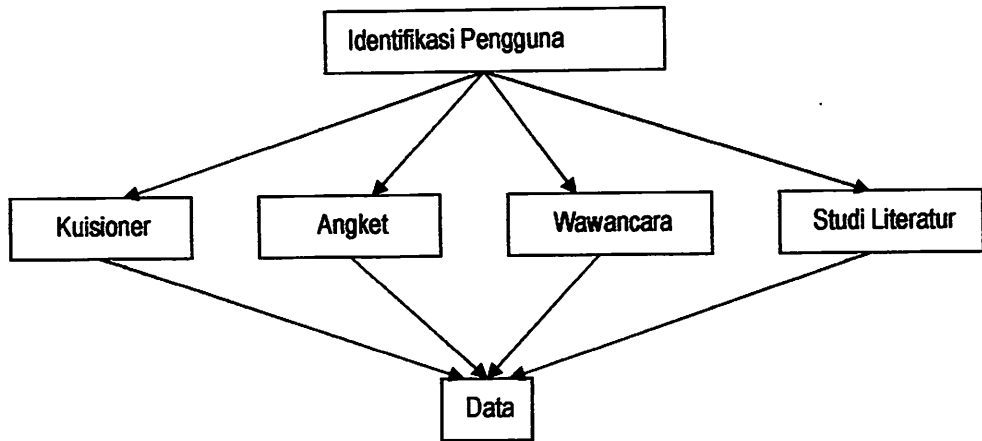
Menurut pustaka [Korth, 1991], sistem manajemen *basisdata* adalah kumpulan (gabungan) dari data yang saling berelasi (yang biasanya dirujuk sebagai suatu *basisdata*) dengan sekumpulan program-program yang mengakses data-data tersebut. *Basisdata Management System* atau Sistem Manajemen Basis data adalah tempat penyimpanan data beserta *users interface* yang dipersiapkan untuk memanipulasi dan administrasi *basisdata*. [Ade, 2000]

II.3.3 Tata Cara Perancangan BasisData

Terdapat tiga tahapan dalam merancang suatu *basisdata*, yaitu :

1. Tahap eksternal, yaitu tahap mengidentifikasi kebutuhan pengguna.

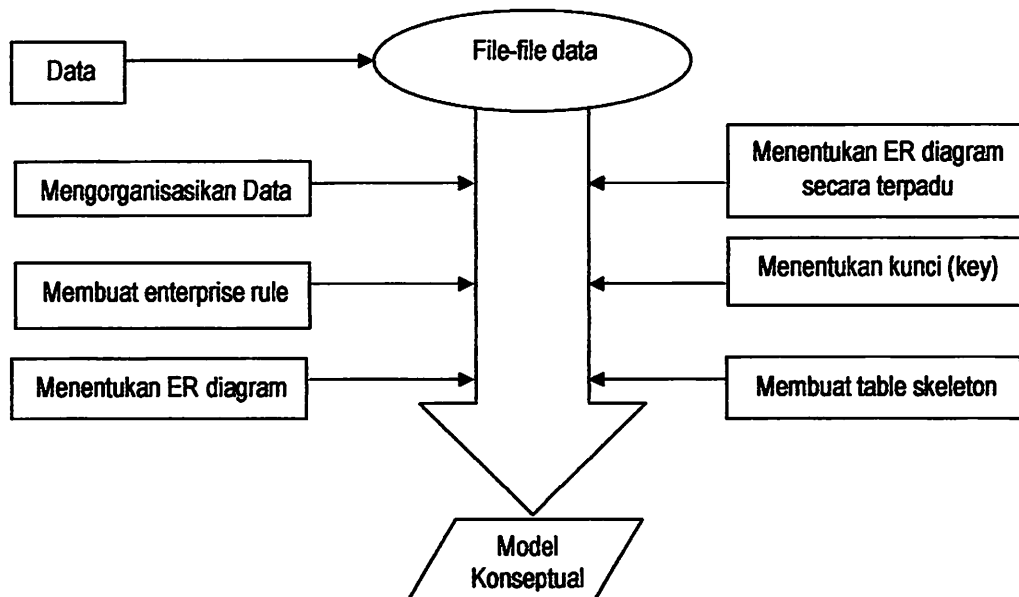
Diagram tahap eksternal adalah sebagai berikut :



Gambar II.3 Tahapan Eksternal Basisdata

2. Tahap konseptual, yaitu tahap mengorganisasi data, memilih, mengelompokkan, menyederhanakan data, menetapkan *enterprise rule* (ER) diagram, menetapkan kunci dan membuat *table skeleton* secara terstruktur.

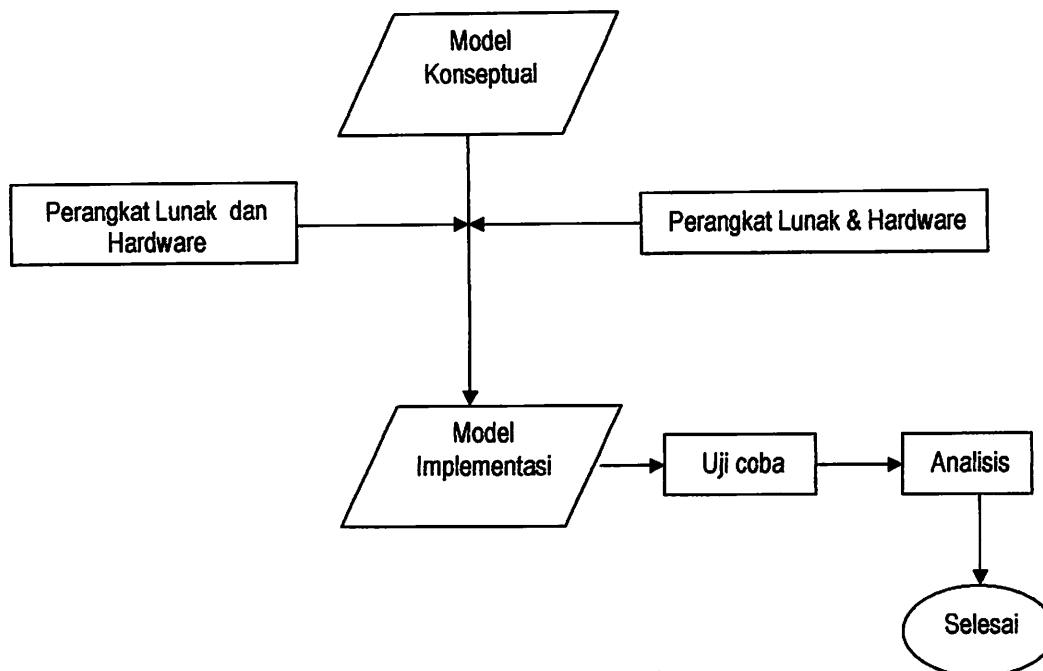
Diagram model konseptual adalah sebagai berikut :



Gambar II.4 Tahapan Konseptual Basisdata

3. Tahap Internal, yaitu tahap mengimplementasi tabel yang telah dirancang kedalam perangkat lunak, kemudian dilakukan uji coba.

Gambar tahap internal adalah sebagai berikut :



Gambar II.5 Tahapan Internal Basisdata

II.3.4 Keuntungan dan Kekurangan Basisdata

Bila dibandingkan dengan sistem pemrosesan *file* yang didukung oleh sistem operasi konvensional, maka penggunaan *basisdata* akan memperoleh keuntungan-keuntungan seperti berikut:

1. Pemusatan kontrol data

Dengan satu DBMS dibawah kontrol satu orang atau kelompok dapat menjamin terpeliharanya standar kualitas data dan keamanan pembatasan pemakaian. Disamping itu adanya konflik dalam persyaratan pemakaian data dapat dinetralkan, serta integritas data dapat terjaga.

2. Pemakaian data bersama

Dengan menggunakan DBMS, informasi yang ada dalam *basisdata* dapat digunakan secara efektif oleh beberapa pemakai dengan kontrol data yang terjaga. Fasilitas penanganan data dalam DBMS juga memberi kemungkinan untuk mengembangkan program aplikasi yang baru (dengan menggunakan fasilitas *basisdata* yang ada).

3. Data yang bebas

Program aplikasi terpisah atau bebas dengan bentuk secara fisik data disimpan dalam komputer.

4. Kemudahan dalam pembuatan program aplikasi baru

Program aplikasi yang baru dan pencarian *basisdata* yang tunggal akan lebih mudah jika menggunakan fasilitas yang ada pada DBMS.

5. Pemakaian secara langsung

Sistem *basisdata* saat ini biasanya menyediakan jendela pemakai, sehingga pemakai (bukan pemogram) dapat melakukan analisis data yang rumit sekalipun. Pada saat yang sama sistem *basisdata* berperan sebagai pengontrol penggunaan dan operasi *basisdata* untuk menjaga konsistensi dan adanya perlindungan pada integritas *basisdata*.

6. Data yang berlebihan dapat dikontrol

Dalam pemrosesan berkas untuk tidap aplikasi menggunakan berkas-berkas yang terpisah. Sehingga tidak jarang akan menghasilkan data yang rangkap (*redundant*). Hal demikian menyebabkan pemborosan biaya. Sedangkan tujuan penggunaan *basisdata* salah satunya adalah apabila dilakukan penambahan data pada salah satu berkas, maka berkas yang lain dengan sendirinya juga ikut diperbaharui. DBMS dapat digunakan untuk menurunkan tingkat *redundancy* dan pengelolaan proses pembaruan data.

7. Pandangan pemakai (*users views*)

DBMS dapat memberikan kemudahan untuk membuat dan memelihara jendela pemakai (*user interface*) sesuai dengan pandangan pemakai terhadap *basisdata*. Sehingga ada kemungkinan *basisdata* yang diakses sama, tetapi jendela pemakai akan berbeda disesuaikan dengan pemahaman tiap pemakai terhadap *basisdata* menurut kebutuhan.

DBMS tidak selalu memberikan keuntungan untuk semua aplikasi pendekatan *basisdata*. Beberapa kelemahannya adalah sebagai berikut:

1. Biaya

Biaya yang digunakan untuk mendapatkan perangkat lunak dan perangkat keras yang tepat sangatlah mahal. Dan paling tidak harus ada untuk

memperoleh kedua perangkat tersebut, termasuk biaya untuk pemeliharaannya (*maintenance cost*) dan penyediaan sumberdaya manusia untuk mengelola *basisdata* tersebut.

2. Sangat kompleks

Sistem *basisdata* lebih kompleks dibanding proses berkas. Menurut teori, semakin kompleks suatu sistem akan semakin mudah terjadi kesalahan dan semakin sulit dalam pemeliharaan data. Dalam prakteknya, DBMS yang baik mampu membuat *back up* secara efektif termasuk pemeliharaan data.

3. Resiko data yang terpusat

Menurut teori, data yang terpusat dalam satu lokasi dengan selalu menjaga adanya data rangkap yang kecil, akan terjadi resiko kehilangan data selama proses aplikasi. Namun demikian, biasanya DBMS mampu menjaga agar resiko ini sangat kecil.

II.3.5 Komponen Sistem Basis Data

Dalam sistem *basisdata* komponen-komponen pokok dapat dibagi menjadi lima bagian, yaitu:

1. Data

Data dalam *basisdata* mempunyai sifat terpadu (*integrated*) dan berbagi (*shared*).

- a. Sifat terpadu, berarti bahwa berkas-berkas data yang ada basis data saling terkait, tetapi kelebihan data tidak akan terjadi atau hanya terjadi sedikit sekali.
- b. Sifat berbagi data, berarti bahwa data dapat dipakai oleh sejumlah pengguna dalam waktu yang bersamaan. Sifat ini biasa terdapat pada sistem *multiuser* (kebalikan dari sistem *singleuser*, yakni suatu sistem yang hanya memungkinkan satu orang yang bisa mengakses suatu data pada suatu waktu).

2. Perangkat lunak

Perangkat lunak dalam DBMS berkedudukan antara *basisdata* (data yang disimpan dalam *hard disk*) dan pengguna. Perangkat lunak inilah yang berperan

melayani permintaan-permintaan pengguna, dimana perangkat ini mempunyai kemampuan utama sebagai berikut:

- a. Kemampuan memasukkan data.
- b. Kemampuan memanipulasi data.
- c. Kemampuan menyimpan data.
- d. Kemampuan menganalisa data.
- e. Kemampuan mengolah data.

3. Perangkat keras

Perangkat keras merupakan peralatan yang diperlukan dalam pemrosesan dan juga menyimpan *basisdata*, yang terdiri atas:

- a. Komputer dengan kapasitas dan kemampuan yang disesuaikan dengan beban.
- b. Alat pemasukan data (*Digitizer, Scanner, dsb*).
- c. Alat pengeluaran data (*Plotter, Printer, Monitor, dsb*).

4. Pengguna

Pengguna dapat diklasifikasikan menjadi tiga kategori diantaranya:

- a. Pengguna akhir, orang yang mengoperasikan program aplikasi yang dibuat oleh pemrograman aplikasi.
- b. Pemrograman aplikasi, orang yang membuat program aplikasi yang menggunakan *basisdata*. Program aplikasi yang dibuat tentu saja sesuai dengan kebutuhan pengguna.
- c. Administrator *basisdata* (*DBA/Basisdata Administrator*), orang yang bertanggung jawab terhadap pengolahan *basisdata*. Secara lebih detail, tugas DBA adalah sebagai berikut:
 - Mendefinisikan *basisdata*.
 - DBA menentukan isi *basisdata*.
 - Menentukan sekuritas *basisdata*.

Setiap pengguna diberi hak akses terhadap basis data secara tersendiri. Tidak semua pengguna bisa menggunakan data yang bersifat sensitif, penentuan hak akses disesuaikan dengan wewenang pengguna dalam organisasi.

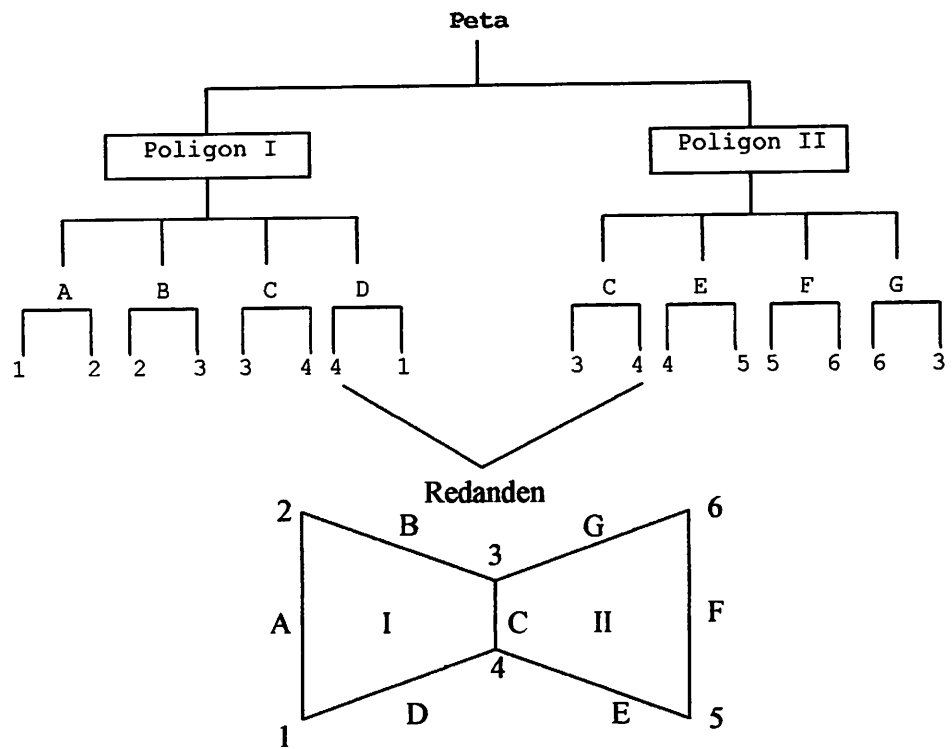
5. Sumber Daya Manusia

Sumber daya manusia merupakan orang yang dapat menjalankan sistem *basisdata* secara maksimal dengan mengembangkan aplikasi sesuai dengan bidang kerja masing-masing, secara global kelima komponen di atas tersebut dapat diminimalkan menjadi tiga komponen yang lebih kompak dalam penggunaannya. Komponen-komponen tersebut meliputi data, sistem (perangkat keras dan lunak) dan sumber daya manusia (pelaksana). Banyak yang menyebut bahwa peranan dalam komponen sumber daya manusia (pelaksana) dan prosedur sebagai *institutional framework* yang memegang peranan yang sangat penting dalam pengoperasian sistem *basisdata* tersebut.

II.3.6 Struktur Basis Data

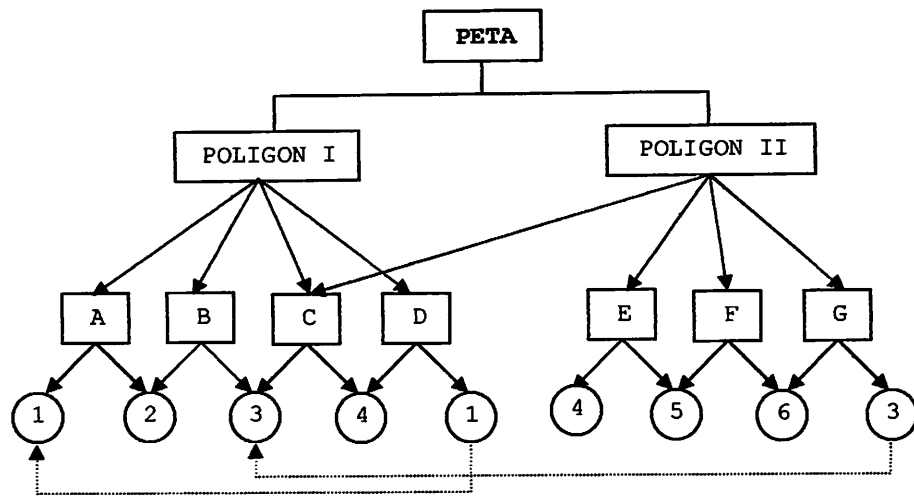
Sebelum membicarakan penyusunan suatu sistem *basisdata*, maka yang perlu ditinjau dalam pembuatan sistem *basisdata* adalah sebagai berikut:

1. Struktur *Basisdata Hirarki*, dibuat pada tahun 1970 – 1980 mempunyai beberapa karakteristik diantaranya :
 - a. Struktur *basisdata*-nya seperti pohon (satu anak hanya mempunyai satu orang tua).
 - b. Sangat cepat dan mudah dalam mendapatkan suatu data.
 - c. Pembentukan kembali struktur dari sebuah *basisdata* adalah kompleks.
 - d. Tidak fleksibel didalam *query* data (pola hanya keatas dan kebawah), tidak bisa akses perpotongan dari kumpulan data).
 - e. Hubungan data *one to one* (1:1) atau *one to many* (1:M) dapat dikerjakan.
 - f. Untuk mengambil data *many to many* (M:N) yang *redunden* harus ada.



Gambar II.6 Struktur Basisdata Hirarki

2. Struktur Basis data *Network*, dibuat pada tahun 1970 – 1980 mempunyai beberapa karakteristik diantaranya:
 - a. Struktur *basisdata*-nya berupa pohon (seorang anak dapat mempunyai lebih dari satu orang tua).
 - b. Semua basis datanya *one to one* (1:1), *one to many* (1:M), *many to many* (M:N) dapat dikuasai..
 - c. Tidak ada data *redunden* tetapi dibutuhkan banyak *pointer* (perpotongan kumpulan data).
 - d. Mudah dan cepat dalam mendapatkan sebuah data.
 - e. Pembentukan kembali struktur dari *basisdata* adalah kompleks.
 - f. Lebih fleksibel didalam *query* data, tetapi lebih sedikit kompleks.

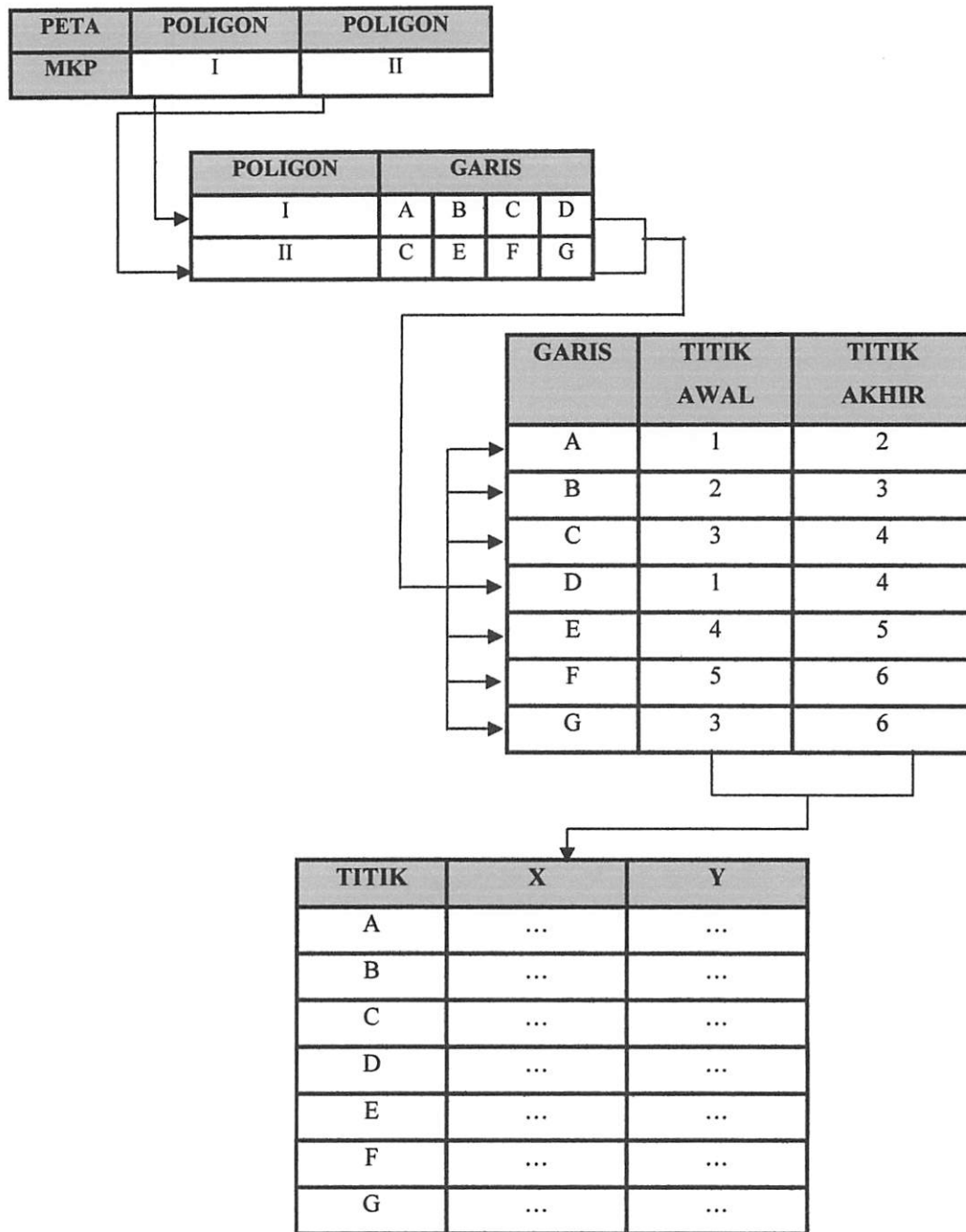


Gambar II.7 Struktur Basisdata Network

3. Struktur *Basisdata Relational*, merupakan model yang paling sederhana, sehingga mudah digunakan dan dipahami oleh pengguna serta yang paling populer pada saat ini. Model ini menggunakan sekumpulan tabel berdimensi dua (yang disebut relasi atau tabel), dengan masing-masing relasi tersusun atas baris dan atribut.

Beberapa karakteristik *basisdata relational* diantaranya :

- a. Penggunaan desain metodologi.
- b. Struktur *basisdata*-nya yang *simple* dan sederhana (semua data disimpan didalam dua dimensional tabel).
- c. Semua basis datanya *one to one* (1:1), *one to many* (1:M), *many to many* (M:N) dapat dikuasai.
- d. Tidak ada data *redunden* (normalisasi tabel).
- e. Pembentukan kembali struktur *basisdata*-nya adalah mudah.



Gambar II.8 Struktur Basis data Relational

4. Struktur *Basisdata Object Oriented*, mempunyai beberapa karakteristik, diantaranya:
 - a. Sangat cocok untuk suatu persoalan atau situasi yang sangat kompleks.
 - b. Teknologi masa depan yang menjanjikan.

c. Masih sedikit tersedia dipasaran.

II.3.7 Model Data Sistem Basis data

Dalam model data konseptual digunakan konsep entitas (*entity*), atribut (*attribute*), dan hubungan (*relationship*). Pengertian ketiga istilah tersebut masing-masing adalah :

1. Entitas (*Entity*), merupakan sebuah obyek atau konsep yang dikenal oleh *file* sebagai sesuatu yang dapat muncul *independent*. Bisa jadi diidentifikasi yang unik dan penggambaran data yang disimpan. Pada model relasional, entitas akan menjadi tabel.
2. Atribut (*attribute*), merupakan keterangan-keterangan yang dimiliki oleh suatu *entity*.
3. Hubungan (*relationship*), bagian dari bumi yang sedang digambarkan atau dimodel basis data, bisa seluruh organisasi atau bagian tertentu.

II.3.8 Konsep Hubungan Antar Entitas (E-R)

Hubungan antara dua *file* atau dua tabel dapat dikategorikan menjadi tiga macam kemungkinan, yaitu:

1. Hubungan satu kesatu (1 : 1), artinya nilai *entity* berhubungan dengan satu nilai *entity* yang lainnya, aturannya adalah sebagai berikut:
 - a. Bila kedua *entity*nya *obligatory*, maka hanya dibuat satu tabel.
 - b. Bila satu *entity* *obligatory* dan yang satu lagi *non-obligatory*, maka harus dibuat 2 tabel masing-masing untuk *entity* tersebut. Kemudian tempatkan *identifier* dari *entity non-obligatory* ke *entity obligatory*.
 - c. Bila kedua *entity*nya *non-obligatory*, maka harus dibuat 3 tabel. Dua tabel untuk masing-masing *entity* tersebut dan satu tabel untuk hubungan kedua *entity* tersebut.
2. Hubungan satu ke banyak (1 : N), artinya satu nilai *entity* berhubungan dengan beberapa nilai *entity* yang lainnya, aturannya adalah sebagai berikut :
 - a. Bila kedua *entity*nya *obligatory*, maka hanya dibuat 2 tabel, masing-masing untuk *entity* tersebut. Kemudian tempatkan *identifier* dari *entity* derajat 1 ke *entity* derajat N.

- b. Bila *entity* derajat banyak *non-obligatory*, maka harus dibuat 3 tabel. Dua tabel untuk masing-masing *entity* tersebut dan satu tabel untuk hubungan kedua *entity* tersebut.
3. Hubungan banyak ke banyak (M : N), artinya beberapa nilai *entity* berhubungan dengan beberapa nilai *entity* yang lainnya. Aturannya adalah sebagai berikut :
 - a. Bila kedua *entity*nya *non-obligatory*, maka hanya dibuat 3 tabel. Dua tabel untuk masing-masing *entity* tersebut dan satu tabel untuk hubungan.
 - b. *Entity Relationship* (ER) diagramnya harus diuraikan dari derajat hubungan (M:N) menjadi derajat hubungan (1:N) dan (N:1).

II.4 Digitasi

Digitasi adalah pekerjaan pengalihan data dari data analog ke data digital, dalam hal ini data umumnya berupa data grafis. *Digitizer* adalah peralatan masukan dari suatu jaringan sistem (komputer) yang menyalurkan lokasi (posisi) X dan Y dari kursor.

II.5 Topologi

Topologi merupakan hubungan yang digunakan untuk mempresentasikan keterkaitan antara *feature-feature* di dalam suatu *coverege*. Topologi dapat dibangun secara otomatis apabila terdapat peta digital. Dengan bantuan ArcInfo topologi dapat dibangun dengan perintah *build* dan *clean*. Semua jenis *feature* dari peta digital yang berupa poligon, garis dan titik. *Build* merupakan perintah untuk pembuatan topologi dan membangun tabel *atribut* dari setiap *feature*. Proses *clean* digunakan untuk membuat interaksi secara otomatis pada setiap pertemuan antar garis pada peta digital.

II.6 Joint Item

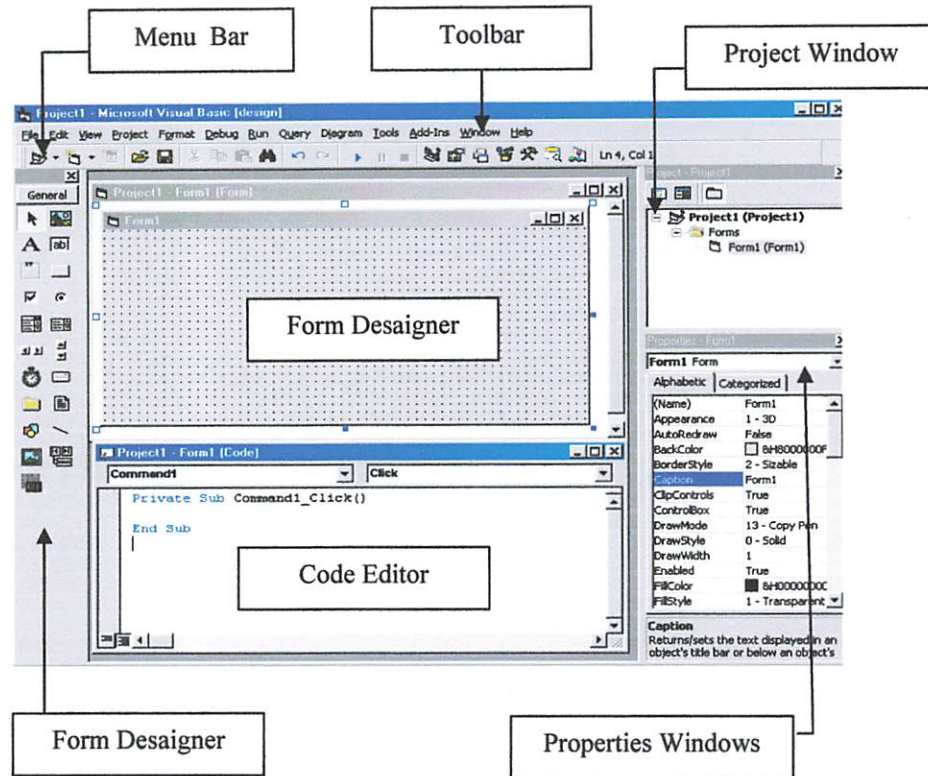
Joint item merupakan proses penggabungan data spasial dan data non spasial. Definisi *item* dan nilai dari kedua *file* tersebut digabungkan untuk membuat *file output*. *Joint item* sering digunakan untuk menambah informasi data non spasial yang ada pada data informasi yang terpisah ke tabel data non spasial *coverege*. Kumpulan dari *item* yang terdapat pada *file output* dibentuk melalui definisi *item* dari *file* yang digabungkan dan yang akan tidak ada pada *file input* yang akan digabungkan. Apabila terdapat *item* yang sama pada kedua *file*, maka definisi dan nilai *file input* yang lebih diutamakan.

II.7 Visual Basic 6.0

MS-Visual Basic adalah bahasa pemrograman yang bekerja dalam lingkup MS-Windows. MS-Visual Basic 6.0 dapat memanfaatkan kemampuan MS-Windows secara optimal. Kemampuannya dapat dipakai untuk merancang program aplikasi yang berpenampilan seperti program aplikasi lainnya berbasis MS-Windows. MS-Visual Basic 6.0 dapat memanfaatkan hampir semua kemudahan dan kecanggihan yang disediakan *system* operasi MS-Windows. Kemampuan MS-Visual Basic 6.0 secara umum adalah menyediakan komponen-komponen yang memungkinkan membuat program aplikasi yang sesuai dengan tampilan dan cara kerja MS-Windows. [M. Agus J. Alam, 1999]

Sistem pemrograman Visual Basic merupakan suatu bahasa pemrograman yang mengkombinasikan kemampuan bahasa *basic* dan piranti desain visual, bahasa ini menyediakan kesederhanaan dan kemudahan pakai tanpa mengorbankan kinerja atau fasilitas grafik yang menyebabkan *windows* menjadi lingkungan kerja yang menyenangkan. Visual Basic adalah suatu bahasa komputer yang mendukung pemrograman *event-driven* yaitu gaya pemrograman yang sangat cocok untuk antar muka pemakai grafis. Pada pemrograman *event-driven* sasarannya adalah menjadikan pemakai sebagai orang yang berkuasa. (Nelson, 1996)

Pemrograman dalam Visual Basic ini berorientasi pada obyek yang didefinisikan sebagai data-data dan sekumpulan metode yang dijalankan. Jenis operasi terhadap obyek tergantung dari jenis komponen yang dipresentasikan oleh obyek tersebut. Obyek yang merepresentasikan area dan posisi misalnya, memiliki operasi yang akan menjawab pertanyaan posisi obyek dalam area. Obyek bersifat *private* artinya hanya prosedur dalam obyek tersebut yang dapat melakukan akses operasi terhadap obyek tersebut.



Gambar II.9 Tampilan Layar Visual Basic

Keterangan tampilan layar Visual Basic adalah sebagai berikut :

1. Menu Bar

Menu bar memiliki 13 menu seperti *file*, *edit*, *view* dan lain-lain yang memiliki akses ke sub menu.

2. Toolbar

Toolbar adalah jalan pintas untuk mengakses menu dengan jalan mengeklik *toolbar* yang bersesuaian dengan suatu menu.

3. Toolbox

Toolbox berfungsi sebagai *tool* menempatkan berbagai macam tombol control di atas *form designer*.

4. Project Windows

Project windows adalah tempat memonitor jumlah *form*, *module*, *class* yang digunakan dalam suatu atau beberapa *project*.

5. Properties Windows

Properties windows adalah tempat untuk mengedit karakteristik dari sebuah obyek yang terdapat di dalam *project*, misalnya ukuran warna, dan *text* dari suatu tombol.

6. Form Designer

Form designer adalah tempat untuk merancang *interface* dari suatu aplikasi.

7. Code Editor

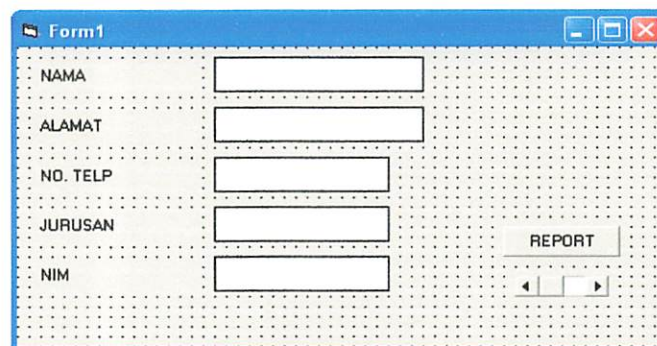
Code editor adalah tempat untuk mengetik perintah dengan menggunakan kode-kode program.

II.7.1 Elemen Utama Visual Basic

Didalam *Visual Basic*, kita bekerja dengan beberapa jendela terbuka hampir setiap waktu. Adapun kelima jendela utama di lingkungan *Visual Basic* adalah sebagai berikut :

1 Form,

Berisi latar belakang program *windows* yang ditulis sehingga dapat menggambar atau meletakkan *item* itu pada *form*.



Gambar II.10 Menu Form

Project windows adalah tempat monitor jendela form windows class yang digunakan dalam suatu proyek project.

5. Properties Windows

Properties windows adalah tempat untuk mengedit karakteristik dari sebuah objek yang terdapat di dalam proyek, misalnya ukuran warna dan text dari suatu tombol.

6. Form Designer

Form designer adalah tempat untuk merancang window dan suatu aplikasi.

7. Code Editor

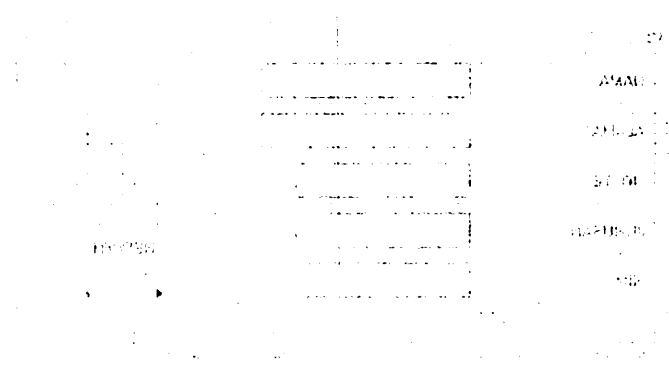
Code editor adalah tempat untuk mengedit perintah dengan menggunakan kode-kode program.

11.7.1 Elemen Utama Visual Basic

Di dalam Visual Basic, kita bekerja dengan beberapa jendela terbuka hampir setiap waktu. Adapun ketiga jendela utama di lingkungan Visual Basic adalah sebagai berikut :

1. Form

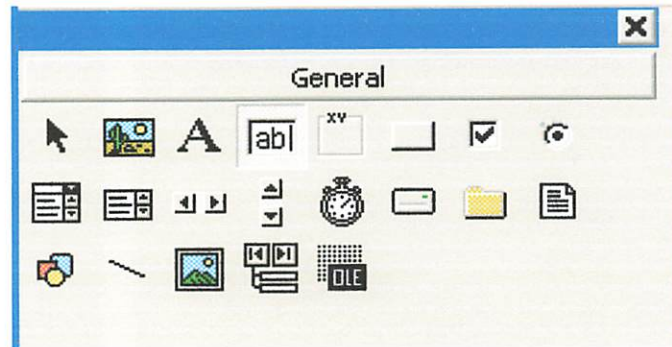
Berisi latar belakang program windows yang ditulis sehingga dapat menggambar atau merancang window pada form.



Gambar 11.10 Form Visual Basic

2. Toolbox

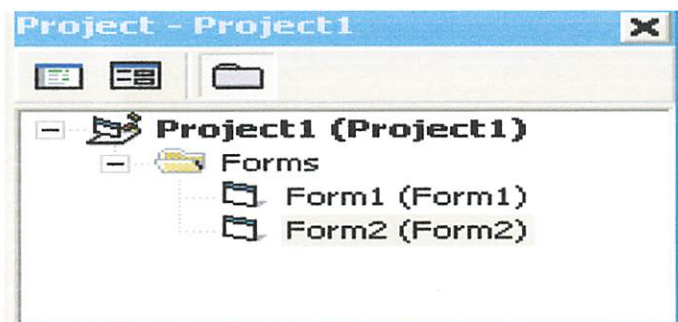
Toolbox berfungsi sebagai tool untuk menempatkan berbagai macam tombol control di atas *form designer*.



Gambar II.11 Menu Toolbox

3. Project

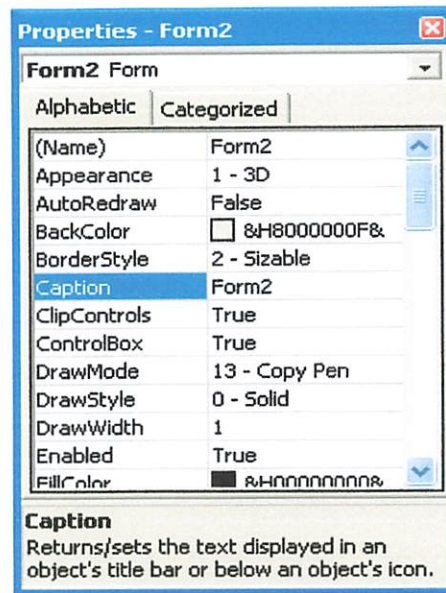
Program *Visual Basic Window* sering berisi beberapa jenis *file* berbeda yang semuanya bekerja dalam satu kesatuan membentuk program tunggal yang dijalankan. Jendela *project* berisi daftar semua *file* aktif yang digunakan secara sederhana dan *file-file* tersebut tempatnya terpisah di dalam *hard disk*.



4. Propertis

Gambar II.12 File Project

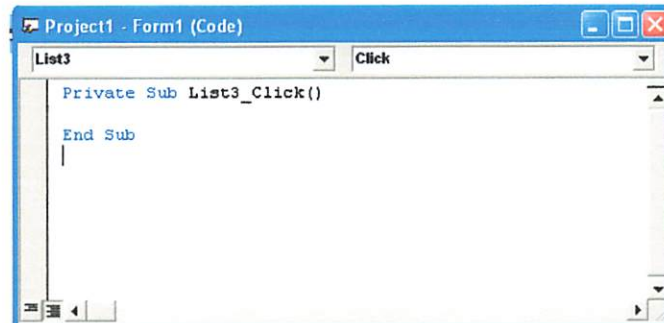
Jendela ini menguraikan setiap elemen individual pada aplikasi kita. Bila kita ingin melihat atau meng-*edit property* dari berbagai *form* atau *control* kita dapat melihat serta meng-*edit* propertinya disatu jendela *property*.



Gambar II.13. Menu Propertis

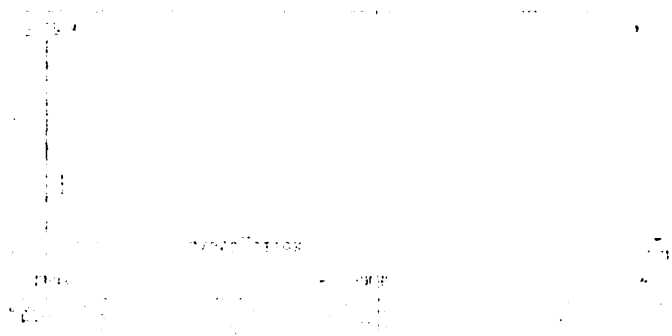
5. Code

Tidak seperti kebanyakan bahasa pemrograman yang lain, kita tidak harus memilih banyak kode jika kita mengembangkan aplikasi di dalam *Visual Basic*. Kode dalam jendela *Code* adalah *Sources Code* Program. Ketika pengguna menjalankan program *Visual Basic* dan *computer* kita menginterpretasikan sebagai *Sources Code* maka *computer* akan mengeksekusi intruksi di dalam *Sources Code* tersebut.



Gambar II.14 Code Form

ԵՊՈՒՆԻ ԳՐԱԿԱԾՆԵՐԻ ԿԱՍԻՆԱԿԱՆ



ՀԱՅԱՍՏԱՆԻ ԳՐԱԿԱԾՆԵՐԻ ԿԱՍԻՆԱԿԱՆ

Կարգի համաձայն հայկական գրականության հետազոտության նպատակն է գիտանալ հայկական գրականության բնույթը, զարգացումը, կառուցվածքը և ստեղծագործական ուղիները:

Սակայն կարևորագույն բնույթի ունենալը կախված է գրականության զարգացումից:

2. Հարց

ԵՊՈՒՆԻ ԳՐԱԿԱԾՆԵՐԻ ԿԱՍԻՆԱԿԱՆ

Հարցի մասին տեղեկությունները բերված են հարցի հարցում և պատասխանում:

Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան
Հարց	Պատասխան

Հարցի մասին տեղեկությունները

Հարցի մասին տեղեկությունները

Begitu banyak jendela yang digunakan didalam aplikasi *Windows*, kita dapat memindah, mengubah ukuran dan menutup kelima jendela. Dengan menggunakan mouse maka semuanya itu dapat dikerjakan dengan mudah. Beberapa contoh dasar (*object Oriented Programing*) yang terdapat dalam *Visual Basic*, antara lain :

1. Property

Property adalah sifat atau ciri yang dimiliki oleh suatu obyek, contohnya : obyek *label* mempunyai warna tulisan merah putih, warna *background* abu-abu. Warna tulisan dan warna *background* disebut *property* dari obyek *label*. Salah satu contoh dari *property object form* dapat dilihat pada tabel II.1.

No.	Property	Keterangan
1	Name	Nama object
2	Picture	Gambar background form
3	Caption	Judul form
4	Left	Jarak kiri dihitung dari layar kiri
5	Top	Jarak atas form dihitung dari atas layar
6	Border Style	Jenis pembahas form (dialog, thin, etc)
7	Record Source	Sumber data form (bisa berupa tabel, query, atau SQL)

Tabel II.1 Property Object Form

Contoh dari *property object textbox* dapat dilihat pada tabel II.2

No.	Property	Keterangan
1	Name	Nama object
2	Value	Isi dari textbox
3	Enable	Dapat digunakan atau tidak
4	Visible	Tampak atau tidak
5	Control Sources	Sumber data textbox (bisa dari field, SQL, Etc)

Tabel II.2 Property Object Toolbox

2. Method

Method ialah suatu aksi yang dapat diterapkan pada suatu *object*. Contohnya : *Textbox1.SetFocus* artinya letak kursor atau fokus di *object textbox1*.

Private Sub Command7_Click0

Me1Text5.SetFocus

End Sub

3. Event

Event adalah kejadian yang menimpa suatu *Object*. *Programmer* dapat mengontrol apa yang akan dilakukan program pada saat suatu kejadian berlangsung. Contohnya : apabila *Command1* di-klik (berarti *Object Event* tulisan (*Caption*) *Command1* berubah dari *Show* menjadi *Hide*. Salah satu contoh *Event textbox* dapat dilihat pada tabel II.3.

No.	Property	Keterangan
1	After Update	Setelah diperbarui
2	On Change	Saat dirubah
3	On Get Focus	Saat mendapat fokus
4	On Lost Focus	Saat kehilangan fokus
5	On Click	Saat di-klik
6	On Key Down	Saat tombol ditekan
7	On Mouse Move	Saat mouse digerakan di atas object

Tabel II.3 Event Object Textbox

Contoh *Event Object Form* dapat dilihat pada tabel II.4.

No.	Property	Keterangan
1	On Load	Saat form di load
2	On Click	Saat form di-klik
3	Timer Interval	Sedang waktu hitungan atau counter
4	On Timer	Saat terjadi pergantian hitngan
5	On Current	Saat terjadi pergantian record aktif
6	On Close	Saat form ditutup
7	Etc	

Tabel II.4 Event Object Form

4. Keyword „Me“

Keyword “Me” dalam kode yang diletakan dalam sebuah *form* dapat diartikan sebagai referensi terhadap *form* itu sendiri. Contohnya : jika ingin merubah tulisan *show* pada *botton* dengan *Event OnClick* menjadi *Hide*, kodenya dapat ditulis sebagai berikut :

```
Private Sub Command4_Click0
```

```
Me ! Command1.Caption = “Hide“
```

Tanda “,” dalam kode di atas mempunyai arti bahwa *Caption* ialah *Property* (dalam kasus lain dapat berupa *Method*) dari *object Command1*. dan tanda “,” bearti *object* yang berada dalam *form “Products“*.

Contoh Event Object Form dapat dilihat pada tabel 11.4.

No	Event Object	Deskripsi
1	On Load	Form Form di-load
2	On Click	Form Form di-klik
3	Timer Interval	Setelah waktu ditentukan akan muncul
4	On Timer	Form terjadi perulangan hitungan
5	On Current	Form terjadi perubahan record aktif
6	On Close	Form Form ditutup

Tabel 11.4 Event Object Form

4. Keyword "Me"

Keyword "Me" dalam kode yang diletakkan dalam sebuah form dapat dituliskan sebagai referensi terhadap form itu sendiri. Contohnya : jika ingin merubah tulisan show pada form dengan form OnClick menjadi Hide, kode yang dapat ditulis sebagai berikut :

```
Private Sub Command1_Click()
```

```
Me1.Command1.Caption = "Hide"
```

Tanda "1" dalam kode di atas merupakan arti bahwa Caption ialah Property (dalam kasus lain dapat berupa Method) dari object Command1 dan tanda "Me" berarti object yang berada dalam form "Form1".

II.7.2 Object Linking And Embedding (OLE)

Kelebihan yang dimiliki oleh *Visual Basic* adalah fasilitas OLE (*Object Linking And Embedding*) yang memungkinkan untuk membuat suatu obyek dalam suatu aplikasi yang berisi data dan aplikasi lain, yang ditempatkan di dalam program *Visual Basic*.

Linking, *object* dari aplikasi lain yang dihubungkan dengan aplikasi *Visual Basic*. Sewaktu pemakai keluar dari aplikasi, *object* yang diperbarui disimpan dalam aplikasi sumbernya. *Object* yang sama dihubungkan dengan beberapa aplikasi yang lain.

Embedding, *object* dari aplikasi lain yang dihubungkan dengan aplikasi *Visual Basic* dan ketika pemakai keluar dari aplikasi, obyek otomatis diperbarui dan disimpan dalam aplikasi *Visual Basic*. Tidak ada aplikasi lain yang mempunyai akses ke data yang di-embed.

II.8 Map Object 2.1

Map Objects 2.1 merupakan salah satu komponen yang dikombinasikan dari obyek *basisdata* dalam SIG yang berguna untuk menyajikan peta. Aplikasi yang digunakan dispesifikasikan terhadap keperluan atau perintah yang dijalankan dalam *Visual Basic 6.0*.

Map Object 2.1 berisi *ActiveX Control (OCX)*, yang merupakan *Map Control* dengan tersedianya lebih dari 50 *ActiveX Automation Objects ActiveX* sehingga dapat digunakan dalam *standart windows*. Program *Map Object 2.1* dapat dijalankan pada *Windows 95*, *Windows 98*, dan *Windows NT 4..*

Dalam *Map Object 2.1* didukung oleh *ActiveX Data Objects (ADO)* yang merupakan model *basisdata* dari *Microsoft* dengan fungsi sebagai alat untuk menjalankan *Universal Data Access (UDA)*. Tujuan dari UDA agar pemrograman *Visual Basic* dapat mengikuti standart *OLEDB* dalam membuat suatu aplikasi. *OLEDB* adalah suatu standar *basisdata* yang digunakan sebagai alat untuk menghubungkan data dari berbagai sumber data, misalnya penyimpanan data dalam sebuah *basisdata*, sebuah *file teks*, atau lembar kerja. Dalam *Map Object* dapat dibuat tabel *object* dengan menghubungkan data dari *Microsoft Acces 2000*

11.7.2 Object Linking and Embedding (OLE)

1. Aplikasi yang dimiliki oleh Visual Basic adalah fasilitas OLE (Object Linking and Embedding) yang memungkinkan untuk membuat suatu objek dalam suatu aplikasi yang berisi data dan aplikasi lain yang ditempatkan di dalam program Visual Basic.

Linking objek dari aplikasi lain yang dihubungkan dengan aplikasi Visual Basic. Setelah pemakai klik dari aplikasi, objek yang dipicu akan disimpan dalam aplikasi sumbernya. Objek yang sama dihubungkan dengan beberapa aplikasi yang lain.

Embedding objek dari aplikasi lain yang dihubungkan dengan aplikasi Visual Basic dan ketika pemakai klik dari aplikasi, objek otomatis dipicu dan disimpan dalam aplikasi Visual Basic. Tidak ada aplikasi lain yang mempunyai akses ke data yang di-embed.

11.8 VBA Object 2.1

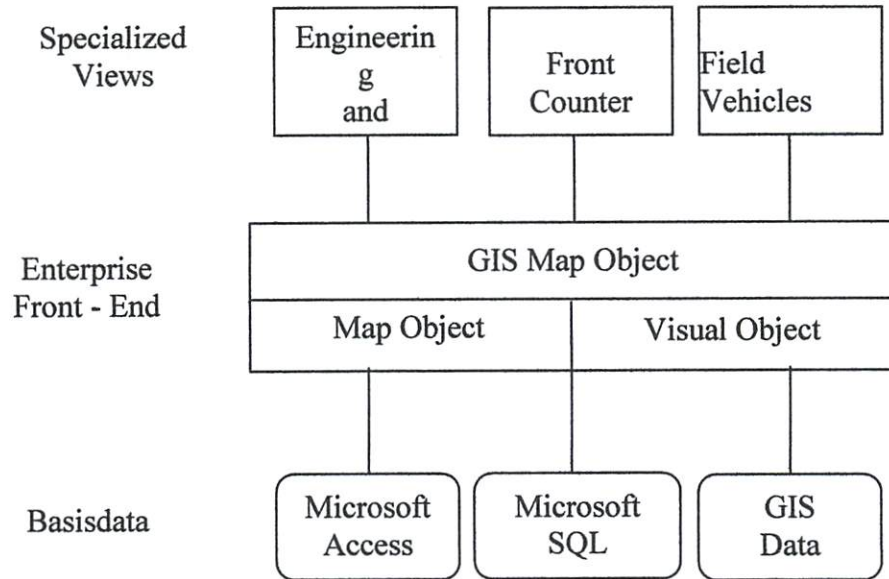
2.1.1. merupakan salah satu komponen yang dikomponisikan dari objek Visual Basic dalam VBA yang berguna untuk menyajikan data. Aplikasi yang digunakan dipertunjukkan terhadap kegunaan atau perintah yang dilakukan dalam Visual Basic 6.0.

2.1.2. berisi Library Class (OCX) yang merupakan VBA Class dengan resourcenya lebih dari 50 Library Automation Object Class. Sehingga dapat digunakan dalam Visual Basic Program Map Object 2.1 dapat dijalankan pada Windows 95 dan Windows NT.

2.1.3. didukung oleh ActiveX Data Object (ADO) yang merupakan model Visual Basic dari Microsoft dengan fungsi sebagai alat untuk menjalankan Query Language (UDA). Tujuan dari UDA agar programmer Visual Basic dapat mengikuti standar OLEDB dalam membuat suatu aplikasi. OLEDB adalah suatu standar Visual Basic yang digunakan sebagai alat untuk menghubungkan data dari berbagai sumber data. Misalnya penyimpanan data dalam sebuah Visual Basic seperti Ms Access atau dalam objek Data Map Object dapat dibuat tabel objek dengan menggunakan Library Class 2000.

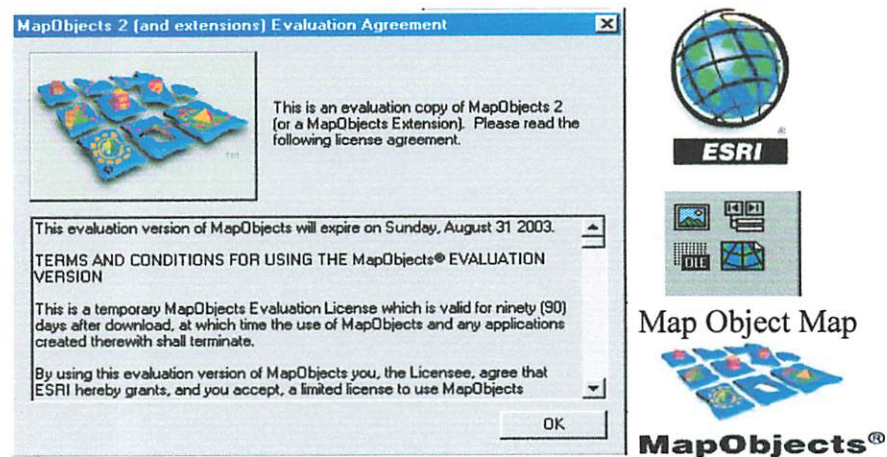
yang menggunakan *Microsoft Jet 4.0 OLEDB Provider* dan *set up AddRelate* untuk menggabungkan informasi ke dalam *Shapefile*. (ESRI, *Map Object Online Reference*, 2001)

Berikut diagram alir hubungan antara Visual Basic 6.0 dengan *Map objects 2.1*, yang terdapat pada gambar di bawah ini :



Gambar II.15 Map Objects System Architecture

Apabila kita menginstallkan Map Objects 2.1 ke dalam Program *Visual Basic* maka akan tampil beberapa *icon* Map Objects 2.1 di dalam *Tollbox*, melalui *icon* ini nantinya kita akan dapat mengaktifkan Map Objects 2.1. Tampilan pembuka apabila kita mengaktifkan program Map Objects 2.1 dapat dilihat pada gambar II.16



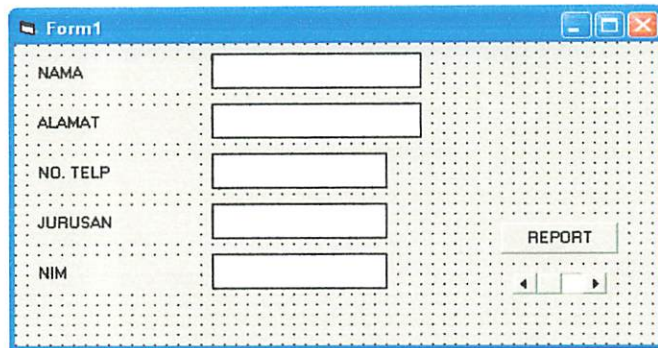
Gambar II.16 Tampilan Map

II.9 Konsep Pembuatan Program Pada Visual Basic

Langkah penyusunan program pada *Visual Basic* secara garis besar adalah :

1. Pembuatan tampilan/*desain form*

Pada umumnya sebuah *form* yang dibuat akan berbentuk persegiempat (kotak) yang berfungsi sebagai *platform* untuk meletakkan obyek yang diperlukan. Penempatan berbagai obyek ini tergantung dari perancang program tersebut. Salah satu contoh *desain* awal sebuah *form* untuk sebuah aplikasi, dapat dilihat pada gambar di bawah ini.



Gambar II.17 Desain Menu Form

2. Penataan property

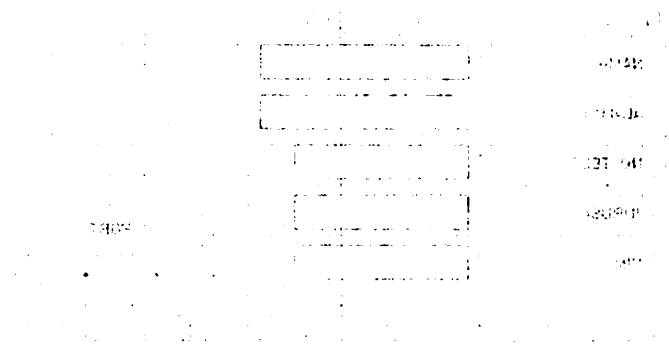
Property Visual Basic adalah mekanisme normal untuk menjelaskan atribut-atribut obyek. Setiap obyek Visual Basic memiliki *property* tertentu yang *setting*-nya mengontrol tampilan obyek dalam suatu aplikasi. Penataan ini meliputi pengaturan *border style*, nama, warna, *font* dan lain-lain. Sebagai contoh *property visible* suatu obyek bias diset ke *true* dan *false* (obyek tampak dan tidak tampak pada suatu aplikasi yang dijalankan). Tampilan *property* dari sebuah obyek yang aktif dapat dilihat pada gambar dibawah ini.

11.9 Konsep Pembuatan Program Pada Visual Basic

Langkah pembuatan program pada Visual Basic secara garis besar adalah :

A. Pembuatan tampilan form

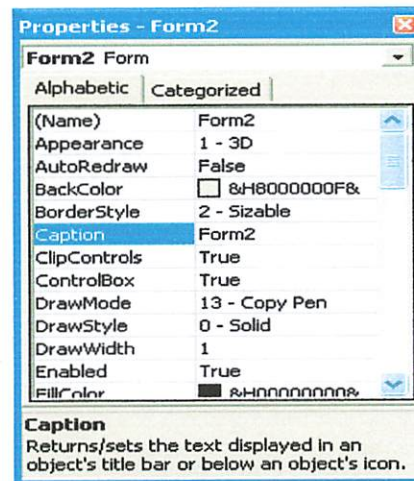
Pada umumnya sebuah form yang dibuat akan berinteraksi dengan objek-objek (kontrol) yang berfungsi sebagai penyedia untuk melakukan objek yang dibutuhkan. Pembuatan objek-objek ini tergantung dari perencanaan program tersebut. Salah satu contoh adalah form untuk sebuah program aplikasi dapat dilihat pada gambar di bawah ini.



Gambar 11.7. Bagan Form Form

B. Pembuatan property

Property Visual Basic adalah mekanisme normal untuk menjelaskan atribut-atribut objek. Setiap objek Visual Basic memiliki property tertentu yang sebelumnya mengontrol tampilan objek dalam suatu aplikasi. Pembuatan ini meliputi pengaturan warna, ukuran, font dan lain-lain. Sebagai contoh property warna suatu objek bisa di set ke warna yang tampak dan tidak tampak pada suatu aplikasi yang dijalankan. Tampilan property dari sebuah objek yang dapat dilihat pada gambar di bawah ini.



Gambar II. 18 Penataan property

3. Prosedur event

Event merupakan kejadian yang akan diterima oleh suatu obyek. Event yang diterima oleh obyek berfungsi untuk menjalankan kode program yang ada di dalam obyek tersebut. Pada pemrograman *event-driven*, perancang program menuliskan sebuah program yang bereaksi terhadap tindakan pemakai.

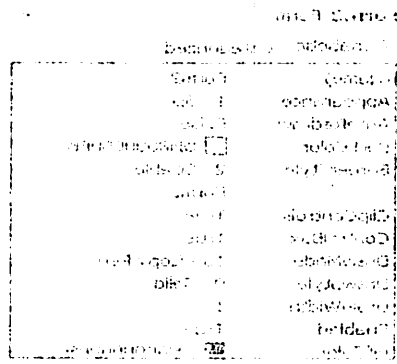
Private Sub Command1_Click

Baris kode program di atas menunjukkan penggunaan *event Click* pada obyek *Command1*, yang mempunyai *irit* apabila obyek *Command1* di-klik maka kode program yang terletak di bawah baris kode program tersebut akan dijalankan.

4. Menuliskan kode

Menentukan perintah-perintah dalam bahasa Visual Basic, pernyataan-pernyataan yang mengontrol operasi program yang merupakan jantung pemrograman.

Termasuk juga deklarasi *variable* yaitu menyisihkan tempat untuk nilai dalam memori *computer*.



Gambar 11.18 Form dan Property

3. Prosesor event

Event merupakan kejadian yang akan diterima oleh suatu objek. Event yang diterima oleh objek berfungsi untuk menjalankan kode program yang ada di dalam objek tersebut. Pada program-program event-driven, programmer menuliskan sebuah program yang bereaksi terhadap tindakan pengguna.

Private Sub Command1_Click

Basis kode program di atas menunjukkan penggunaan event Click pada objek Command1 yang mempunyai nilai sebagai objek Command1_Click. Maka kode program yang terletak di bawah basis kode program tersebut akan dijalankan.

4. Menuliskan kode

Menentukan perintah-perintah dalam bahasa Visual Basic, pernyataan-pernyataan yang mengontrol operasi program yang merupakan jantung program.

Formasok juga deklarasi variabel yang menuliskan tempat untuk nilai dalam memori komputer.

```
Private Sub Command1_Click0
```

```
    Picture1.Visible = True
```

```
End Sub
```

```
Private Sub Command2_Click0
```

```
    Picture1.Visible = False
```

```
End Sub
```

5. Menjalankan program

Setelah menuliskan kode dilakukan pengetesan program dengan menekan tombol F5, *desain* program dapat ditinjau lagi dan dapat ditambahkan aksesoris, *label*, kata-kata sehingga memudahkan pemakai untuk menggunakan program tersebut.

BAB III

PELAKSANAAN PENELITIAN

III.1. Lokasi Penelitian

Kabupaten Gresik terletak 7° 32' - 8° 39' Lintang Selatan dan 112° 51' - 113° 44' Bujur Timur. Secara administrasi wilayah Kabupaten Gresik mempunyai batas-batas sebagai berikut :

- Sebelah Utara : Laut Jawa
- Sebelah Timur : Selat Madura
- Sebelah Selatan : Kab. Mojokerto, Kab. Sidoarjo dan Kota Surabaya
- Sebelah Barat : Kabupaten Lamongan dan Mojokerto.

Luas wilayah Kabupaten Gresik 1.192,90 Km² merupakan daerah yang mempunyai luas relatif kecil dibanding kabupaten lain di Propinsi Jawa Timur. Sedangkan Pembagian wilayah administrasi Kabupaten Gresik adalah :

- Jumlah Kecamatan sebanyak 18
- Jumlah Desa/Kelurahan : 356

Di Kabupaten Gresik Terdapat dua buah sungai besar yang melintas di Kabupaten Gresik, yaitu Sungai Bengawan Solo dan Kali Lamong.

Selain sungai, potensi air permukaan di Kabupaten Gresik tersebar di beberapa waduk dan telaga yang ada di Kabupaten Gresik. Waduk dan telaga tersebut pada umumnya merupakan waduk tadah hujan. Waduk-waduk utama dengan luasan cukup besar Sedangkan luas areal seluruh waduk di Kabupaten Gresik adalah 1.079,1 ha. Waduk-waduk tersebut tersebar di 15 kecamatan yang secara keseluruhan diperkirakan dapat mengairi area seluas 11.086,00 ha (*Dinas PU, 2001*).

III.2. Data dan Alat yang digunakan

Materi atau bahan penelitian yang digunakan dalam penelitian ini terdiri dari data spasial dan data non spasial, dengan spesifikasi sebagai berikut:

III.2.1 Bahan Penelitian

Data Penelitian yang digunakan dalam penelitian meliputi :

1. Data Spasial

- Peta Administrasi Kabupaten Gresik skala 1 : 25.000
- Peta Lokasi waduk hasil survey lapangan. dengan GPS Navigasi
- Peta Jaringan Jalan Kabupaten Gresik skala 1 : 25.000

2. Data Non Spasial

- Data atribut batas administrasi :
 1. Nama Kecamatan
 2. Nama Desa
- Data jalan
- Data atribut Waduk
 1. Nama Telaga
 2. Fungsi Telaga
 3. Di bangun Tahun
 4. Kapasitas Tampung air (M3)
 5. Lokasi Telaga :
 - Desa
 - Kecamatan
 6. Bangunan :
 - Trap / Tangga cuci
 - Plengsengan
 7. Pannjang Tanggul Telaga (m)
 8. Kedalaman Rata – Rata (m)
 9. Kondisi Saat ini (%)

10. Penggunaan kepala keluarga (KK)

- Foto - foto

III.1. 2 Alat Penelitian

1. Perangkat lunak

- Auto Cad 2000
- Visual Basic 6.0
- Map Object 2.1
- Arc info
- Arc View 3.2
- Microsoft Access 2000
- Ozi_Explorer

2. Perangkat Keras

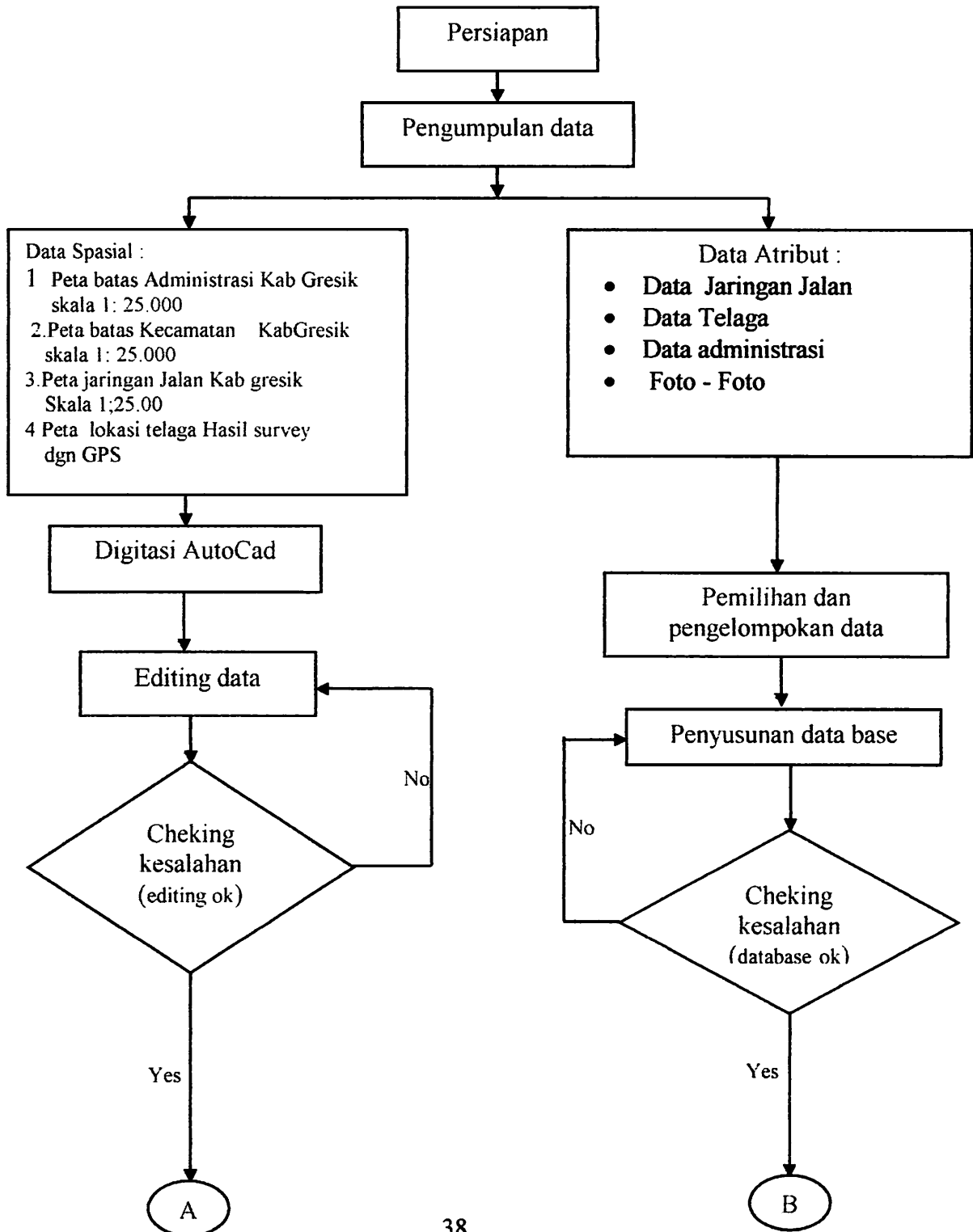
1. Hardware

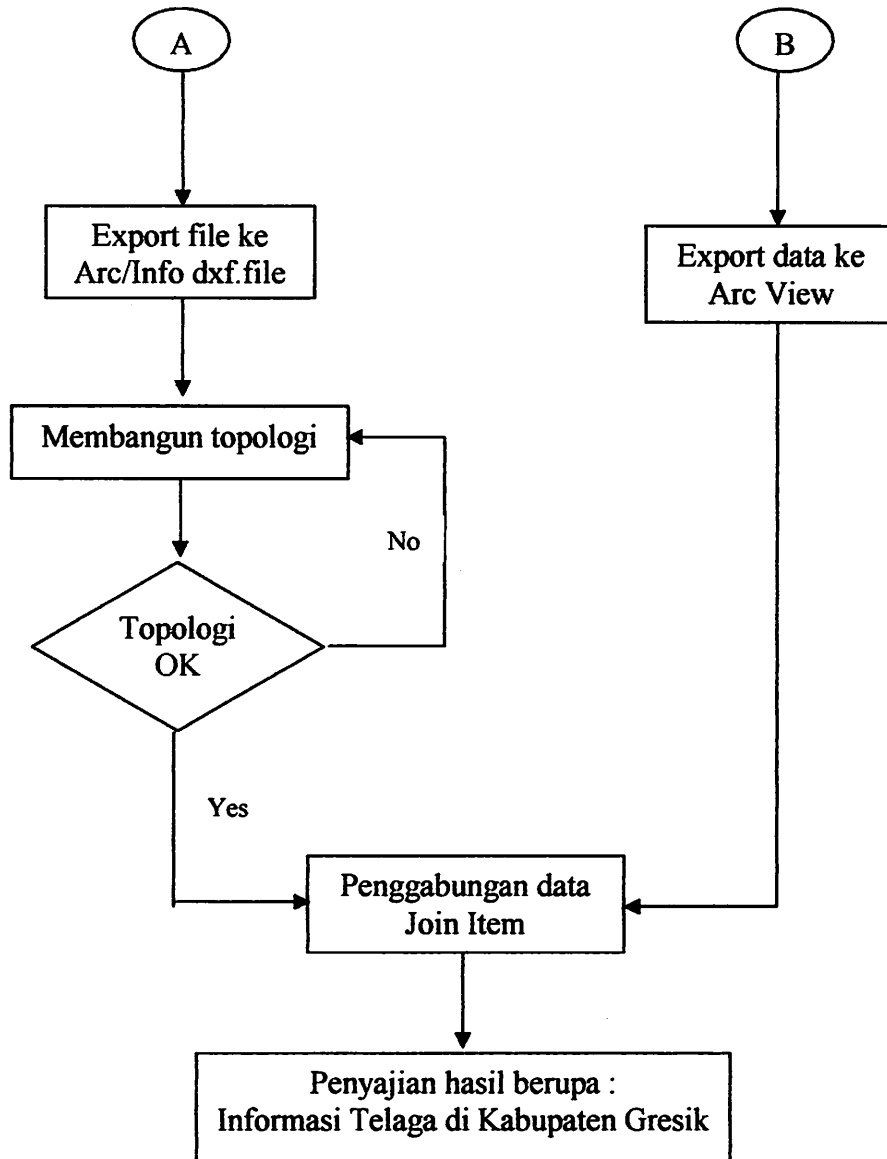
- Monitor
- CPU (*Control Processing Unit*)
 - Pentium AMD Atlon
 - RAM 128 MB
 - Hard Disk 40 GB
- Keyboard
- Mouse
- GPS III +
- Digitizer
- Printer Canon S6300 SP, Canon BJC 2100 SP
- CD ROM 52x
- Kamera Digital

III.4. Metodologi Penelitian

Langkah kerja penelitian yang digunakan dapat dilihat pada diagram alir dengan keterangan sebagai berikut :

Bagan Alir Penelitian





Gambar Bagan Alir Penelitian

III .5. Pengertian Bagan Alir Penelitian

1. Persiapan

Dalam persiapan penelitian ini meliputi semua persiapan data baik data spasial maupun data non spasial serta perangkat keras dan perangkat lunak yang akan digunakan dalam penelitian.

2. Pengumpulan data

Pengumpulan data yang digunakan dalam hal ini baik data spasial maupun data non spasial.

a. Data Spasial

Data spasial yang digunakan untuk penelitian ini diantaranya adalah peta Batas administrasi kabupaten Gresik . Peta ini di buat dengan peta dasar Topografi yang dilengkapi dengan peta – peta penunjang lainnya. Peta-peta yang dibutuhkan dalam mengerjakan pembuatan penginformasian Telaga di kabupaten gresik berikut :

- ✓ Peta Batas Administrsi kabupaten Gresik Sumber Bakosortanal Skala1: 25.000
- ✓ Peta Batas Administrsi Kecamatan Kabupaten Gresik sumber Bakosortanal Skala1: 25.000
- ✓ Peta Jaringan Jalan Gresik Sumber Bakosortanal Skala1: 25.000
- ✓ Peta Lokasi Telaga hasil survey Lapangan dengan GPS

2.1. Persiapan Basis Data Spasial

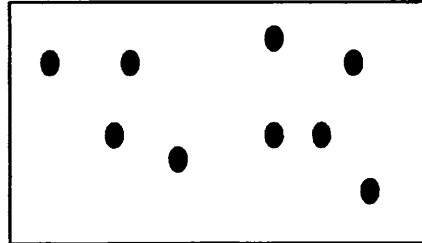
2.1.1. Digitasi

Data spasial disini adalah data berupa gambar yang berhubungan dengan lokasi, bentuk dan hubungan antar unsur. Proses dari pemasukan data spasial dimulai dengan pemasangan data spasial pada meja digitizer, yang kemudian dilakukan proses pendigitasian.

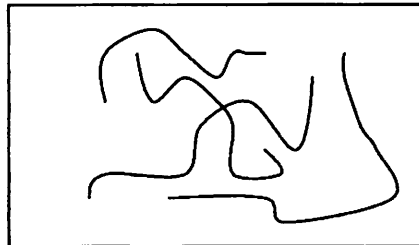
Waktu pendigitasian tergantung dari banyaknya peta yang didigitasi, dan hasil digitasi dapat langsung ditampilkan pada layar

monitor, sehingga dapat dengan mudah membenahi jika ada kesalahan dalam melakukan digitasi. Data spasial pada sistem vektor terdiri dari :

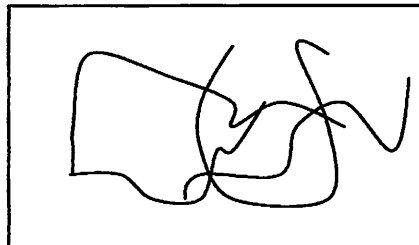
- ✓ Titik, dengan format sepanjang koordinat (x,y) yang tidak memiliki dimensi panjang dan luas



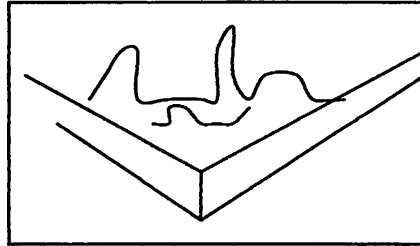
- ✓ Garis, kumpulan pasangan-pasangan koordinat yang mempunyai titik awal dan titik akhir serta mempunyai dimensi panjang tetapi tidak mempunyai dimensi luas.



- ✓ Area/Poligon, kumpulan pasangan-pasang koordinat yang mempunyai titik awal dan titik akhir sama serta mempunyai dimensi panjang dan luas.

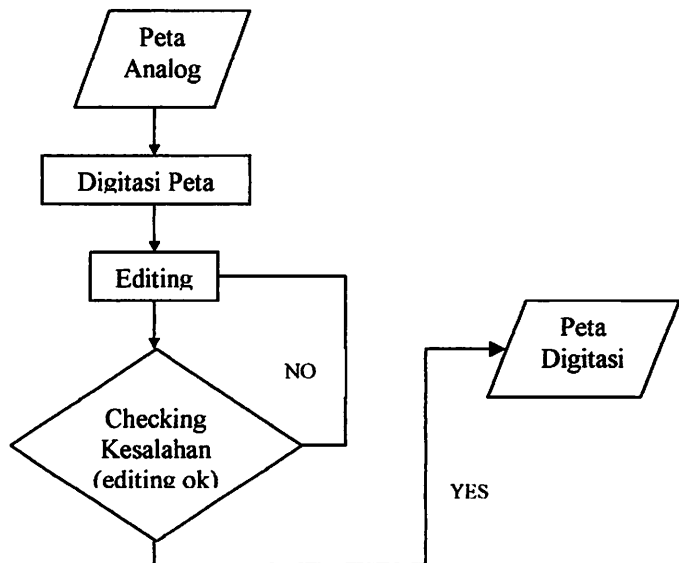


- ✓ Permukaan, area dengan besaran (x,y,z) yang mempunyai dimensi luas, panjang dan ketinggian



Pekerjaan pendigitasian dimaksudkan untuk mengkonfersi format data, yaitu dari data grafis menjadi data digital yang dapat disimpan dan dikelola oleh komputer. Alat yang digunakan untuk pendigitasian dinamakan digitazer. Sebelum dilakukan pendigitasian terlebih dahulu dilakukan penyusunan klasifikasi data terhadap semu data yang sesuai dengan karakteristik dari masing-masing kenampakan dan tema informasi yang dimilikinya, meliputi feature titik, arc dan poligon.

Proses dari digitasi perangkat lunak Autocad adalah sebagai berikut :



2.1.1.1. Persiapan Digitasi

Didalam persiapan ini yang kita lakukan adalah mengumpulkan data spasial yang akan kita digitasi, seperti peta topografi, peta jaringan jalan

2.1.1.2. Menghubungkan Komputer dengan Digitiser

Yang pertama kali harus dilakukan untuk membuat suatu sistem informasi adalah pemasukan data. Untuk data spasial dilakukan proses digitasi yang disimpan dengan ekstensi DWG, baru kemudian dilakukan editing pada masing-masing layer.

Digitasi data spasial dilakukan pada meja digitizer, dengan perangkat lunak (software) Autocad, dimana peta yang pertama kali didigit adalah peta dasar, baru kemudian peta-peta yang lain yang memiliki format sama dengan peta dasar.

2.1.1.3. Kalibrasi Meja Digitizer

Sebelum mulai melakukan digitasi peta grafis, perlu dilakukan proses kalibrasi. Proses kalibrasi ini dilakukan untuk mengecek kesalahan dalam memasukkan koordinat peta melalui digitizer dengan koordinat peta yang dimasukkan melalui keyboard.

2.1.1.4. Proses Pembuatan Layer

Pada tahapan inilah proses digitasi dimulai. Dalam mendigit data spasial, disini harus dibuat/didigit per-lapis (layer) supaya mudah dalam proses pengolahan data.

2.1.1.5. Editing Hasil Digitasi

Setelah pendigitasian dengan menggunakan Autocad sudah selesai, masih perlu dilakukan pengecekan ulang, untuk menghindari kesalahan-kesalahan yang terjadi. Setelah pendigitasian dilakukan pengkoreksian dan pendigitasian sudah memenuhi syarat, maka langkah selanjutnya adalah mengekspor data digitasi dari Autocad ke Arc/info (DWG ke ArcInfo) dengan menyimpan dengan format **DXF**.

2.1.2. Membuat Coverage dalam ArcInfo

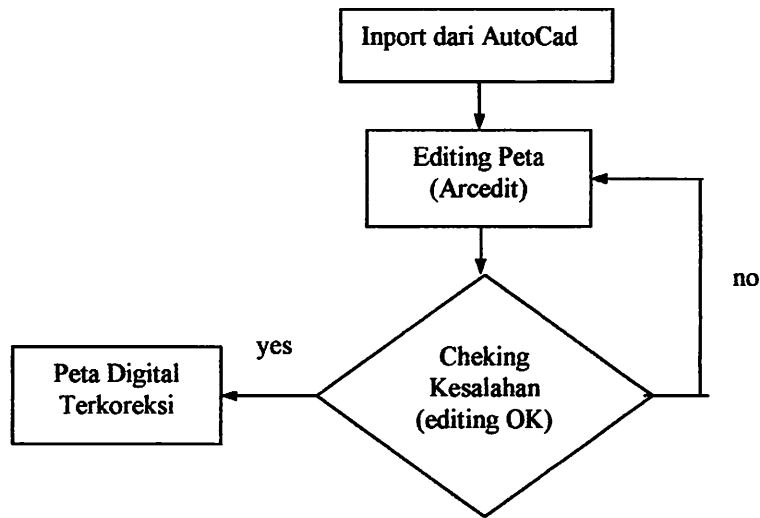
Setelah peta hasil digitasi diedit dan disimpan dalam extension **DXF**, kemudian membuat coverage dalam ArcInfo dari file DXF yang sudah ada dilakukan dengan menggunakan perintah :

DXFARC [FILE_DXF] [coverage]

2.1.2.1. Editing

Peta yang diimport dari AutoCad sangat mungkin mengandung banyak kesalahan. Sehingga koreksi data dilakukan untuk memperoleh kesesuaian peta hasil digitasi dengan peta asli. Akibat kelelahan dan kejenuhan dari pelaksanaan digitasi maka seringkali muncul kesalahan digitasi, seperti bentuk garis, titik maupun poligon yang tidak tepat (melencong dari posisi sebenarnya). Kesalahan-kesalahan digitasi lain yang seringkali terjadi adalah *pseudo node*, *dangling node* dan kesalahan penempatan tabel.

Proses koreksi data grafis atau data spasial dengan menggunakan perangkat lunak ArcInfo diperlihatkan diagram :



Proses editing di dalam ArcInfo dilakukan dengan menggunakan perintah **ARCEDIT**. Kesalahan-kesalahan yang sering ditemui dalam proses editing antara lain : kekurangan tabel, overshoot, undersoot.

Contoh daftar untuk mengoreksi kesalahan :

Daftar koreksi	Edit Feature
Data ditambah dengan mendigitasi	
1. Menambahkan kekurangan tabel	LABEL
Data dikoreksi dengan menggunakan layar grafik	
2. Mengoreksi overshoot	ARC
3. Mengoreksi undeshoot	NODE
4. Memperbaiki poligon terbuka	NODE
5. Menambah User-ID label	LABEL

Tabel : Daftar Koreksi

b. Data Non Spasial

Untuk pemasukan data non spasial dilakukan dengan perangkat lunak Microsoft excel sesuai dengan informasi masing-masing peta. Data-data atribut ini disusun dalam bentuk tabel dan masing-masing unsur yang berbeda diberi ID (identitas) yang unik yaitu tidak sama antara satu dengan yang lainnya. Dalam pemberian ID tersebut sama dengan nomer label yang

diberikan pada setiap data spasial (titik, garis dan poligon) pada proses editing.

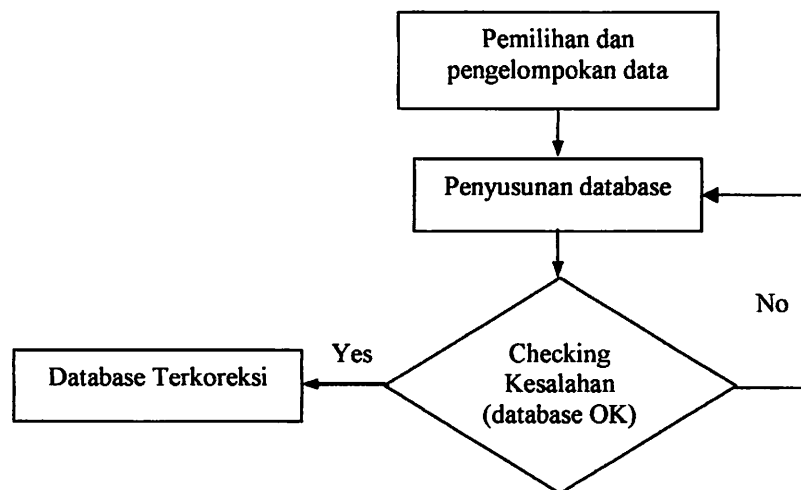
Input data atribut kedalam basis data yaitu menggabungkan data atribut dengan data spasial dalam satu file. Dalam penelitian ini data atribut/entitas yang akan dimasukkan atau dibuat adalah :

- ✓ Data Batas Administrasi Kabupaten Gresik
- ✓ Data Batas Administrasi Kecamatan Kabupaten Gresik
- ✓ Data Jaringan jalan Kabupaten Gresik
- ✓ Data Profil Telaga Di kabupaten Gresik

Sebelum dilakukan penyusunan data atribut terlebih dahulu dilakukan pemilahan dan pengelompokan data berdasarkan jenis dan macamnya.

3. Editing dan Checking Kesalahan

Setelah menyusun database selesai, maka dilakukan proses editing untuk data tersebut. Hal ini dilakukan agar data yang sudah disusun tidak terdapat kesalahan dan kemudian dilanjutkan dengan proses checing data atribut, apabila masih ada data yang kurang maka dilakukan proses penyusunan kembali.



4. Pembentukan Topologi

Untuk mendapatkan hubungan spasial antar feature pada peta digital, digambarkan dengan menggunakan topologi. Topologi adalah hubungan yang digambarkan untuk menyajikan persambungan antara pertemuan feature. Pada pembuatan peta secara digital topologi menentukan hubungan diantara feature, mengidentifikasi poligon yang bersebelahan dan dapat menentukan bentuk suatu feature misalnya, feature poligon (area) yang merupakan kumpulan feature lainnya, yaitu feature garis. Ada dua pilihan menu yang dapat digunakan untuk pembuatan topologi suatu coverage, yaitu **Clean** dan **Build**.

Clean adalah menu untuk membentuk struktur data topologi sekaligus dengan fasilitas koreksi terhadap kesalahan-kesalahan sederhana (nimer error). Sedangkan **Build** berfungsi untuk membentuk topologi tanpa melakukan perubahan terhadap data grafis, jadi build tidak menambah atau merubah informasi.

Langkah pembuatan topologi :

- [ARC] **Clean** (nama coverage) (enter)
- [ARC] **Build** (nama coverage) (enter)

Demikian pembentukan topologi untuk semua coverage.

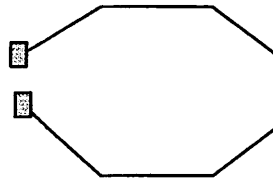
Untuk editing data pada ArcInfo antara lain meliputi :

- Menghilangkan undershoot dan overshoot
- Memberikan Id pada coverage
- Memperbesar dan memperkecil feature

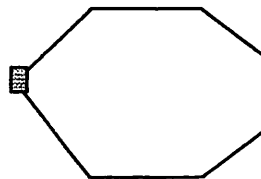
4.1. Menghilangkan Undershoot dan Overshoot

Undershoot terjadi karena kurang penjangnya line sehingga tidak membentuk poligon tertutup, untuk menghilangkan dapat dilakukan dengan cara :

- Tampilkan coverage yang akan diedit pada **Arcedit**
- Tampilkan node danglennya sehingga diketahui poligon yang mengalami undershoot dengan perintah **drawen noden dangle;draw [enter]**
- Setelah tampak node dangle seperti di bawah ini :

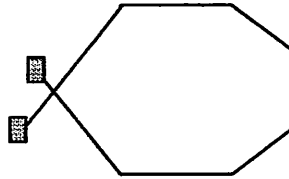


- Ketik **EF Node [enter]**
- Ketik **Move [enter]**
- Klik node pertama yang akan digabung dengan mouse
- Pilih **Select**
- (Where point to move) klik node kedua yang akan digabung sehingga akan tampil seperti dibawah ini

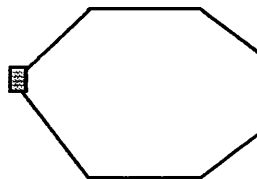


Sedangkan untuk menghilangkan overshoot (garis yang berlebihan pada poligon) dapat dilakukan dengan cara :

- Tampilkan fitur yang mengalami overshoot
- Tampilkan dangle dengan cara **Drawen node dangle [enter]**, seperti dibawah ini
- Ketik **ef arc [enter]**



- Ketik **Sel box [enter]**, lalu select bagian yang mengalami overshoot
- Ketik **delete [enter]**, sehingga akan tampil seperti dibawah ini

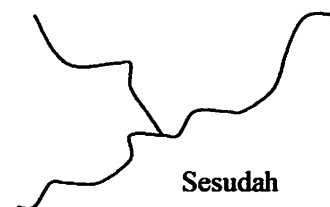
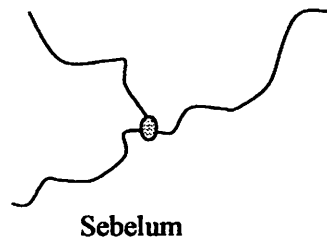


4.2. Pemberian Id

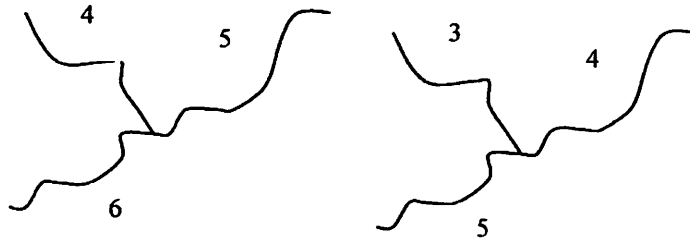
Tampilkan pada software ArcView harus diberi Id. Cara pemberian Id yaitu :

Tampilkan coverage yang akan diberi Id pada Arcedit dengan mengetikkan **Edit nama coverage all;draw[enter]**

- Kemudian ketikkan **Drawen node Off [enter]** untuk mengaktifkan titik-titik perpotongan (node), hal ini tidak dapat dilakukan apabila node tersebut tidak mengganggu dalam pemberian Id.



- Kemudian tampilkan Id pada coverage dengan mengetikkan **Drawen arc Id;draw [enter]**
- Setelah semua Id tampak, maka mulai tentukan dari mana akan dilakukan pemberian Id. Id yang akan dimasukkan harus sesuai dengan tabel atribut yang dibuat.
- Kemudian ketikkan **ef arc;sel;calculate sid = 5;draw [enter]**
- Setelah itu tempatkan kursor mouse pada yang akan diganti Id-nya
- Setelah semua Id selesai dimasukkan ketikkan **Save [enter]** untuk menyimpan hasil dari pergantian Id



4.3. Pemberian Label

Langkah kerja pemberian tabel (Id) dapat dilakukan dengan cara :

- Tampilkan coverage yang akan diberi label
- Ketik : **ef lab (enter)**
- Ketik : **add (enter)**
- Pilih : **1 (add label)**
- Klik poligon yang akan diberi label (Id)
- Jika label yang akan diberikan sesuai dengan pilihan maka ketik : **add (enter)**
- Pilih : **8 (digitazing option) (enter)**
- Pilih : **1 (new user Id) (enter)**
- Klik poligon yang akan diberi label

- Untuk cheking kesalahan label ketik : **drawen label id;draw**
(enter)

4.4. Memperbesar dan memperkecil feature

Jika dalam editing feature ada object yang tidak jelas maka object harus diperbesar. Untuk memperbesar dan memperkecil featur dapat dilakukan dengan cara :

- Tampilkan featur yang akan diedit
- Ketik : **mape*;draw** (enter)
- Select object yang akan diperbesar
- Sedangkan untuk memperkecil kedalam ukuran semula, ketik : **mape default;draw** (enter)

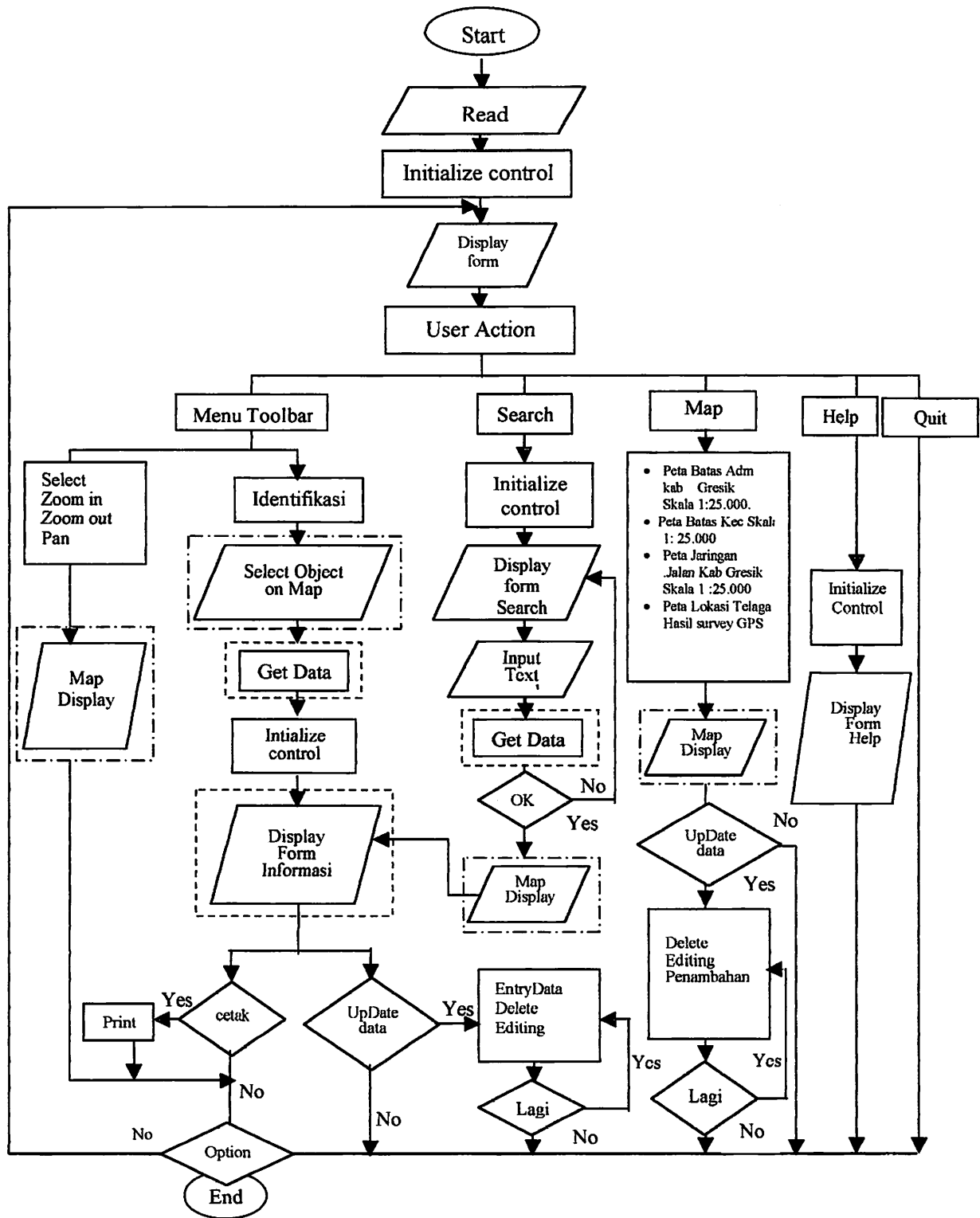
5. Penggabungan Data (Join Item)

Setelah export data dbf dilakukan maka langkah selanjutnya dalah melakukan penggabungan atau join item terhadap data atribut yang sudah diexport dengan data spasial di ArcView. Join item merupakan perintah arc yang digunakan untuk menggabungkan file data ke table atribut. Langkah-langkah penggabungan ini adalah sebagai berikut :

- ✓ Aktifkan software ArcView
- ✓ Muncul kotak dialog Untitled, kemudian klik New pada kotak dialog Untitled dan muncul View 1, lalu klik Add Theme (icon bergambar tanda tambah)
- ✓ Memilih file yang akan ditampilkan pada kotak View 1, kemudian klik ok
- ✓ Mengklik theme tabel, sehingga akan muncul atribut dari file (coverage) yang dipilih
- ✓ Kemudian mengklik tables pada untitled, lalu klik add
- ✓ mengklik ID dari tables, kemudian ID Attribute Of (name file)
- ✓ Kemudian mengklik Toolbar, untuk menggabungkan dau ID tersebut

3.2 Tahapan Pembuatan Program

Diagram Alir Rencana Program



Gambar3.1 Diagram Alir Pembuatan Program

- Melalui proses penvimanan data
- - - - - Melalui proses menggunakan MapObject
- _____ Melalui proses menggunakan Visual Basic

6.1. Keterangan diagram alir program

Untuk program informasi telaga maka ada beberapa tahapan yang perlu di lalui yang diagram alirnya perlu di lihat pada gambar, yang penjelasannya sebagai berikut:

- 1 Read project data : Sub sistem ini merupakan panggilan data yang telah di proses/yang telah di bentuk sebelumnya.
- 2 Initialize control : Sub sistem ini merupakan tahap mengenali perintah (control) untuk menjalankan program
- 3 Display main form : Sub sistem yang menampilkan form tampilan dari pembacaan data spasial.
- 4 Uses action : Sub sistem yang dimana pengguna (user) menampilkan dan mencari informasi yang tersedia di dalam program.
- 5 Menu dan Toolbar : Icon yang berfungsi untuk menampilkan informasi langsung dari peta spasial yang ada diprogram.
- 6 Search : Menu perintah pencarian dalam Visual Basic sesuai dengan perintah yang dijalankan.
- 7 Help : Sub sistem merupakan menu bantuan yang berisikan petunjuk (guide) menjalankan program.
- 8 Input Query : Sub sistem yang diajukan untuk manentukan data mana saja yang akan disimpan atau ditampilkan didalam basis data dan menentukan bagaimana data tersebut direalisasikan.
- 9 Updating data : Sub sistem yang menunjukkan adanya perintah pembaruan data atau perubahan data secara berkala sehingga tidak menghilangkan informasi yang sebelumnya.
10. Display form Informasi : Sub sistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data dalam bentuk soft copy

Microsoft Access 2000. Untuk tiap unsur yang berbeda diberi nomor ID yang berbeda dengan tipe text.

Langkah-langkah dalam mendesain basisdata ini, yaitu :

1. Di dalam kotak dialog *Microsoft Access*, pilih *Blank Database* dan ketik nama *file* yang akan dibuat. Klik *Create* untuk menampilkan jendela databasenya.
2. Klik *Desain View* untuk menuliskan *field name* yang akan dibuat di setiap tabel.
3. Isikan ID atau keterangan yang lainnya dalam setiap *field name* dan pada kolom *Data Type* pilih *number* atau *text*.
4. Simpan hasil desain tabel dengan nama sesuai dengan entitinya.

III. 8. Pembuatan Aplikasi Pencarian

Aplikasi pencarian pada program ini juga dibuat dengan menggunakan program *Visual Basic* yang berfungsi untuk mempermudah pencarian lokasi dari setiap telaga, kecamatan dan kelurahan, yang ada.

6. Pembuatan Program

Pembuatan Program merupakan proses terakhir yang bertujuan untuk menampilkan hasil akhir yang telah dikerjakan oleh peneliti. Hasil akhir ini berupa informasi telaga di Kabupaten Gresik

III.6. Pendesain Tampilan Informasi Telaga Di Kabupaten Gresik

Dalam pembuatan informasi telaga di kabupaten Gresik ini kita menggunakan program Visual Basic dan MapObject. Program Visual Basic digunakan untuk membuat tampilan (form) sedangkan MapObject digunakan untuk menampilkan data spasial, karena Visual Basic tidak memiliki fasilitas untuk menampilkan data spasial.

Program Informasi telaga didesain dengan tampilan yang sederhana mungkin agar dapat mudah dimengerti dengan semua pengguna yang membutuhkan informasi tentang telaga yang ada di Kabupaten Gresik. Dalam program ini menyediakan peta batas administrasi kabupaten, batas administrasi kecamatan, peta jaringan jalan yang berada di Kabupaten Gresik

Selain data spasial yang ada, program ini juga menyediakan data non spasial, meliputi identitas dari masing-masing objek dan fasilitas objek tersebut.

III.7. Pembuatan Basis Data Telaga Di Kabupaten Gresik

Pembuatan basis data ini dengan menggunakan bantuan Microsoft Access 2000, yang kemudian dimasukkan kedalam Visual Basic. *Form* basis data telaga ini menyajikan berbagai informasi Profil telaga yaitu letak telaga, letak kecamatan, letak letak desa. *Form* ini dapat dijalankan dengan mudah yaitu dengan menekan tombol *updating* data pada program atau juga dapat dijalankan melalui tombol *grafis*.

Pada basisdata Telaga ini, selain pengguna dapat memperoleh informasi tentang telaga, pengguna juga dapat melakukan perbaikan data, penambahan data baru, analisa data, dan penyimpanan data. kesemuanya tersebut dapat dijalankan dengan menekan tombol yang tersedia pada *form* basisdata telaga

Data non-spasial atau data atribut disusun dalam bentuk tabel-tabel dan masing-masing unsur yang berbeda diberi ID (identitas) yang unik (perbedaan identifikasi data satu dengan yang lain). Pembuatan basisdata ini menggunakan

BAB IV

HASIL DAN PEMBAHASAN

IV.1 Penyajian Peta Batas Administrasi

Hasil yang diperoleh dari penelitian ini adalah tampilan data Spasial dan data Non Spasial informasi Telaga Di kabupaten Gresik. Sehingga memungkinkan para pengguna informasi dapat dengan mudah mengakses data informasi Telaga di Kabupaten Gresik

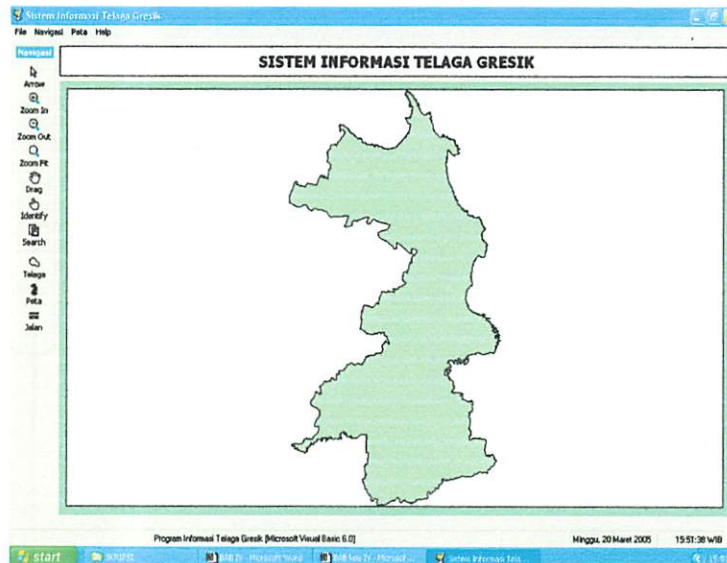
IV.2 Penyajian Peta Batas Administrasi

Penyajian peta pada program informasi Telaga di Kabupaten Gresik

ini terdiri dari :

a. Peta Batas Kota Malang

Merupakan peta batas administrasi Kabupaten Gresik beserta beberapa data atribut seperti data jumlah Kecamatan, jumlah Kelurahan .Seperti terlihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Peta Batas Kabupaten Gresik

BAB IV

HASIL DAN PEMBAHASAN

IV.1. Penyajian Peta Batas Administrasi

Hasil yang diperoleh dari penelitian ini adalah tampilan data spasial dan data Non Spasial informasi Tenda Di Kabupaten Gresik. Sehingga memungkinkan para pengguna informasi dapat dengan mudah mengakses data informasi di Kabupaten Gresik

IV.2. Penyajian Peta Batas Administrasi

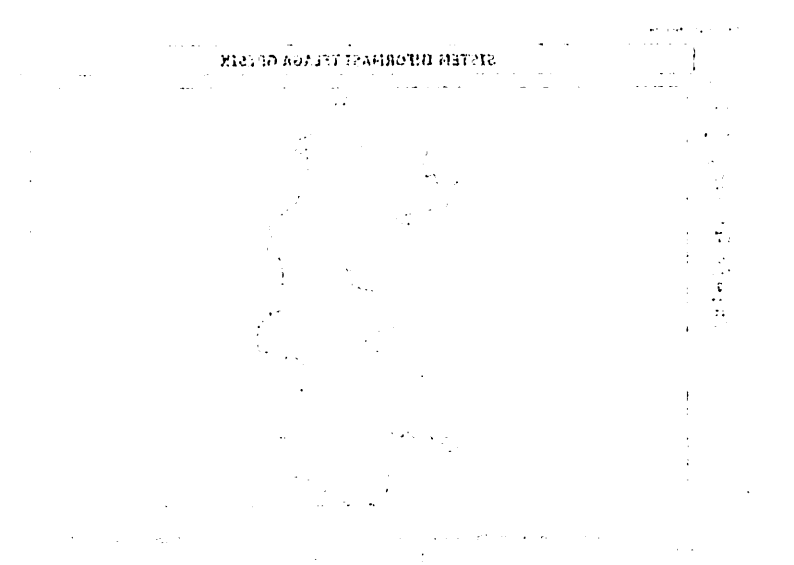
Penyajian peta pada program informasi Tenda di Kabupaten Gresik

ini terdiri dari :

a. Peta Batas Kota Malang

Menampilkan peta batas administrasi Kabupaten Gresik beserta beberapa data atribut seperti data jumlah Kecamatan, jumlah Kelurahan. Seperti terlihat pada

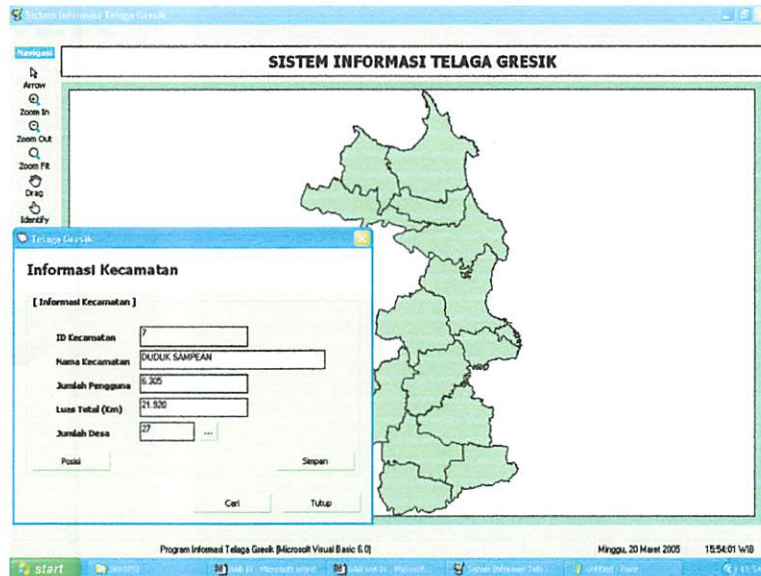
gambar 4.1 dibawah ini.



Gambar 4.1 Peta Batas Kabupaten Gresik

b. Peta Batas Kecamatan

Merupakan peta batas kecamatan yang ada di Kabupaten Gresik beserta data atribut Id_kecamatan, nama kecamatan, jumlah pengguna telaga (kk), jumlah kelurahan. Dengan informasi ini pengguna dapat mengetahui lokasi dari masing-masing kecamatan yang ada kota Malang beserta data atribut lainnya dengan mudah cukup dengan mengklik di masing-masing batas administrasi atau dengan menggunakan fasilitas “pencarian” yang tersedia pada program. Seperti terlihat pada gambar 4.2 dibawah ini.



Gambar 4.2 Peta Batas Kecamatan

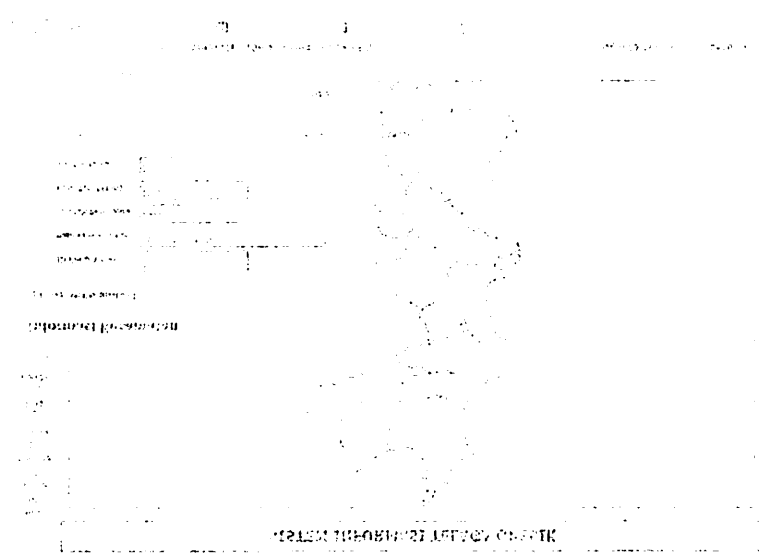
c. Peta Batas Kelurahan

Merupakan peta batas kelurahan yang ada di Kabupaten Gresik beserta data atribut pendukung seperti halnya pada peta batas kecamatan, yaitu data atribut Id_Kelurahan, nama kecamatan, jumlah telaga per kelurahan. Dengan informasi ini, pengguna bisa mencari lokasi kelurahan yang ada di kabupaten Gresik, baik itu

Բանցիտայի քիչ աստիճանի խոչընդոտման կայուն ծրագրի շրջանում կարևորագույն ժողովրդագրական և սոցիալական խնդիրների լուծման համար կարևորագույն է համարվում Երևանի մարզի կազմակերպչական և կրթական ծրագրի իրականացումը։

Ե. Երևանի մարզի կազմակերպչական

Շարքի 4.3 Երևանի մարզի կազմակերպչական

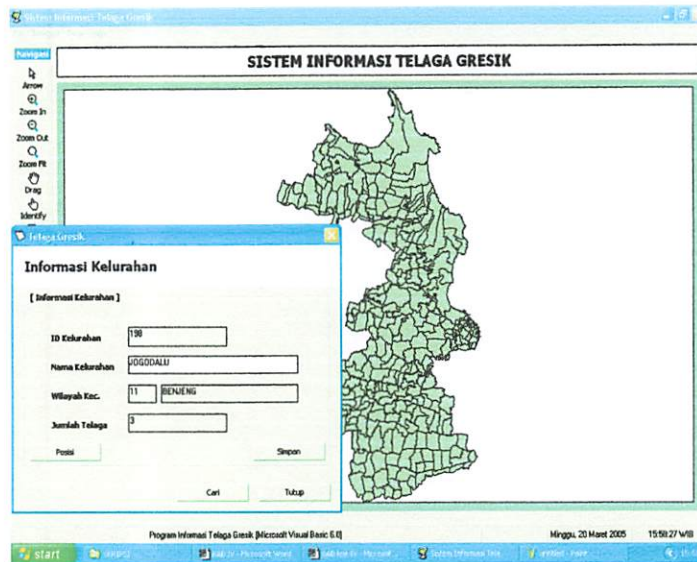


Շարքի 4.3 շարքի 4.3

Երևանի մարզի կազմակերպչական ծրագրի իրականացումը կարևորագույն է համարվում Երևանի մարզի կազմակերպչական և կրթական ծրագրի իրականացումը։

Զ. Երևանի մարզի կազմակերպչական

dengan langsung daerah yang dimaksud pada peta maupun dengan mengetik nama kelurahan pada kota pencarian yang tersedia dibawah peta. Pencarian lokasi juga dapat dilakukan melalui fasilitas “pencarian” yang terdapat pada program. Seperti terlihat pada gambar 4.3 dibawah ini.

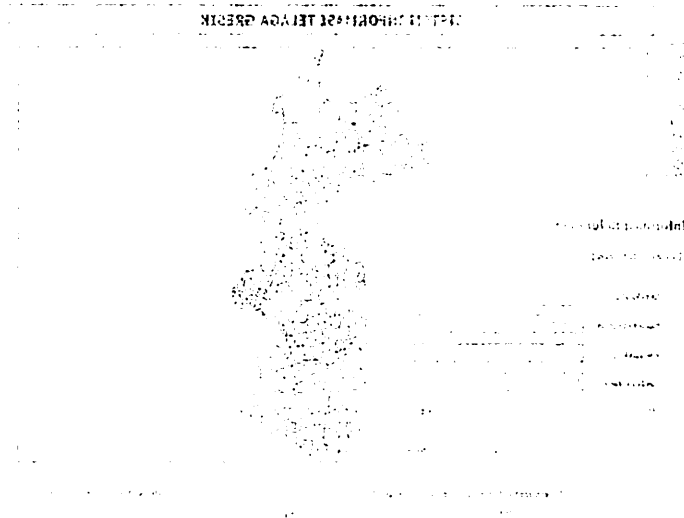


Gambar 4.3 Peta Batas Kelurahan

d. Peta Jaringan Jalan

Merupakan peta jaringan jalan yang ada Kabupaten Gresik. Berisi Id_ jalan Seperti halnya pada peta batas kecamatan dan kelurahan. Untuk mengetahui lokasi jalan dapat dilakukan dengan beberapa cara, yaitu dengan mengklik langsung pada peta jalan yang dimaksud, melalui *textbox* pencarian yang terdapat di bawah peta atau melalui fasilitas “pencarian” yang terdapat pada program. Seperti terlihat pada gambar 4.4 dibawah ini.

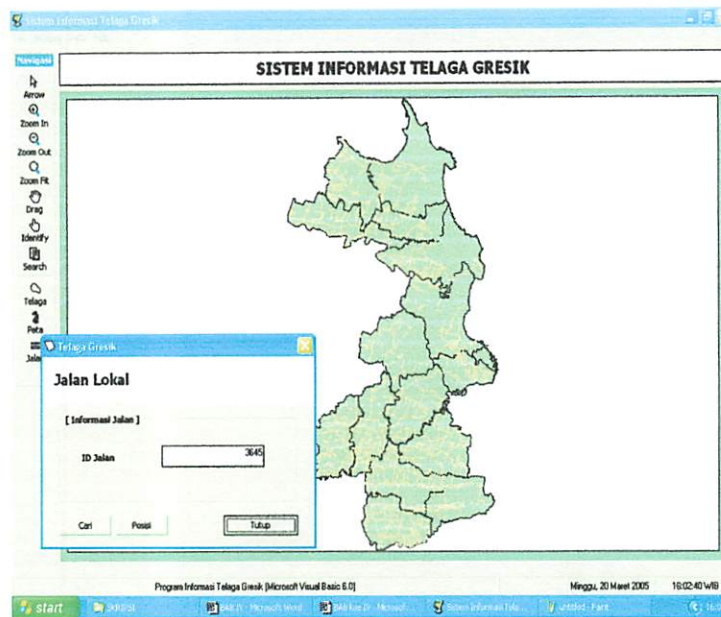
dengan langsung daerah yang dimaksud pada peta maupun dengan mengetik nama kelurahan pada kota pencairan yang tersedia dibawah peta. Pencairan lokasi juga dapat dilakukan melalui fasilitas "pencairan" yang terdapat pada program. Seperti terlihat pada gambar 4.3 dibawah ini.



Gambar 4.3 Peta Batas Kelurahan

d. Peta Jaringan Jalan

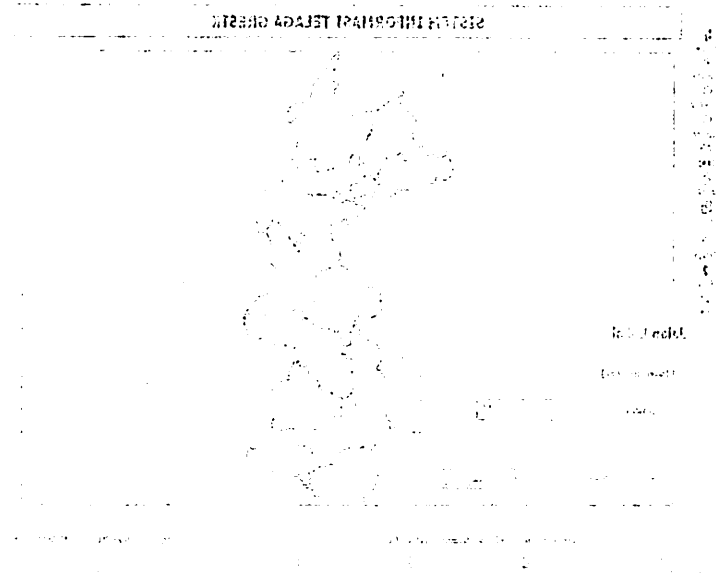
Merupakan peta jaringan jalan yang ada Kabupaten Gresik. Berisi id_jalan seperti halnya pada batas kecamatan dan kelurahan. Untuk mengetahui lokasi jalan dapat dilakukan dengan beberapa cara yaitu dengan mendklik langsung pada peta jalan yang dimaksud, melalui keypad pencairan yang terdapat di bawah peta atau melalui fasilitas "pencairan" yang terdapat pada program. Seperti terlihat pada gambar 4.4 dibawah ini.



Gambar 4.4 Peta Jaringan Jalan

e. Peta Lokasi Telaga

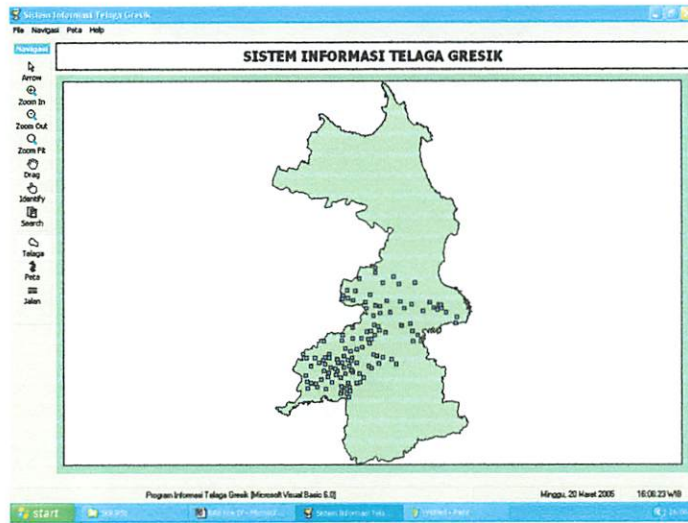
Merupakan peta lokasi Telaga yang ada di kabupaten Gresik.. Pada peta lokasi Telaga ini, pengguna mengetahui lokasi masing-masing Telagadengan cara mengklik langsung pada peta, mencari melalui *textbox* pencarian yang tersedia dibawah peta, juga melalui fasilitas “pencarian” yang terdapat pada program. Apabila lokasi Telaga tersebut ditemukan, maka Telaga tersebut akan berkedip atau menyala selama sekitar 3 detik. Dengan fasilitas ini diharapkan para pengguna dapat mengetahui secara cepat informasi posisi dan lokasi Telaga yang dimaksud. Seperti terlihat pada gambar 4.5



Gambar 4.4 Peta jaringan jalan

e. Peta Lokasi Telaga

Menunjukkan peta lokasi Telaga yang ada di Kabupaten Gresik. Pada peta lokasi Telaga ini, pengguna menggunakan lokasi masing-masing Telaga dengan cara mengklik langsung pada peta mencari melalui vector pencarian yang tersedia dibawah peta juga melalui fasilitas "pencarian" yang terdapat pada program. Apabila lokasi Telaga tersebut ditemukan, maka Telaga tersebut akan berkedip atau menyala selama sekitar 3 detik. Dengan fasilitas ini diharapkan para pengguna dapat mengetahui secara cepat informasi posisi dan lokasi Telaga yang dimaksud. Seperti terlihat pada gambar 4.5.



Gambar 4.5 Peta Lokasi Telaga di kabupaten Gresik

4.1 Penyajian Data Atribut Telaga di Kabupaten Gresik

Di dalam setiap data spasial Telaga, memiliki data atribut yang berisi berbagai informasi profil Telaga. Adapun data atribut terdiri atas : Id_ telaga, Nama Telaga Lokasi Kecamatan, Lokasi Kelurahan, Daya tampung air, fungsi, jumlah pengguna Luas total area telaga. Seperti terlihat pada gambar 4.6 di bawah ini.

Gambar 4.6 Data atribut Telaga di kabupaten Gresik

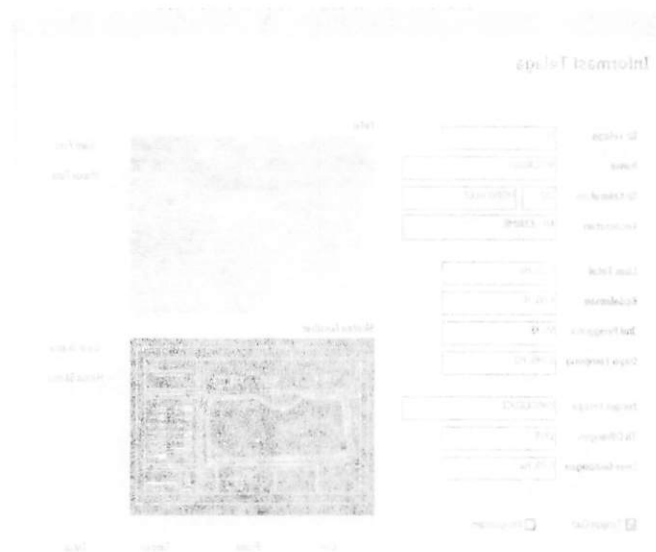
Pada *form* basisdata telaga, selain memiliki fungsi menyajikan berbagai informasi tentang Telaga juga dapat dilakukan *updating data* yaitu apabila terdapat



Gambar 4.5 Peta Lokasi Telaga di Kabupaten Gresik

4.1 Penyajian Data Atribut Telaga di Kabupaten Gresik

Di dalam setiap data spasial telaga, memiliki data atribut yang berisi berbagai informasi profil telaga. Adapun data atribut terdiri atas : `id_telaga`, `Nama Telaga`, `Lokasi Kecamatan`, `Lokasi Kelurahan`, `Daya tampung air`, `fungsi`, `jumlah pengguna`. Luas total area telaga. Seperti terlihat pada gambar 4.6 di bawah ini.



Gambar 4.6 Data atribut Telaga di kabupaten Gresik

Pada form basisdata telaga, selain memiliki fungsi menyajikan berbagai informasi tentang Telaga juga dapat dilakukan `update` data yaitu apabila terdapat

perbaikan pada data yang ada dan penambahan data atribut baru sesuai dengan tahun ajaran baru.

Adapun cara untuk melakukan *updating data*, yaitu :

1. Bukalah jendela Telaga yang akan dilakukan *updating data*, Klik icon *data* untuk masuk ke dalam database Pondok Pesantren.
2. Cari nama Telaga yang akan dilakukan *updating*, dengan melalui tombol *cari* untuk memunculkan *form* pencarian.
3. Setelah menemukan Telaga yang dimaksud, klik tombol *edit*, agar *updating data* dapat dilakukan.
4. Setelah selesai melakukan *updating*, baik itu pengeditan maupun penambahan data baru sesuai dengan tahun ajaran, maka dilakukan penyimpanan data dengan menekan tombol *simpan*.
5. Apabila *updating data* telah selesai dilakukan maka *form* database Telaga dapat ditutup dengan menekan tombol *tutup*.

Dengan adanya informasi data atribut dari masing-masing Telaga diharapkan pengguna dapat mengetahui secara cepat dan lengkap informasi yang berhubungan dengan Telaga tersebut. Untuk dijadikan bahan masukan dalam pengambilan keputusan

4.3 Penyajian Pencarian

Untuk memudahkan pengguna dalam mencari lokasi/posisi Telaga yang diinginkan, maka disediakan *form* pencarian. Bentuk *form* pencarian ini terbagi atas dua bagian, yaitu :

berisikan pada data yang ada dan penambahan data akibat dari sesuai dengan tahun ajaran baru.

Adapun cara untuk melakukan update data yaitu :

1. Bukalah jendela Tglga yang akan dilakukan update data. Klik icon data untuk masuk ke dalam database Pondok Pesantren.

2. Cari nama Tglga yang akan dilakukan update, dengan menekan tombol cari untuk menemukan form pencarian.

3. Setelah menemukan Tglga yang dimaksud, klik tombol edit agar update data dapat dilakukan.

4. Setelah selesai melakukan update, baik itu pengeditan maupun penambahan data baru sesuai dengan tahun ajaran, maka dilakukan penyimpanan data dengan menekan tombol simpan.

5. Apabila update data telah selesai dilakukan maka form database Tglga dapat ditutup dengan menekan tombol www.

Dengan adanya informasi data akibat dari masing-masing Tglga diharapkan pengguna dapat mengetahui secara cepat dan lengkap informasi yang berhubungan dengan Tglga tersebut. Untuk dilakukan bahan masukan dalam pengembangan ke depannya.

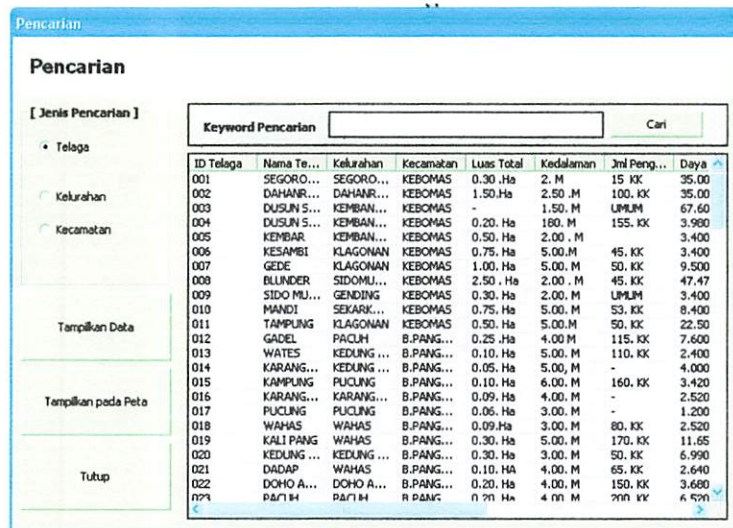
4.3. Penyajian Pencarian

Untuk memudahkan pengguna dalam mencari lokasi-posisi Tglga yang diinginkan, maka disediakan form pencarian. Bentuk form pencarian ini terbagi atas

dua bagian, yaitu :

1. Pencarian dengan menggunakan *form* pencarian

Form pencarian berisikan *kriteria pencarian*, agar pengguna dapat memilih informasi apa yang ingin diketahui posisinya. Yang kedua yaitu *Daftar Pencarian*, yaitu daftar obyek yang berada pada setiap kriteria, yang ketiga yaitu *Hasil Pencarian*, berisi obyek yang dicari / diinginkan oleh pengguna., contoh penggunaannya : apabila dalam *kriteria pencarian* pengguna memilih kecamatan, maka secara otomatis di dalam *daftar pencairan* berisi daftar kecamatan yang ada di Kabupaten Gresik kemudian apabila pengguna memilih salah satu dari kecamatan tersebut, misalnya kecamatan Cerme maka di dalam *hasil pencarian* akan tertera Cerme



Gambar 4.7 Penyajian Form Pencarian

Dengan fasilitas pencarian ini, diharapkan pengguna program ini mendapatkan kemudahan dalam menemukan atau mencari setiap lokasi batas-batas administrasi maupun lokasi masing-masing Pondok Pesantren yang ada di kota Malang.

4.4 Penyajian Help

Fasilitas *Help* (pertolongan) ini dibuat dengan maksud untuk membantu pengguna untuk mengetahui bagaimana cara menjalankan atau mengoperasikan program informasi Pondok Pesantren ini.⁶⁰ Sehingga apabila pengguna mengalami

1. Pencarian dengan menggunakan form pencarian

Form pencarian berbasis kriteria pencarian agar pengguna dapat memilih informasi apa yang ingin diketahuinya. Yang kedua yaitu *Differ Pencarian*, yaitu daftar objek yang berada pada setiap kriteria yang ketiga yaitu *Visual Pencarian*, berisi objek yang dicari & diinginkan oleh pengguna. contoh penggunaannya : apabila dalam kriteria pencarian pengguna memilih kecamatan maka secara otomatis di dalam *Visual Pencarian* berisi daftar kecamatan yang ada di Kabupaten Gresik kemudian apabila pengguna memilih salah satu dari kecamatan tersebut misalnya kecamatan Cerme maka di dalam *Visual Pencarian* akan tertera

Cerme

Pencarian		[Kecamatan]	
+ Test			
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	42
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	51	51	51
52	52	52	52
53	53	53	53
54	54	54	54
55	55	55	55
56	56	56	56
57	57	57	57
58	58	58	58
59	59	59	59
60	60	60	60
61	61	61	61
62	62	62	62
63	63	63	63
64	64	64	64
65	65	65	65
66	66	66	66
67	67	67	67
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	76	76	76
77	77	77	77
78	78	78	78
79	79	79	79
80	80	80	80
81	81	81	81
82	82	82	82
83	83	83	83
84	84	84	84
85	85	85	85
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	94
95	95	95	95
96	96	96	96
97	97	97	97
98	98	98	98
99	99	99	99
100	100	100	100

Gambar 4.7. Penyajian Form Pencarian

Dengan fasilitas pencarian ini, diharapkan pengguna program ini mendapatkan kemudahan dalam menemukan atau mencari setiap lokasi batas-batas administrasi maupun lokasi masing-masing Pondok Pesantren yang ada di Kota Malang.

4.4 Penyajian Help

Fasilitas *Web* (petolongan) ini dibuat dengan maksud untuk membantu pengguna untuk mengetahui bagaimana cara menjalarkannya atau mengoperasikan program informasi Pondok Pesantren ini. Sehingga apabila pengguna mengalami

kesulitan dalam menjalankan program maka pengguna dapat tertolong dengan adanya fasilitas *help* ini. Fasilitas *help* ini dibagi dalam beberapa bagian untuk memudahkan pengguna dalam memahami program, yaitu menu utama, menu Pondok Pesantren, basisdata Pondok Pesantren, dan pencarian. Adapun bentuk dari fasilitas *help* ini dapat dilihat pada gambar 4.8 di bawah ini.



Gambar 4.8 Penyajian Help

Kesulitan yang timbul pada saat pembuatan program antara lain :

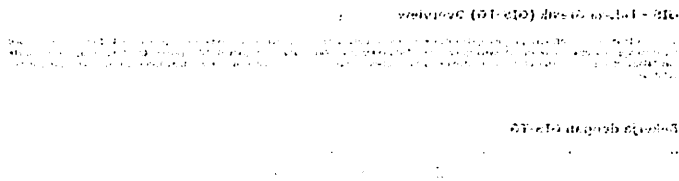
- Pada tahap pembuatan program dengan *Visual Basic*, *design* tampilan program secara keseluruhan seperti penempatan tombol-tombol perintah sehingga terlihat sesuai dengan obyek menu yang ada pada *form*.
- Pada tahapan pembuatan penyajian informasi posisi Telaga yang digabungkan dengan *basisdata (atribut)* yang dibuat oleh *software* yang berbeda.

Untuk menyelesaikan kesulitan-kesulitan di atas perlu dicarikan cara pemecahannya, antara lain :

- *Desain form* banyak diilhami dari program-program yang sudah ada dan saran dari *user/pengguna/masyarakat*.

kesulitan dalam menjalankan program maka pengguna dapat menolong dengan adanya fasilitas help ini. Fasilitas help ini dibagi dalam beberapa bagian untuk memudahkan pengguna dalam memahami program. yaitu menu utama menu fondok Pesantren, basisdata fondok Pesantren dan pencarian. Adapun bentuk dari fasilitas help ini dapat dilihat pada gambar 4.8 di bawah ini.

Gambar 4.8



Gambar 4.8 Fasilitas Help

- Kesulitan yang timbul pada saat pembuatan program antara lain :
- Pada tahap pembuatan program dengan Visual Basic design tampilan program secara keseluruhan seperti penempatan tombol-tombol perintah sehingga terlihat sesuai dengan objek menu yang ada pada form.
 - Pada tahapan pembuatan penyajian informasi posisi Tglaga yang digabungkan dengan basisdata (array) yang dibuat oleh software yang berbeda.
- Untuk menyelesaikan kesulitan-kesulitan di atas perlu dicarikan cara pemecahannya antara lain :
- Desain form banyak diambil dari program-program yang sudah ada dan saran dari penggunaannya sangat.

- Pada penyusunan *basisdata* yang benar dan disesuaikan dengan atributnya masing-masing maka dengan beberapa perintah dari *Visual Basic*, hal ini dapat dijalankan sesuai dengan kebutuhan yang diperlukan.

IV.5 Kelemahan Dan Kelebihan Pada Penggunaan Program Visual Basic 6.0 Dan Map Object 2.1

IV.5.1 Kelemahan Program

Dalam penggunaan program pada penelitian ini, diketahui beberapa kelemahan seperti :

- *Map Object 2.1* merupakan program penelitian untuk menampilkan data spasial, dimana program *Map Object 2.1* ini harus di *up-date* dengan waktu tertentu (memiliki batas waktu penggunaan / *expired date*)

IV.5.2 Kelebihan Program

- Pada penelitian ini merupakan penelitian dengan penggunaan program yang masih terbilang baru dalam suatu penyajian informasi, dimana *Map Object* sebagai penyajian peta (posisi) dan dengan mengoptimalkan perintah-perintah dalam bahasa pemrograman *Visual Basic*.
- Mampu melakukan pencarian data yang cepat dan sistem *updating* yang mudah dipahami dan langsung pada informasi obyek yang akan di *update*.
- Pada data spasial, informasi obyek yang telah ter-*updating* dapat diketahui dan terhubung langsung (*Zoom*) sebagai obyek terpilih.
- Penggunaan tombol-tombol yang mudah dipahami dan *aplikatif* pada pengguna dan dapat dilakukan desain tampilan yang lebih menarik dan variatif, seperti pemberian *icon*, foto dan lain sebagainya.
- Program dapat melakukan *updating* data spasial, tanpa harus mengerjakan dari awal .

- Pada penyusunan prosedur yang benar dan disesuaikan dengan kebutuhan masing-masing maka dengan perintah dari Visual Basic ini dapat dijalankan sesuai dengan kebutuhan yang diperlukan.

IV.5. Kelemahan Dan Kelebihan Pada Penggunaan Program Visual Basic 6.0

Dan Map Object 2.1

IV.5.1 Kelemahan Program

Dalam penggunaan program pada penelitian ini, diketahui beberapa kelemahan seperti :

- Map Object 2.1 merupakan program penelitian untuk menampilkan data spasial. dimana program Map Object 2.1 ini harus di update dengan waktu tertentu (memiliki batas waktu penggunaan / expired date)

IV.5.2 Kelebihan Program

- Pada penelitian ini merupakan penelitian dengan penggunaan program yang masih terbilang baru dalam suatu penyajian informasi, dimana Map Object sebagai penelitian peta (posisi) dan dengan mengoptimalkan perintah-perintah dalam bahasa pemrograman Visual Basic.
- Mampu melakukan pencarian data yang cepat dan sistem aplikasi yang mudah dipahami dan langsung pada informasi objek yang akan di akses.
- Pada data spasial, informasi objek yang telah ter-wykwat dapat diketahui dan terhubung langsung (zoom) sebagai objek terdapat.
- Penggunaan tombol-tombol yang mudah dipahami dan aplikasi pada penggunaan dan dapat dilakukan desain tampilan yang lebih menarik dan variatif seperti pemberian color foto dan lain sebagainya.
- Program dapat melakukan aplikasi data spasial, tanpa harus mengerjakan data

BAB V

PENUTUP

V.1 Kesimpulan

Dengan adanya kemajuan teknologi, maka komputer dapat digunakan untuk pembuatan sistem *basis data* spasial maupun *basis data* non spasial .

Dari pelaksanaan tugas akhir ini dapat ditarik kesimpulan sebagai berikut :

1. Bahasa Program *Visual Basic* yang dihubungkan dengan *Map Object 2.1* mampu untuk mempermudah pencarian posisi Telaga di kabupaten Gresik. Data yang di informasikan berupa data spasial dan non spasial yang dapat di *update* dan di *edit* secara langsung
2. Pada program informasi Telaga ini mempunyai kemampuan :
 - a. Pada menu peta menampilkan informasi yang variatif yang antara lain menu peta Kota, Kecamatan, Kelurahan, Jalan dan Peta lokasi Telaga
 - b. Pada menu *file* menampilkan informasi tentang data Kabupaten Gresik, data Kecamatan, data Kelurahan, data Jalan dan khusus untuk data Profil Telaga . bisa dilakukan pembaruan dan penghapusan data (*up dating*).
 - c. Pada menu pencarian menampilkan pencarian secara cepat tentang data yang mungkin diperlukan oleh pengguna, yang antara lain seperti data kota, data kecamatan, data kelurahan, data jalan dan data Telaga kabupaten Gresik
 - d. Dengan adanya menu bantuan diharapkan bisa memudahkan pengguna dalam menjalankan aplikasi program
3. Dari hasil penelitian ini dapat di ketahui jumlah telaga yang tersebar di Kabupaten Gresik sejumlah 122 telaga yang terletak di 6 Kecamatan Dari 16 yang ada di kabupaten Gresik antara lain : Kecamatan Balong Panggang, Kecamatan Manyar, Kecamatan Benjeng, Kecamatan Cerme, Kecamatan Kebo Mas, Kecamatan Duduk sampean

4. Kecamatan yang paling banyak mempunyai telaga adalah Kecamatan Benjeng dengan jumlah 34 telaga, Sedangkan kecamatan yang paling sedikit mempunyai telaga adalah Kecamatan Manyar dengan jumlah 1 telaga.
5. Telaga yang memiliki luas terbesar adalah telaga Jono yang terletak di Kecamatan Cerme dengan luas 4 Ha, sedangkan yang terkecil adalah telaga dusun sumber yang terletak di Kecamatan Kebomas 1 Ha.
6. Di Kabupaten Gresik memiliki 122 telaga yang memiliki fungsi untuk mandi sejumlah 33 buah telaga, untuk cuci / mandi 37 buah telaga, untuk minum 25 buah telaga, untuk MCK (mandi cuci, kakus) 5 buah telaga, untuk kolam pemancingan 16 buah telaga, untuk wisata 6 buah telaga,

V.2 Saran

Paket program yang dihasilkan dari penelitian ini masih jauh dari sempurna, akan tetapi bisa dijadikan sebagai titik awal dari penggunaan komputer guna membantu para pengguna data agar nantinya dapat dijadikan salah satu dasar dalam pengambilan keputusan untuk meningkatkan kualitas dan kuantitas Telaga yang ada di Kabupaten Gresik. Oleh karena itu untuk lebih memaksimalkan dari paket program penyajian informasi Telaga di kabupaten Gresik ini hendaknya ada pengembangan lebih lanjut dari pada program ini yang dilengkapi dengan sistem informasi geografi, sehingga pengguna dapat mengetahui informasi-informasi baik itu data spasial dan non spasial yang berhubungan dengan telaga di kabupaten Gresik

Daftar pustaka

1. Ir. Iman Subarkah Bangunan Air, Penerbit Idea Dharma Agung Bandung
2. Ir CD Soemanto, B.I.E. Dipl. Hidrologi Teknik Penerbit Usaha Nasional Surabaya
3. Dinas PU PENGAIRAN Derah Tingkat II Kabupaten Gresik Profil Waduk Dan Telaga
4. Drs. Erman Mawardi .Dipl. AIT, Ir.M. Memed ,Dipl. H.E, APU. Desain Hidraulik BENDUNG TETAP. Penerbit ALBETA
5. Step by step microsoft visual basic 6.0, michael harvorson , penerbit pt. Elexmedia komputindo
6. Buku latihan visual basic 6.0 ,Ario Suryo kusumo, Penerbit PT. Elexmedia Komputindo
7. Tip dan trik pemrograman Visual Basic 6.0, Ir.Pamungkas. PT. Elexmedia Komputindo.
8. Mengolah Data Spasial dengan Map Info Profesional , I wayan Nuarsa , Penerbit Andi Yogyakarta
9. Sistem Informasi Geografis Menggunakan MapInfo, Eko Bidiyanto Penerbit Andi, Yogyakarta

FORM 1

```
Private Sub Form_Resize()  
    Image1.Left = 0  
    Image1.Top = 0  
    Image1.Width = Me.Width  
    Image1.Height = Me.Height  
  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    Set Me.Image1.Picture = Nothing  
End Sub  
  
Private Sub Image1_Click()  
    Unload Me  
End Sub
```

FORM CARI

```
Dim strsql As String  
Private rs As DAO.Recordset  
  
Private Sub cmdCari_Click()  
    If Option1(0).Value = True Then  
        Cari "Telaga", txtKey  
    ElseIf Option1(1).Value = True Then  
        Cari "Kelurahan", txtKey  
    ElseIf Option1(2).Value = True Then  
        Cari "kecamatan", txtKey  
    End If  
    txtKey = ""  
End Sub  
  
Private Sub cmdKeluar_Click()  
    Unload Me  
    frmMain.Enabled = True  
End Sub  
  
Private Sub cmdTampilData_Click()  
    If Option1(0).Value = True Then  
        frmMain.PetaWilayah = "kabupaten"  
        frmMain.TampilJalan = False  
        frmMain.TampilTelaga = True  
        frmMain.RefreshMap  
        frmTelaga.GetData Val(LV.SelectedItem)  
    ElseIf Option1(1).Value = True Then  
        frmMain.TampilJalan = False  
        frmMain.TampilTelaga = False
```

```
frmMain.PetaWilayah = "kelurahan"  
frmMain.RefreshMap  
  
frmWilayah.GetData "kelurahan", Val(LV.SelectedItem)  
Elseif Option1(2).Value = True Then  
    frmMain.TampilJalan = False  
    frmMain.TampilTelaga = False  
    frmMain.PetaJalan = "kecamatan"  
    frmMain.RefreshMap  
    frmWilayah.GetData "kecamatan", Val(LV.SelectedItem)  
End If  
Unload Me  
frmMain.Enabled = False  
End Sub  
  
Private Sub cmdTampilPosisi_Click()  
    Dim recs As mapobjects2.Recordset  
    Dim shp As Object  
    Dim exp As String  
  
    Me.Hide  
    If Option1(0).Value = True Then  
        frmMain.TampilTelaga = True  
        frmMain.TampilJalan = False  
        frmMain.RefreshMap  
        exp = "Telaga_ID = " & Val(LV.SelectedItem)  
        Set recs = frmMain.Map1.Layers("telaga").SearchExpression(exp)  
  
    Elseif Option1(1).Value = True Then  
        frmMain.PetaWilayah = "kelurahan"  
        frmMain.TampilTelaga = False  
        frmMain.TampilJalan = False  
        frmMain.RefreshMap  
        exp = "ID_Kelurah = " & LV.SelectedItem  
        Set recs = frmMain.Map1.Layers("kelurahan").SearchExpression(exp)  
  
    Elseif Option1(2).Value = True Then  
        frmMain.PetaWilayah = "kecamatan"  
        frmMain.TampilTelaga = False  
        frmMain.TampilJalan = False  
        frmMain.RefreshMap  
        exp = "ID_Kecamat = " & LV.SelectedItem  
        Set recs = frmMain.Map1.Layers("kecamatan").SearchExpression(exp)  
  
    End If  
  
    If Not recs.EOF Then  
        Set shp = recs.Fields("Shape").Value  
        frmMain.Refresh  
    End If  
End Sub
```



```
        frmMain.Map1.FlashShape shp, 3
    End If

    Me.Show

End Sub

Private Sub cmdX_Click()
    Unload Me
    frmMain.Enabled = True
End Sub

Private Sub Form_Load()
    frmMain.Enabled = False
    Option1_Click 0
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Set rs = Nothing
End Sub

Private Sub Form_Unload(Cancel As Integer)
    frmMain.Enabled = True
End Sub

Private Sub LV_DbClick()
    cmdTampilData_Click
End Sub

Private Sub LV_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmdTampilData_Click
End Sub

Private Sub Option1_Click(Index As Integer)
    Select Case Index
        Case 0
            Cari "Telaga"
        Case 1
            Cari "Kelurahan"
        Case 2
            Cari "Kecamatan"
    End Select
End Sub

Sub Cari(FindWhat As String, Optional KeyWord As String)
    Dim LI As ListItem
    On Error Resume Next
    With LV
        .ColumnHeaders.Clear
```

```
keycari = "*" & KeyWord & "*"
Select Case UCase(FindWhat)
Case "TELAGA"
.ColumnHeaders.Add , , "ID Telaga", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Nama Telaga", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Kelurahan", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Kecamatan", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Luas Total", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Kedalaman", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Jml Pengguna", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Daya Tampung", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Fungsi", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Th Dibangun", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Luas Genangan", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Tangga Cuci", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Plengsengan", 1000, lvwColumnLeft
strsql = "SELECT Telaga.*, kelurahan>Nama, kecamatan>Nama " & _
"FROM kecamatan INNER JOIN (kelurahan INNER JOIN Telaga ON
kelurahan.IDKel = Telaga.IDKel) " & _
"ON kecamatan.IDKec = kelurahan.IDKec "

If KeyWord <> "" Then
strsql = strsql & _
"WHERE str(telaga.telaga_id) LIKE " & keycari & " OR telaga.nama_telaga LIKE
" & keycari & " OR kecamatan.nama LIKE " & keycari & " OR kelurahan.nama LIKE " &
keycari & " OR telaga.fungsi_telaga LIKE " & keycari & " OR telaga.th_dibangun LIKE " &
keycari & " "
End If
strsql = strsql & "ORDER BY Telaga.Telaga_ID ASC"
Set rs = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
.ListItems.Clear
While Not rs.EOF
Set LI = .ListItems.Add()
LI.Text = Format(rs("Telaga_ID"), "000")
LI.SubItems(1) = rs("Nama_Telaga")
LI.SubItems(2) = rs("Kelurahan>Nama")
LI.SubItems(3) = rs("Kecamatan>Nama")
LI.SubItems(4) = rs("Luas_Total")
LI.SubItems(5) = rs("Kedalaman")
LI.SubItems(6) = rs("Jml_Pengguna")
LI.SubItems(7) = rs("Daya_Tampung")
LI.SubItems(8) = rs("Fungsi_Telaga")
LI.SubItems(9) = rs("Th_Dibangun")
LI.SubItems(10) = rs("Luas_Genangan")
If rs("Tangga_Cuci").Value = True Then
LI.SubItems(11) = "Ada"
Else
LI.SubItems(11) = "Tidak Ada"
End If
End While
```

```
If rs("Plengsengan").Value = True Then
    LI.SubItems(12) = "Ada"
Else
    LI.SubItems(12) = "Tidak Ada"
End If
LI.Tag = rs("Telaga_ID")
rs.MoveNext
Wend

Case "KELURAHAN"
.ColumnHeaders.Add , , "ID Kelurahan", 1000, lvwColumnLeft
.ColumnHeaders.Add , , "Nama Kelurahan", 3000, lvwColumnLeft
.ColumnHeaders.Add , , "Kecamatan", 3000, lvwColumnLeft
strsql = "SELECT kelurahan.*, kecamatan.Nama " & _
        "FROM kecamatan INNER JOIN kelurahan ON kecamatan.IDKec = kelurahan.IDKec
"
If KeyWord <> "" Then
    strsql = strsql & _
        "WHERE str(kelurahan.idkel) LIKE " & keycari & " OR kelurahan.nama LIKE " &
keycari & " OR kecamatan.nama LIKE " & keycari & " "
End If
strsql = strsql & "ORDER BY Kelurahan.IDKel ASC"
Set rs = DBSIG.OpenRecordset(strsql, dbOpenDynaset)

.ListItems.Clear
While Not rs.EOF
    Set LI = .ListItems.Add()
    LI.Text = Format(rs("IDKel"), "000")
    LI.SubItems(1) = rs("Kelurahan.Nama")
    LI.SubItems(2) = rs("Kecamatan.Nama")
    LI.Tag = rs("IDKel")
    rs.MoveNext
Wend

Case "KECAMATAN"
.ColumnHeaders.Add , , "ID Kecamatan", 1200, lvwColumnLeft
.ColumnHeaders.Add , , "Nama Kecamatan", 3000, lvwColumnLeft
.ColumnHeaders.Add , , "Luas Wilayah (KM)", 2000, lvwColumnLeft
strsql = "SELECT * FROM Kecamatan "
If KeyWord <> "" Then
    strsql = strsql & _
        "WHERE nama LIKE " & keycari & " OR str(IDKec) LIKE " & keycari & " "
End If
strsql = strsql & "ORDER BY IDKec ASC "
Set rs = DBSIG.OpenRecordset(strsql, dbOpenDynaset)

.ListItems.Clear
While Not rs.EOF
    Set LI = .ListItems.Add()
```

```
        LI.Text = Format(rs("IDKec"), "000")
        LI.SubItems(1) = rs("Nama")
        LI.SubItems(2) = rs("Luas")
        LI.Tag = rs("IDKec")
        rs.MoveNext
    Wend
End Select
End With
End Sub

Private Sub Picture1_Click()

End Sub

Private Sub txtKey_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmdCari_Click
End Sub

Private Sub LV_ColumnClick(ByVal ColumnHeader As _
    MSCComctlLib.ColumnHeader)
    ' Sort according to data in this column.
    If LV.Sorted And _
        ColumnHeader.Index - 1 = LV.SortKey Then
        ' Already sorted on this column, just invert the sort order.
        LV.SortOrder = 1 - LV.SortOrder
    Else
        LV.SortOrder = lvwAscending
        LV.SortKey = ColumnHeader.Index - 1
    End If
    LV.Sorted = True
End Sub
```

FORM HELP

```
Private Sub Form_Load()
    wb1.Navigate App.Path & "\help\help.htm"
End Sub

Private Sub Form_Resize()
    If Me.WindowState <> vbMinimized Then
        wb1.Left = 120
        wb1.Top = 120
        wb1.Height = Me.ScaleHeight - 240
        wb1.Width = Me.ScaleWidth - 240
    End If
End Sub
```

FORM JALAN

```
Dim strsql As String
Private rsJLAlteri As DAO.Recordset
Private rsJLLokal As DAO.Recordset
Private rsJLTol As DAO.Recordset
Private rsJLKabupaten As DAO.Recordset
Private rsJLKolektor As DAO.Recordset
Private vPetaJ As String
```

```
Public Sub GetData(PetaJ As String, ID As Integer)
```

```
On Error Resume Next
```

```
vPetaJ = PetaJ
```

```
Select Case PetaJ
```

```
Case "jalan_alteri"
```

```
LabTitle.Caption = "Jalan Alteri"
```

```
strsql = "SELECT * FROM jalan_alteri"
```

```
Set rsJLArteri = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
```

```
rsJLArteri.FindFirst "IDRuas = " & ID
```

```
txtID = ID
```

```
Me.Show , frmMain
```

```
Exit Sub
```

```
Case "jalan_lokal"
```

```
LabTitle.Caption = "Jalan Lokal"
```

```
strsql = "SELECT * FROM jalan_lokal"
```

```
Set rsJLLokal = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
```

```
rsJLLokal.FindFirst "IDRuas = " & ID
```

```
txtID = ID
```

```
Me.Show , frmMain
```

```
Exit Sub
```

```
Case "jalan_kolektor"
```

```
LabTitle.Caption = "Jalan Kolektor"
```

```
strsql = "SELECT * FROM jalan_kolektor"
```

```
Set rsJLKolektor = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
```

```
rsJLKolektor.FindFirst "IDRuas = " & ID
```

```
txtID = ID
```

```
Me.Show , frmMain
```

```
Exit Sub
```

Case "jalan_tol"

LabTitle.Caption = "Jalan Tol"

strsql = "SELECT * FROM jalan_tol"

Set rsJLTol = DBSIG.OpenRecordset(strsql, dbOpenDynaset)

rsJLTol.FindFirst "IDRuas = " & ID

txtID = ID

Me.Show , frmMain

Exit Sub

Case "jalan_kabupaten"

LabTitle.Caption = "Jalan Kabupaten"

strsql = "SELECT * FROM jalan_kabupaten"

Set rsJLKabupaten = DBSIG.OpenRecordset(strsql, dbOpenDynaset)

rsJLKabupaten.FindFirst "IDRuas = " & ID

txtID = ID

Me.Show , frmMain

Exit Sub

End Select

End Sub

Private Sub cmdCari_Click()

Unload Me

frmCari.Show , frmMain

End Sub

Private Sub cmdPosisi_Click()

Dim recs As mapobjects2.Recordset

Dim shp As Object

Dim rect As mapobjects2.Rectangle

Dim exp As String

Me.Hide

Select Case vPetaJ

Case "jalan_alteri"

exp = "jalan_ = " & txtID

Set recs = frmMain.Map1.Layers("jalan_alteri").SearchExpression(exp)

Case "jalan_lokal"

exp = "idruas = " & txtID

Set recs = frmMain.Map1.Layers("jalan_lokal").SearchExpression(exp)

Case "jalan_kolektor"

exp = "kolektor_ = " & txtID

```
Set recs = frmMain.Map1.Layers("jalan_kolektor").SearchExpression(exp)
Case "jalan_tol"
  exp = "tol_id = " & txtID
  Set recs = frmMain.Map1.Layers("jalan_tol").SearchExpression(exp)
Case "jalan_kabupaten"
  exp = "jln_kab_id = " & txtID
  Set recs = frmMain.Map1.Layers("jalan_kabupaten").SearchExpression(exp)
End Select

If Not recs.EOF Then
  Set shp = recs.Fields("Shape").Value
  frmMain.Refresh
  frmMain.Map1.FlashShape shp, 3
End If
Me.Show
End Sub

Private Sub cmdTutup_Click()
  Unload Me
End Sub

Private Sub Form_Load()
  frmMain.Enabled = False
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
  Set rsJLAlteri = Nothing
  Set rsJLLokal = Nothing
  Set rsJLTol = Nothing
  Set rsJLKabupaten = Nothing
  Set rsJLKolektor = Nothing
End Sub

Private Sub Form_Unload(Cancel As Integer)
  frmMain.Enabled = True
End Sub

Private Sub Picture1_Click()

End Sub
```

FORM MAIN

```
Public PetaJalan As String
Public PetaWilayah As String
Public TampilJalan As Boolean
Public TampilTelaga As Boolean
```

```
Private Sub cmdPicJalan_Click()
    picJalan.Visible = False
End Sub
```

```
Private Sub cmdPicPeta_Click()
    picPeta.Visible = False
End Sub
```

```
Private Sub Form_Load()
    PetaWilayah = "kabupaten"
    TampilJalan = False
    TampilTelaga = False
    RefreshMap
    Set MapRect = Map1.Extent
End Sub
```

```
Private Sub Form_Resize()
    If Me.WindowState = vbMinimized Then Exit Sub
    If Me.Height < (480 * 15) Then Me.Height = (480 * 15)
    If Me.Width < (640 * 15) Then Me.Width = (640 * 15)
```

```
    picBG.Left = (2 * Toolbar1.Left) + Toolbar1.Width
    picBG.Top = 840
    picBG.Width = Me.ScaleWidth - picBG.Left - 120
    picBG.Height = Me.ScaleHeight - picBG.Top - StatusBar1.Height - 240
```

```
    labJudul.Top = 240
    labJudul.Left = picBG.Left
    labJudul.Width = picBG.Width
```

```
    Shape2.Left = labJudul.Left
    Shape2.Width = labJudul.Width
```

```
    With Me.Map1
        .Left = 120
        .Top = 120
        .Width = picBG.Width - 240 - 120
        .Height = picBG.Height - 240 - 120
```

```
    End With
End Sub
```



```
Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 2 Then
        PopupMenu mnuNav
    Else
        Select Case Map1.MousePointer
            Case moZoomIn
                Set Map1.Extent = Map1.TrackRectangle
                Set MapRect = Map1.Extent
            Case moZoomOut
                Dim rect As mapobjects2.Rectangle
                Set rect = Map1.Extent
                rect.ScaleRectangle 2
                Map1.Extent = rect
                Set MapRect = Map1.Extent
            Case moPan
                Map1.Pan
                Set MapRect = Map1.Extent
            Case moIdentify
                ' DoIdentify X, Y
            Case moHotLink
                DoIdentifyWilayah X, Y
        End Select
    End If
End Sub
```

```
Sub DoIdentifyWilayah(X As Single, Y As Single)
    Dim MyLayer As New MapLayer
    Dim MyPoint As New Point
    Dim recs As mapobjects2.Recordset
    Dim shp As Variant
```

```
    If TampilTelaga Then
        Set MyLayer = Map1.Layers("telaga")
        Set MyPoint = Map1.ToMapPoint(X, Y)

        ' perform the search
        If MyLayer.shapeType = moPolygon Then
            Set recs = MyLayer.SearchShape(MyPoint, moPointInPolygon, "")
        Else
            Set recs = MyLayer.SearchByDistance(MyPoint, Map1.ToMapDistance(100), "")
        End If

        If Not recs.EOF Then
            Set shp = recs.Fields("shape").Value
            Map1.Refresh ' force redraw of map
            Map1.FlashShape shp, 1 ' flash the state'
            frmTelaga.GetData recs.Fields("Telaga_id")
        End If
    Exit Sub
```

End If

If TampilJalan Then

 If optJalan(0).Value = True Then

 Set MyLayer = Map1.Layers("jalan_alteri")

 Elseif optJalan(1).Value = True Then

 Set MyLayer = Map1.Layers("jalan_lokal")

 Elseif optJalan(2).Value = True Then

 Set MyLayer = Map1.Layers("jalan_kolektor")

 Elseif optJalan(3).Value = True Then

 Set MyLayer = Map1.Layers("jalan_tol")

 Elseif optJalan(4).Value = True Then

 Set MyLayer = Map1.Layers("jalan_kabupaten")

 End If

Else

 Select Case LCase(PetaWilayah)

 Case "kelurahan"

 Set MyLayer = Map1.Layers("kelurahan")

 Case "kecamatan"

 Set MyLayer = Map1.Layers("kecamatan")

 Case "kabupaten"

 Set MyLayer = Map1.Layers("kabupaten")

 End Select

End If

Set MyPoint = Map1.ToMapPoint(X, Y)

If MyLayer.shapeType = moPolygon Then

 Set recs = MyLayer.SearchShape(MyPoint, moPointInPolygon, "")

Else

 Set recs = MyLayer.SearchByDistance(MyPoint, Map1.ToMapDistance(100), "")

End If

If Not recs.EOF Then

 Set shp = recs.Fields("shape").Value

 Map1.Refresh

 Map1.FlashShape shp, 1

If TampilJalan = True Then

 Select Case LCase(PetaJalan)

 Case "jalan_alteri"

 frmJalan.GetData PetaJalan, recs.Fields("Jalan_")

 Case "jalan_lokal"

 frmJalan.GetData PetaJalan, recs.Fields("IDRuas")

 Case "jalan_kolektor"

 frmJalan.GetData PetaJalan, recs.Fields("KOLEKTOR_")

 Case "jalan_tol"

 frmJalan.GetData PetaJalan, recs.Fields("TOL_ID")

 Case "jalan_kabupaten"

```
        frmJalan.GetData PetaJalan, recs.Fields("JLN_KAB_ID")
    End Select
Else
    Select Case LCase(PetaWilayah)
        Case "kelurahan"
            frmWilayah.GetData PetaWilayah, recs.Fields("ID_KELURAH")
        Case "kecamatan"
            frmWilayah.GetData PetaWilayah, recs.Fields("ID_KECAMAT")
    End Select
End If

End If
End Sub

Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
' StatusBar1.Panels(2).Text = "X : " & X & " ; Y : " & Y
End Sub

Private Sub mnuAbout_Click()
    frmAbout.Show 1, Me
End Sub

Private Sub mnuExit_Click()
    Unload Me
End

End Sub

Private Sub mnuHelp_Click()
    frmHelp.Show , Me
End Sub

Private Sub mnuNavArrow_Click()
    Toolbar1_ButtonClick Toolbar1.Buttons(1)
End Sub

Private Sub mnuNavDrag_Click()
    Toolbar1_ButtonClick Toolbar1.Buttons(5)
End Sub

Private Sub mnuNavIdentify_Click()
    Toolbar1_ButtonClick Toolbar1.Buttons(6)
End Sub

Private Sub mnuNavSearch_Click()
    Toolbar1_ButtonClick Toolbar1.Buttons(7)
End Sub

Private Sub mnuNavZoomFit_Click()
    Toolbar1_ButtonClick Toolbar1.Buttons(4)
```

End Sub

```
Private Sub mnuNavZoomIn_Click()  
    Toolbar1_ButtonClick Toolbar1.Buttons(2)  
End Sub
```

```
Private Sub mnuNavZoomOut_Click()  
    Toolbar1_ButtonClick Toolbar1.Buttons(3)  
End Sub
```

```
Private Sub mnuPetaAlteri_Click()  
    optJalan(0).Value = True  
End Sub
```

```
Private Sub mnuPetaJLKabupaten_Click()  
    optJalan(4).Value = True  
End Sub
```

```
Private Sub mnuPetaKabupaten_Click()  
    optPeta(2).Value = True  
End Sub
```

```
Private Sub mnuPetaKecamatan_Click()  
    optPeta(1).Value = True  
End Sub
```

```
Private Sub mnuPetaKelurahan_Click()  
    optPeta(0).Value = True  
End Sub
```

```
Private Sub mnuPetaKolektor_Click()  
    optJalan(2).Value = True  
End Sub
```

```
Private Sub mnuPetaLokal_Click()  
    optJalan(1).Value = True  
End Sub
```

```
Private Sub mnuPetaTanpaJalan_Click()  
    optJalan(5).Value = True  
End Sub
```

```
Private Sub mnuPetaTelaga_Click()  
    If mnuPetaTelaga.Checked = False Then  
        mnuPetaTelaga.Checked = True  
        Toolbar1.Buttons(9).Value = tbrPressed  
        Toolbar1_ButtonClick Toolbar1.Buttons(9)  
    Else  
        mnuPetaTelaga.Checked = False
```

```
Toolbar1.Buttons(9).Value = tbrUnpressed
Toolbar1_ButtonClick Toolbar1.Buttons(9)
End If
End Sub
```

```
Private Sub mnuPetaTol_Click()
    optJalan(3).Value = True
End Sub
```

```
Private Sub optJalan_Click(Index As Integer)
    picJalan.Visible = False
    If Index = 5 Then
        TampilJalan = False
    Else
        TampilJalan = True
    End If
    RefreshMap
    Map1.Extent = MapRect
End Sub
```

```
Private Sub optPeta_Click(Index As Integer)
    Select Case Index
        Case 0: PetaWilayah = "kelurahan"
        Case 1: PetaWilayah = "kecamatan"
        Case 2: PetaWilayah = "kabupaten"
    End Select
    picPeta.Visible = False
    RefreshMap
    Map1.Extent = MapRect
End Sub
```

```
Private Sub Timer1_Timer()
    StatusBar1.Panels(3).Text = Format(Now, " dddd, dd MMMM yyyy ")
    StatusBar1.Panels(4).Text = Format(Now, " hh:mm:ss ") & "WIB"
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSCComctlLib.Button)
    picPeta.Visible = False
    picJalan.Visible = False
    Select Case Button.Index
        Case 1 'arrow
            Map1.MousePointer = moArrow
        Case 2 'zoom in
            Map1.MousePointer = moZoomIn
        Case 3 'zoom out
            Map1.MousePointer = moZoomOut
        Case 4 'zoom fit
            Map1.Extent = Map1.FullExtent
            Map1.MousePointer = moArrow
    End Select
End Sub
```

```
Set MapRect = Map1.FullExtent
Case 5 'drag
  Map1.MousePointer = moPan
Case 6 'identify
  Map1.MousePointer = moHotLink
Case 7
  'search
  Map1.MousePointer = moArrow
  frmCari.Show , Me
Case 8 'separator
Case 9 'telaga
  If Button.Value = tbrPressed Then
    mnuPetaTelaga.Checked = True
    TampilTelaga = True
  Else
    mnuPetaTelaga.Checked = False
    TampilTelaga = False
  End If
  RefreshMap
  Map1.Extent = MapRect
Case 10 'peta
  picPeta.Visible = True
Case 11 'jalan
  picJalan.Visible = True
End Select
End Sub

Public Sub RefreshMap()
Dim ML As MapLayer

Map1.Layers.Clear

Select Case LCase(PetaWilayah)
  Case "kelurahan"
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("kelurahan")
    ML.Symbol.Color = &HC0FFC0
    Map1.Layers.Add ML
  Case "kecamatan"
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("kecamatan")
    ML.Symbol.Color = &HC0FFC0
    Map1.Layers.Add ML
  Case "kabupaten"
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("kabupaten")
    ML.Symbol.Color = &HC0FFC0
    Map1.Layers.Add ML
End Select
```

```
If TampilJalan Then
  If optJalan(0).Value = True Then
    TampilJalan = True
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("jalan_alteri")
    ML.Symbol.Color = &H8080FF
    Map1.Layers.Add ML
    PetaJalan = "jalan_alteri"
  ElseIf optJalan(1).Value = True Then
    TampilJalan = True
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("jalan_lokal")
    ML.Symbol.Color = &H8080FF
    Map1.Layers.Add ML
    PetaJalan = "jalan_lokal"
  ElseIf optJalan(2).Value = True Then
    TampilJalan = True
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("jalan_kolektor")
    ML.Symbol.Color = &H8080FF
    Map1.Layers.Add ML
    PetaJalan = "jalan_kolektor"
  ElseIf optJalan(3).Value = True Then
    TampilJalan = True
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("jalan_tol")
    ML.Symbol.Color = &H8080FF
    Map1.Layers.Add ML
    PetaJalan = "jalan_tol"
  ElseIf optJalan(4).Value = True Then
    TampilJalan = True
    Set ML = New MapLayer
    ML.GeoDataset = DC.FindGeoDataset("jalan_kabupaten")
    ML.Symbol.Color = &H8080FF
    Map1.Layers.Add ML
    PetaJalan = "jalan_kabupaten"
  End If
End If

If TampilTelaga Then
  TampilTelaga = True
  Set ML = New MapLayer
  ML.GeoDataset = DC.FindGeoDataset("telaga")
  ML.Symbol.Color = vbBlue
  Map1.Layers.Add ML
End If

Map1.Refresh
```

Set ML = Nothing

End Sub

FORM SPLASH

Private Sub Form_Click()

End

End Sub

Private Sub Form_Unload(Cancel As Integer)

SetupData

frmMain.Show

End Sub

Private Sub Label2_Click()

End Sub

Private Sub Timer1_Timer()

Unload Me

End Sub

FORM TELAGA

Private strSQL As String

Private rs As DAO.Recordset

Private rsLV1 As DAO.Recordset

Private Sub cmdCari_Click()

Unload Me

frmCari.Show , frmMain

End Sub

Private Sub cmdClose_Click()

Unload Me

End Sub

Sub GetData(ID As Integer)

On Error Resume Next

Me.Show , frmMain

strSQL = "SELECT * FROM Telaga WHERE Telaga_ID = " & Val(ID)

Set rs = DBSIG.OpenRecordset(strSQL, dbOpenDynaset)


```
If rs.EOF And rs.BOF Then
```

```
txtIDTelaga = ID
```

```
Me.Show , frmMain
```

```
Exit Sub
```

```
End If
```

```
txtIDTelaga = rs("TELAGA_ID")
```

```
txtNamaTelaga = rs("NAMA_TELAGA")
```

```
txtDayaTampung = rs("DAYA_TAMPUNG")
```

```
txtFungsi = rs("FUNGSI_TELAGA")
```

```
txtIDKelurahan = rs("IDKEL")
```

```
txtJmlPengguna = rs("JML_PENGGUNA")
```

```
txtKedalaman = rs("KEDALAMAN")
```

```
txtLuas = rs("LUAS_TOTAL")
```

```
txtLuasGenangan = rs("LUAS_GENANGAN")
```

```
txtTahun = rs("TH_DIBANGUN")
```

```
If rs("TANGGA_CUCI") = True Then
```

```
chkTangga.Value = vbChecked
```

```
Else
```

```
chkTangga.Value = vbUnchecked
```

```
End If
```

```
If rs("PLENGSENGAN") = True Then
```

```
chkPl.Value = vbChecked
```

```
Else
```

```
chkPl.Value = vbUnchecked
```

```
End If
```

```
Set Image1.Picture = Nothing
```

```
Set Image2.Picture = Nothing
```

```
txtFoto1 = ""
```

```
txtFoto2 = ""
```

```
txtFoto1.Text = rs("Foto")
```

```
txtFoto2.Text = rs("Sketsa")
```

```
Image1.Picture = LoadPicture(txtFoto1)
```

```
Image2.Picture = LoadPicture(txtFoto2)
```

```
strsql = "SELECT * FROM Kelurahan, Kecamatan WHERE (Kelurahan.IDKel = " &  
Val(txtIDKelurahan) & ") AND (Kelurahan.IDKec = Kecamatan.IDKec)"
```

```
Set rs = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
```

```
txtNamaKelurahan = rs("kelurahan.Nama")
```

```
txtNamaKecamatan = rs("kecamatan.nama")
```

End Sub

Private Sub cmdFoto_Click(Index As Integer)

On Error Resume Next

Select Case Index

Case 0:

cd1.CancelError = True

cd1.Filter = "All Supported Image Files

(*.*bmp;*.jpg;*.jpeg;*.gif;*.wmf)|*.bmp;*.jpg;*.jpeg;*.gif;*.wmf|Windows Bitmap
(*.*bmp)|*.bmp|JPG File (*.jpg)|*.jpg|Gif File (*.gif)|*.gif| WMF file (*.wmf)|*.wmf"

cd1.FilterIndex = 0

cd1.ShowOpen

txtFoto1 = cd1.FileName

Image1.Picture = LoadPicture(txtFoto1)

Case 1:

Set Image1.Picture = Nothing

txtFoto1 = " "

Case 2:

cd1.CancelError = True

cd1.Filter = "All Supported Image Files

(*.*bmp;*.jpg;*.jpeg;*.gif;*.wmf)|*.bmp;*.jpg;*.jpeg;*.gif;*.wmf|Windows Bitmap
(*.*bmp)|*.bmp|JPG File (*.jpg)|*.jpg|Gif File (*.gif)|*.gif| WMF file (*.wmf)|*.wmf"

cd1.FilterIndex = 0

cd1.ShowOpen

txtFoto2 = cd1.FileName

Image2.Picture = LoadPicture(txtFoto2)

Case 3:

Set Image2.Picture = Nothing

txtFoto2 = " "

End Select

End Sub

Private Sub cmdPosisi_Click()

Dim recs As mapobjects2.Recordset

Dim shp As Object

Dim rect As mapobjects2.Rectangle

Dim exp As String

' frmMain.TampilJalan = True

' frmMain.RefreshMap

Me.Hide

exp = "Telaga_ID = " & Val(txtIDTelaga) & ""

Set recs = frmMain.Map1.Layers("telaga").SearchExpression(exp)

If Not recs.EOF Then

Set shp = recs.Fields("Shape").Value

```
frmMain.Refresh
frmMain.Map1.FlashShape shp, 3
End If
Me.Show
End Sub
```

```
Private Sub cmdSimpan_Click()
```

```
Dim vTg, vPl As Boolean
```

```
If MsgBox("Simpan data Telaga?", vbYesNo + vbQuestion) = vbNo Then Exit Sub
strsql = "SELECT Telaga_id FROM Telaga WHERE Telaga_ID = " & txtIDTelaga
Set rs = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
```

```
If chkPl.Value = vbChecked Then
```

```
    vPl = True
```

```
Else
```

```
    vPl = False
```

```
End If
```

```
If chkTangga.Value = vbChecked Then
```

```
    vTg = True
```

```
Else
```

```
    vTg = False
```

```
End If
```

```
If rs.BOF And rs.EOF Then
```

```
    'save
```

```
strsql = "INSERT INTO Telaga " & _
        "(Telaga_id, nama_telaga, idkel, " & _
        "luas_total, kedalaman, jml_pengguna, " & _
        "daya_tampung, fungsi_telaga, th_dibangun, " & _
        "luas_genangan, tangga_cuci, plengsengan, foto, sketsa) " & _
        "Values(" & Val(txtIDTelaga) & ", " & _
        """" & txtNamaTelaga & """, " & _
        Val(txtIDKelurahan) & ", " & _
        """" & txtLuas & """, " & _
        """" & txtKedalaman & """, " & _
        """" & txtJmlPengguna & """, " & _
        """" & txtDayaTampung & """, " & _
        """" & txtFungsi & """, " & _
        """" & txtTahun & """, " & _
        """" & txtLuasGenangan & """, " & _
        vTg & ", " & _
        vPl & ", " & _
        """" & txtFoto1 & """, " & _
        """" & txtFoto2 & """)"
```

```
Else
```

```
    'update
```

```
strsql = "UPDATE Telaga " & _
        "SET nama_telaga = "" & txtNamaTelaga & """, " & _
```

```
"idkel = " & Val(txtIDKelurahan) & ", " & _  
"luas_total = " & txtLuas & ", " & _  
"kedalaman = " & txtKedalaman & ", " & _  
"jml_pengguna = " & txtJmlPengguna & ", " & _  
"daya_tampung = " & txtDayaTampung & ", " & _  
"fungsi_telaga = " & txtFungsi & ", " & _  
"th_dibangun = " & txtTahun & ", " & _  
"luas_genangan = " & txtLuasGenangan & ", " & _  
"tangga_cuci = " & vTg & ", " & _  
"plengsengan = " & vPl & ", " & _  
"foto = " & txtFoto1 & ", " & _  
"sketsa = " & txtFoto2 & " " & _  
"WHERE Telaga_ID = " & Val(txtIDTelaga)
```

End If

DBSIG.Execute strsql

MsgBox "Data Telaga telah tersimpan.", vbInformation

Unload Me

End Sub

Private Sub Form_Load()

With LV

.ColumnHeaders.Add , , "ID Kel", 800, lvwColumnLeft

.ColumnHeaders.Add , , "Nama Kel", 1800, lvwColumnLeft

.ColumnHeaders.Add , , "Kecamatan", 1500, lvwColumnLeft

strsql = "SELECT * FROM Kelurahan, Kecamatan WHERE Kelurahan.idkec =
kecamatan.idkec ORDER BY kelurahan.idkel ASC"

Set rsLV1 = DBSIG.OpenRecordset(strsql, dbOpenDynaset)

.ListItems.Clear

While Not rsLV1.EOF

Set LI = .ListItems.Add()

LI.Text = rsLV1("IDKel")

LI.SubItems(1) = rsLV1("Kelurahan.Nama")

LI.SubItems(2) = rsLV1("kecamatan.Nama")

LI.Tag = rsLV1("IDKel")

rsLV1.MoveNext

Wend

End With

frmMain.Enabled = False

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

Set rs = Nothing

End Sub

Private Sub Form_Unload(Cancel As Integer)

frmMain.Enabled = True

End Sub

```
Private Sub Image1_DblClick()  
    On Error Resume Next  
    Form1.Image1.Picture = LoadPicture(txtFoto1)  
    Form1.Show 1, Me  
End Sub  
  
Private Sub Image2_DblClick()  
    On Error Resume Next  
    Form1.Image1.Picture = LoadPicture(txtFoto2)  
    Form1.Show 1, Me  
End Sub  
  
Private Sub LV_Click()  
    txtIDKelurahan = Val(LV.SelectedItem)  
    txtNamaKelurahan = LV.SelectedItem.ListSubItems(1)  
    txtNamaKecamatan = LV.SelectedItem.ListSubItems(2)  
    LV.Visible = False  
End Sub  
  
Private Sub LV_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then LV_Click  
    If KeyAscii = 27 Then LV.Visible = False  
    txtIDKelurahan.SetFocus  
End Sub  
  
Private Sub LV_LostFocus()  
    LV.Visible = False  
End Sub  
  
Private Sub txtIDKelurahan_Click()  
    LV.Visible = True  
    LV.SetFocus  
End Sub  
  
Private Sub txtIDKelurahan_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then txtIDKelurahan_Click  
End Sub
```

FORM WILAYAH

```
Dim strsql As String
Private rsKel As DAO.Recordset
Private rsTelaga As DAO.Recordset
Private rsKec As DAO.Recordset

Public Sub GetData(PetaW As String, ID As Integer)
On Error Resume Next

Select Case UCase(PetaW)
Case "KELURAHAN"
LabTitle.Caption = "Informasi Kelurahan"
frKelurahan.Visible = True
frKec.Visible = False

strsql = "SELECT * FROM Kelurahan, Kecamatan WHERE (Kelurahan.IDKel = " & ID & ")
AND (Kelurahan.IDKec = Kecamatan.IDKec)"
Set rsKel = DBSIG.OpenRecordset(strsql, dbOpenDynaset)

If rsKel.EOF And rsKel.BOF Then
txtIDKel = ID
Me.Show , frmMain
Exit Sub
End If

txtIDKel = rsKel("IDKel")
txtNamaKel = rsKel("kelurahan.Nama")
txtKec1 = rsKel("kelurahan.IDKec")
txtKec2 = rsKel("kecamatan.nama")

strsql = "Select COUNT(Telaga_ID) AS JmlTlg FROM Telaga WHERE IDKel = " & ID
Set rsTelaga = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
txtJmlTelaga = rsTelaga("JmlTlg")

Me.Show , frmMain
Exit Sub

Case "KECAMATAN"
LabTitle.Caption = "Informasi Kecamatan"
frKelurahan.Visible = False
frKec.Visible = True

strsql = "SELECT * FROM Kecamatan"
Set rsKec = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
rsKec.FindFirst "IDKec = " & ID

If rsKec.NoMatch Then
```

```
txtIDKec = ID
Me.Show , frmMain
Exit Sub
End If
```

```
txtIDKec = rsKec("IDKec")
txtNamaKec = rsKec("Nama")
txtLuas = rsKec("Luas")
txtJml = rsKec("JML_Pelaya")
```

```
strsql = "Select COUNT(IDKel) AS JmlKel FROM Kelurahan WHERE IDKec = " & ID
Set rsTelaga = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
txtJmlKel = rsTelaga("JmlKel")
```

```
Me.Show , frmMain
Exit Sub
End Select
End Sub
```

```
Private Sub cmdCari_Click()
Unload Me
frmCari.Show , frmMain
End Sub
```

```
Private Sub cmdPosisiKec_Click()
Dim recs As mapobjects2.Recordset
Dim shp As Object
Dim rect As mapobjects2.Rectangle
Dim exp As String
```

```
' frmMain.RefreshMap
Me.Hide
exp = "ID_Kecamat = " & Val(txtIDKec) & ""
Set recs = frmMain.Map1.Layers("kecamatan").SearchExpression(exp)
If Not recs.EOF Then
Set shp = recs.Fields("Shape").Value
frmMain.Refresh
frmMain.Map1.FlashShape shp, 3
End If
Me.Show
End Sub
```

```
Private Sub cmdSaveKel_Click()
'simpan kelurahan
If MsgBox("Simpan data Kelurahan?", vbYesNo + vbQuestion) = vbNo Then Exit Sub
strsql = "SELECT * FROM kelurahan WHERE IDkel = " & Val(txtIDKel)
Set rsKel = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
If rsKel.BOF And rsKel.EOF Then
strsql = "INSERT INTO kelurahan " & _
```

```
        "(IDkel, Nama) " & _
        "VALUES (" & Val(txtIDKel) & ", " & txtNamaKel & " ")"
    Else
        strsql = "UPDATE kelurahan " & _
        "SET Nama = " & txtNamaKel & " " & _
        "WHERE IDkel = " & Val(txtIDKel)
    End If
    DBSIG.Execute strsql
    MsgBox "Data Kelurahan telah tersimpan.", vbInformation
    Unload Me
End Sub

Private Sub cmdSimpanKec_Click()
    'simpan kecamatan
    If MsgBox("Simpan data Kecamatan?", vbYesNo + vbQuestion) = vbNo Then Exit Sub
    strsql = "SELECT * FROM Kecamatan WHERE IDKec = " & Val(txtIDKec)
    Set rsKec = DBSIG.OpenRecordset(strsql, dbOpenDynaset)
    If rsKec.BOF And rsKec.EOF Then
        strsql = "INSERT INTO Kecamatan " & _
        "(IDkec, Nama) " & _
        "VALUES (" & Val(txtIDKec) & ", " & txtNamaKec & " )"
    Else
        strsql = "UPDATE kecamatan " & _
        "SET Nama = " & txtNamaKec & " , " & _
        "JML_PELAYA = " & txtJml & " , " & _
        "Luas = " & txtLuas & " " & _
        "WHERE IDkec = " & Val(txtIDKec)
    End If
    DBSIG.Execute strsql
    MsgBox "Data Kecamatan telah tersimpan.", vbInformation
    Unload Me
End Sub

Private Sub cmdTutup_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    frmMain.Enabled = False
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Select Case LabTitle.Caption
        Case "Informasi Kelurahan"
            Set rsKel = Nothing
            Set rsTelaga = Nothing
        Case "Informasi Kecamatan"
            Set rsKec = Nothing
    End Select
```


End Sub

Private Sub Form_Unload(Cancel As Integer)

frmMain.Enabled = True

End Sub

Private Sub frKelurahan_DragDrop(Source As Control, X As Single, Y As Single)

End Sub

Private Sub txtPosisiKel_Click()

Dim recs As mapobjects2.Recordset

Dim shp As Object

Dim rect As mapobjects2.Rectangle

Dim exp As String

' frmMain.RefreshMap

Me.Hide

exp = "ID_Kelurah = " & Val(txtIDKel) & ""

Set recs = frmMain.Map1.Layers("kelurahan").SearchExpression(exp)

If Not recs.EOF Then

Set shp = recs.Fields("Shape").Value

frmMain.Refresh

frmMain.Map1.FlashShape shp, 3

End If

Me.Show

End Sub

