

SKRIPSI

**PEMBUATAN SISTEM INFORMASI
USAHA KECIL DAN MENENGAH (UKM)
DENGAN MENGGUNAKAN
VISUAL BASIC 6.0 DAN MAP OBJECT 2.1
(Studi Kasus : Kabupaten Pasuruan)**



Disusun Oleh :

Dian Widhianto

99.25.091

**JURUSAN TEKNIK GEODESI S1
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI NASIONAL**

MALANG

2006

1941223

REPRODUCED FROM THE
ORIGINAL SOURCE AND FROM THE
MANUFACTURING SOURCE
THE TOTAL COST HAS NOT BEEN
REPRODUCED FROM THE

1941223
REPRODUCED FROM THE
ORIGINAL SOURCE

REPRODUCED FROM THE
ORIGINAL SOURCE AND FROM THE
MANUFACTURING SOURCE
THE TOTAL COST HAS NOT BEEN
REPRODUCED FROM THE

1941223

LEMBAR PERSETUJUAN

**PEMBUATAN SISTEM INFORMASI
USAHA KECIL DAN MENENGAH (UKM)
DENGAN MENGGUNAKAN
VISUAL BASIC 6.0 DAN MAP OBJECT 2.1
(Studi Kasus : Kabupaten Pasuruan)**

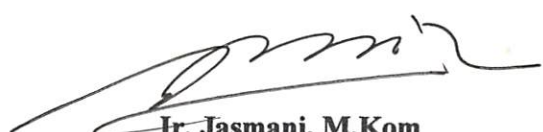
**TUGAS AKHIR
(SKRIPSI)**

**Diajukan untuk Memenuhi Persyaratan dalam Mencapai Gelar Sarjana
Teknik Strata Satu (S-1)**


**Oleh :
DIAN WIDHIANTO
99.25.091**

Menyetujui :

Dosen Pembimbing I


Ir. Jasmani, M.Kom

Dosen Pembimbing II


Ir. Leo Pantimena, M.Sc



**Mengetahui
Ketua Jurusan Teknik Geodesi S-1**

17/10/06


Hery Purwanto, ST, MSc

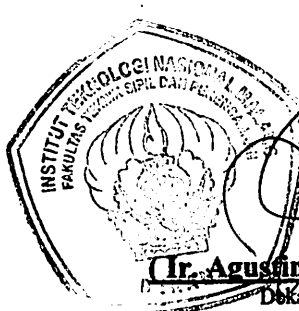
LEMBAR PENGESAHAN


Dipertahankan di depan Panitia Penguji Tugas Akhir Jurusan Teknik Geodesi,
Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang, dan
diterima untuk memenuhi sebagai persyaratan untuk memperoleh gelar sarjana

S-1 Teknik Geodesi :


Pada hari / Tanggal : Jum'at / 29 September 2006

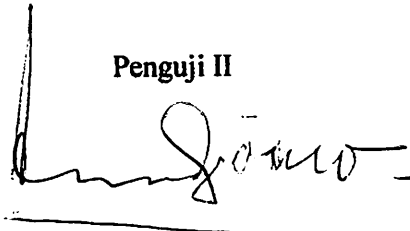
Panitia Ujian Tugas Akhir :

 **Ketua**
(Ir. Agustina Nurul Hidayati, MTP)
Dekan FTSP ITN Malang


Sekretaris

(Hery Purwanto, ST., MSc)
Ketua Jurusan Teknik Geodesi S-1

Anggota Penguji :

Penguji I

Ir. Leo Pantimena, M.Sc

Penguji II

R. Moeljono, SH.MH.UM

Penguji III


Silvester Sari Sai, ST. MT

KATA PENGANTAR

Puji syukur Alhamdulillah kepada ALLAH SWT dengan rahmat dan hidayah-NYA penulis dapat menyelesaikan Tugas Akhir ini sebagai syarat kelulusan studi yang ditempuh di Institut Teknologi Nasional pada Jurusan Teknik Geodesi S-1.

Dalam Tugas Akhir ini penulis melakukan penelitian yang berjudul :

“ Pembuatan Sistem Informasi Usaha Kecil dan Menengah (UKM) di Kabupaten Pasuruan Dengan Menggunakan Visual Basic 6.0 dan Map Object 2.1”.

Tugas Akhir ini disusun atas kerja sama yang baik dari berbagai pihak yang telah membantu penulis yang berupa saran dan respon positif sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan baik. Untuk itu penulis ingin menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Ibu Ir. Agustina Nurul Hidayati, MTP selaku Dekan Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.
3. Bapak Hery Purwanto, ST., Msc selaku Ketua Jurusan Teknik Geodesi Institut Teknologi Nasional Malang.
4. Bapak Christian Tongam Siahaan, ST selaku Sekretaris Jurusan Teknik Geodesi Institut Teknologi Nasional Malang.
5. Bapak Ir. Jasmani, M. Kom dosen pembimbing pertama pada penulisan Tugas Akhir ini.
6. Bapak Ir. Leo Pantimena, MSc, selaku dosen pembimbing kedua pada penulisan Tugas Akhir ini.

7. Bapak, Ibuk dan Adik-adikku sekeluarga yang telah memberikan dukungan baik moril maupun doa kepada penulis sehingga dapat terselesainya Tugas Akhir ini.
8. Mbah-mbahku, Om-omku, Bulek-bulekku, Pak De yang selalu memberikan semangat dan perhatiannya kepada penulis dalam menyelesaikan Tugas Akhir ini.
9. Teman-temanku Heri, Nanang, Aripin, Ero, Ronny, Marjono, Gobas, Dwi Kiki dan lain-lain yang tidak dapat penulis sebutkan satu-persatu trimakasih do'anya
10. Semua teman-teman Jurusan Teknik Geodesi S-1 ITN Malang yang banyak membantu dalam penulisan dan pengolahan data pada Tugas Akhir ini.

Dalam Laporan Tugas Akhir ini penulis menyadari sepenuhnya bahwa apa yang telah penulis kerjakan ini masih sangat jauh dari kesempurnaan, oleh karena itu penulis mengharapkan saran dari para pembaca.

Akhir kata penulis mengucapkan banyak terima kasih dan penghargaan yang sebesar-besarnya atas semua dukungan yang telah diberikan serta respon positif dari para pembaca. Semoga Laporan Tugas Akhir ini dapat lebih bermanfaat di masa yang akan datang khususnya mahasiswa Teknik Geodesi.

Malang, Oktober 2006

Penulis

DAFTAR ISI

Lembar Persetujuan	i
Lembar Pengesahan	ii
Kata Pengantar	iii
Daftar Isi	v
Daftar Gambar dan Tabel.....	ix

BAB I PENDAHULUAN

I.1. Latar Belakang.....	1
I.2. Maksud dan Tujuan.....	2
I.3. Pendekatan Masalah	2
I.4. Batasan Masalah	2
I.5. Rumusan Masalah.....	2
I.6. Manfaat Penelitian.....	3

BAB II LANDASAN TEORI

2.1. Pengertian Usaha Kecil dan Menengah.....	4
2.2. Definisi Sistem informasi Geografi (SIG).....	5
2.2.1. Komponen SIG.....	5
2.2.1.1. Data Input/Data Masukan.....	7
2.2.1.1.1. Data Spasial.....	8
2.2.1.1.2. Data NonSpasial.....	9
2.2.1.2. Penyimpanan dan Pemanggilan Data.....	9
2.2.1.3. Data Manipulasi dan Analisa.....	10

2.2.1.4. Menampilkan Produk SIG.....	10
2.3. Basis Data.....	10
2.3.1. Sistem Manajemen Basis Data.....	11
2.3.2. Keuntungan dan Kekurangan Basisdata.....	11
2.3.3 Komponen Sistem Basis Data	13
2.3.4. Struktur Basis	15
2.3.5. Model Data Sistem Basis	20
2.4. Visual Basic 6.0.....	22
2.4.1. Elemen Utama Visual Basic	22
2.5. Map Object 2.1.....	24
2.6. Desain Program Visual Basic 6.0 dan Map Object.....	24

BAB III PELAKSANAAN PENELITIAN

3.1. Deskripsi Wilayah Penelitian.....	25
3.1.1. Keadaan Umum Kabupaten Pasuruan.....	25
3.2. Materi Penelitian.....	26
3.3. . Alat Penelitian	27
3.3.1. Tahapan Penelitian.....	30
3.4. Tahapan Pembuatan Program.....	35
3.5. Pemasukan Data Spasial.....	38
3.6. Penyimpanan.....	46
3.7.Pembentukan Topologi.....	47
3.8. Editing Topologi.....	50
3.9. Pembuatan Data Atribut.....	54

3.9.1. Proses Operasi Data Atribut.....	55
3.9.2. Penggabungan Data Atribut	56
3.9.3. Convert File.....	58
3.10. Desain Tampilan Informasi Usaha Kecil dan Menengah.....	59
3.10.1. Start/Enter.....	59
3.10.2. Display Main Form.....	62
3.10.3. Membuat Desain Tampilan Peta(Map Display).....	67
3.10.4. Mendesain Program Untuk Pencarian(Search).....	68
3.10.5. Desain Program Untuk Penggambaran dan Update Data...	69
3.10.6. Pembuatan Program Bantuan(Help).....	71

BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian.....	75
4.1.1. Visualisasi Informasi Dengan Visual Basic.....	75
4.2. Pembahasan Hasil.....	76
4.2.1. Penyajian Peta.....	76
4.2.2. Penyajian Aplikasi Data.....	74
4.2.3. Pelaksanaan Updating data.....	75
4.2.3.1. Updating data Non Spasial.....	75
4.2.3.2. Updating data Spasial.....	76
4.2.4. Penyajian Aplikasi Pencarian.....	77
4.2.4.1. Pencarian dengan mengklik objek peta.....	77
4.2.4.2. Pencarian dengan mengklik toolbar Cari.....	79
4.2.5. Menambah atau menghapus layer.....	80

4.2.6. Mencari Query.....	80
4.3. Kendala Yang Timbul dan Cara Mengatasi Kendala Dalam Pembuatan Program Dengan Menggunakan Visual Basic.....	81
4.3.1. Kendala yang timbul pada saat pembuatan program.....	81
4.3.2. Mengatasi Kendala-kendala Dalam Pembuatan Program Dengan Menggunakan Visual Basic.....	81
4.4. Keuntungan Pembuatan Program Dengan Visual Basic dan Map Object.....	82
4.5. Analisa Hasil.....	84

BAB V PENUTUP

5.1 Kesimpulan.....	87
5.2. Saran.....	88

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar2.1	Komponen SIG	6
Gambar 2.2.	Model Data Raster	9
Gambar 2.3.	Model Data Vektor	9
Gambar 2.4	Struktur Basisdata Hirarki	16
Gambar 2.5	Struktur Basisdata Network	17
Gambar 2.6.	Struktur Basisdata Relational	19
Gambar 2.7	Tampilan Layar Visual BasiC.....	22
Gambar 3.1.	Peta Lokasi Pasuruan diwilayah Jawa Timur	25
Gambar 3.2.	Tampilan AutoCad 2000.....	28
Gambar 3.3.	Tampilan ARCinfo 3.5	29
Gambar 3.4.	Tampilan Awal Pada ArcView versi 3..1.....	29
Gambar 3.5	Menu AutoCad 2000.....	38
Gambar 3.6	Tampilan AutoCad.....	38
Gambar 3.7	Tampilan menu Layer	39
Gambar 3.8	Extend sesudah dan sebelum	41
Gambar 3.9	Trim sesudah dan sebelum	42
Gambar 3.10	Pedit (join) sesudah dan sebelum.....	43
Gambar 3.11	Move sesudah dan sebelum	44
Gambar 3.12	Fillet sesudah dan sebelum	44
Gambar 3.13	Chamfer sesudah dan sebelum	45
Gambar 3.14	Esport Data	46
Gambar 3.15.	Tampilan Menu Save	47

Gambar 3.16 Tampilan Coverage Yang Akan Diedit	51
Gambar 3.17 Tampilan Kesalahan Pada Coverage.....	52
Gambar 3.18 Penyusunan Data Atribut	55
Gambar 3.19 Proses Export data atribut	56
Gambar 3.20 Penggabungan Data	57
Gambar 3.21 Convert File	58
Gambar 3.22 : Tampilan Awal Program Visual Basic	59
Gambar 3.23 : Tampilan di Lingkungan	60
Gambar 3.24 : Kotak Dialog Componets	60
Gambar 3.25 Penambahan Kontrol Pada ToolBox	62
Gambar 3.26 Desain Form Utama	62
Gambar 3.27 : Kotak Dialog Menu Editor.....	63
Gambar 3.28 Property Pages	64
Gambar 3.29 : Kotak Dialog Property Page pada Kontrol ToolBar	65
Gambar 3.30 : Kotak Dialog Property Page pada Kontrol CoolBar	66
Gambar 3.31 : Desain Form frmShp Untuk Menampilkan Peta.....	67
Gambar 3.32 : Desain Form Pencarian	68
Gambar 3.33 : Desain Form updating Data site pline perumahan	69
Gambar 3.34 : Desain Form updating Data fasilitas umum.....	70
Gambar 4.1 Desain Form Utama Informasi Usaha Kecil dan Menengah.....	72
Gambar 4.2 Form Utama Usaha Kecil dan Menengah	73
Gambar 4.3a Desain Form informasi Usaha Kecil dan Menengah	74
Gambar 4.3b Desain Form informasi foto Usaha Kecil dan Menengah.....	74
Gambar 4.4 Form Editi Data UKM	76

Gambar 4.5 Form C ari Usaha Kecil dan Menengah.....	79
Gambar 4.6 Menu File.....	80
Gambar 4.7 Menu Query.....	81

DAFTAR TABEL

Tabel 2.1 Jendela Visual Basic 6.0	23
--	----

DAFTAR DIAGRAM

Diagram 3.1 Diagram Alir Penelitian	34
Diagram 3.2 Diagram alir Pembuatan Program	37

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kabupaten Pasuruan mengalami perkembangan yang cukup pesat khususnya di bidang ekonomi. Hal ini dapat dilihat dari semakin banyaknya usaha-usaha kecil dan menengah di daerah itu. Selain itu dukungan dari pemerintah juga menjadi salah satu penyebab utama berkembangnya usaha-usaha kecil dan menengah, dengan program kredit bantuan modal untuk Usaha Kecil dan Menengah juga ikut membantu dalam mengatasi masalah permodalan, sehingga semakin lama semakin banyak Usaha Kecil dan Menengah (UKM) baru.

Dengan adanya perkembangan yang cukup pesat di bidang ekonomi khususnya Usaha Kecil dan Menengah, maka dibutuhkan suatu informasi mengenai perencanaan dan strategi pengembangan UKM serta informasi terkait lokasi serta letak kawasan potensi UKM di wilayah Kabupaten Pasuruan, agar masyarakat baik di Kabupaten maupun diluar Kabupaten Pasuruan dapat mengakses informasi maupun data dengan cepat dan mudah. Dengan perkembangan teknologi di bidang komputer penyajian informasi UKM dapat dikerjakan dengan menggunakan bahasa pemrograman. Oleh karena itu informasi UKM tersebut perlu disajikan dalam suatu program atau perangkat lunak yang memudahkan pencarian informasi dan memberikan keleluasaan masyarakat untuk menggali informasi mengenai perkembangan UKM yang ada di Kabupaten Pasuruan. Dan informasi tersebut tidak hanya berguna bagi masyarakat pasuruan saja tetapi juga berguna bagi daerah lain dalam mencari informasi.

1.2. Maksud dan Tujuan

Dari latar belakang diatas, maksud dan tujuan penelitian ini adalah membuat sistem informasi Usaha Kecil dan Menengah (UKM) di Kabupaten Pasuruan dengan menggunakan Visual Basic 6.0 dan Map Object 2.1 yang memiliki kemampuan untuk mengolah data, baik menambah, mengedit dan menghapus data spasial dan atribut.

1.3. Pendekatan Masalah

Pendekatan masalah pada penelitian ini adalah merencanakan suatu sistem informasi Usaha Kecil dan Menengah (UKM) di Kabupaten Pasuruan dengan menggunakan bahasa pemrograman Visual Basic 6.0 dan Map Object 2.1.

1.4. Batasan Penelitian

Pada penelitian ini masalah yang akan dibahas terbatas pada penggunaan software Visual Basic 6.0 dan Map Object 2.1 untuk dapat menyajikan data informasi Usaha Kecil dan Menengah (UKM) di Kabupaten Pasuruan.

1.5. Rumusan Masalah

Dalam pembuatan Laporan ini terdapat masalah yang dapat diangkat sebagai berikut :

- Bagaimana cara membuat program “ Penyajian Informasi UKM di Kabupaten Pasuruan “dengan memanfaatkan Visual Basic 6.0 ?
- Apakah Visual Basic 6.0 dan Map Object 2.1 dapat digunakan dalam penyelesaian masalah visualisasi Informasi UKM di Kabupaten Pasuruan?
- Apakah Map Object 2.1 dapat menampilkan data input berupa peta dalam processing pembuatan program ?

1.6. Manfaat Penelitian

Manfaat yang di dapat dalam penelitian ini adalah:

1. Masyarakat dengan mudah mendapatkan informasi dan data Usaha Kecil dan Menengah (UKM) di Kabupaten Pasuruan.
2. Dapat melakukan *updating* data apabila ada pembaruan data tanpa mengurangi informasi yang ada sebelumnya.

BAB II

LANDASAN TEORI

Sistem perekonomian di Indonesia perlu terus dikembangkan dan ditingkatkan. Hal ini dapat dilakukan salah satunya dengan cara memberdayakan usaha-usaha mikro, kecil dan menengah. Hal ini dapat dilihat dari potensi yang dimiliki negara kita dalam mengembangkan sistem perekonomian kerakyatan.

2.1. Pengertian Usaha Kecil dan Menengah

Usaha Kecil dan Menengah (UKM) adalah suatu bentuk usaha yang dimana modal usahanya kurang dari 10 milyar rupiah, dengan jumlah tenaga kerja antara satu orang sampai dengan seratus orang dan dimiliki oleh seorang warga negara Indonesia. UKM menurut jumlah tenaga kerjanya dapat dibedakan menjadi tiga yaitu usaha mikro dengan jumlah tenaga kerja kurang dari lima orang, selanjutnya usaha kecil dengan jumlah tenaga kerja antara enam sampai dua puluh orang dan terakhir usaha menengah dengan jumlah tenaga kerja dua puluh sampai dengan seratus orang.

Usaha Kecil dan Menengah adalah salah satu usaha yang menganut sistem kerakyatan, dimana dalam melakukan produksi dilakukan oleh lebih dari dua orang. Selain itu UKM adalah usaha dalam bentuk informal, sehingga dalam melakukan kegiatan yang dibutuhkan hanya keterampilan saja bukan tingkat pendidikan seseorang. Bidang-bidang usahanya biasanya meliputi kerajinan tangan, pembudidayaan tanaman dan jasa.

Dalam perkembangan dunia usaha saat ini UKM adalah usaha yang sangat menjanjikan karena banyaknya barang kebutuhan yang tidak dapat dipenuhi atau diproduksi oleh perusahaan besar. Selain itu UKM juga menyerap tenaga kerja yang cukup banyak sehingga dapat mengurangi pengangguran. (Supena, 2001: 12)

2.2. Definisi Sistem informasi Geografi (SIG)

SIG merupakan suatu sistem berbasis komputer yang mempunyai kemampuan untuk mengumpulkan, menyimpan, memanipulasi dan menyajikan informasi mengenai tempat-tempat yang terletak dipermukaan bumi, mengenai posisi dimana suatu obyek terletak dipermukaan bumi dan informasi mengenai keterangan-keterangan (*atribut*) yang terdapat dipermukaan bumi yang posisinya diberikan dan diberitahu. (Leo Pantimena, 1997:27)

2.2.1. Komponen SIG

SIG merupakan suatu sistem yang terdiri dari komponen – komponen yang terkumpul dan membentuk suatu kesatuan yang saling terkait. Terdiri dari 5 komponen pembangun SIG (Deny Charter, Irma Agtrisari, 2001:44) yaitu :

1. Data

Data merupakan informasi utama yang menunjang SIG, terdiri dari:

- Data spasial yang berupa peta topografi, peta tematik, dan peta-peta lainnya

➤ Data Non Spasial meliputi nama, atribut serta data-data penunjang

2. Perangkat Keras (*Hardware*)

Hardware merupakan seperangkat fasilitas berupa komputer yang dilengkapi dengan alat bantu sebagai pendukung pekerjaan seperti printer,ploter,dan scanner.

3. Perangkat lunak (*Software*)

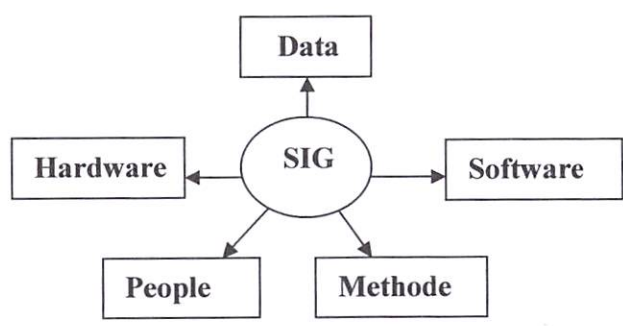
Software merupakan media yang digunakan dalam memproses data untuk menghasilkan informasi yang diinginkan .Didalam software terdapat perintah – perintah tertentu untuk memproses data input.

4. Methode / Operasional SIG

Merupakan operasi pengerjaan SIG dengan tata cara yang berurutan dalam proses pembuatan SIG

5. People / Operator

Operator adalah personal atau manusia yang bertugas menjalankan SIG



(Deny Charter,Irma Agtrisari ,Desain dan aplikasi SIG).

Gambar2.1 Komponen SIG

2.2.1.1. Data Input/Data Masukan

Data merupakan komponen yang sangat penting, karena merupakan dasar dalam penyediaan informasi bagi pemakai. Data *input* SIG dapat berupa:

- Data dari foto udara
- Data dari penginderaan jauh dan image prosesing
- Data dari peta
- Data dari tabular
- Data survey lapangan

Adapun tipe-tipe data input SIG meliputi:

1. Jaringan titik geodesi

Tingkat ketelitian jaringan titik kontrol

2. Unsur-unsur topogrsfi

Jalan, Jalan kereta api, lapangan terbang, jembatan, bangunan, kuburan, danau tambak, sungai, hutan, dan lain-lain.

3. Unsur-unsur kadastral

Persil tanah dan atributnya

4. Unsur-unsur batas luasan

Batas kota, batas kecamatan, batas kelurahan, batas desa, batas perencanaan, dan lain-lain.

5. Unsur-unsur utilitas

Jaringan telepon, air minum, pembuangan air, listrik dan lain-lain.

6. Zone sosial ekonomi

Tingkat kepadatan penduduk, tingkat kesejahteraan, dan lain-lain.

Dari tipe-tipe data masukan diatas, secara garis besar data input/masukan dibedakan menjadi dua yaitu data spasial dan data non spasial.

2.2.1.1.1. Data Spasial

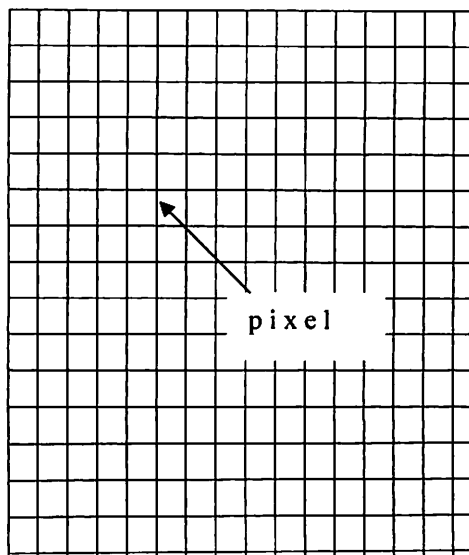
Data spasial merupakan data yang berisi informasi tentang lokasi, bentuk dan hubungan antar unsur geografinya. Data input spasial berupa data dari peta, dimana data dari peta sifatnya hardcopy (*analog*), untuk itu perlu mengubah data tersebut menjadi digital dengan cara mendigit data data dari peta. Digitasi ini merupakan cara yang paling umum digunakan untuk memasukkan data spasial. Data digital merupakan data yang format datanya sudah digital sehingga tidak perlu dikonversi lagi. Tipe data spasial yang paling umum digunakan adalah:

- Model Data Vektor

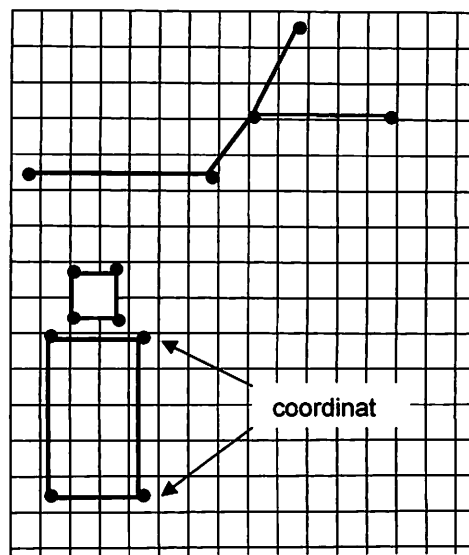
Suatu model data yang diperoleh dari hasil digitasi, dengan menggunakan luasan, garis dan titik untuk menampilkan obyek.

- Model Data Raster

Data yang diperoleh dari hasil scanner, pada sistem ini setiap elemen geografi yang disimpan dalam bentuk grid sel yang teratur. Struktur data dinyatakan dalam bentuk sel yang terbentuk atas baris dan kolom dari kiri atas, setiap sel mempunyai satu nilai dari setiap sel terisi informasi. Grup dari sel mewakili suatu unsur-unsur.



Gambar 2.2. Model Data Raster



Gambar 2.3. Model Data Vektor

2.2.1.1.2. Data NonSpasial

Data Non Spasial adalah data yang berupa dengan angka, teks, atau gambar yang berhubungan dengan unsur spasial. Data atribut biasanya disimpan dalam bentuk tabel, yang biasa disebut data tabular. Data tersebut bisa didapatkan dengan metode survey langsung dilapangan (data primer) atau menurunkan data dari laporan-laporan yang terdahulu.

2.2.1.2. Penyimpanan dan Pemanggilan Data

Penyimpanan dan pemanggilan data tergantung dari bagaimana data diorganisasi atau diatur didalam media penyimpanan data. Ada satu atau lebih *file* data yang disimpan di dalam sebuah cara yang terstruktur, seperti hubungan antara *Item* data yang berbeda. Penyimpanan data pengorganisasian data berdasarkan

jenis data dan struktur data, data spasial disimpan dalam *file* grafis sedangkan data *non spasial* disimpan dalam bentuk tabel-tabel.

2.2.1.3. Data Manipulasi dan Analisa

Fungsi ini sangat penting untuk membentuk informasi dari SIG. Keinginan pemakai (*user need*) sangat berperan besar sekali dalam menentukan model dan sebagai konsekuensinya analisa dari fungsi-fungsi SIG untuk melaksanakan pengarsipan, penentuan persyaratan-persyaratan informasi yang akan ditampilkan. Jadi pemakai (*User*) Keterlibatannya sangat penting selama perencanaan informasi desain sistem dan pengetesan.

2.2.1.4. Menampilkan Produk SIG

Produk dari SIG dapat ditampilkan dalam bentuk peta, tabel. Keduanya dapat disajikan pada *Hardcopy* (diatas kertas) dan *Softcopy* (didalam disket, cd-rom, dan lain-lain). *User* juga sangat berperan dalam menentukan bentuk keluaran yang dibutuhkan.

2.3. Basis Data

Basis data 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

tabelnya/struktur data dan relasi-relasi) dalam membentuk bangunan informasi yang penting (*enterprise*) (E. Prahasta, 2001: 41).

2.3.1. Sistem Manajemen Basis Data

Menurut pustaka (Korth, 1991:67), sistem manajemen basis data adalah kumpulan (gabungan) dari data yang saling berelasi (yang biasanya dirujuk sebagai suatu basis data) dengan sekumpulan program-program yang mengakses data-data tersebut. *Basis data Manajemen Sistem* atau Sistem Manajemen Basis data adalah tempat penyimpanan data beserta *users interface* yang dipersiapkan untuk memanipulasi dan administrasi basis data. (Ade, 2000:55).

DBMS akan berarti paket perangkat lunak (tanpa basisdata) *general purpose (pre-written computer program)* yang digunakan untuk membangun sistem basisdata tertentu. Dengan demikian, menurut pustaka ini DBMS adalah bagian dari sistem basisdata.

2.3.2. Keuntungan dan Kekurangan Basisdata

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

1. Pemakaian data bersama

Dengan menggunakan DBMS, informasi yang ada dalam basis data dapat digunakan secara efektif oleh beberapa pemakai dengan kontrol data yang terjaga.

2. 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.

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 dapat melakukan analisis data yang rumit sekalipun.

6. Data yang berlebihan dapat dikontrol

Pada tahap pemrosesan berkas untuk tiap aplikasi menggunakan berkas-berkas yang terpisah. Sehingga tidak jarang akan menghasilkan data yang rangkap (*redundant*) sehingga terjadi pemborosan biaya. DBMS dapat digunakan untuk menurunkan tingkat *redundancy* dan proses pembaruan data.

7. Pandangan pemakai

DBMS dapat memberikan kemudahan untuk membuat dan memelihara jendela pemakai (*user interface*) sesuai dengan pandangan pemakai terhadap basisdata. Sehingga ada kemungkinan basis data yang diakses sama, tetapi

jendela pemakai akan berbeda disesuaikan dengan pemahaman tiap pemakai terhadap basisdata menurut kebutuhan.

Dalam pemakaiannya DBMS juga memiliki beberapa kelemahan, diantaranya:

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 pemeliharannya (*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.

2.3.3 Komponen Sistem Basis Data

Dalam sistem basis data komponen-komponen pokok dapat dibagi menjadi lima bagian, yaitu:

1. Data

Data dalam basis data mempunyai sifat terpadu 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 harddisk) dan pengguna. Perangkat lunak inilah yang berperan melayani permintaan-permintaan pengguna, dimana perangkat ini mempunyai kemampuan utama untuk memasukkan data, memanipulasi data, menyimpan data, menganalisa data dan mengolah data

3. Perangkat keras

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

- a. Komputer
- b. Alat pemasukan data (Digitizer, Scanner)
- c. Alat pengeluaran data (Printer, Monitor)

4. Pengguna

Pengguna dapat diklasifikasikan menjadi tiga kategori diantaranya:

- a. Pengguna akhir adalah orang yang mengoperasikan program aplikasi
- b. Pemrograman aplikasi adalah orang yang membuat program aplikasi yang menggunakan basis data. Program aplikasi yang dibuat tentu saja sesuai dengan kebutuhan pengguna.
- c. *DBA/Basisdata Administrator* adalah orang yang bertanggung jawab terhadap pengolahan basis data.

Dalam pemakaian basis data setiap 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 basis data secara maksimal dengan mengembangkan aplikasi sesuai dengan bidang kerja masing-masing dan berfungsi sebagai *institutional framework* yang memegang peranan yang sangat penting dalam pengoperasian sistem basis data.

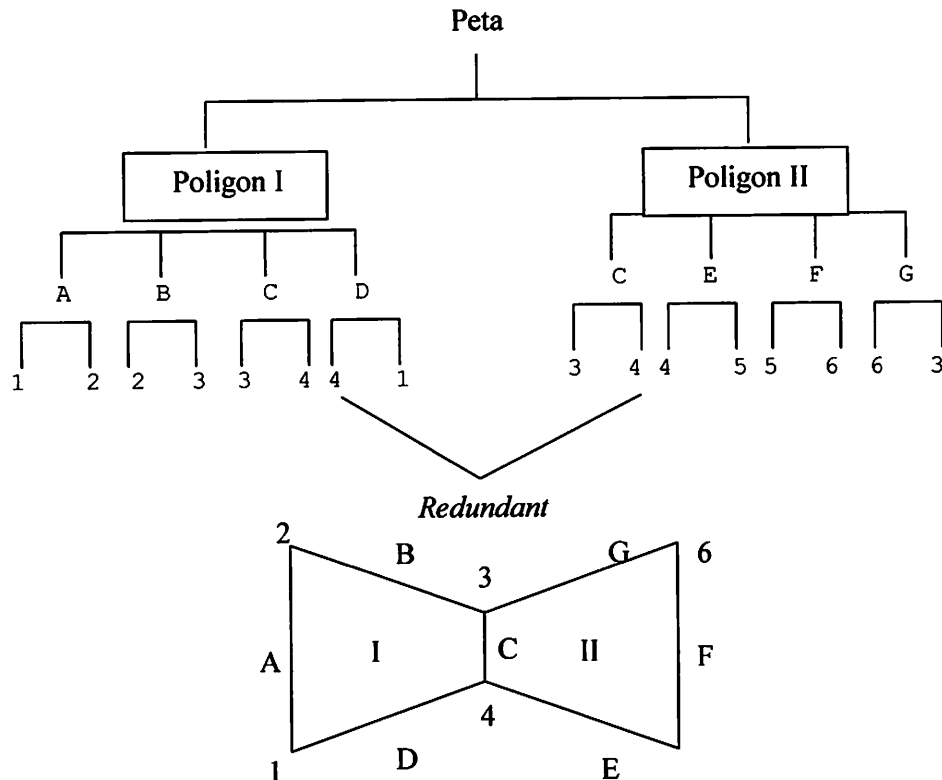
2.3.4. Struktur Basis Data

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

1. Struktur Basisdata *Hirarki*, (1970 – 1980) mempunyai beberapa ciri -ciri :
 - a. Struktur basisdatanya seperti pohon.
 - 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 *redundant* harus ada.

Contoh dari *struktur basisdata hirarki* dapat dilihat pada gambar 2.4.

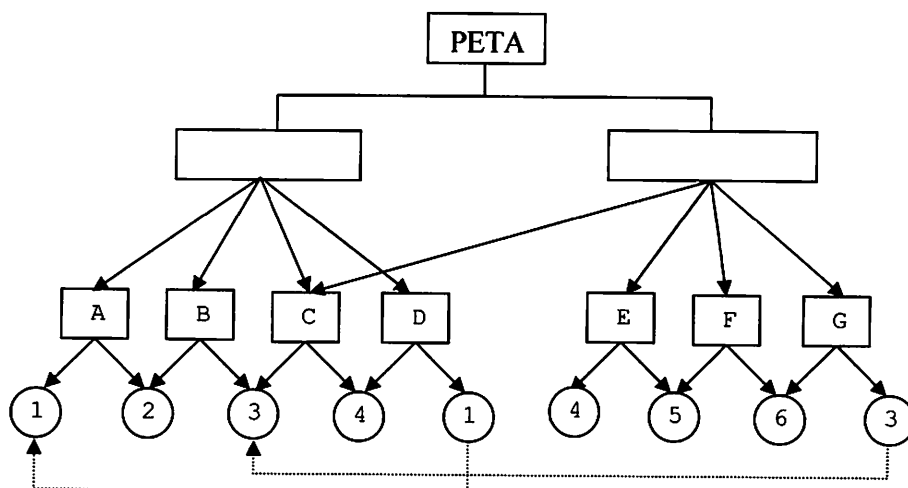


Gambar 2.4 Struktur Basisdata Hirarki

- 2. Struktur Basisdata *Network* (1970 – 1980) mempunyai ciri-ciri diantaranya:
 - a. Struktur basis datanya berupa pohon (seorang anak dapat mempunyai lebih dari satu orang tua).

- b. Hubungan antar entity : *one to one* (1:1), *one to many* (1:M), *many to many* (M:N) dapat dikuasai..
- c. Tidak ada data *redundant* 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.

Contoh dari *struktur basisdata network* dapat dilihat pada gambar 2.5.



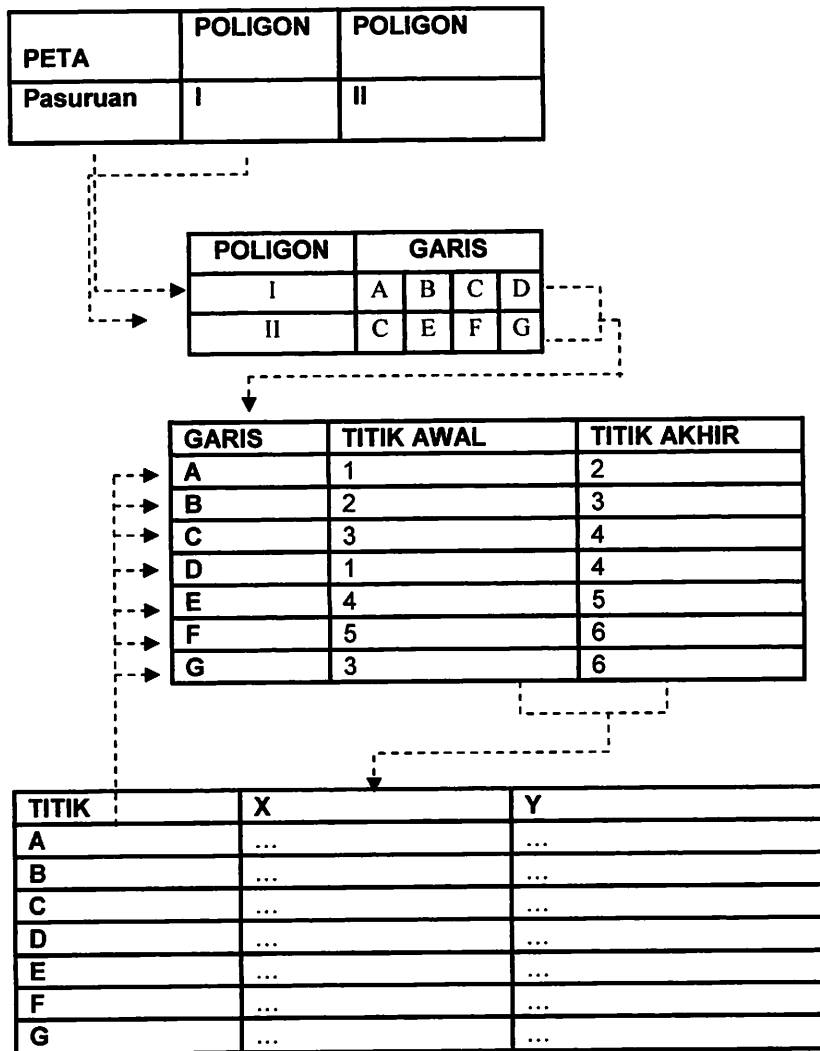
Gambar 2.5 Struktur Basisdata Network

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 attribut.

Beberapa karakteristik basisdata relational diantaranya:

- a. Penggunaan desain metodologi.
- b. Struktur basisdatanya yang *simple* dan sederhana (semua data disimpan didalam dua dimensional tabel).
- c. Semua basisdatanya *one to one* (1:1), *one to many* (1:M), *many to many* (M:N) dapat dikuasai.
- d. Tidak ada data *redundant* (normalisasi tabel).
- e. Pembentukan kembali struktur basisdatanya adalah mudah.
- f. Sangat baik dan *standard query language* (SQL).

Contoh dari *struktur basisdata relational* dapat dilihat pada gambar 2.6.



Gambar 2.6. Struktur Basisdata Relational

4. Struktur Basisdata *Object Oriented*, mempunyai ciri -ciri, diantaranya:

- a. Sangat cocok untuk suatu persoalan atau situasi yang sangat kompleks.
- b. Teknologi masa depan yang menjanjikan.

2.3.5. Model Data Sistem Basis Data

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

- Entity (*entitas*), Sebuah objek atau konsep yang dikenal oleh enterprise sebagai sesuatu yang dapat muncul independent. Bisa jadi diidentifikasi yang unik dan penggambaran data yang disimpan. Pada model relasional, entitas akan menjadi tabel.
- Atribut (*attribute*), merupakan keterangan-keterangan yang dimiliki oleh suatu entity.
- Hubungan (*relationship*), Bagian dari bumi yang sedang digambarkan atau dimodel database, bisa seluruh organisasi atau bagian tertentu.

Konsep Hubungan antar entitas (E-R)

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

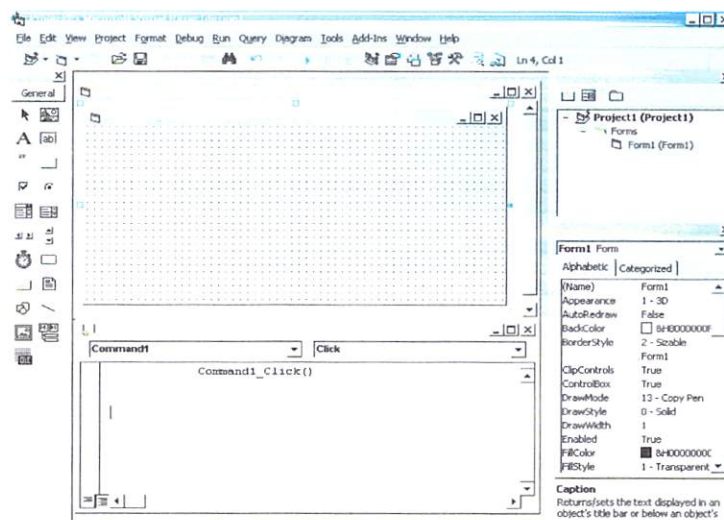
1. *Hubungan satu kesatu* (1 : 1), artinya nilai entiti berhubungan dengan satu nilai entiti yang lainnya, aturannya adalah sebagai berikut:
 - a. Bila kedua entitynya obligatory, maka hanya dibuat satu tabel.

- 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 entitynya 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 entitynya 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 entitynya 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).

2.4. Visual Basic 6.0

Visual Basic merupakan bahasa pemrograman berbasis windows. Kelebihan dari visual Basic adalah dapat memanfaatkan fasilitas MS-Windows secara optimal, menyediakan obyek-obyek sehingga berguna dan mudah dipakai ,dapat diterapkan pada jaringan ,didesain dengan arsitektur terbuka dan banyak vendor yang menyediakan fasilitas tambahan untuk mempermudah programmer membuat suatu aplikasi.(*M.Agus .J.Alam,Pt.Elex Media Komputindo*)

Tampilan layar program Microsof Visual Basic 6.0 dapat dilihat pada gambar 2.7.



Gambar 2.7 Tampilan Layar Visual Basic

2.4.1 Elemen Utama Visual Basic

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

No	Jendela	Keterangan
1	Form	<i>Form</i> merupakan latar belakang program <i>windows</i> yang ditulis. Anda menggambar dan meletakkan <i>item</i> itu pada <i>form</i> , sehingga pengguna program terbiasa melihat dan berinteraksi.
2	Toolbox	<i>Toolbox</i> berisi alat-alat yang diperlukan program, hal ini nampaknya jelas tetapi kita perlu tahu bahwa alat-alat pada Visual Basic lebih sering disebut kontrol-kontrol. Kontrol-kontrol ini yang kita tempatkan pada <i>Form</i> .
3	Project	Dalam jendela <i>project</i> terdapat daftar semua <i>file</i> aktif yang digunakan. Jendela <i>project</i> berisi uraian <i>file</i> sederhana, tetapi semua <i>file</i> tersebut tempatnya terpisah di dalam <i>harddisk</i> .
4	Properties	Terdapat elemen individual pada pembuatan program. Bila kita ingin melihat atau mengedit <i>property</i> dari berbagai form atau kontrol kita dapat melihat serta mengedit properti-nya di satu jendela properti
5	Code	<i>Code</i> adalah <i>source code</i> program. Ketika pengguna menjalankan program Visual Basic dan komputer kita menginterpretasikan sebagai <i>source code</i> maka komputer akan mengeksekusi instruksi didalam <i>source code</i> tersebut.

Tabel 2.1 Jendela Visual Basic 6.0.

2.5. Map Object 2.1

Selain dari Visual Basic juga tersedia fasilitas Active control yang berisi kontrol tambahan yang digunakan untuk mengembangkan aplikasi yang dibuat. Active control merupakan file yang berekstensi OCX yang diletakkan pada toolbox.

Map Object merupakan fasilitas dari Visual Basic yang berguna untuk pengolahan data spasial/peta yang diolah dan dikontrol melalui source code dengan bahasa pemrograman Visual Basic. Dari Map Object ini dapat dibuat suatu aplikasi Sistem Informasi Geografi, dan dalam penelitian ini akan dibuat Sistem Informasi Bank Perkreditan Rakyat. Pada Map Object terdapat komponen icon yang disebut dengan komponen OCX Map Object yang merupakan salah satu fasilitas Visual Basic yang berbentuk file OCX yang kegunaannya untuk menampilkan peta dalam pembuatan program Sistem Informasi Geografi.

2.6. Desain Program Visual Basic 6.0 dan Map Object 2.1

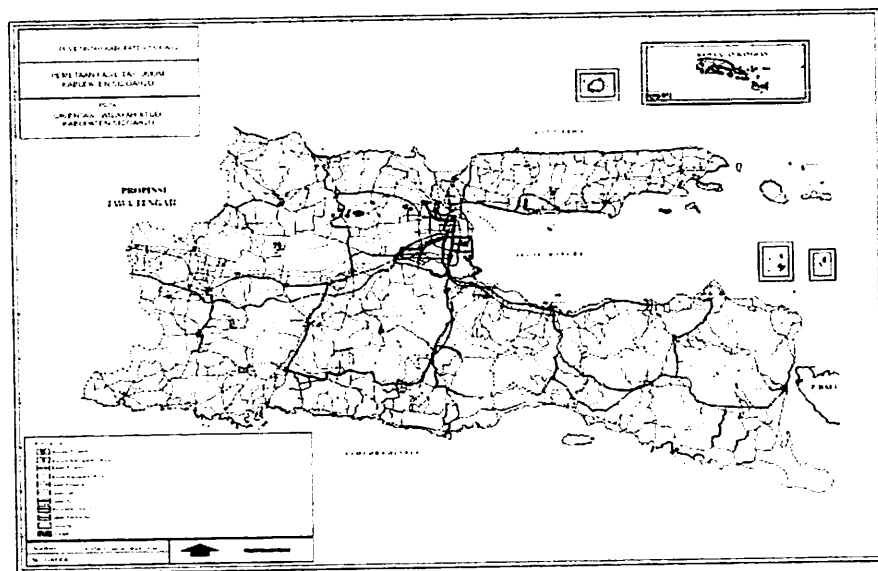
Dalam Map Object 2.1 didukung oleh ActiveX Data Object (ADO) yang merupakan model basis data dari Microsoft dengan fungsi sebagai alat untuk menjalankan Universal Data Access (UDA). Tujuan dari UDA agar pemrograman Visual Basic dapat mengikuti standart OLEDB dalam membuat suatu aplikasi OLEDB adalah suatu standart basis data yang digunakan sebagai alat untuk menghubungkan data dari berbagai sumber data, misalnya penyiompanan data dalam sebuah basis data, sebuah file teks, atau lembar kerja. Dalam Map Object dapat dibuat tabel object dengan menghubungkan data dari SQL Server yang menggabungkan informasi ke dalam Shapefile.

BAB III

PELAKSANAAN PENELITIAN

3.1. Diskripsi Wilayah Penelitian

Diskripsi merupakan gambaran umum penelitian mencakup kondisi fisik lokasi dan wilayah Kabupaten Pasuruan yang terletak di propinsi Jawa Timur.



Gambar 3.1. Peta Lokasi Pasuruan diwilayah Jawa Timur

3.1.1. Keadaan Umum Kabupaten Pasuruan

Kabupaten Pasuruan mempunyai luas wilayah kurang lebih sebesar 1113.434,65 ha dan secara geografis terletak pada posisi 112°6' - 113°1' Bujur Timur dan antara 7°5' - 7°9' Lintang Selatan. Kabupaten Pasuruan merupakan salah satu kabupaten di Propinsi Jawa Timur yang berjarak ± 40 Km sebelah Selatan dari ibukota Surabaya dan berbatasan dengan :

- Sebelah Utara : Kabupaten Sidoarjo, Kota Pasuruan dan Selat Madura
- Sebelah Timur : Kabupaten Probolinggo dan Kabupaten Lumajang
- Sebelah Selatan : Kabupaten Malang dan Kabupaten Lumajang
- Sebelah Barat : Kabupaten Mojokerto

(Sumber BPN : Kabupaten Pasuruan Dalam Angka 2001)

3.2.Materi Penelitian

Materi atau data – data yang digunakan dalam penelitian meliputi :

1. Data Spasial

- Peta Administrasi Kabupaten Pasuruan skala 1 : 25.000 sumber BAKOSURTANAL 2002
- Peta Administrasi tiap Kecamatan di Pasuruan skala 1 : 25.000
- Peta Jaringan Jalan Kabupaten Pasuruan skala 1 : 25.000
- Lokasi Usaha Kecil dan Menengah (UKM) hasil survei lapangan dengan menggunakan GPS Navigasi

2. Data Non Spasial

- Data atribut batas administrasi :
 1. Data Batas Kabupaten
 2. Nama Kecamatan
 3. Nama Kelurahan
- Data atribut UKM:
 1. Data Profil UKM
 2. Data Kepegawaiaan

3. Data Pemasaran
4. Data Produksi
5. Data Foto

3.3. Alat Penelitian

Alat yang digunakan dalam penelitian kali ini meliputi

1. Hardware

- Monitor
- CPU (*Control Processing Unit*)
 - Pentium IV 1,6 GHz
 - RAM 128 MB
 - Floopy Drive 1.44 MB
 - CD ROM 52x
 - Hard Disk 20 GB
- Keyboard
- Mouse
- Printer
- Scanner
- GPS Garmin III

2. Software

- AutoCad 2000
- Arcview 3.1
- Arc Info 3.5
- Visual Basic 6.0
- Microsoft Acces 2000

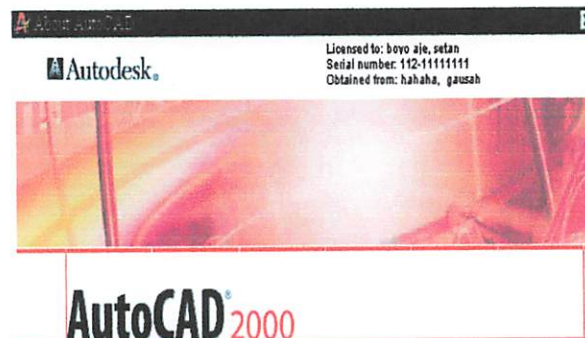
- Map Object 2.1

3. Peralatan lainnya

- Alat tulis
- Kamera foto
- Sepeda motor dan perlengkapannya

□ AutoCad 2000

AutoCad 2000 adalah software yang berbasis Computer Aided Design (CAD). Penggunaan AutoCad 2000 dalam bidang geodesi dimanfaatkan untuk pembuatan peta digital dalam survei dan pemetaan. AutoCad 2000 mampu mendesain bentuk gambar 2D dan 3D. AutoCad 2000 digunakan sebagai media penggambaran grafis dan untuk mengubah data analog menjadi data digital melalui proses digitasi. Tampilan awal AutoCAD seperti pada gambar 3.2.



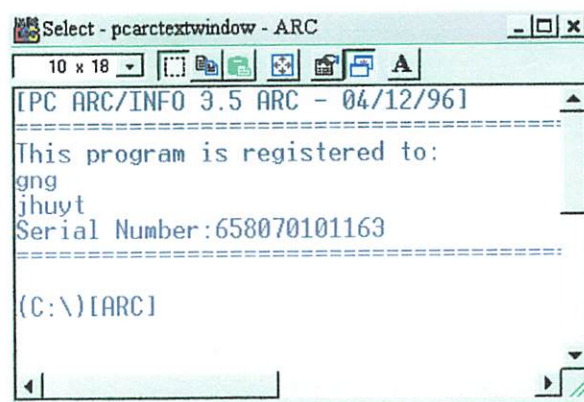
Gambar 3.2. Tampilan AutoCad 2000

□ Arc Info 3.5

Arc Info 3.5 adalah software yang dikembangkan oleh ESRI dan berbasis Sistem informasi Geografis yang dirancang untuk proses pemetaan sehingga dapat menghasilkan informasi spasial. Arc Info 3.5 digunakan untuk membangun

topologi dengan perintah *Build* dan *Clean* serta dalam pemberian ID (*labelling*).

Tampilan Arc Info 3.5 dapat dilihat pada gambar 3.3.



Gambar 3.3. Tampilan ArcInfo 3.5

□ Arc View 3.1

Arc View 3.1 merupakan suatu software yang memiliki kemampuan untuk melakukan visualisasi, meng-*explore*, menjawab *query* (baik *basisdata* spasial maupun non spasial), menganalisis data secara geografis . Arc View 3.1 merupakan software dalam pengolahan Sistem Informasi Geografis dan pemetaan yang telah dikembangkan oleh ESRI. Pada penelitian ini ArcView digunakan sebagai media setelah proses topologi di Arc Info 3.5. Tampilan awal pada perangkat lunak ArcView 3.1 dapat dilihat pada gambar 3.4.



Gambar 3.4. Tampilan Awal Pada ArcView versi 3..1.

3.3.1. Tahapan Penelitian

Pada penelitian ini terdapat tahapan dalam penelitian , yang telah disusun dalam diagram alir penelitian pada diagram 3.1, yaitu sebagai berikut :

1. Persiapan

Persiapan merupakan tahap awal dalam proses penelitian ini. Didalam tahap persiapan merupakan tahap yang sangat berperan dalam keberhasilan penelitian, karena tahap ini berisikan perencanaan penelitian yang meliputi program yang akan digunakan, data yang diperlukan dalam penelitian, serta literatur-literatur yang akan digunakan sebagai referensi dalam penelitian.

2. Pelaksanaan

Tahap pelaksanaan dilakukan apabila tahap perencanaan sudah tersusun . Pelaksanaan disini terbagi menjadi :*Pengumpulan data , peninjauan lokasi , pengolahan data.*

a. Pengumpulan Data

Data yang digunakan berupa data spasial dan non spasial yang dikumpulkan untuk proses penelitian.

- **Data Spasial**

Data spasial pada penelitian ini yaitu peta Kabupaten Pasuruan skala 1 : 25.000, yang sudah melalui proses digitasi dan disimpan dalam format (*.DWG). Peta digital yang dihasilkan merupakan rangkaian dari beberapa *layer* .

- **Data Non Spasial**

Data Non Spasial pada penelitian ini yaitu data atribut mengenai profil UKM yang ada, serta data mengenai produk hasil dari UKM tersebut.

b. Peninjauan lokasi

Setelah didapatkan data spasial yang ada kemudian dilakukan peninjauan lokasi, kriteria pekerjaannya yaitu penentuan titik dengan menggunakan GPS. Mencari informasi produk, pemasaran dan jumlah pegawai.

c. Pengolahan data

- **Data Spasial**

Pengolahan Data spasial dilakukan pada Peta dari (*.DWG) di simpan ke (*.DXF) agar dapat diproses dalam pembuatan topologi. Kemudian data ditampilkan dalam Arc View 3.1 dan disimpan ke format (*.shp), yang setiap data telah memiliki identitas tersendiri seperti batas administrasi desa, batas administrasi kecamatan, batas administrasi kabupaten dan jalan yang didapat dari beberapa layer yang ada.

- **Data Non Spasial**

Proses pengolahan Data Non Spasial dikelompokkan dan disusun ke suatu *database* dengan memanfaatkan *Microsoft Access 2000* melalui proses input data tabel, penentuan enterprise rule, normalisasi tabel dan pembuatan Query.

3. Penggabungan Data (*Joint item*)

Joint item dilakukan untuk menggabungkan data spasial dan non spasial

4. Pembuatan Program

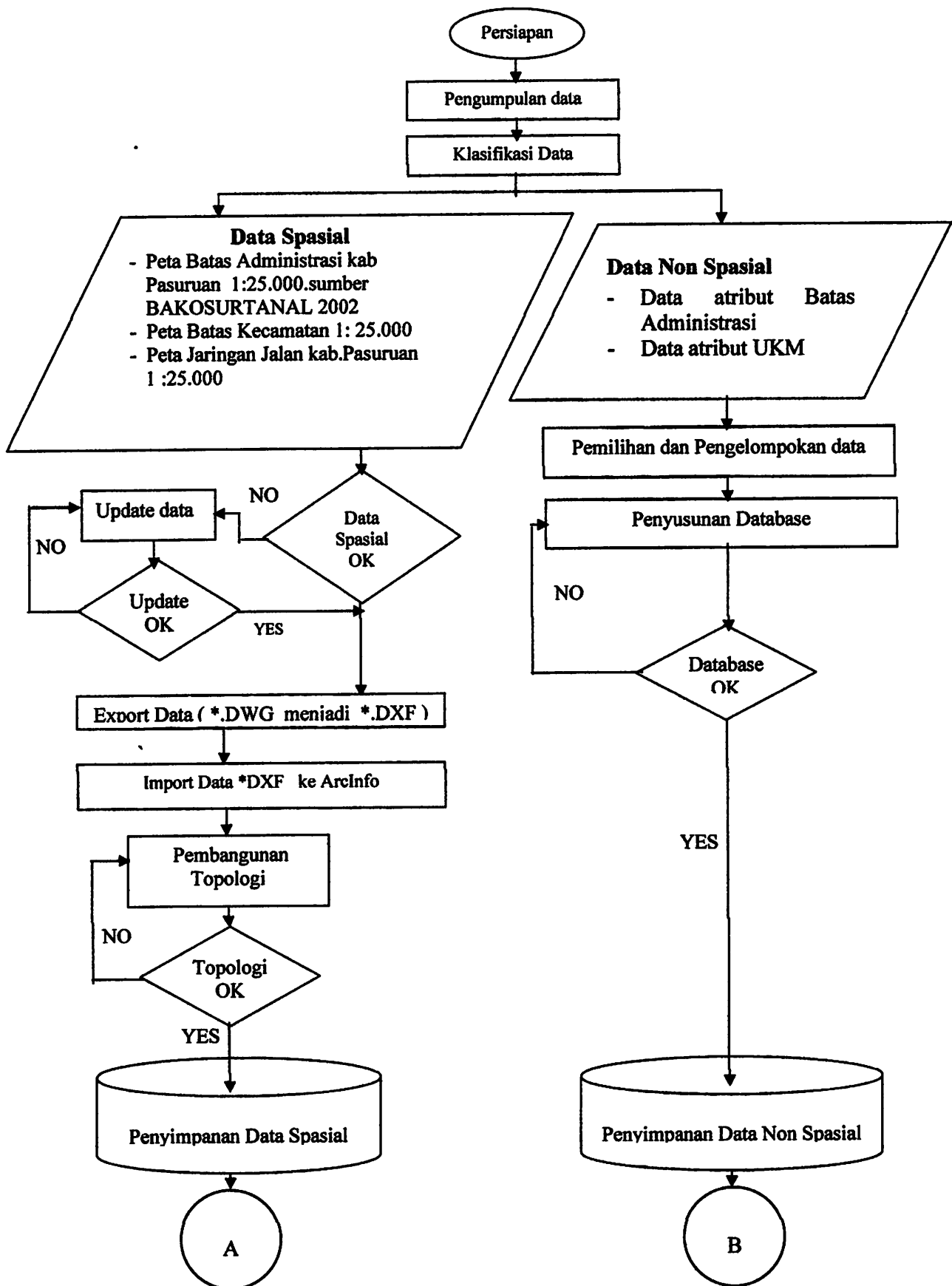
Setelah didapat data spasial dengan format (*.shp) dan data non spasial, kemudian dilakukan penggabungan di dalam program yang akan dibuat dengan menggunakan program Microsoft Visual Basic 6.0 dan MapObject 2.1, dilanjutkan dengan membuat program Penyajian informasi Usaha Kecil dan Menengah (UKM) itu sendiri sehingga dihasilkan tampilan menarik dan mudah dalam operasinya serta mempercepat proses pencarian informasi yang dibutuhkan .

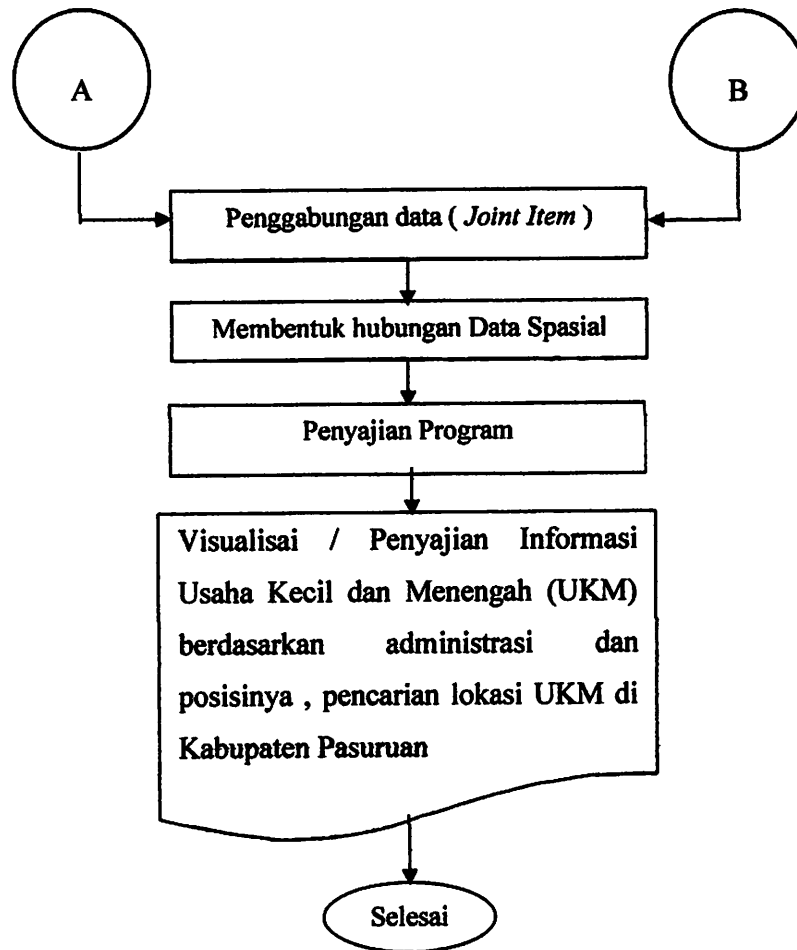
5. Penyajian program

Setelah program yang telah selesai dibuat dan sudah sesuai dengan keinginan penulis, kemudian dilakukan user opinion untuk mengetahui sejauh mana tingkat kebutuhan yang diperlukan (*user*) pengguna sebelum program ini diberikan kepada para pengguna (*user*) agar mudah dalam pengoperasian dan cakupan informasi yang ditampilkan.

6. Informasi Usaha Kecil dan Menengah di Kabupaten Pasuruan

Setelah dilewatinya proses *user opinion* maka program ini sudah mencapai hasil akhir yang diinginkan oleh semua pihak dalam hal ini penulis dan pengguna (*user*) yaitu program penyajian informasi Usaha Kecil dan Menengah di Kabupaten Pasuruan. Penyajian informasi berupa visualisasi hasil yang berisi informasi lengkap dan tersusun dalam format basis data yang terkait dengan spasial data berupa peta.





Diagramr 3.1. Diagram alir Penelitian

3.4 Tahap Pembuatan Program

Dalam pembuatan program penyajian informasi Usaha Kecil dan Menengah di Kabupaten Pasuruan terdapat beberapa tahapan yang perlu dilalui, yang diagram alirnya dapat dilihat pada diagram 3.2, yang dapat dijelaskan :

1. Read Project Data

Sub sistem ini adalah pemanggilan data yang telah diproses sebelumnya.

2. Initialize Control

Sub Sistem ini merupakan tahap mengenali perintah (*control*) untuk menjalankan program.

3. Display Form

Sub sistem yang menampilkan *form* tampilan dari pembacaan data spasial

4. User Action

Sub sistem yang dimana pengguna (*user*) menampilkan dan mencari informasi yang tersedia di dalam program.

5. Search

Menu perintah pencarian dalam Visual Basic 6.0 sesuai dengan perintah yang dijalankan.

6. Menu & Toolbar

Icon yang berfungsi untuk menampilkan informasi langsung dari peta spasial yang ada di program.

7. Help

Sub sistem merupakan menu bantuan yang berisikan petunjuk (*guide*) menjalankan program.

8. Menu & Command

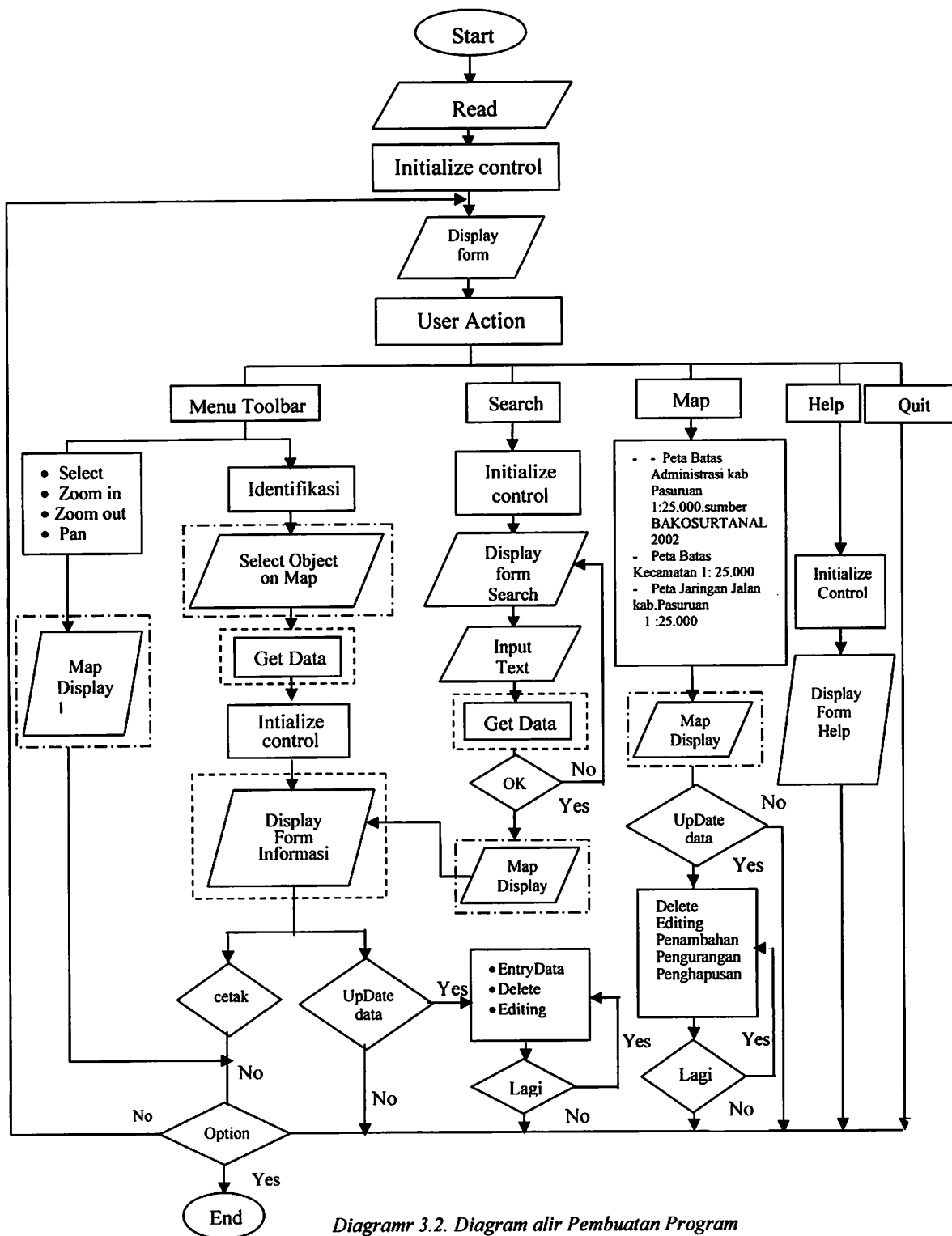
Icon yang berfungsi untuk menjalankan perintah untuk menampilkan informasi Usaha Kecil dan Menengah dan batas administrasi serta jaringan jalan.

9. Input Query

Sub sistem yang diajukan untuk menentukan data mana saja yang akan disimpan atau ditampilkan di dalam *basisdata* dan menentukan bagaimana data tersebut direlasikan.

10. Data Updating

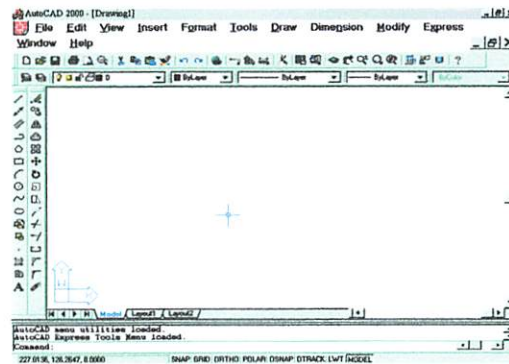
Sub sistem yang menunjukkan adanya perintah pembaruan data atau perubahan data secara berkala sehingga tidak menghilangkan informasi yang sebelumnya.



Diagramr 3.2. Diagram alir Pembuatan Program

3.5 Pemasukan Data Spasial

Pemasukan data spasial adalah proses merubah dari *analog* menjadi *digital* dengan cara mendigitasi peta tersebut dengan menggunakan *software AutoCad 2000*. Tampilan jendela *software AutoCad 2000* dapat dilihat pada gambar 3.5.

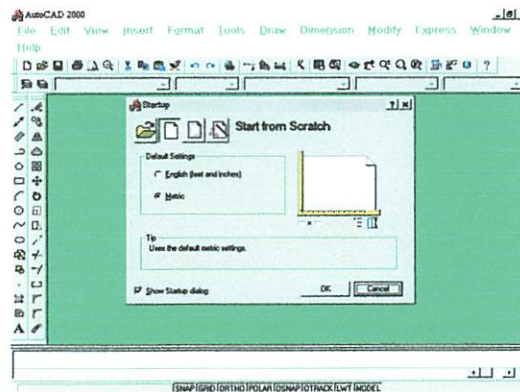


Gambar 3.5 Menu AutoCad 2000

Adapun tahap-tahap pendigitasiannya menggunakan metode *onscreen* yaitu :

1. Buka AutoCad 2000

Tampilan layar AutoCad 2000



Gambar 3.6 Tampilan AutoCad

Setelah proses diatas akan muncul tampilan AutoCad 2000,yang terdiri dari beberapa toolbar,baris status/dibagian bawah,layar gambar ,dan jendela perintah (Command).

2.Penggunaan Perintah

Penggunaan perintah dalam pelaksanaan operasional kerja dilakukan dengan Menu,Toolbar dan pengetikan perintah pada Command line.

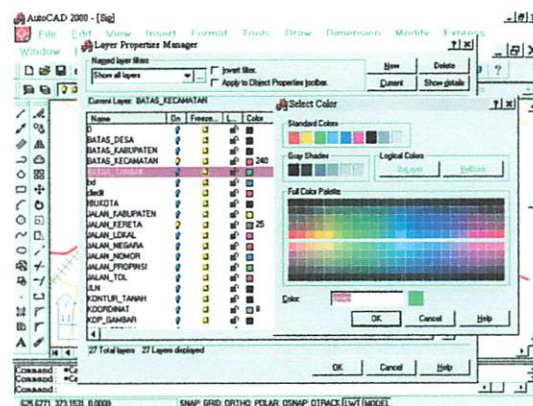
3.Membuka file

Untuk membuka gambar yang tersimpan pada Hardisk ,disket atau CD. Dapat dipilih menu **File** ,pilih **Open** ,atau lebih singkatnya mengklik icon pada Toolbar, lalu **Browse** untuk memilih lokasi dimana file gambar tadi berada.

4 .Pembuatan Layer

4.1. Membuat Layer Baru

Buka menu **Layer** dari Menu **Format** , atau klik dari toolbar



Gambar 3.7 Tampilan menu Layer

Keterangan:

- New : Untuk membuat layer baru
- Delete : Untuk menghapus layer baru
- 0 (On / Off) : Untuk menghidupkan/mematikan layer .

F (Freeze/Thaw) : Untuk membuka / menutup layer .(Freeze : layer tidak ditampilkan dilayar , thaw : Layer ditampilkan dilayar)

L (Lock/Unlock) : Untuk mengunci layer sehingga object yang digambar dengan layer ini tidak dapat diseleksi atau membebaskan (gambar kunci terbuka) suatu layer sehingga object yang digambar dengan layer ini dapat diseleksi kembali.

C (Color) : Untuk memilih warna

L (linetype) : Untuk memilih pola garis

Dalam keadaan awal hanya terdapat pola garis continuous tetapi AutoCad menyediakan beberapa pola garis dalam file ACADICO.LIN. Untuk memilih pola garis , harus menyiapkan (*Load*) pola yang diperlukan dalam kotak select linetype . Apabila memilih keseluruhan pola dilakukan *select All*

4.2. Memberlakukan Layer (Layer Current)

Layer dibuat untuk menggambarkan berbagai kondisi (warna, pola garis dan lain-lain). Pemberlakukan suatu layer dapat juga dilakukan melalui Object Properties Toolbar

4.3. Membuat Object dengan layer Current

Untuk mengganti layer , pilih object dan tandai object . kemudian pilih *Make Object's Layer Current* pada object Property Toolbar

5. Proses Editing

Proses editing meliputi tahap pengkoordinasian hasil dari proses digitasi untuk mengetahui terjadinya kesalahan dan memperbaiki timbulnya kesalahan tersebut. Kesalahan tersebut seperti tidak menyambung garis yang melewati batas, bentuk sungai, dan masih

banyak lagi. Untuk itu harus dilakukan editing agar hasil yang diperoleh akan lebih baik.

Perintah yang dilakukan dalam proses editing seperti:

1. EXTEND

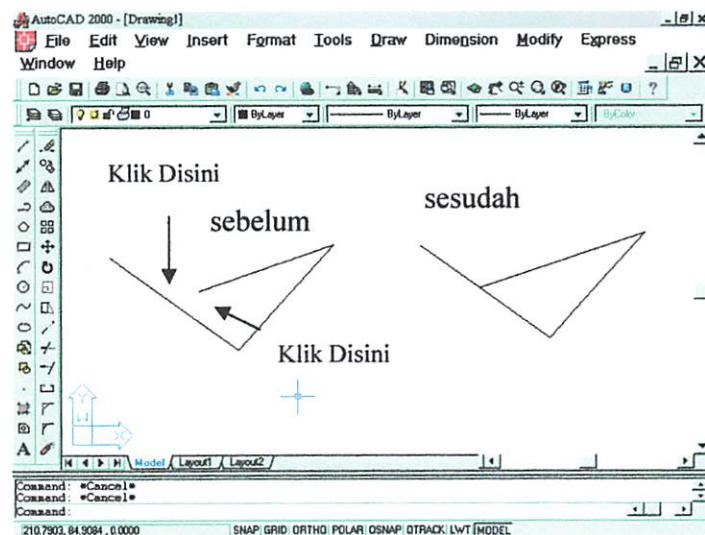
Digunakan untuk memperpanjang suatu objek gambar sampai pada batas yang ditentukan.

```
Command: _extend
Current settings: Projection=UCS Edge=None
Select boundary edges ...
Select objects: Specify opposite corner: 0 found
Select objects: 1 found

Select objects:

Select object to extend or [Project/Edge/Undo]:
Select object to extend or [Project/Edge/Undo]:

Command: |
```



Gambar 3.8 *Extend sesudah dan sebelum*

2. TRIM

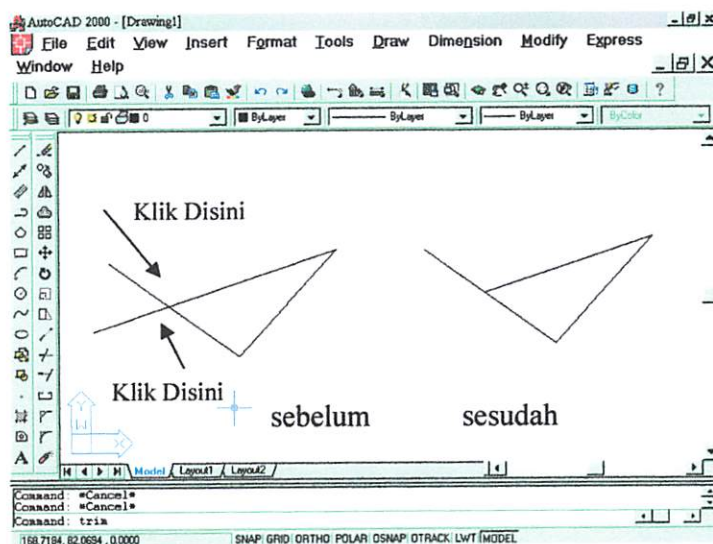
Trim adalah perintah untuk menghilangkan bagian dari suatu obyek gambar yang dibatasi oleh garis pembatas.

```
Command: _trim
Current settings: Projection=UCS Edge=None
Select cutting edges ...
Select objects: 1 found

Select objects:

Select object to trim or [Project/Edge/Undo]:
Select object to trim or [Project/Edge/Undo]:

Command: |
```



Gambar 3.9 Trim sesudah dan sebelum

3. PEDIT

Pedit digunakan untuk mengedit garis seperti menyambung 2 buah garis menjadi satu garis.


```

Command: pe
PEDIT Select polyline:
Enter an option [Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype
gen/Undo]: j

Select objects: 1 found

Select objects: 1 found, 2 total

Select objects:

4 segments added to polyline

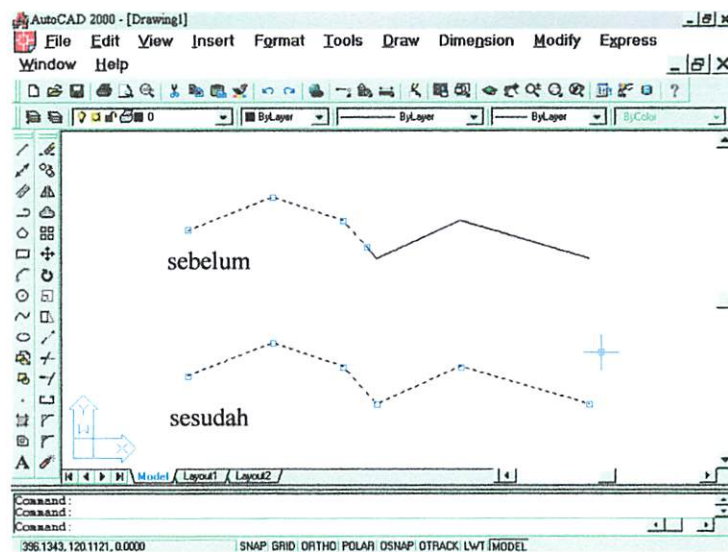
Enter an option [Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype
gen/Undo]:

```

```

Command:

```



Gambar 3.10 Pedit (join) sesudah dan sebelum

4. MOVE

Digunakan untuk memindahkan obyek dari suatu lokasi ke lokasi yang lain.

```

Command: move

```

```

Select objects: 1 found

```

```

Select objects:

```

```

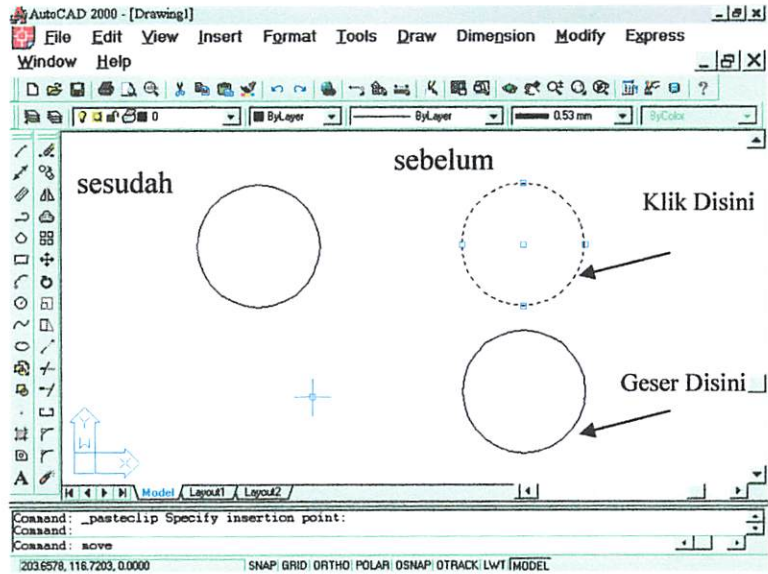
Specify base point or displacement: Specify second point of displacement or
<use first point as displacement>:

```

```

Command: |

```

Gambar 3.11 Move sesudah dan sebelum

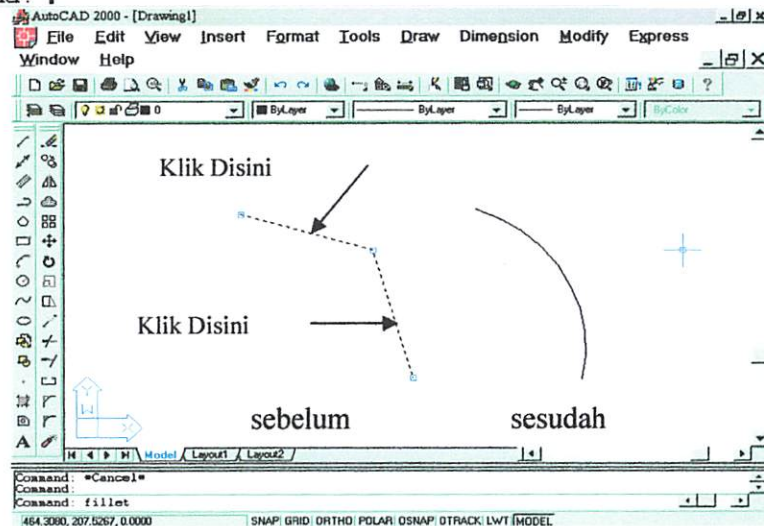
5. FILLET

Digunakan untuk memperhalus pertemuan antara garis dengan radius tertentu.

Command: fillet

Current settings: Mode = TRIM, Radius = 0.5000
 Select first object or [Polyline/Radius/Trim]:
 Select second object:

Command: |



Gambar 3.12 Fillet sesudah dan sebelum

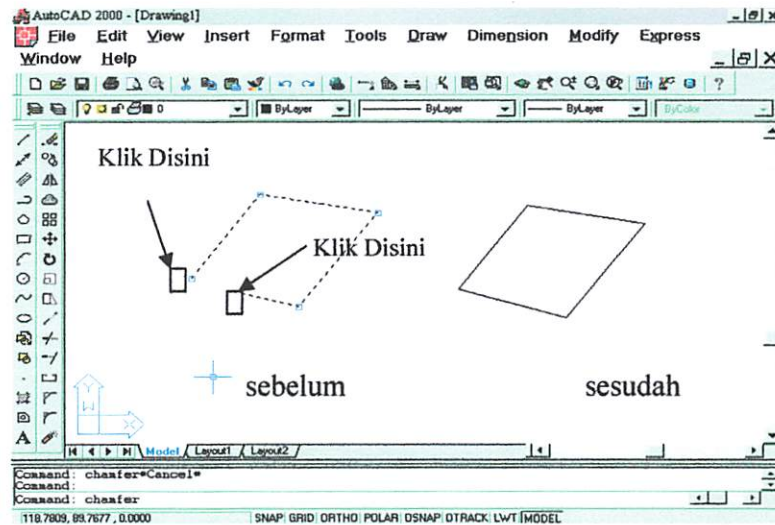
6. CHAMFER

Digunakan untuk membentuk hubungan antara dua garis yang berpotongan

Command: chamfer

(TRIM mode) Current chamfer Dist1 = 0.5000, Dist2 = 0.5000
Select first line or [Polyline/Distance/Angle/Trim/Method]:
Select second line:

Command:



Gambar 3.13 Chamfer sesudah dan sebelum

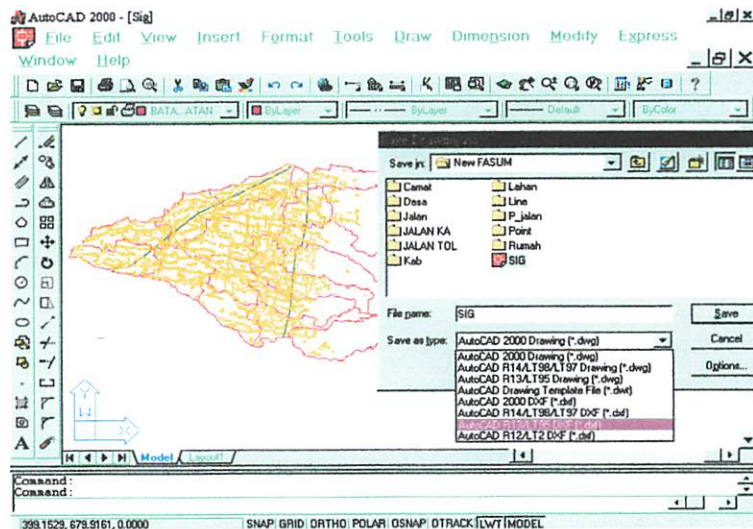
7. EXPORT DATA

Export ini dilakukan untuk mendapatkan data dengan format yang sebelumnya berformat DWG. Hal ini dilakukan karena untuk dapat membuka dan membaca data pada program Arc/Info diperlukan data dengan format DXF. Adapun cara untuk mengexport data dari DWG ke DXF adalah sebagai berikut :

1. Data yang akan di-*ekspor* masih terbuka pada program AutoCAD, kemudian memilih menu *file* setelah itu klik *Eksport*.
2. Setelah muncul menu ekspor data isikan nama *file* yang dikehendaki, seteh itu memilih *Save As* dengan tipe eksrention **DXF**

3. Klik tombol *Save*.

Tampilan jendela untuk eksport data terdapat pada gambar 3.15.

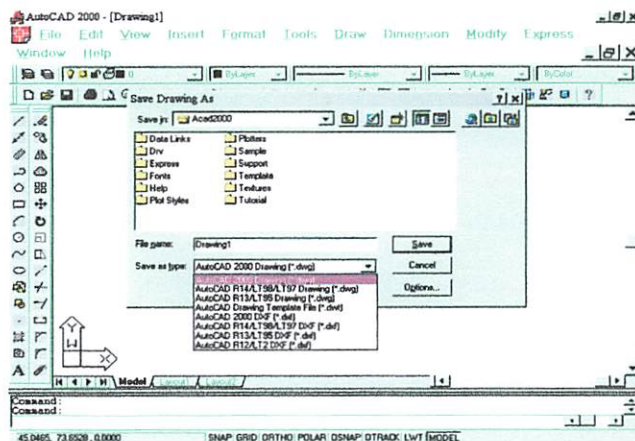


Gambar 3.14 Eksport Data

3.6.. Penyimpanan

Saat melakukan proses pekerjaan pada AutoCad 2000 sebaiknya dilakukan penyimpanan secara berkala, hal ini dilakukan untuk mengantisipasi terjadinya kerugian / kehilangan pada gambar bila terjadi putusnya hubungan listrik atau kapasitas disket/hardisk yang sudah penuh .

Perintahnya : Buka File .pilih *Save*



Gambar 3.15. Tampilan Menu Save

Pada penyimpanan data gambar dilakukan pemilihan nama ekstensi agar mudah melakukan import data. Secara otomatis AutoCAD 2000 memilih DWG sebagai ekstensi pembentuknya.

3.7 Pembuatan Topologi

Untuk mendapatkan hubungan spasial antar *feature* pada peta digital, digambarkan dengan menggunakan topologi. Topologi adalah hubungan yang digunakan untuk menyajikan persambungan antar pertemuan *feature*.

Setelah dilakukan ekspor data dari **DWG** ke **DXF** kemudian dilakukan pembuatan topologi (*coverage*) dalam pembuatan topologi ini menggunakan dua (2) cara yaitu **CLEAN** dan **BUILD**. Tahapan pekerjaan yang dilakukan dalam pembuatan topologi adalah sebagai berikut :

1. *Import* data dari hasil ekspor pada AutoCAD yang ber-*extension* DXF.

Adapun caranya adalah :

- a. Membuka program **Arc/Info 3.5 for DOS** dengan terlebih dahulu komputer diset di MS DOS.

Mengetikkan perintah ARC pada prompt C, yaitu C:\ARC <Enter>

- b. Setelah muncul logo Arc/Info dan sudah berada dalam program tersebut, kemudian mengeditkan perintah untuk *import* data :

KEC.SBX [SHP] KECAMA~1.DXF

11 file(s) 510,781 bytes

12 dir(s) 508,350,464 bytes free

(D:\MYDOC~1\ARMEN\PETA)[ARC]dxfarc kecam~1 KEC

[PC ARC/INFO 3.5 DXFARC - 04/12/96]

Enter layer names and options (type END or \$REST when done)

=====

Enter the 1st layer and options: kecamatan

Enter the 2nd layer and options:

Character string expected.

Done entering layer names and options (Y/N)? y

Do you wish to use the above layers and options (Y/N)? y

Processing KECAMA~1.DXF...

No labels, killing XCODE...

18 Arcs written.

0 Labels written.

0 Annotations written.

0 Annotation levels.

(D:\MYDOC~1\ARMEN\PETA)[ARC]

2. Pembentukan topologi

Data yang sudah diimport kemudian dibentuk topologinya dengan menggunakan perintah sebagai berikut :

```
(D:\MYDOC~1\ARMENPETA)[ARC]build KEC
```

```
[PC ARC/INFO 3.5 BUILD - 04/12/96]
```

```
Building polygons...
```

```
Sorting input file...
```

```
Sorting label file...
```

```
Processing...
```

```
Assigning final IDs...
```

```
Writing arc file...
```

```
Generating polygon report...
```

```
Creating attribute file for KEC.
```

```
Sorting User-IDs...
```

```
Merging record 18
```

```
(D:\MYDOC~1\ARMENPETA)[ARC]
```

```
(D:\MYDOC~1\ARMENPETA)[ARC]clean KEC
```

```
[PC ARC/INFO 3.5 CLEAN - 04/12/96]
```

```
Cleaning KEC.
```

```
Sorting...
```

```
CLNSRT Ver 3.5.1
```

```
Copyright (C) 1996 by
```

```
Environmental Systems Research Institute
```

```
380 New York Street
```

Redlands, CA 92373
All Rights Reserved Worldwide.
Intersecting...
Assembling Polygons...
Sorting input file...
Sorting label file...
Processing...
Assigning final IDs...
Writing arc file...
Generating polygon report...
Creating PAT...
Sorting User-IDs...
Merging record 18
(D:\MYDOC~1\ARMEN\PETA)[ARC]

3.8 Editing Topologi

Editing topologi merupakan salah satu tahap untuk memperbaiki kesalahan yang dibuat ketika digitasi peta. Jika kesalahan ini tidak diperbaiki dengan benar, maka perhitungan luas, analisa data peta berikutnya tidak valid. Proses *editing* ini dilakukan di *Arcedit*. Adapun langkah-langkah *editing* topologi adalah :

(D:\MYDOC~1\ARMEN\PETA)[ARC]arcedit
[PC ARC/INFO 3.5 ARCEDIT - 04/12/96]
Serial Communications Driver - Version 5.0
COM1 (IRQ04 Level - I/O Port 3F8)

ARCEDIT Ver 3.5.1

Copyright (C) 1997 by

Environmental Systems Research Institute

380 New York Street

Redlands, CA 92373

All Rights Reserved Worldwide.

: disp 4

(C:\)[ARC] ARCEDIT <Enter>

1. Memanggil coverage yang akan diedit :

: mapex KEC

: editcov KEC

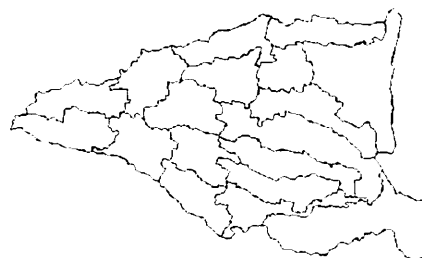
The edit coverage is now D:\MYDOC~1\ARMEN\PETA\KEC

The Map extent is not defined

Defaulting the map extent to the BND of

D:\MYDOC~1\ARMEN\PETA\KEC

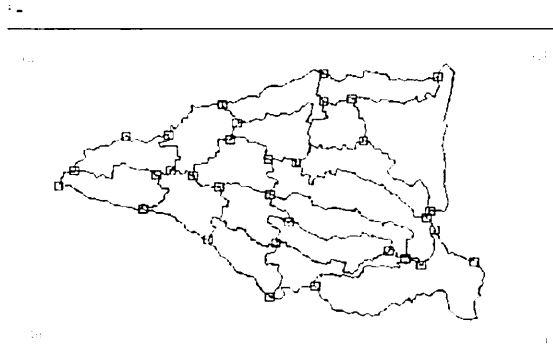
: drawn all;draw



Gambar 3.16 Tampilan Coverage Yang Akan Diedit

2. Mendeteksi dan menampilkan kesalahan pada *coverage*

: *drawen node dangle;draw* <Enter>



Gambar 3.17 Tampilan Kesalahan Pada Coverage

3. Memperbaiki kesalahan pada *coverage* :

- *Overshoot* (menghilangkan kelebihan garis)

: *ef arc* <Enter>

: *select box* <Enter>

: *delete* <Enter>

: *draw* <Enter>

- *Undershoot* (menyambungkan garis atau memindahkan *node* ke *node* lain)

: *ef node* <Enter>

: *move* <Enter>

- Memilih *node* yang akan dipindahkan lalu klik ke *node* tujuan kemudian

tekan angka 2

: *draw* <Enter>

4. Pemberian *User-ID* atau nilai *label*

: *eflabel* <Enter>

: *add* <Enter>

5. Merubah nilai *label*

: *eflabel* <Enter>

: *select* <Enter>

: *calculate* <nama cover_id> = nilai yang benar <Enter>

: *draw* <Enter>

6. Menghapus nilai *label* yang lebih dari satu

: *eflabel* <Enter>

: *select many* <Enter>

: *delete* <Enter>

: *draw* <Enter>

Setelah semua *editing* telah selesai, dilanjutkan dengan menyimpan hasil *editing* tersebut dan kemudian keluar dari *Arcedit* dengan mengetikkan **Quit** <Enter>, dan dilanjutkan dengan membuat topologi dari hasil *editing* tersebut.

[PC ARC/INFO 3.5 BUILD - 04/12/96]

Building polygons...

Sorting input file...

Leaving the ARC EDITOR ...

Serial Communications Driver - Version 5.0

***** COM1 Driver Removed *****

(D:\MYDOC~1\ARMEN\PETA)[ARC]build KEC

Sorting label file...

Processing...

Assigning final IDs...

Writing arc file...

Generating polygon report...

Creating attribute file for KEC.

Sorting User-IDs...

Merging record 18

(D:\MYDOC~1\ARMEN\PETA)[ARC]

3.9 Pembuatan Data Atribut

Untuk pemasukan data atribut dilakukan dengan perangkat lunak Microsoft Access. Data atribut ini disusun dalam bentuk tabel untuk masing-masing unsur data yang berbeda. Untuk masing-masing data atribut diberi nomor ID (identitas) yang berbeda antara satu dengan yang lainnya. Pemberian nomor ID tersebut disamakan dengan nomor label yang diberikan pada tiap data spasial dalam proses pemberian label.

3.9.1 Proses Operasi Data Atribut

Sebelum melakukan penyusunan data atribut terlebih dahulu dilakukan pemilihan dan pengelompokan data berdasarkan jenis dan macamnya, kemudian dilakukan proses penyusunan data atribut. Proses pekerjaan ini sangat penting dimana kesalahan pada tahap ini akan menyebabkan kesalahan yang lebih besar pada tahap pekerjaan selanjutnya. dan pemberian informasi tidak teratur dan akurat. Adapun langkah kerjanya adalah sebagai berikut :

1. Aktifkan perangkat lunak *Microsoft Excel* dan dari menu *File* pilih *New*.
2. Isikan tabel sesuai tujuan pembuatan data atribut seperti pada gambar berikut

Tampilan jendela untuk proses penyusunan data atribut terdapat pada gambar 3.18.

IDPerum	IDDesa	Nama Developer	Nama Perumahan
1	225	Bersama makmur abadi	Mufara Cino Groho
3	10	Delta Sejahtera Prima	Bluru Permai
4	11	Eka Surya Mutiara	Megaresri Permai
5	12	Gemilang Panca Realty	Pondok Jati
6	13	Megatopmas Nusantara	Pondok Mutiara
7	14	Sariguna Primatita	Taman Pinang Indah
8	15	Sinar gini Cemerlang	Pondok Sidokara Asri
9	16	Trisada Windu	Gading Fejir 2
10	17	Cipta Griya Sejahtera	Tanggalunan Sumokali
11	18	Awala Prima Intra Makmur	Pondok Sidokara Indah
12	19	Bintang Karyasama	Groho Kuncoro
13	2	Bhakti Pertiwi	Griya Permata Hijau
14	20	Benteng Tunggal	Sda Permai
15	21	Bersatu Sukses Sejahtera	PP Lagi Indah
16	22	Berkat Candra Mas	Delta Sari Indah
17	23	Bahagia santosa mega	Wakarya Binangun
18	24	Citra mutiara indah	Rawin Wisma Weru Indah
19	25	Ciputra Delta	Kapuh Permai
20	26	Ganda prima realty	Wisma Tropoda
21	27	Galra kanaka harum	Griya Mejan Senioso
22	28	Hijau Teduh Lestari	Pondok Candra
23	29	Hasana Damai Putera	Griya Peban Asri
24	3	Jagatsya prakarsa	Bumi Intan Permai
25	30	Kharisma indoperkasa	Griya Sedati Indah
26	31	Karya Makmur	Merpati Nusantara Airt

Gambar 3.18 Penyusunan Data Atribut

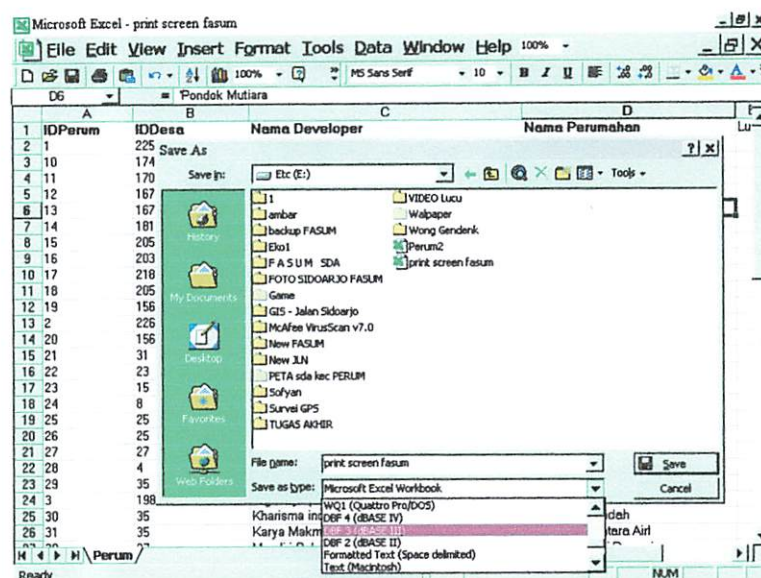
Setelah penyusunan data atribut selesai, maka dilakukan proses *editing* untuk data tersebut. Dimana hal ini dilakukan untuk data yang sudah disusun tidak terdapat kesalahan. Setelah itu dilanjutkan dengan proses *cheking* data atribut,

apabila masih ada data yang kurang maka dilakukan penyusunan data atribut kembali. Jika sudah lengkap dan benar maka dilanjutkan pada proses *export* data atribut, dimana *export* data dari *Microsoft Excel* ke *ArcView* dengan extention *.dbf. adapun langkah kerjanya sebagai berikut :

Pilih menu file lalu klik submenu Save as

Save as type pilih DBF 3 (Dbase III), kemudian klik Save

Tampilan jendela proses *export* data atribut terdapat pada gambar 3.19.



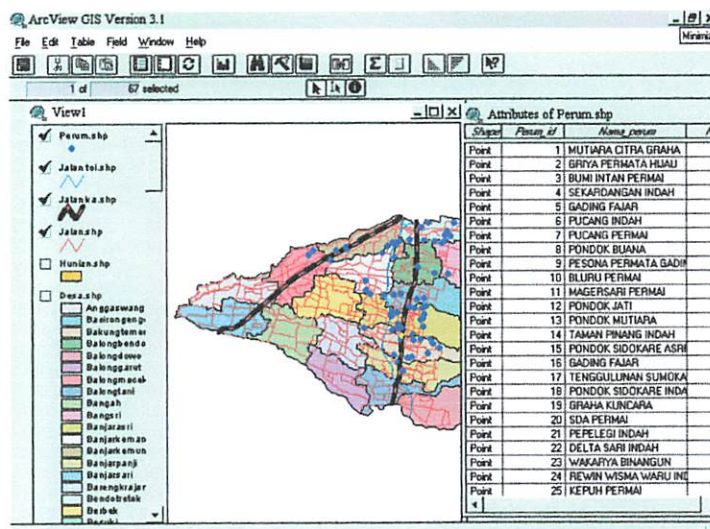
Gambar 3.19 Proses Export data atribut

3.9.2 Penggabungan Data Atribut

Penggabungan data atau *join item* adalah untuk menggabungkan data atribut (dalam *database*) dengan data spasial. Penggabungan data ini dilakukan pada *software ArcView*, yang di-*join* adalah ID dari masing-masing data, sehingga dapat dilakukan analisa berdasarkan 2 data yang telah digabungkan tersebut. Adapun cara penggabungannya adalah sebagai berikut :

1. Mengaktifkan software ArcView
2. Klik *New* pada kotak dialog *Untitled*, akan tampil *View 1*, setelah itu klik *Add Theme*
3. Memilih *coverage* yang akan ditampilkan pada kotak *View 1*, kemudian klik *OK*
4. Klik *Theme table*, maka akan tampil atribut dari *coverage*.
5. Klik *Tables* pada *Untitled*, kemudian klik *add* dan memilih *file* dari *database*.
6. Klik *ID* dari *file database*, kemudian klik *ID Attribute Of* (nama *coverage*)
7. Setelah itu klik *toolbars Join* atau memilih menu *Table* kemudian klik *Join*, untuk menggabungkan dua ID dari data tersebut.

Tampilan jendela untuk penggabungan data terdapat pada gambar 3.20.



Gambar 3.20 Penggabungan Data

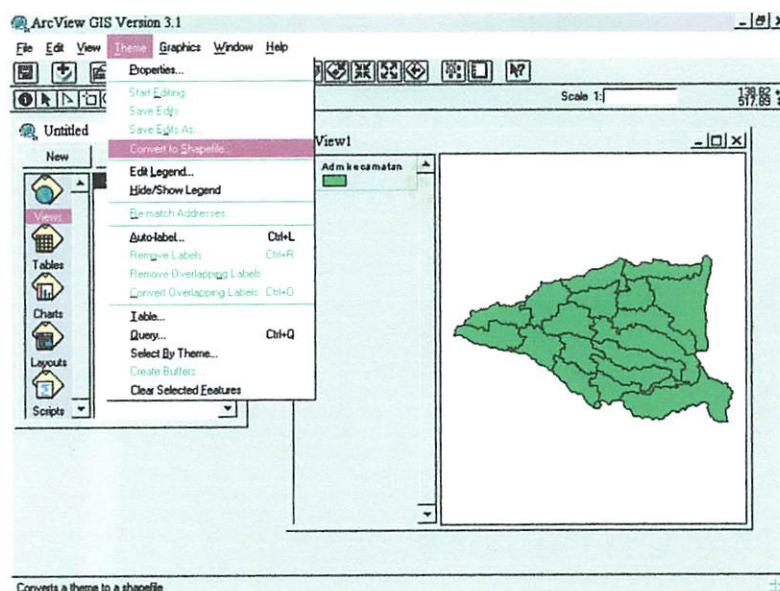
3.9.3 Convert File

Setelah data spasial dan data atribut telah selesai dilakukan penggabungan, kemudian data hasil penggabungan tersebut harus di *convert* ke dalam *ekstension* *.shp agar *file* tersebut dapat dibaca oleh program Map Objects 2.1.

Adapun langkah-langkah untuk meng-*convert file* tersebut, yaitu :

1. Mengaktifkan Arc View
2. Membuka *file* yang akan di-*convert*, contohnya : Kecamatan.apr
3. Setelah *file* dibuka, kemudian klik *theme* yang berada ada pada menu, lalu klik *Convert to Shapefile* untuk membuka formnya.
4. Isikan nama *file* pada *File Name*, lalu dimana *file* tersebut akan diletakkan dengan mengganti *directories*..

Tampilan jendela untuk *convert file* dapat dilihat pada gambar 3.21.



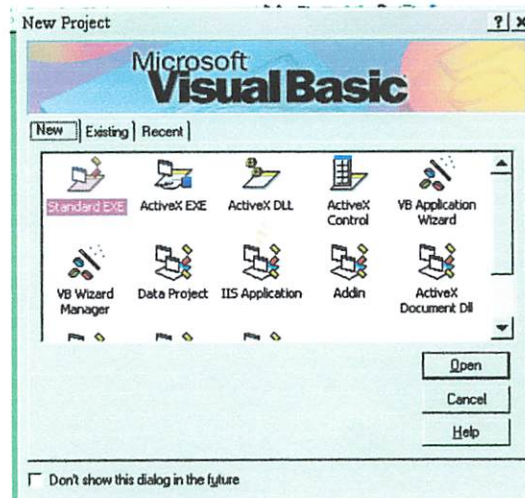
Gambar 3.21 Convert File

3.10 Desain Tampilan Informasi Usaha Kecil dan Menengah

3.10.1 Start/Enter

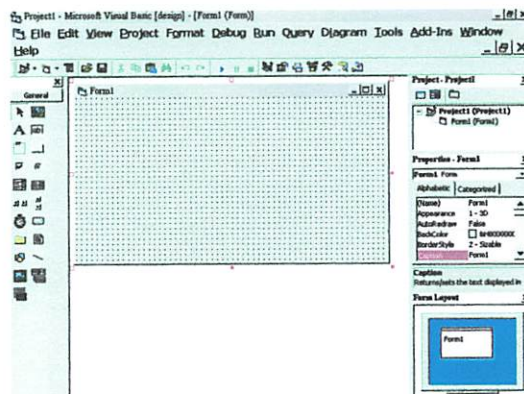
Start merupakan memulai menjalankan program Visual Basic yang merupakan bahasa pemrograman yang akan digunakan didalam penelitian ini. Adapun langkah-langkahnya adalah :

- Jalankanlah program *Visual Basic* yang telah ada pada OS (*Operating System*) yang digunakan (dalam penelitian ini peneliti menggunakan OS *Windows*), bila program *Visual Basic* belum ada, maka installah terlebih dahulu.
- Setelah program *visual basic* dijalankan, maka akan tampil kotak *dialog New Project*, kemudian pilihlah *Standart.exe* dan diakhiri dengan menekan tombol *Open*



Gambar 3.22 : Tampilan Awal Program Visual Basic

- Maka akan tampil *form* yang merupakan tempat untuk *men-desain* program yang akan dibuat.

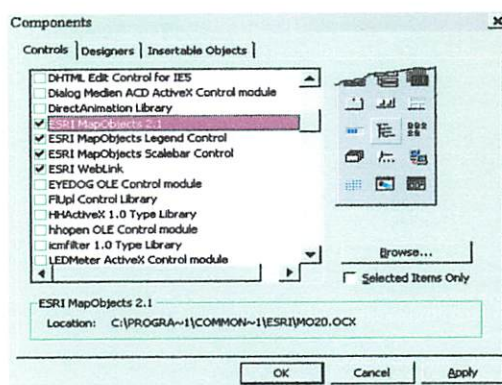


Gambar 3.23 : Tampilan di Lingkungan

Initialize Kontrol

Pada *initialize control* ini merupakan pengenalan dan pemanggilan program *visual basic* pada kontrol-kontrol yang akan digunakan dalam penelitian ini. Untuk menambahkan kontrol-kontrol dapat dilakukan dengan cara :

1. Pada menu *visual basic*, pilihlah menu *Project* kemudian klik *Components*, atau dengan menekan kontrol T (^T)
2. Maka akan tampil kotak *dialog Components*



Gambar 3.24 : Kotak Dialog Componets

3. Pilihlah dan tandai kontrol komponen yang akan digunakan, penelitian ini menggunakan kontrol-kontrol

- a. *AdvancedControl.ButtonEx*
- b. *Crystal Report Viewer Control*
- c. *Esri MapObject 2.1*
- d. *Esri MapObject 2.1 Legend Control*
- e. *Esri MapObject 2.1 Scalebar Control*
- f. *Microsoft ADO Data Control 6.0 (SP4) (OLEDB)*
- g. *Microsoft Common Dialog Control 6.0*
- h. *Microsoft Data Bound List Control 6.0*
- i. *Microsoft DataGrid Control 6.0 (SP5)(OLEDB)*
- j. *Microsoft DataList Control 6.0 (OLEDB)*
- k. *Microsoft FlexGrid Control 6.0*
- l. *Microsoft Grid Control*
- m. *Microsoft Internet Controls*
- n. *Microsoft Tabbed Dialog Control 6.0*
- o. *Microsoft Windows Common Control 5.0 (SP2)*
- p. *Microsoft Windows Common Control 6.0 (SP4)*
- q. *Microsoft Windows Common Control -3 6.0*

4. Maka akan ada penambahan kontrol pada kontrol toolbox

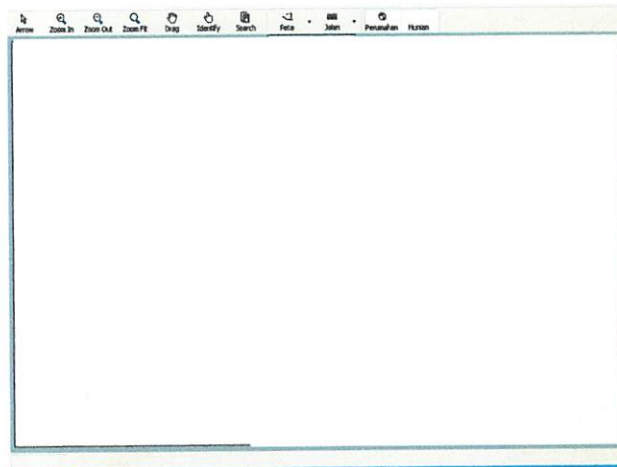


Gambar 3.25 Penambahan Kontrol Pada ToolBox

3.10.2 Display Main Form

Pada tahap *display main form* ini merupakan *desain form main* pada program yang akan dibuat. Langkah kerja yang harus dilakukan adalah :

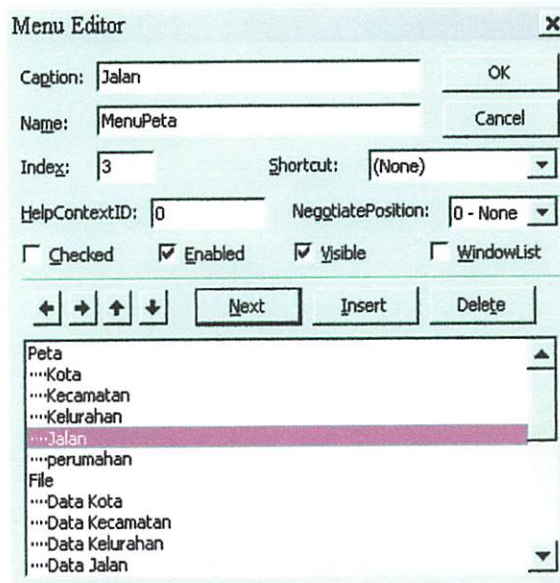
1. Buatlah *desain main form* seperti dibawah ini dengan keterangan sebagai berikut



Gambar 3.26 Desain Form Utama

a. Membuat *Menu Bar*

- 1) Kliklah menu *Tools* kemudian pilih *Menu Editor*
- 2) Pada kolom *Menu Editor*, ketik judul menu dan *submenu* sebagai berikut :

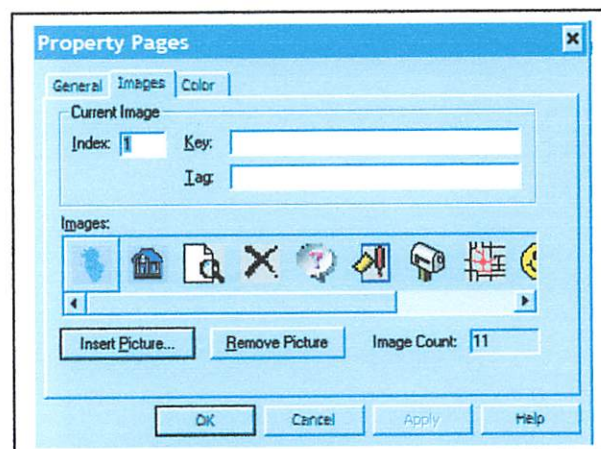


Gambar 3.27 : Kotak Dialog Menu Editor

- 3) Cara mengisi *input* Menu Editor adalah sebagai berikut
- a) Untuk menu utama isilah kolom *input* *Caption* dan Nama, misalnya *&File*
 - b) Untuk pengisian menu berikutnya, kliklah baris kosong dibawah menu yang telah terisi pada *listbox* atau klik tombol *Next*
 - c) Bila antara menu akan disisipi menu, kliklah menu kelua pada *listbox* dan kliklah tombol *Insert*
 - d) Untuk submenu :
 - Tempatkanlah *kursor* panjang pada baris kosong dibawah menu
 - Kliklah tombol →

- Isikan kolom *input Caption* dan *Name*
- e) Submenu ditandai oleh empat buah titik (...) yang akan muncul bila menekan tombol →
- f) Subsubmenu ditandai oleh tanda delapan buah titik (.....) yang akan tampil bila menekan tombol → dua kali
- g) Subsubsub ditandai oleh tanda dua belas titik (.....) yang akan tampil bila menekan tombol → tiga kali
- h) Untuk garis pemisah antara menu, ketiklah *karakter - (-)*
- i) Untuk bantuan tombol *keyboard*, misalkan Ctrl+O, isilah kolom *Shortcut*
- j) Karakter & memberi arti bahwa karakter sesudahnya ditandai garis bawah.
- k) Kliklah tombol OK bila semuanya telah selesai dimasukkan.

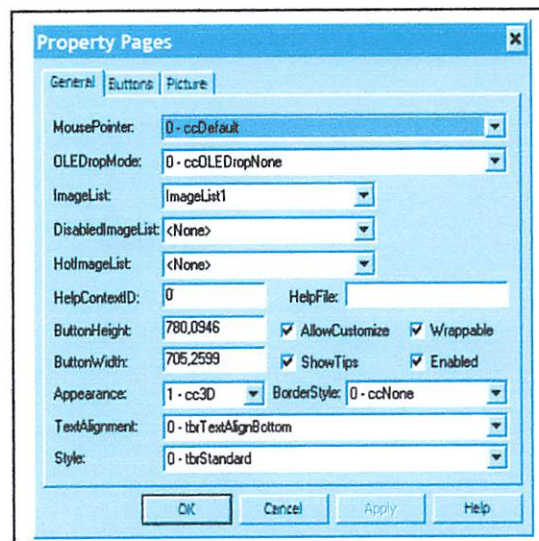
b. Membuat Tool Bar



Gambar 3.28 Property Pages

Langkah-langkahnya adalah sebagai berikut :

1. Untuk *ImageList* yang berfungsi untuk memasukkan ikon yang akan digunakan pada *toolbar* nantinya, klik kananlah pada kontrol *ImageList* dan pilihlah *Properties*, maka akan tampil kotak *dialog Property Pages*.
2. Untuk kontrol *ToolBar* yang berfungsi untuk tampilan maupun perintah tombol yang akan dipakai, tempatkan pada *form main*.
3. Aturilah sedemikian rupa sehingga *desain* tampilan *form main* nantinya bagus untuk dilihat.



Gambar 3.29 : Kotak Dialog Property Page pada Kontrol

4. Masukkan gambar ikon yang akan dibuat kedalam kontrol *ImageList*.

Cara pemasukkannya yaitu :

Untuk Tombol Pertama

Index : 1

Key : Satuan Peta

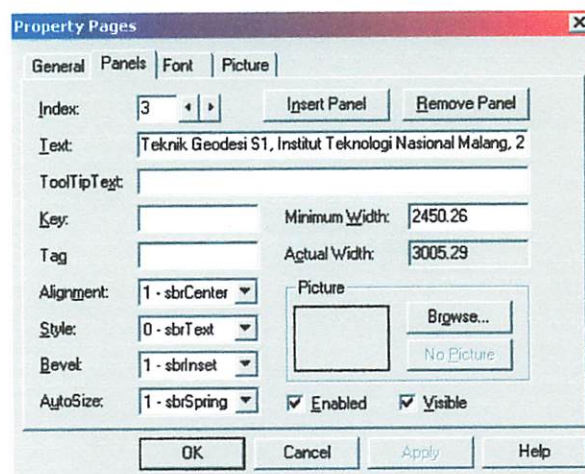
ToolTip Text : Satuan Peta

Image : 21 (pilihlah gambar ikon yang akan ditampilkan)

5. Untuk tombol-tombol yang lain juga sama cara pemasukkan perintah dan *image*-nya seperti langkah ke 4

c. Membuat *CoolBar*

Untuk membuat *CoolBar* dengan cara mengambil kontrol *CoolBar* pada *TollBox* yang ada dan diletakkan pada bagian *form main* paling bawah.



Gambar 3.30 : Kotak Dialog Property Page pada Kontrol CoolBar

Untuk memasukkan apa saja yang akan ditampilkan pada *Coolbar* sama dengan cara untuk memasukkan *icon* ke *ToolBar*.

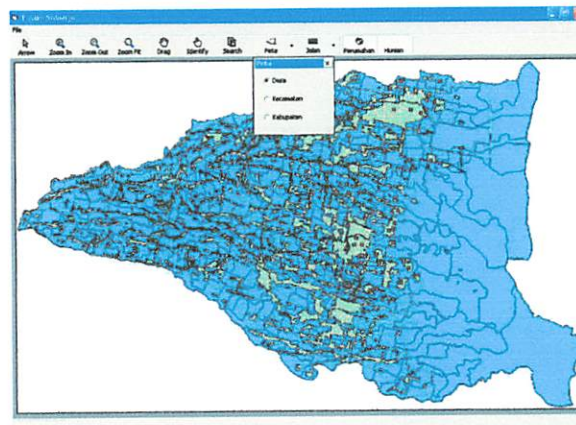
d. Kode Program yang digunakan pada penelitian ini adalah:

Terlampir pada halaman Kode Program.

3.10.3 Membuat Desain Tampilan Peta (*Map Display*)

Untuk mendesain tampilan peta diperlukan beberapa kontrol yang digunakan. Langkah-langkah dalam mendesain tampilan peta ini adalah :

1. Membuat atau menambahkan *form* baru pada *project* dengan cara mengklik kanan pada jendela *project* kemudian pilih *Add* dan klik *form*
2. Setelah *form* baru tampil, kemudian mengubah *Caption* dan *Name* pada jendela *propertis*. Untuk *Caption* dan *Name* diketikkan *frmShp*.
3. Buatlah *form* seperti pada gambar dibawah ini dengan kontrol-kontrol yang ada dan dengan keterangan tabel dibawah ini



Gambar 3.31 : Desain Form *frmShp* Untuk Menampilkan Peta

4. Masukkanlah isi *properti* kontrol pada jendela *properties* sesuai dengan langkah kerja diatas.

5. Sedangkan kode program yang digunakan pada frmShp adalah :

sesuai terlampir pada halaman kode program

3.10.4 Mendesain Program Untuk Pencarian (*Search*)

Untuk mendesain program yang berfungsi untuk pencarian suatu obyek (desa ,kecamatan ,jalan ,fasilitas umum dan perumahan). Langkah-langkah untuk mendesain program pencarian ini adalah

1. Membuat *form* baru dan *desain*-lah *form* tersebut seperti pada gambar berikut ini, dengan ketentuan-ketentuan seperti pada tabel 3.3

ID Desa	Nama Desa	Kecamatan
001	Segrotambak	Sedati
002	Pranti	TAMAN
003	Bebek	WARU
004	Wadungasii	WARU
005	Bunguresih	WARU
006	Tambaksumur	WARU
007	Semampir	TAMAN
008	Janti	WARU
009	Tambaksawah	TAMAN
010	Wedoro	WARU
011	Tambakoso	WARU
012	Kepukkiiman	WARU
013	Tambakemandi	TAMAN
014	Gitikemandi	TAMAN
015	Kedungrejo	WARU
016	Kreimbangan	TAMAN
017	Buncilan	TAMAN
018	Pukungan	TAMAN
019	Tambakrejo	WARU
020	Kalanganyor	TAMAN
021	Medaeng	WARU
022	Medaeng	WARU

Gambar 3.32 : Desain Form Pencarian

2. Kode program yang diberikan adalah:

Sesuai dengan yang terlampir pada halaman kode program

3.10.5 Desain Program Untuk Penggambaran dan Update Data

Desain program untuk penggambaran dan *update* data, merupakan bagian dari menu gambar grafik maupun tombol perintah gambar grafik yang terdapat pada *form main*. Sedangkan untuk penambahan data yang berbentuk suatu area dipisahkan, karena data *input* yang digunakan berdasarkan pada koordinat (penggambaran menggunakan data koordinat). Adapun langkah-langkah pembuatan *form update* data adalah :

1. Buatlah form baru dan letakkan kontrol-kontrol pada form serta isikan nilai propertinya seperti pada tabel 3.5
2. Membuat tampilan site pline yang terdiri dari form-form yang dapat diupdate .

Gambar 3.33 : Desain Form updating Data site pline perumahan

3. Setelah ditampilkan site pline, area tiap faslitas umum pada satu perumahan dapat ditampilkan informasinya.

The screenshot shows a software window titled "Fasilitas Umum" with a blue header. On the left, there is a photo of a church building. Below the photo is a text box for the file location: "F:\Projects\GIS - Fasilitas Sidoarjo\Images\01-20-02.jpeg", and two buttons: "Ganti Foto" and "Hapus Foto". On the right, there is a form titled "Informasi Fasilitas" with the following fields:

ID Fasilitas	F-20-02
Nama Fasilitas	GEREJA
Jenis	TEMPAT IBADAH
Pemanfaatan	BERBADA41
Luas	600 M2
Pos Kamling	<input checked="" type="checkbox"/> Ada / <input type="checkbox"/> Tidak Ada
Batas Barat	BLOK J
Batas Timur	BLOK G
Batas Utara	TAMAN
Batas Selatan	POS KAMLING

At the bottom right of the form are four icons: "Cari" (magnifying glass), "Posisi" (globe), "Simpan" (floppy disk), and "Tutup" (close button).

Gambar 3.34 : Desain Form updating Data fasilitas umum

4. Kode yang diberikan pada program ini adalah “

Sesuai dengan yang terlampir pada halaman kode program

3.10.6 Pembuatan Program Bantuan (*Help*)

Help merupakan bantuan atau panduan bagi para pengguna untuk dapat menjalankan atau mengoperasikan program ini dengan baik dan lancar. *Form help* ini dibuat dengan bantuan Microsoft Front Page 2000, untuk kemudian dimasukkan kedalam *form help*.

Form help ini dapat dijalankan dengan menekan tombol *help* yang berada di dalam program. Tampilan jendela proses pembuatan *form help* pada Microsoft Front Page 2000.

BAB IV

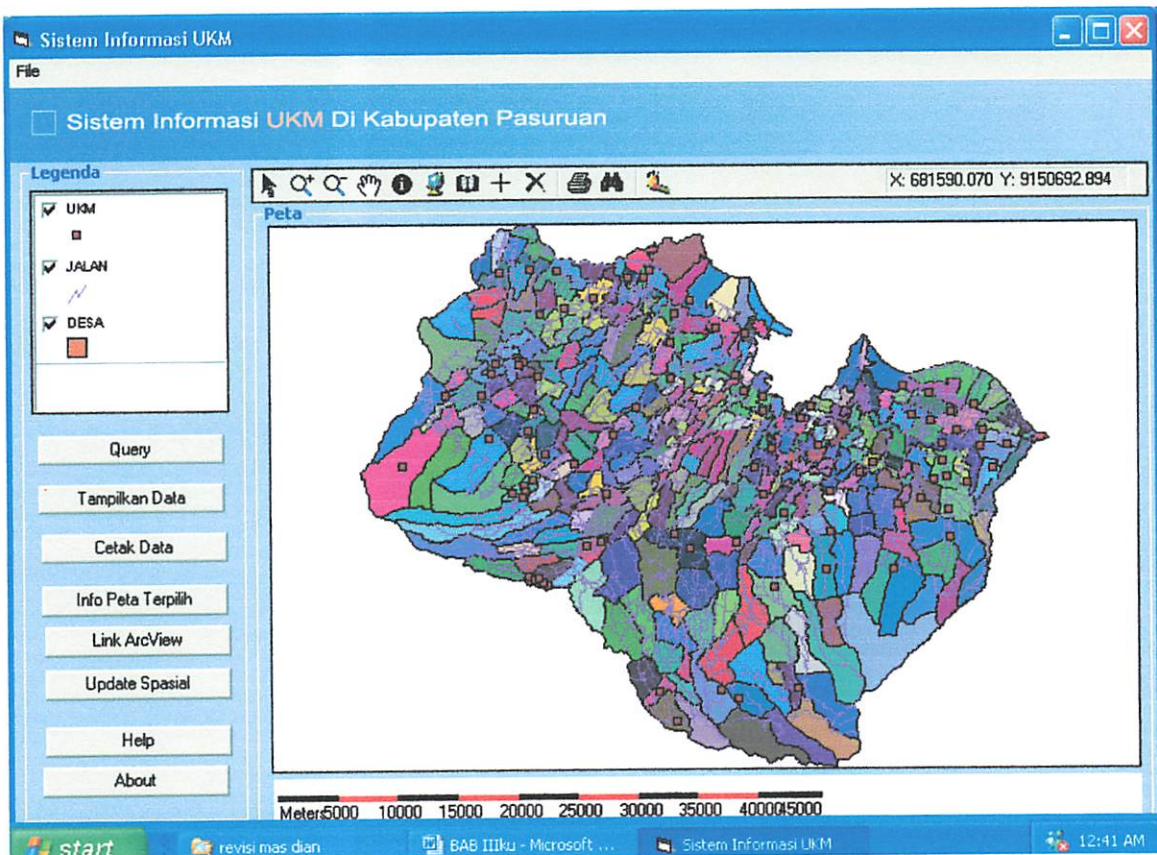
HASIL DAN PEMBAHASAN

Hasil yang diperoleh dari penelitian ini adalah tampilan data spasial dan data non spasial informasi Usaha Kecil dan Menengah di Kabupaten Pasuruan.

4.1. Hasil Penelitian

4.1.1. Visualisasi Informasi Dengan Visual Basic

Form utama untuk aplikasi ini berupa toolbar untuk peta, menu untuk mengetahui informasi dari obyek *toolbar* untuk memilih obyek yang diinginkan, pencarian, tabel informasi nama Usaha Kecil dan Menengah dan menu Tutup.

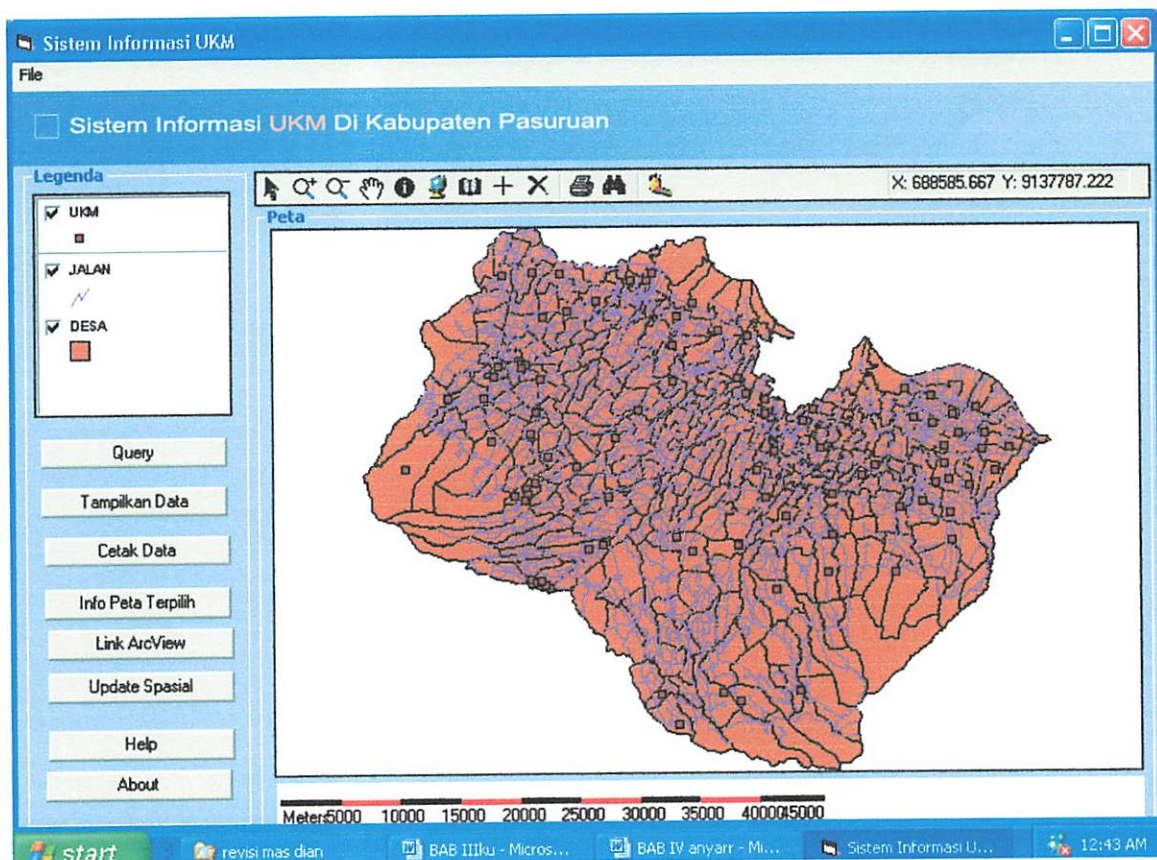


Gambar 4.1 Desain Form Utama Informasi Usaha Kecil dan Menengah

4.2. Pembahasan Hasil

4.2.1. Penyajian Peta

Pada aplikasi penyajian peta, data yang dapat ditampilkan adalah peta Kabupten, peta Kecamatan, peta Desa, peta Jalan dan peta Lokasi Usaha Kecil dan Menengah.



Gambar 4.2 Form Utama Usaha Kecil dan Menengah

4.2.2. Penyajian Aplikasi Data

Aplikasi data ini meliputi data kecamatan, data kelurahan, data jalan, data informasi Usaha Kecil dan Menengah. Pada aplikasi ini terdapat menu update data, informasi foto serta dilengkapi dengan fasilitas penyimpanan data.

Informasi	
Profil UKM	
Nama Pemilik	Sutikno
Perusahaan	UD. Sinar Kayu Mas
Alamat	Desa Karangrejo Purwosari Telp.(0343)612838
Jenis Usaha	Pengolahan Kayu
Total Produksi/Bln	Rp. 22190000
Total Omze/Thn	Rp. 296000000
Total Asset/Thn	Rp. 32000000
Total Tenaga Kerja	8 Orang
Pemasaran	Domestik
Desa	Karangrejo
Kecamatan	Purwosari
Keterangan :	Data Diambil Tahun 2004

Buttons: Lihat Foto, Edit Data, Close

Gambar 4.3a Desain Form informasi Usaha Kecil dan Menengah



Gambar 4.3b Desain Form informasi foto Usaha Kecil dan Menengah

4.2.3. Pelaksanaan Updating data

Proses Updating pada program ini dilakukan pada data non spasial dan Data Spasial, dapat dijelaskan :

4.2.3.1. Updating data Non Spasial :

Proses Updating data Non Spasial, yaitu :

1. Cari nama Perusahaan atau UKM yang akan dilakukan *updating* dengan mencari pada menu *cari*, atau masuk ke menu *Informasi* , Klik icon titik atau posisi yang akan di *update* kemudian klik *edit* untuk masuk ke dalam database Usaha Kecil dan Menengah.
2. Kemudian dilakukan *updating* sesuai dengan data informasi Usaha Kecil dan Menengah yang baru.
3. Setelah selesai melakukan *updating*, baik itu pengeditan maupun penambahan data baru, maka dilakukan penyimpanan data dengan menekan tombol *simpan*. Dan apabila data foto perlu di *update* juga dapat dilakukan dengan mengklik *edit foto*, kemudian cari *file* foto terbaru kemudian klik *simpan*.
4. Apabila *updating* data Usaha Kecil dan Menengah telah selesai dilakukan maka dapat ditutup dengan menekan tombol *Close*.

Profil UKM	
No UKM	18
Nama Pemilik	Pudjo Sakti
Perusahaan	UD. Dewi Sri
Alamat	Dusun Glatik Rt.3/Rw.VI GlagahSari Sukorejo Te
Jenis Usaha	Wig dan Sanggul
Total Produksi/Bln	Rp. 72065000
Total Omze/Thn	Rp. 964300000
Total Asset/Thn	Rp. 180000000
Total Tenaga Kerja	170 Orang
Pemasaran	Domestik
Desa	Glagahsari
Keterangan :	Data Diambil Tahun 2002

Buttons: Edit Foto, Simpan, Close

Gambar 4.4 Form Editi Data UKM

4.2.3.2. Updating data Spasial :

Proses Updating data Spasial yaitu :

A. Cara I

1. Memasukkan data spasial yang baru ke dalam autocad untuk mengubah data analog menjadi data digital dengan cara mendigitasi peta.
2. Pembuatan topologi dengan ArcInfo.
3. Setelah itu data di convert ke dalam ekstension *.shp agar dapat dibaca oleh Map Object.

B. Cara II

1. Klik tombol *Tambah Spatial* pada form utama program.

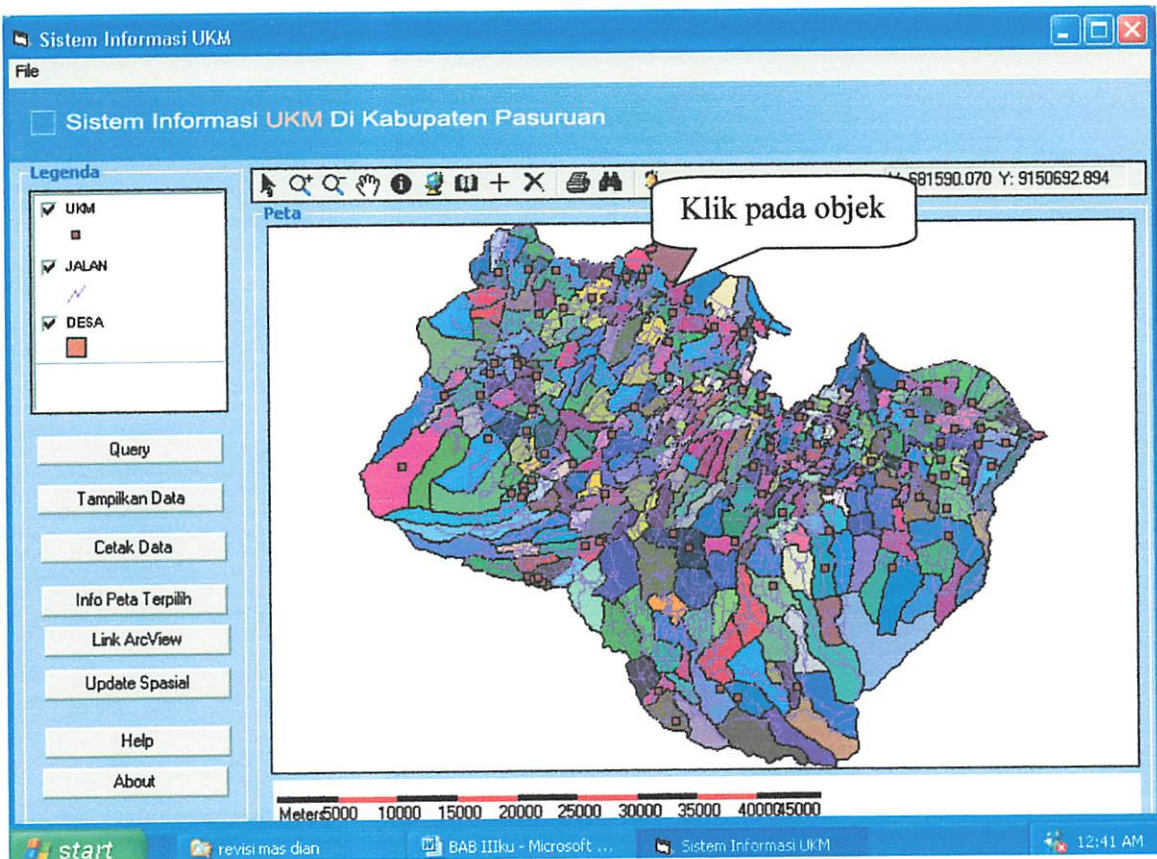
2. Klik pada posisi koordinat peta yang diinginkan.
3. Untuk mengisi data dari Usaha Kecil dan Menengah yang baru maka, masuk ke menu *Informasi* , Klik icon titik atau posisi yang baru kemudian klik *edit* untuk masuk ke dalam database Usaha Kecil dan Menengah.
4. Kemudian dilakukan *updating* sesuai dengan data informasi Usaha Kecil dan Menengah yang baru.
5. Setelah selesai melakukan *updating*, baik itu pengeditan maupun penambahan data baru, maka dilakukan penyimpanan data dengan menekan tombol *simpan*. Dan data foto dimasukkan dengan cara mengklik *edit foto*, kemudian cari *file* foto terbaru kemudian klik *simpan*.
6. Apabila *updating* data Usaha Kecil dan Menengah telah selesai dilakukan maka dapat ditutup dengan menekan tombol *Close*.

4.2.4. Penyajian Aplikasi Pencarian

4.2.4.1. Pencarian dengan mengklik objek pada peta

Untuk memudahkan pengguna dalam mencari lokasi atau posisi Bank Perkreditan Rakyat yang diinginkan, maka pengguna cukup melakukan langkah-langkah seperti tersebut di bawah ini :

1. Mengaktifkan menu utama, kemudian mengklik *toolbar Informasi* yang terdapat pada toolbar yang berada dibawah.
2. Kemudian klik posisi atau lokasi yang diinginkan, maka akan muncul informasi mengenai UKM tersebut.



Hasil dari kegiatan diatas:

Informasi	
Profil UKM	
Nama Pemilik	Sutikno
Perusahaan	UD. Sinar Kayu Mas
Alamat	Desa Karangrejo Purwosari Telp.(0343)612838
Jenis Usaha	Pengolahan Kayu
Total Produksi/Bin	Rp. 22190000
Total Omze/Thn	Rp. 296000000
Total Asset/Thn	Rp. 32000000
Total Tenaga Kerja	8 Orang
Pemasaran	Domestik
Desa	Karangrejo
Kecamatan	Purwosari
Keterangan :	Data Diambil Tahun 2004



4.2.4.2. Pencarian dengan mengklik toolbar Cari

1. Mengklik *toolbar Cari* yang terdapat di bawah menu utama.
2. Setelah itu akan muncul menu pencarian sesuai dengan apa yang diinginkan.
3. Selanjutnya klik apa yang dicari.
4. Kemudian tekan *OK*, maka item yang kita cari akan didapat dengan berkedipnya posisi pada tampilan utama.

Search Criteria	Search Results
TOTALKERJA	'UD. Ecco Prayogo'
NO	'CV. Wonokaya Sejati'
TOTALOMZE	'CV. Dwi Karya'
PERUSAHAAN	'UD. Yulita'
KET	'UD. Sumber Rejeki'
PEMILIK	'UD. Arjasa'
JENIS	'UD. Sinar Kayu Mas'
TOTALASET	'Percetakan dan Sabl'
ALAMAT	'UD. Setia Kawan'
PEMASARAN	'UD. Dewi Sri'
IDDESA	'UD. Al-Veles'
TOTALPROD	'Foxi 73 Seni Bordir'

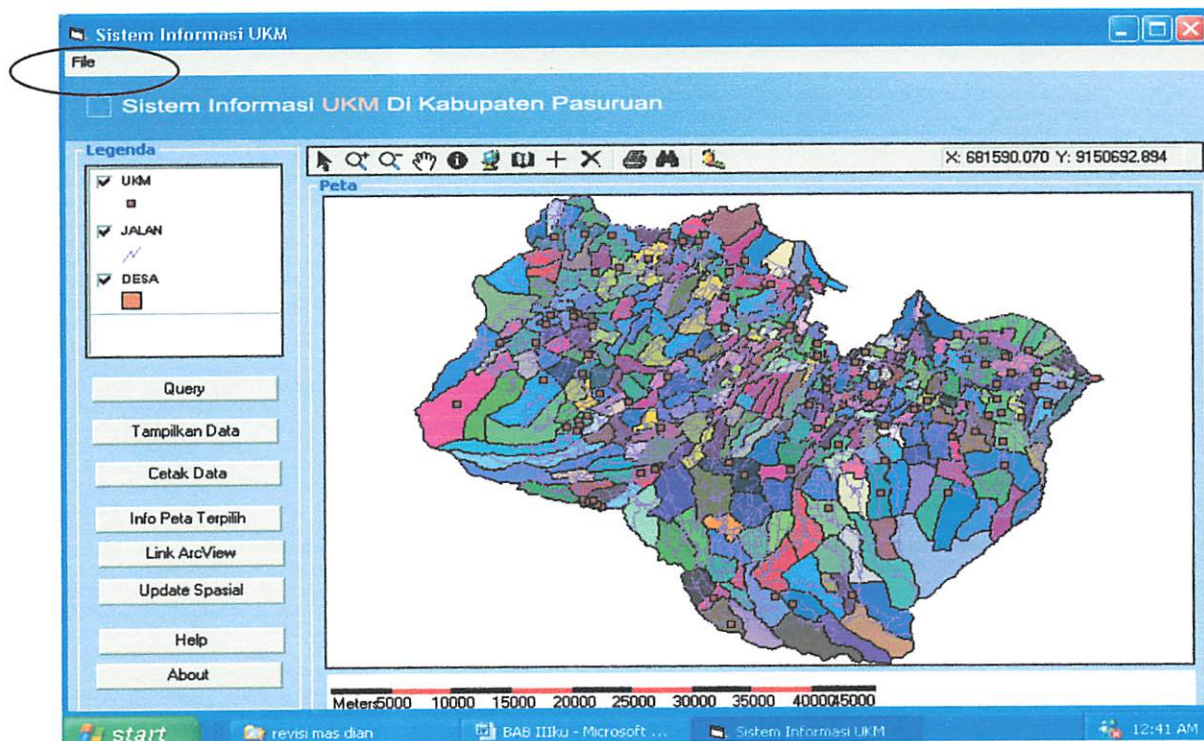
Search Field: PERUSAHAAN='UD. Yulita'

Buttons: Keterangan, Ok

Gambar 4.5 Form Cari Usaha Kecil dan Menengah

4.2.5. Menambah atau menghapus layer

1. Mengklik menu *File* yang terletak pada pojok kiri atas menu utama.
2. Maka akan muncul menu *Add Themes* yang artinya menambah *layer* dan *Delete Themes* yang artinya menghapus *layer*.
3. Kemudian apabila ingin menambah *layer* klik menu *Add Themes* maka *layer* yang akan kita tambahkan dapat ditampilkan.
4. Begitu juga apabila kita ingin menghapus *layer* juga dapat dilakukan dengan mengklik menu *Delete Themes* maka kita dapat menghapus *layer* yang ada pada legenda peta.

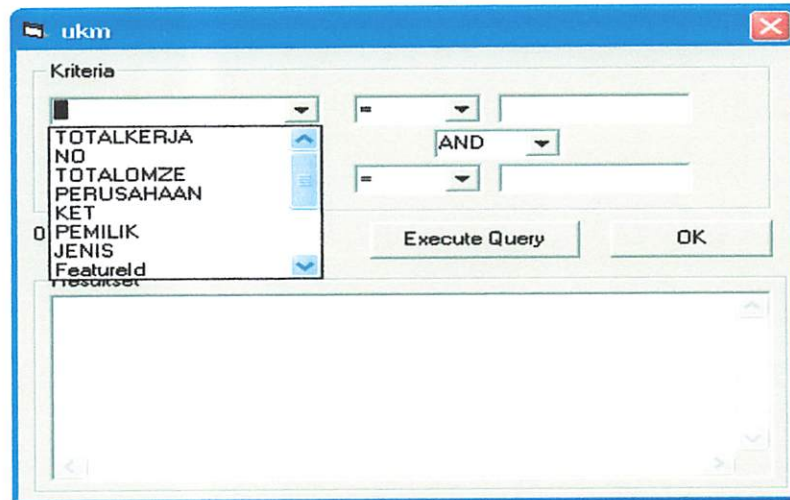


Gambar 4.6 Menu File

4.2.6. Mencari Query

1. Mengklik menu *File* yang terletak pada pojok kiri atas menu utama.
2. Maka akan muncul menu *Query*.

3. Kemudian mengklik menu *Query*, selanjutnya kita dapat mencari *query* sesuai dengan yang kita inginkan.



Gambar 4.7 Menu Query

4.3. Kendala Yang Timbul dan Cara Mengatasi Kendala Dalam Pembuatan Program Dengan Menggunakan Visual basic

4.3.1. Kendala yang timbul pada saat pembuatan program antara lain :

1. 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*.
2. Pada tahapan pembuatan penyajian informasi posisi Bank Perkreditan Rakyat yang digabungkan dengan *basisdata (atribut)* yang dibuat oleh *software* yang berbeda.
3. Banyaknya bahasa pemrograman yang digunakan.
4. Membutuhkan kemampuan memainkan logika yang tinggi.

4.3.2. Mengatasi Kendala-kendala Dalam Pembuatan Program Dengan Menggunakan Visual Basic

Dalam mengatasi kendala-kendala dalam pembuatan program dengan Visual Basic, ada beberapa hal yang perlu diperhatikan, yaitu :

1. Seorang programmer harus sabar dan teliti di dalam penulisan suatu program.
2. *Desain form* banyak diilhami dari program-program yang sudah ada dan saran dari *user* atau pengguna.
3. 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.
4. Bila terjadinya kesalahan pada penulisan program, biasanya visual basic akan mengenalinya dengan menunjukkannya suatu pesan ataupun arah cursor.
5. Diharapkan seorang programmer memiliki pengetahuan maupun pustaka yang lebih banyak tentang penulisan kode program.
6. Tip dan trik juga dirasakan sangat perlu di dalam penulisan program.

4.4. Keuntungan Pembuatan Program Dengan Visual Basic dan Map Object

Suatu program dikatakan baik apabila program tersebut dapat menyelesaikan suatu masalah dari permasalahan yang ada tanpa meninggalkan keindahan pada tampilannya (*interface*).

Oleh sebab itu banyak manfaat dan keuntungan yang didapat pada pembuatan program dengan menggunakan Visual Basic, yaitu:

1. Dapat membuat suatu program yang digunakan untuk membantu suatu pekerjaan-pekerjaan perencanaan, pencarian suatu obyek, penyimpanan data, manipulasi data, pemanggilan data maupun analisa data.

2. Visual Basic dapat menghasilkan program-program berorientasi obyek.
3. Dapat membuat suatu program pada bidang-bidang keilmuan, seperti bidang kegeodesian.
4. Perintah-perintah yang digunakan untuk menjalankan program yang dihasilkan oleh Visual Basic dapat diganti dengan bahasa Indonesia (menghasilkan program yang berbasis bahasa Indonesia), sehingga pemakai dapat dengan mudah mengerti dan menjalankan program tersebut.
5. Visual Basic merupakan bahasa pemrograman yang familiar dengan program-program yang ada, terutama dengan program-program yang berhubungan dengan kegeodesian, seperti ArcView, ArcObject, ArcMap, AutoCAD, MapInfo, MapObject dan lain sebagainya.
6. Karena Visual Basic merupakan bahasa pemrograman yang familiar, banyak perusahaan-perusahaan pembuat program yang memberikan fasilitas pengembangan (*Open Source Code*) untuk para pengembang program.
7. Fasilitas-fasilitas kontrol pada Visual Basic yang digunakan dalam pembuatan program cukup banyak.
8. MapObject merupakan suatu program berbeasis GIS yang memberikan fasilitas bagi para pengembang (*Open Source Code*), yang dapat digunakan maupun dikembangkan oleh pengembang dengan menggunakan bahasa pemrograman Visual Basic.

4.5. Analisa Hasil

Dari hasil pembuatan program sistem informasi Usaha Kecil dan Menengah di Kabupaten Pasuruan maka dapat diketahui penyebaran atau posisi dari Usaha Kecil dan

Menengah di Kabupaten Pasuruan. Lokasi Usaha Kecil dan Menengah menyebar di seluruh di Kabupaten Pasuruan. Sehingga hal ini dapat memajukan perekonomian di semua wilayah kecamatan yang ada di Kabupaten Pasuruan, dan dapat menarik investor untuk berinvestasi dengan bekerjasama dengan para pengusaha UKM di Kabupaten Pasuruan.

Adapun data mengenai lokasi dan informasi Usaha Kecil dan Menengah di Kabupaten Pasuruan yang dapat diketahui dari pembuatan program ini yaitu:

1. UKM dengan total produksi tertinggi

- UD. Dua Gunung Mas
- Pemilik H. Mustopo
- Alamat: Desa Randupitu Gempol

2. UKM dengan total produksi terendah

- UD. Sabar
- Pemilik: Markum
- Alamat: Rt4/I Desa Dayu Rejo Prigen

3. UKM dengan total omzet tertinggi

- UD. Dua Gunung Mas
- Pemilik H. Mustopo

- **Alamat: Desa Randupitu Gempol**

4. UKM dengan total omzet terendah

- **UD. Ecco Prayogo**
- **Pemilik: Fakhri Syahroni**
- **Jl. Kedondong RT3/III Sumber Gedang Padaan**

5. UKM dengan total asset tertinggi

- **UD. Dua Gunung Mas**
- **Pemilik H. Mustopo**
- **Alamat: Desa Randupitu Gempol**

6. UKM dengan total asset terendah

- **Keramik Telaga Rejeki**
- **Pemilik: Hasan Basori**
- **Alamat: Desa Kertosari Purwosari**

7. UKM dengan total tenaga kerja tertinggi

- **UD. Dewi Sri**
- **Pemilik: Pudjo Sakti**

- **Alamat: Dusun Glatik RT3/VI Desa Glagahsari Purworejo**

8. UKM dengan total tenaga kerja terendah

- **UD. Warung dowo Jaya**
- **Pemilik: Siti Sarofah**
- **Alamat: Desa Warung Dowo Pohjentrek**

BAB V

PENUTUP

5.1 Kesimpulan

Dari pelaksanaan tugas akhir ini terdapat beberapa kesimpulan yaitu :

1. Dengan pembuatan program penyajian informasi Usaha Kecil dan Menengah ini maka dapat digunakan untuk mencari data atribut Usaha Kecil dan Menengah dan mengetahui posisinya.
2. Dari hasil penelitian ini dapat diketahui bahwa jumlah Usaha Kecil dan Menengah yang tersebar di 24 kecamatan di Kabupaten Pasuruan adalah 90 Usaha Kecil dan Menengah.
3. Usaha Kecil dan Menengah sebagian besar terdapat di Kecamatan Purwosari dengan 8 UKM.
4. Pada program informasi Usaha Kecil dan Menengah ini mempunyai keunggulan diantaranya yaitu:
 - a. Program ini dibuat sesederhana mungkin sehingga para pengguna (*users*) dapat dengan mudah melakukan pencarian informasi mengenai Usaha Kecil dan Menengah yang diinginkan.
 - b. Dalam Program ini, informasi dan objek yang diinginkan ditampilkan dalam satu form sehingga pengguna dapat langsung mengetahui informasi dan posisi dari Usaha Kecil dan Menengah yang diinginkan.
 - c. Pengguna dapat melakukan pencarian informasi dari Usaha Kecil dan Menengah yang diinginkan dengan mengklik objek Usaha Kecil dan

Menengah yang ada pada peta maupun mengklik nama Usaha Kecil dan Menengah yang diinginkan yang ada di Tombol pencarian dan selanjutnya peta akan menyala dan secara otomatis akan keluar form informasi dari Usaha Kecil dan Menengah tersebut.

5.2. Saran

Program yang dihasilkan dalam penelitian ini masih jauh dari kesempurnaan namun bisa dijadikan sebagai titik awal dari (*user*) pengguna komputer untuk membantu proses pencarian informasi data yang nantinya menjadi dasar dalam pengambilan keputusan untuk mendapatkan informasi Usaha Kecil dan Menengah. Untuk membuat suatu Program yang lebih maksimal hendaknya ada pengembangan lebih lanjut dari pada program ini yang dilengkapi dengan sistem informasi geografi, sehingga (*user*) pengguna dapat mengetahui informasi-informasi data spasial dan non spasial yang berhubungan dengan Usaha Kecil dan Menengah di Kabupaten Pasuruan.

DAFTAR PUSTAKA

- Ade. 2000. Basis Data. PT. Jaya Utama, Jakarta**
- Deni Charter, Irma Agtisari. 2001. Desain dan Aplikasi SIG. PT Budi Jaya, Jogjakarta**
- E, Prahasta. 2001. Sistem Informasi Geografi. PT. Gaya Baru, Jogjakarta**
- Korth. 1991. Data Base. London Press, London**
- M. Agus J. Alam. 2001. Desain Basis Data. PT. Elex Media Komputindo, Jakarta**
- Pantimena, Leo. 1997. Diktat Sistem Informasi Geografi. ITN, Malang**
- Supena.2001.UKM dalam Modernisasi. PT. Tiga Karya , Bandung**

LAMPIRAN

```

Dim GisUkm As New Setyo
Dim IdK As String
Public Indeks As Integer
Dim g_grid As New SnappingGrid
Dim g_editLayer As New EditLayer
Dim addStatus As Boolean
Dim delStatus As Boolean
' dragging variables
Dim g_dragger As DragFeedback
Public RecVirtual As MapObjects2.Recordset
Public RecTerpilih As MapObjects2.Recordset
Public Recs As MapObjects2.Recordset
Dim WithEvents Con As ADODB.Connection
Dim WithEvents rst As ADODB.Recordset
Dim cmd As ADODB.Command
Dim p As MapObjects2.Point
Dim pts As MapObjects2.Points
Public Sub connect()
Set Con = New ADODB.Connection
Con.CursorLocation = adUseClient
Con.Provider = "microsoft.jet.oledb.4.0"
Con.Open "Data Source=" & App.Path & "\database\ukmdb.mdb"
DataEnvironment1.Connection1.Open Con
End Sub

```

```
Private Sub cmdTitle_Click()
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
FormAktif = 2
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim td As TEXT_DESC
```

```
Dim lf As LOGFONT
```

```
'Mengatur Tombol
```

```
abmContainerSubclass Me.hWnd
```

```
abmButtonSubclassEx Me.hWnd, cmdTitle.hWnd, BS_PUSHBUTTON
```

```
abmContainerSetting Me.hWnd, SET_SPECIAL_EFFECT, 3
```

```
abmContainerSetting Me.hWnd, SET_SPECIAL_EFFECT_FACTOR, 2
```

```
abmContainerSetting Me.hWnd, SET_SHAPE, 0
```

```
abmContainerSetting Me.hWnd, SET_SURFACE, 5
```

```
abmContainerSetting Me.hWnd, SET_COLOR_BACKGROUND, &HFFFFFF '&H808080
```

```
'&HFFC0C0
```

```
abmContainerSetting Me.hWnd, SET_COLOR_MOUSEOVER, &HFFFFFF '&HC0E0FF
```

```
'RGB(192, 192, 192)
```

```
' get the current settings from the Exit button
```

```
abmGetTextSettingOnPos cmdTitle.hWnd, TEXTPOS_CAPTION, td
```

```
abmGetFontText cmdTitle.hWnd, TEXTPOS_CAPTION, lf
```

```
td.colorNormal = &H800000
```



```

td.colorNormal = &H800000

abmGetFontExt cmdTitle.Wnd, TEXTPOS_CAPTION, ff
abmGetTextSettingOnPos cmdTitle.Wnd, TEXTPOS_CAPTION, ff
' get the current settings from the Exit button

RGB(192, 192, 192)
abmContainerSetting Me.Wnd, SET_COLOR_MOUSEOVER, &HFFFFFF, &HC00FF
&HFFC00
abmContainerSetting Me.Wnd, SET_COLOR_BACKGROUND, &HFFFFFF, &H808080
abmContainerSetting Me.Wnd, SET_SURFACE, 5
abmContainerSetting Me.Wnd, SET_SHAPE, 0
abmContainerSetting Me.Wnd, SET_SPECIAL_EFFECT_FACTOR, 2
abmContainerSetting Me.Wnd, SET_SPECIAL_EFFECT, 3
abmButtonSubclass Me.Wnd, cmdTitle.Wnd, BS_PUSHBUTTON
'Mengatur Tombol
Dim ff As LOGFONT
Dim fd As TEXT_DESC
Private Sub Form_Load()
FormAktif = 2
End Sub
Private Sub Form_Activate()
End Sub
Private Sub cmdTitle_Click()
End Sub

DataEnvironment.Connection.Open Con
Con.Open "Data Source=" & App.Path & "/database/ukmdb.mdb"
Con.Provider = "microsoft.jet.oledb.4.0"
Con.CursorLocation = adUseClient
Set Con = New ADODB.Connection
Public Sub connect()
Dim pts As MapObjects2.Points
Dim p As MapObjects2.Point
Dim cmd As ADODB.Command
Dim WithEvents rst As ADODB.Recordset
Dim WithEvents Con As ADODB.Connection
Public Recs As MapObjects2.Recordset
Public Rectripth As MapObjects2.Recordset
Public RectVirtual As MapObjects2.Recordset
Dim g_dragger As DragFeedback
' dragging variables
Dim delstatus As Boolean
Dim addstatus As Boolean
Dim g_editlayer As New EditLayer
Dim g_grid As New snappingGrid
Public Indeks As Integer
Dim Idk As String
Dim Gisukm As New Setyo

```

```
td.iEffect = EFFECT_RAISED
td.nTransparentFactor = 50
abmContainerTextSetting Me.hWnd, TEXTPOS_CAPTION, td, 0
```

```
'Mengatur huruf pada form utama
lf.lfHeight = 12
lf.lfFaceName = "Times New Roman"
abmContainerFontSetting Me.hWnd, TEXTPOS_CAPTION, lf
```

```
'memasukan data
Set GisUkm.Peta = Map1
FileData = peth & "Map/"
Set dc = GisUkm.Koneksi(FileData)
```

```
GisUkm.AddLayer "desa"
GisUkm.AddLayer "jalan"
GisUkm.AddLayer "ukm"
```

```
Indeks = 0
legMapDisp.setMapSource Map1
legMapDisp.LoadLegend True
legMapDisp.ShowAllLegend
```

```
Set g_ActiveLayer = Map1.Layers(0)
Call AturLayer
legMapDisp.Active(Indeks) = True
```

```
sbScaleBar1.MapUnits = muMeters
sbScaleBar1.ScaleBarUnits = suMeters
Call RefreshScale(sbScaleBar1, Map1)
Map1.Extent = Map1.FullExtent
Map1.MousePointer = moArrow
addStatus = False
delStatus = False
```

```
End Sub
```

```
Private Sub cmdKeluar_Click()
```

```
Unload Me
```

```
End Sub
```

```
Public Sub RefreshScale(AEScaleBar As ScaleBar.sbScaleBar, MOMap As Map)
```

```
Dim MapExt As sbExtent
Dim Pageext As sbExtent
On Error Resume Next
Set MapExt = AEScaleBar.MapExtent
Set Pageext = AEScaleBar.PageExtent
```

```
MapExt.MinX = MOMap.Extent.Left
MapExt.MinY = MOMap.Extent.Bottom
MapExt.MaxX = MOMap.Extent.Right
MapExt.MaxY = MOMap.Extent.Top
```

```
MapExt.MaxY = MOMap.Extent.Top
MapExt.MaxX = MOMap.Extent.Right
MapExt.MinY = MOMap.Extent.Bottom
MapExt.MinX = MOMap.Extent.Left

Set PageExt = AEscalabar.PageExtent
Set MapExt = AEscalabar.MapExtent
On Error Resume Next
Dim PageExt As sBExtent
Dim MapExt As sBExtent
Public Sub RefreshScale(AEscalabar As Escalabar.sBEscalabar, MOMap As Map)
    End Sub
    Unload Me
    Private Sub cmdKekuar_Click()
        End Sub
```

```
delstatus = False
addstatus = False
Map1.MousePointer = moArrow
Map1.Extent = Map1.FullExtent
Call RefreshScale(sBEscalabar, Map1)
sBEscalabar.ScaleBarUnits = sumMeters
sBEscalabar.MapUnits = mMeters
legMapDisq.Active(Indeks) = True
Call AddLayer
Set g_ActiveLayer = Map1.Layers(0)
legMapDisq.ShowAllLegend
legMapDisq.LoadLegend True
legMapDisq.setMapSource Map1
Indeks = 0
```

```
GisUkm.AbdLayer "ukm"
GisUkm.AbdLayer "jalan"
GisUkm.AbdLayer "desa"
```

```
Set dc = GisUkm.Koneksi(Filedata)
Filedata = path & "Map\
Set GisUkm.Peta = Map1
'memasukan data
```

```
abmC ontainerFontSetting Me.hWnd, TEXTPOS_CAPTION, ff
E.HFaceName = "Times New Roman"
H.Height = 12
Mengatur huruf pada form utama
```

```
abmC ontainerTextSetting Me.hWnd, TEXTPOS_CAPTION, ff, 0
H.TransparentFactor = 50
H.Effect = EFFECT_RAISED
```

```
Pageext.MinX = MOMap.Left / Screen.TwipsPerPixelX
Pageext.MinY = MOMap.Top / Screen.TwipsPerPixelY
Pageext.MaxX = (MOMap.Left + MOMap.Width) / Screen.TwipsPerPixelX
Pageext.MaxY = (MOMap.Top + MOMap.Height) / Screen.TwipsPerPixelY
```

```
AEScaleBar.Refresh
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    Unload FUTama
```

```
    Set GisUkm = Nothing
```

```
End Sub
```

```
Private Sub Map1_AfterTrackingLayerDraw(ByVal hDC As stdole.OLE_HANDLE)
```

```
    Dim sym As New MapObjects2.Symbol
```

```
    Dim RecSet As MapObjects2.Recordset
```

```
    Dim lyr As MapObjects2.MapLayer
```

```
    Dim IPoly As Integer
```

```
    Dim ps As MapObjects2.Point
```

```
    If Map1.MousePointer = moCross Then
```

```
        Set lyr = Map1.Layers(0)
```

```
        Set sym = lyr.Symbol
```

```
        Set RecSet = lyr.Records
```

```
        If Not pts Is Nothing Then
```

```
            For IPoly = 0 To pts.Count - 1
```

```
                Map1.DrawShape pts(IPoly), sym
```

```
                If RecSet.Updatable Then
```

```
                    RecSet.AddNew
```

```
                    If RecSet.EditMode = moEditAdd Then
```

```
                        Set ps = pts(IPoly)
```

```
                        RecSet("Shape") = ps
```

```
                        RecSet.Update
```

```
                    End If
```

```
                Else
```

```
                    MsgBox "AddNew failed."
```

```
                End If
```

```
            Next
```

```
        End If
```

```
        Set pts = Nothing
```

```
    Else
```

```
        Set pts = New MapObjects2.Points
```

```
    End If
```

```
        'If Indeks = 0 Then
```

```
            ' Call DrawRecordset(Recs, moMagenta, moSolidFill)
```

```
        'End If
```

```
End Sub
```

```
PageExt.MinX = MOMap.Left \ Screen.TwipsPerPixelX  
PageExt.MinY = MOMap.Top \ Screen.TwipsPerPixelY  
PageExt.MaxX = (MOMap.Left + MOMap.Width) \ Screen.TwipsPerPixelX  
PageExt.MaxY = (MOMap.Top + MOMap.Height) \ Screen.TwipsPerPixelY
```

```
Ascender.Refresh
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Unload Forms
```

```
Set GsUkm = Nothing
```

```
End Sub
```

```
Private Sub Map_AfterTrackLayerDraw(ByVal hdc As Stdole.Ole_Handle)
```

```
Dim sym As New MapObjects.Symbol
```

```
Dim Recset As MapObjects.Recordset
```

```
Dim lyr As MapObjects.MapLayer
```

```
Dim lpt As Integer
```

```
Dim ps As MapObjects.Point
```

```
If Map.MousePointer = mCross Then
```

```
Set lyr = Map.Layers(0)
```

```
Set sym = lyr.Symbol
```

```
Set Recset = lyr.Recordset
```

```
If Not pts Is Nothing Then
```

```
For lpt = 0 To pts.Count - 1
```

```
Map.DrawShape pts(lpt), sym
```

```
If Recset.Updateable Then
```

```
Recset.AddNew
```

```
If Recset.EditMode = mEditAdd Then
```

```
Set ps = pts(lpt)
```

```
Recset("Shape") = ps
```

```
Recset.Update
```

```
End If
```

```
Else
```

```
MsgBox "AddNew failed."
```

```
End If
```

```
Next
```

```
End If
```

```
Set pts = Nothing
```

```
Else
```

```
Set pts = New MapObjects.Points
```

```
End If
```

```
If Indexes = 0 Then
```

```
Call DrawRecordset(Recs, mMsgArea, mSolidFill)
```

```
End If
```

```
End Sub
```

```
Private Sub Map1_BeforeLayerDraw(ByVal Index As Integer, ByVal hDC As  
stdole.OLE_HANDLE)
```

```
    If Index = Map1.Layers.Count - 1 Then  
        Call RefreshScale(sbScaleBar1, Map1)  
        StatusBar1.Panels(2) = GisUkm.updateScale
```

```
        Indeks = Index  
    End If  
End Sub
```

```
Private Sub Frame3_DragDrop(Source As Control, X As Single, Y As Single)
```

```
End Sub
```

```
Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
On Error GoTo ErrorHandler
```

```
Dim NmLyrSem As String
```

```
Dim curPoint As Point
```

```
Dim curX As Double
```

```
Dim curY As Double
```

```
Dim kode As String
```

```
Dim hasil As String
```

```
    Indeks = legMapDisp.getActiveLayer
```

```
    If Button = 1 Then
```

```
        GisUkm.KondisiMouse X, Y
```

```
        If Indeks < 0 Then
```

```
            Indeks = 0
```

```
            legMapDisp.Active(Indeks) = True
```

```
        End If
```

```
        Set g_ActiveLayer = Map1.Layers(Indeks)
```

```
        If Map1.MousePointer = moArrowQuestion Then
```

```
            Set Recs = GisUkm.RecShp(g_ActiveLayer, X, Y)
```

```
            Call DeletePoint
```

```
        End If
```

```
        If Map1.MousePointer = moIdentify Then
```

```
            NmLyrSem = g_ActiveLayer.Name
```

```
        If NmLyrSem = "ukm" Then
```

```
            Set Recs = GisUkm.RecShp(g_ActiveLayer, X, Y)
```

```
            Set g_recSelone = GisUkm.RecShp(g_ActiveLayer, X, Y)
```

```
            If Not Recs Is Nothing Then
```

```
                Map1.FlashShape Recs("Shape").Value, 3
```

```
                kode = Recs!no
```

```
            Call connect
```

```
            Set rst = New ADODB.Recordset
```

```
            rst.Source = "select * from viewukm where nomor ='" & kode & "'"
```

```
            rst.ActiveConnection = Con
```

```
            rst.CursorLocation = adUseClient
```

```

Private Sub Map1_BeforeLayerDraw(ByVal Index As Integer, ByVal HDC As
stdole.OLE_HANDLE)
    If Index = Map1.Layers.Count - 1 Then
        Call RefreshScale(pScaleBar, Map1)
        StatusBar.Panels(2) = GISUKM.updatescale
    End If
End Sub

```

```

Private Sub Frames3_DragDrop(Source As Control, X As Single, Y As Single)
End Sub

```

```

Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    On Error GoTo ErrorHandler
    Dim NumLayers As String
    Dim curPoint As Point
    Dim curX As Double
    Dim curY As Double
    Dim kode As String
    Dim hasil As String
    Index = legMapDisp.getActiveLayer
    If Button = 1 Then

```

```

        GISUKM.kondisiMouse X, Y
        If Index < 0 Then
            Index = 0
            legMapDisp.Active(Index) = True
        End If
        Set g_ActiveLayer = Map1.Layers(Index)
        If Map1.MousePointer = moArrowQuestion Then
            Set Recs = GISUKM.Recs(p_g_ActiveLayer, X, Y)
            Call DeletePoint

```

```

        End If
        If Map1.MousePointer = moIdentify Then
            NumLayers = g_ActiveLayer.Name

```

```

            If NumLayers = "ukm" Then
                Set Recs = GISUKM.Recs(p_ActiveLayer, X, Y)
                Set g_recstone = GISUKM.Recs(p_g_ActiveLayer, X, Y)
                If Not Recs Is Nothing Then
                    Map1.FlashShape Recs("shape").Value, 3
                    kode = Recs.no

```

```

                Call connect
                Set rst = New ADO.DB.Recordset
                rst.Source = "select * from viewukm where nomor = " & kode & ""
                rst.ActiveConnection = Con
                rst.CursorLocation = adUseClient

```

```

rst.LockType = adLockPessimistic
rst.CursorType = adOpenStatic
rst.Open
If rst.RecordCount > 0 Then
    FInformasi.Text1.Text = rst!pemilik
    FInformasi.Text2.Text = rst!perusahaan
    If rst!alamat <> "" Then
        FInformasi.Text3.Text = rst!alamat
    End If
    FInformasi.Text4.Text = rst!jenis
    FInformasi.Text5.Text = "Rp. " & rst.Fields!totalprod
    FInformasi.Text6.Text = "Rp. " & rst.Fields!totalomze
    FInformasi.Text7.Text = "Rp. " & rst.Fields!totalaset
    FInformasi.Text8.Text = rst.Fields!totalkerja & " Orang"
    FInformasi.Text9.Text = rst.Fields!pemasaran
    FInformasi.Text10.Text = rst.Fields!namadesa
    FInformasi.Text11.Text = rst.Fields!namakec
    FInformasi.Text12.Text = rst.Fields!ket
    FInformasi.Text13.Text = rst.Fields!nomor
    rst.Close
    DataEnvironment1.Connection1.Close
    FInformasi.Show
    Map1.Refresh

    End If
    Else
    Unload FInformasi
    Exit Sub

    End If
End If
Dim fld As MapObjects2.Field
Dim Luas As Double
If NmLyrSem = "desa" Then
    Set Recs = GisUkm.RecShp(g_ActiveLayer, X, Y)

    If Not Recs Is Nothing Then
        Map1.FlashShape Recs("Shape").Value, 3

        For Each fld In Recs.Fields
            If fld.Type = moPolygon Then
                Luas = fld.Value.Area
            End If
        Next
        MsgBox "Nama = " & Recs!Nama & vbCrLf & _
            "Luas Area =" & FormatNumber(Luas, 2), vbOKOnly, "Desa"

    End If
End If

Elseif Map1.MousePointer = moCross Then
    Set curPoint = Map1.ToMapPoint(X, Y)

```



```

Set cursorint = Map1.ToMapPoint(X, Y)
ElseIf Map1.MousePointer = mcrCross Then
End If
End If
"User Area = " & FormatNumber(Luas, 2, vbOKOnly, "Desa")
MsgBox "Nama = " & RecName & vbCrLf & _
Next
End If
Luas = fld.Value.Area
If fld.Type = mofPolygon Then
For Each fld In Recs.Fields
Map1.EraseShape Recs("shape").Value.3
If Not Recs.IsNothing Then
Set Recs = GSet(RecShape(g_ActiveLayer, X, Y)
If NumLayers = "Desa" Then
Dim Luas As Double
Dim fld As MapObjects2.Field
End If
End If
End If
Exit Sub
Until EndFormasi
Else
End If
Map1.Refresh
Formasi.Show
DataEnvironment1.Connection1.Close
Set Close
Formasi.Text3.Text = rstField1Nameor
Formasi.Text12.Text = rstField1Kot
Formasi.Text11.Text = rstField1NamaDesa
Formasi.Text10.Text = rstField1NamaDesa
Formasi.Text9.Text = rstField1Permasaran
Formasi.Text8.Text = rstField1TataKerja & "Orang"
Formasi.Text7.Text = "Rp." & rstField1Jumlahaset
Formasi.Text6.Text = "Rp." & rstField1Jumlahtanah
Formasi.Text5.Text = "Rp." & rstField1Jumlahperod
Formasi.Text4.Text = rstJenis
End If
Formasi.Text3.Text = rstJumlah
If rstJumlah < "0" Then
Formasi.Text2.Text = rstPermasaran
Formasi.Text1.Text = rstPermisik
If rstRecordCount > 0 Then
rst.Open
rst.CursorType = adOpenStatic
rst.LockType = adLockPessimistic

```

```

curX = curPoint.X
curY = curPoint.Y
With frmUpdate_Koordinat
    .txtX.Text = Format(curX, "0.000")
    .txtY.Text = Format(curY, "0.000")
End With
' Map1.TrackingLayer.AddEvent curPoint, 0
Dim lyr As MapObjects2.MapLayer
Set lyr = Map1.Layers(0)

```

```

If lyr.shapeType = moShapeTypePoint Then

```

```

    If Not pts Is Nothing Then
        Set p = Map1.ToMapPoint(X, Y)
        pts.Add p
    
```

```

End If

```

```

End If

```

```

Map1.TrackingLayer.Refresh True
Elseif Map1.MousePointer = moArrow Then
    Dim idkec As String
    NmLyrSem = g_ActiveLayer.Name
    If NmLyrSem = "ukm" Then
        Set Recs = GisUkm.RecShp(g_ActiveLayer, X, Y)
        Set g_recSelone = GisUkm.RecShp(g_ActiveLayer, X, Y)
        If Recs Is Nothing Then
            Unload FInformasi
            Exit Sub
        Elseif Recs.Count > 0 Then
            Dim JRecord As Long
            Dim hasilcek As String
            Map1.FlashShape Recs("Shape").Value, 3
            kode = Recs!no
            Call connect
            Set rst = New ADODB.Recordset
            rst.Source = "select * from tukm where tukm.nomor =" & kode & ""
            rst.ActiveConnection = Con
            rst.CursorLocation = adUseClient
            rst.LockType = adLockPessimistic
            rst.CursorType = adOpenStatic
            rst.Open
            If Not (rst.BOF And rst.EOF) Then
                hasilcek = rst.Fields!perusahaan
                JRecord = rst.RecordCount
            End If
            rst.Close
            DataEnvironment1.Connection1.Close
            If JRecord > 0 Then

```

```

        curX = curPointX
        curY = curPointY
        With frmUpdate_Koordinat
            txtX.Text = Format(curX, "0.000")
            txtY.Text = Format(curY, "0.000")
        End With
        Map1.TrackingLayer.AddEvent curPoint, 0
        Dim lyr As MapObjects2.MapLayer
        Set lyr = Map1.Layers(0)

        If lyr.ShapeType = moshapeTypePoint Then

            If Not pts Is Nothing Then
                Set p = Map1.ToMapPoint(X, Y)
                pts.Add p
            End If

            End If

        Map1.TrackingLayer.Refresh True
        ElseIf Map1.MousePointer = moArrow Then
            Dim idrec As String
            Dim lyrsem = g_ActiveLayerName
            If Not lyrsem = "ukm" Then
                Set Recs = GIsUkm.Recshp(g_ActiveLayer, X, Y)
                Set g_recstone = GIsUkm.Recshp(g_ActiveLayer, X, Y)
                If Recs Is Nothing Then
                    Unload FrmFormasi
                    Exit sub
                ElseIf Recs.Count > 0 Then
                    Dim RRecord As Long
                    Dim hshidrec As String
                    Map1.HshShape Recs("Shape").Value, 3
                    Kode = RRecord
                    Call connect
                    Set rst = New ADOODB.Recordset
                    rst.Source = "select * from tukm where tukm.nomer = '" & Kode & "'"
                    rst.ActiveConnection = Con
                    rst.CursorLocation = adUseClient
                    rst.LockType = adLockPessimistic
                    rst.CursorType = adOpenStatic
                    rst.Open
                    If Not (rst.EOF And rst.EOF) Then
                        hshidrec = rst.Fields("hshidrec").Value
                        RRecord = rst.RecordCount
                    End If
                    rst.Close
                End If
            End If
        End If
    End Sub

```

```
MsgBox "Nama UKM = " & hasilcek & "", vbOKOnly + vbInformation, "Informasi"
```

```
Else  
With EditUKM  
.Caption = "Tambahkan Data UKM Baru"  
.Show  
End With  
End If
```

```
End If  
End If  
End If
```

```
Else  
PopupMenu mnu, , X + 2750, Y + 800  
End If  
Exit Sub
```

```
ErrorHandle:
```

```
rst.Close  
DataEnvironment1.Connection1.Close  
MsgBox Err.Description  
Exit Sub  
End Sub
```

```
Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
'This procedure updates the coordinate display in the status bar.
```

```
Dim curPoint As Point
```

```
Dim curX As Double
```

```
Dim curY As Double
```

```
'Convert screen coordinates to map coordinates
```

```
Set curPoint = Map1.ToMapPoint(X, Y)
```

```
curX = curPoint.X
```

```
curY = curPoint.Y
```

```
'If map coordinates are large, suppress digits to right of decimal place.
```

```
Dim cx As String, cy As String
```

```
cx = curX
```

```
cy = curY
```

```
cx = Left(cx, InStr(cx, ".") + 3)
```

```
cy = Left(cy, InStr(cy, ".") + 3)
```

```
StatusBar1.Panels(1) = "X: " & cx & " Y: " & cy
```

```
Call RefreshScale(sbScaleBar1, Map1)
```

```
End Sub
```

```
Private Sub AturLayer()
```

```
Set frmLayerSymbol.lyr = g_ActiveLayer
```

```
Set frmLayerSymbol.Recs = frmLayerSymbol.lyr.Records
```

```
Set frmLayerSymbol.tDesc = frmLayerSymbol.Recs.TableDesc
```

```
Set frmLayerSymbol.flds = frmLayerSymbol.Recs.Fields
```

```
Call frmLayerSymbol.PopulateNewUVlegend("Nama")
```

```
legMapDisp.LoadLegend
```

```
Map1.Refresh
```

```

Map.Refresh
legislapdispl.loadlegend
Call frmLayerSymbol.PopulateNewUvwLegend("Nama")
Set frmLayerSymbol.Fields = frmLayerSymbol.Record.Fields
Set frmLayerSymbol.TableDesc = frmLayerSymbol.Record.TableDesc
Set frmLayerSymbol.Record = frmLayerSymbol.Record
Set frmLayerSymbol.Layer = g_ActiveLayer
Private Sub AtnLayer()
End Sub
Call RevealCaption(AppCaption, Map)
StatusBar.Labels(1) = "X: " & cx & " Y: " & cy
cy = Left(cy, Instr(cy, ".") + 3)
cx = Left(cx, Instr(cx, ".") + 3)
cy = curY
cx = curX
Dim cx As String, cy As String
If map coordinates are large, suppress digits to right of decimal place.
curY = curPoint.Y
curX = curPoint.X
Set curPoint = Map.ToMapPoint(X, Y)
'Convert screen coordinates to map coordinates
Dim curY As Double
Dim curX As Double
Dim curPoint As Point
This procedure updates the coordinate display in the status bar.
Private Sub Map_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
End Sub
End Sub
Map.BoxEnt.Description
Exit Sub
DataEnvironment.Connection.Close
Set Close
End Sub
End Sub
End Sub
Else
Population map, X + 2750, Y + 800
End If
End Sub
End If
End If
End If
Caption = "Tampilkan Data UKM Bar"
Show
With EditUKM
Caption = "Tampilkan Data UKM Bar"
Show
End With
End If
End If
End If
End If
MagBox "Nama UKM = " & hasilcek & ", vboKOnly + vplnformation, Informasi"

```

```

Set frmLayerSymbol.lyr = Nothing
Set frmLayerSymbol.Recs = Nothing
Set frmLayerSymbol.tDesc = Nothing
Set frmLayerSymbol.flds = Nothing
Unload frmLayerSymbol
End Sub

Private Sub legMapDisp_AfterSetLayerVisible(Index As Integer, isVisible As Boolean)
    Map1.Refresh
End Sub
Private Sub legMapDisp_LayerDbClick(Index As Integer)
    Set g_ActiveLayer = Map1.Layers(Index)
    Indeks = Index
    If g_ActiveLayer.LayerType = moImageLayer Then
        MsgBox "Sorry, you cannot set properties for an image layer." & _
            vbCrLf & "Try setting properties for a layer with vector data."
        Exit Sub
    End If
    FormAktif = 2
    frmLayerSymbol.Show 1
End Sub

Private Sub mnuAddTheme_Click()
    Dim strFileTitle As String, sDataSetName As String
    Dim sFileDirectory As String
    Dim dc As New MapObjects2.DataConnection
    Dim lyr As New MapObjects2.MapLayer

    CommonDialog1.Filter = "Shapefiles (*.shp) | *.shp"
    CommonDialog1.ShowOpen

    strFileTitle = CommonDialog1.FileTitle 'just the name with the shp extension

    If strFileTitle <> "" Then

        With CommonDialog1
            sFileDirectory = Left$(.FileName, InStr(.FileName, .FileTitle) - 1)

        End With

        With dc
            .Database = sFileDirectory 'the directory containing the selected file

            If .connect Then
                sDataSetName = Mid(strFileTitle, 1, Len(strFileTitle) - 4) 'remove the extension
                Set lyr.GeoDataset = dc.FindGeoDataset(sDataSetName)
                Map1.Layers.Add lyr
                Map1.Refresh
                legMapDisp.LoadLegend True
            Else
                MsgBox "Data Connection error #" & dc.ConnectError, vbExclamation
            End If
        End With
    End If

```


End With

End If

Set dc = Nothing

Set lyr = Nothing

End Sub

Private Sub mnuDelTheme_Click()

Dim ans, Index As Integer

Index = legMapDisp.getActiveLayer

**ans = MsgBox("Apakah Anda yakin ingin menghapus/menghilangkan layer ini?", vbYesNo +
vbQuestion, _**

"Konfirmasi", "", 0)

If ans = vbYes Then

Map1.Layers.Remove Index

Map1.Refresh

legMapDisp.LoadLegend True

End If

End Sub

Private Sub mnuExit_Click()

Unload Me

End Sub

Private Sub mnuNormal_Click()

Map1.MousePointer = moArrow

Toolbar1.Buttons(1).Value = tbrPressed

End Sub

Private Sub mnuPerbesar_Click()

Map1.MousePointer = moZoomIn

Toolbar1.Buttons(2).Value = tbrPressed

End Sub

Private Sub mnuPerkecil_Click()

Map1.MousePointer = moZoomOut

Toolbar1.Buttons(3).Value = tbrPressed

End Sub

Private Sub mnuQuery_Click()

'Dim judul As String

Set g_ActiveLayer = Map1.Layers(legMapDisp.getActiveLayer)

With QueryDialog

.Caption = g_ActiveLayer.Name

.Show

End With


```

End With
End If
Set dc = Nothing
Set lyr = Nothing

End Sub

Private Sub mnuDefHome_Click()
Dim ans As Integer
index = legMapDisp.getActiveLayer
ans = MsgBox("Apakah Anda yakin ingin menghapus/menghilangkan layer ini?", vbYesNo +
vbQuestion, "Konfirmasi", 0)
If ans = vbYes Then
Map1.Layers.Remove index
Map1.Refresh
legMapDisp.LoadLegend True
End If
End Sub

Private Sub mnuExit_Click()
Unload Me
End Sub

Private Sub mnuNormal_Click()
Map1.MousePointer = moArrow
ToolBar1.Buttons(1).Value = vbPressed
End Sub

Private Sub mnuPerbesar_Click()
Map1.MousePointer = moZoomIn
ToolBar1.Buttons(2).Value = vbPressed
End Sub

Private Sub mnuPerkecil_Click()
Map1.MousePointer = moZoomOut
ToolBar1.Buttons(3).Value = vbPressed
End Sub

Private Sub mnuQuery_Click()
Dim judul As String
Set g_ActiveLayer = Map1.Layers(legMapDisp.getActiveLayer)
With QueryDialog
.Caption = g_ActiveLayer.Name
.Show
End With

```

End Sub

Private Sub mnuTampil_Click()

Map1.Extent = Map1.FullExtent

Call RefreshScale(sbScaleBar1, Map1)

End Sub

Private Sub DeletePoint()

Dim cc As Integer

Dim res As New ADODB.Recordset

'MsgBox Recs!no

If Not Recs Is Nothing Then

cc = MsgBox("Apa Anda Yakin?", vbExclamation + vbYesNo, "Konfirmasi")

If cc = vbYes Then

Recs.Delete

Set Recs = Nothing

Map1.TrackingLayer.Refresh True

Map1.Refresh

Else

Exit Sub

End If

End If

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSCComctlLib.Button)

On Error Resume Next

Dim LayerPasuruan As MapObjects2.MapLayer

If Not pts Is Nothing Then

Set pts = Nothing

End If

Select Case Button.Index

Case 1: Map1.MousePointer = moArrow

Case 2: Map1.MousePointer = moZoomIn

Case 3: Map1.MousePointer = moZoomOut

Case 4: Map1.MousePointer = moPan

Case 5: Map1.MousePointer = moIdentify

Case 6: Map1.Extent = Map1.FullExtent

Case 7: FTabel.Show

Case 8:

Map1.MousePointer = moCross

Set pts = New MapObjects2.Points

Case 9: Map1.MousePointer = moArrowQuestion

Case 11

FormAktif = 1

PrinterForm.Show

Case 12

Fcari.Tampil_data Map1.Layers("ukm").Records, "ukm"

Fcari.Show

```

End Sub

Private Sub mainTampil_Click()
    Map1.Extent = Map1.FullExtent
    Call RefreshScaleBar(Map1)
End Sub

Private Sub DeletePoint()
    Dim cc As Integer
    Dim res As New ADODB.Recordset
    MsgBox Kerasno
    If Not Res Is Nothing Then
        cc = MsgBox("Apakah Anda Yakin?", vbExclamation + vbYesNo, "Konfirmasi")

        If cc = vbYes Then
            Res.Delete
            Set Res = Nothing
            Map1.RefreshLayer.Refresh True
            Map1.Refresh
        Else
            Exit Sub
        End If
    End If
End Sub

Private Sub Toolbar_Button_Click(ByVal Button As MSComctlLib.Button)
    On Error Resume Next
    Dim LayerPapanan As MapObject2.MapLayer
    If Not pts Is Nothing Then
        Set pts = Nothing
    End If

    Select Case Button.Index
        Case 1: Map1.MousePointer = moArrow
        Case 2: Map1.MousePointer = moZoomIn
        Case 3: Map1.MousePointer = moZoomOut
        Case 4: Map1.MousePointer = moPan
        Case 5: Map1.MousePointer = moIdentify
        Case 6: Map1.Extent = Map1.FullExtent
        Case 7: Label.Show
        Case 8:
            Map1.MousePointer = moCross
            Set pts = New MapObject2.Points
        Case 9: Map1.MousePointer = moArrowQuestion

        Case 11
            FormAktif = 1
            PrinterForm.Show
        Case 12
            FormTampil_data Map1.Layers("ukm").Records, "ukm"
            Form.Show
    End Select

```

```

    Case 14: Unload Me
    End Select
End Sub
Private Sub mnuGeser_Click()
    Map1.MousePointer = moPan
    Toolbar1.Buttons(4).Value = tbrPressed
End Sub

Private Sub mnuInformasi_Click()
    Map1.MousePointer = moIdentify
    Toolbar1.Buttons(5).Value = tbrPressed
End Sub

Sub DrawRecordset(ByVal Recs1 As MapObjects2.Recordset, Color, style)
    ' draw the features of a RecordSet
    If Not Recs1 Is Nothing Then
        Dim sym As New Symbol
        sym.Color = Color
        If style = moTransparentFill Then sym.OutlineColor = Color
        sym.style = style
        Map1.DrawShape Recs1, sym
    End If
End Sub

' Button Styles (as defined by Windows)
Public Const BS_PUSHBUTTON = 0
Public Const BS_CHECKBOX = 2
Public Const BS_RADIOBUTTON = 4

' Container level settings
Public Const SET_SHAPE = 0
Public Const SET_ROUNDNESS = 1
Public Const SET_SLANT = 2
Public Const SET_SURFACE = 3
Public Const SET_COLOR_BACKGROUND = 4
Public Const SET_COLOR_FOREGROUND = 5
Public Const SET_COLOR_MOUSEOVER = 6
Public Const SET_COLOR_FOCUS = 7
Public Const SET_COLOR_INTERIOR = 8
Public Const SET_COLOR_TRANSP = 9
Public Const SET_COLOR_CONTAINER_BACK = 10
Public Const SET_COLOR_HIGHLIGHT = 11
Public Const SET_COLOR_SHADOW = 12
Public Const SET_COLOR_TEXTURE_MIX = 13
Public Const SET_COLOR_BACKGROUND_DISABLED = 14
Public Const SET_COLOR_FOREGROUND_DISABLED = 15
Public Const SET_COLOR_BACKGROUND_PRESSED = 16
Public Const SET_PICT_XOFFSET = 17
Public Const SET_PICT_YOFFSET = 18
Public Const SET_PICT_POS = 19
Public Const SET_PICT_TRANSP_MODE = 20
Public Const SET_PICT_FIT_IN_BTN = 21

```

```

ЪРРНС СОУЛ ГЕЛ ЪСЛ ЪЛ ЪЛ ЪЛ = 31
ЪРРНС СОУЛ ГЕЛ ЪСЛ ЪКЪИЪЪ МОДЕ = 30
ЪРРНС СОУЛ ГЕЛ ЪСЛ ЪОС = 18
ЪРРНС СОУЛ ГЕЛ ЪСЛ ЪОУСЕЛ = 18
ЪРРНС СОУЛ ГЕЛ ЪСЛ ЪОУСЕЛ = 12
ЪРРНС СОУЛ ГЕЛ СОГОК ЪУСКЕКОВИД ЪИГСЕД = 16
ЪРРНС СОУЛ ГЕЛ СОГОК ЪОУЕКЕКОВИД ДИЪУВЕД = 12
ЪРРНС СОУЛ ГЕЛ СОГОК ЪУСКЕКОВИД ДИЪУВЕД = 11
ЪРРНС СОУЛ ГЕЛ СОГОК ЪЕХИЪЪЕ ЪИХ = 13
ЪРРНС СОУЛ ГЕЛ СОГОК ЪИУДОМ = 13
ЪРРНС СОУЛ ГЕЛ СОГОК ЪИСНГІСНІ = 11
ЪРРНС СОУЛ ГЕЛ СОГОК СОИЪИЕК ЪУСК = 10
ЪРРНС СОУЛ ГЕЛ СОГОК ЪКЪИЪЪ = 8
ЪРРНС СОУЛ ГЕЛ СОГОК ЪИЛЕУОК = 8
ЪРРНС СОУЛ ГЕЛ СОГОК ЪОУС = 1
ЪРРНС СОУЛ ГЕЛ СОГОК ЪОУСЕОУЕК = 6
ЪРРНС СОУЛ ГЕЛ СОГОК ЪОУЕКЕКОВИД = 2
ЪРРНС СОУЛ ГЕЛ СОГОК ЪУСКЕКОВИД = 1
ЪРРНС СОУЛ ГЕЛ ЪИУЕАСЕ = 3
ЪРРНС СОУЛ ГЕЛ ЪИУИЛ = 3
ЪРРНС СОУЛ ГЕЛ ЪОВИДИУЕЕ = 1
ЪРРНС СОУЛ ГЕЛ ЪИУЕ = 0
, Contingent level settings

```

```

ЪРРНС СОУЛ ВЕ ЪАДИОВУЛЛОИ = 1
ЪРРНС СОУЛ ВЕ ЪЧЕКЪОХ = 3
ЪРРНС СОУЛ ВЕ ЪОУЕВУЛЛОИ = 0
, Window styles (as defined by Windows)

```

```

End sub
End If
  MsgBox MsgBoxStyle Key1 & Chr(Asc(Key2))
  & Chr(Asc(Key3)) = Key3
  If Key3 = vbEnter Then Print Chr(Asc(Key4)) = Key4
  Chr(Asc(Key5)) = Key5
  Dim Key As VbKey
  If Not Key Is vbEnter Then
    , that, the features of a keyboard
  End MsgBoxBox(Style Key1 & Chr(Asc(Key2)) & Chr(Asc(Key3)) & Chr(Asc(Key4)))

```

```

End sub
  MsgBox MsgBoxStyle vbYes = vbYes
  MsgBox MsgBoxStyle = vbYes
  MsgBox MsgBoxStyle vbYes = vbYes

```

```

End sub
  MsgBox MsgBoxStyle vbYes = vbYes
  MsgBox MsgBoxStyle = vbYes
  MsgBox MsgBoxStyle vbYes = vbYes

```

```

End sub
  Case 1: MsgBox Yes

```

Public Const SET_PICT_KEEP_RATIO = 22
Public Const SET_PICT_MOVE = 23
Public Const SET_BTN_RAISED = 24
Public Const SET_SMOOTH_EDGES = 25
Public Const SET_FOCUS_MODE = 26
Public Const SET_SPECIAL_EFFECT = 27
Public Const SET_SPECIAL_EFFECT_FACTOR = 28
Public Const SET_CAPTION_WORDWRAP_PERC = 29
Public Const SET_CLICKABLE = 30
Public Const SET_TEXTURE_MODE = 31
Public Const SET_TEXTURE_LIGHT = 32
Public Const SET_TEXTURE_BEVEL_FACTOR = 33
Public Const SET_FLAT_PILLOW_FACTOR = 34
Public Const SET_SHADOW_MODE = 35
Public Const SET_COLOR_FOREGROUND_MOUSEOVER = 36
Public Const SET_TEXTURE_MDX_PERC = 37
Public Const SET_COLOR_FOREGROUND_PRESSED = 38
Public Const SET_PICT_VISIBLE = 39
Public Const SET_ORIENTATION = 40
Public Const SET_SPOTLIGHT_OFFSETX = 41
Public Const SET_SPOTLIGHT_OFFSETY = 42
Public Const SET_SPOTLIGHT_RESIZEW = 43
Public Const SET_SPOTLIGHT_RESIZEH = 44
Public Const SET_USEANTIALIAS = 45

' Shapes

Public Const SHAPE_RECTANGLE = 0
Public Const SHAPE_ROUNDRECT = 1
Public Const SHAPE_PARALLEL = 2
Public Const SHAPE_ELLIPSE = 3
Public Const SHAPE_GEM = 4
Public Const SHAPE_CUSTOM = 5

' Grades

Public Const GRD_VERYHIGH = 0
Public Const GRD_HIGH = 1
Public Const GRD_MEDIUM = 2
Public Const GRD_LOW = 3
Public Const GRD_VERYLOW = 4

' Surface types

Public Const SURFACE_HARD = 0
Public Const SURFACE_SOFT = 1
Public Const SURFACE_FLAT = 2
Public Const SURFACE_FLATPIL = 3
Public Const SURFACE_SUNKEN = 4
Public Const SURFACE_PLASTIC = 5
Public Const SURFACE_LUCID = 6
Public Const SURFACE_NEON = 7
Public Const SURFACE_AQUA = 8
Public Const SURFACE_GEL = 9
Public Const SURFACE_PLASTIC_SHAPED = 10

Public Const SURFACE_PLASTIC_SHAPED = 10
Public Const SURFACE_GLZ = 9
Public Const SURFACE_AQUA = 8
Public Const SURFACE_NEON = 7
Public Const SURFACE_LUCID = 6
Public Const SURFACE_PLASTIC = 5
Public Const SURFACE_SUNKEN = 4
Public Const SURFACE_FLAT = 3
Public Const SURFACE_SOFT = 1
Public Const SURFACE_HARD = 0

' Surface types

Public Const GRD_VERYLOW = 4
Public Const GRD_LOW = 3
Public Const GRD_MEDIUM = 2
Public Const GRD_HIGH = 1
Public Const GRD_VERYHIGH = 0

' Grades

Public Const SHAPE_CUSTOM = 5
Public Const SHAPE_GEM = 4
Public Const SHAPE_ELLIPSE = 3
Public Const SHAPE_PARALLAL = 2
Public Const SHAPE_ROUNDRECT = 1
Public Const SHAPE_RECTANGLE = 0

' Shapes

Public Const UESHAPEMATERIALS = 45
Public Const SPOTLIGHT_RESIZEH = 44
Public Const SPOTLIGHT_RESIZEW = 43
Public Const SPOTLIGHT_OFFSETY = 42
Public Const SPOTLIGHT_OFFSETX = 41
Public Const SET_ORIENTATION = 40
Public Const SET_PICT_VISIBLE = 39
Public Const SET_COLOR_FOREGROUND_PRESSED = 38
Public Const SET_TEXTURE_MIX_PERC = 37
Public Const SET_COLOR_FOREGROUND_MOUSEOVER = 36
Public Const SET_SHADOW_MODE = 35
Public Const SET_FLAT_PILLOW_FACTOR = 34
Public Const SET_TEXTURE_BRIVIL_FACTOR = 33
Public Const SET_TEXTURE_LIGHT = 32
Public Const SET_TEXTURE_MODE = 31
Public Const SET_CLICKABLE = 30
Public Const SET_CAPTION_WORDWRAP_PERC = 29
Public Const SET_SPECIAL_EFFECT_FACTOR = 28
Public Const SET_SPECIAL_EFFECT = 27
Public Const SET_FOCUS_MODE = 26
Public Const SET_SMOOTH_EDGES = 25
Public Const SET_BTN_RAISED = 24
Public Const SET_PICT_MOVE = 23
Public Const SET_PICT_KEEP_RATIO = 22

Public Const SURFACE_NEON_SHAPED = 11
Public Const SURFACE_AQUA_SHAPED = 12

' Text positions

Public Const TEXTPOS_CAPTION = 0 ' Center-Middle
Public Const TEXTPOS_LT = 1 ' Left-Top
Public Const TEXTPOS_CT = 2 ' Center-Top
Public Const TEXTPOS_RT = 3 ' Right-Top
Public Const TEXTPOS_LM = 4 ' Left-Middle
Public Const TEXTPOS_RM = 5 ' Right-Middle
Public Const TEXTPOS_LB = 6 ' Left-Bottom
Public Const TEXTPOS_CB = 7 ' Center-Bottom
Public Const TEXTPOS_RB = 8 ' Right-Bottom

' Picture positions

Public Const PICTPOS_CM = 0 ' Center-Middle
Public Const PICTPOS_LT = 1 ' Left-Top
Public Const PICTPOS_CT = 2 ' Center-Top
Public Const PICTPOS_RT = 3 ' Right-Top
Public Const PICTPOS_LM = 4 ' Left-Middle
Public Const PICTPOS_RM = 5 ' Right-Middle
Public Const PICTPOS_LB = 6 ' Left-Bottom
Public Const PICTPOS_CB = 7 ' Center-Bottom
Public Const PICTPOS_RB = 8 ' Right-Bottom
Public Const PICTPOS_TILE = 9 ' Tile
Public Const PICTPOS_STRETCH = 10 ' Stretch
Public Const PICTPOS_SIZETOCONTENT = 11 ' Size to content

' Transparent modes

Public Const TRANSPMODE_USECOLOR = 0
Public Const TRANSPMODE_USELEFTTOP_PIXEL = 1

' Text effects

Public Const EFFECT_NONE = 0
Public Const EFFECT_RAISED = 1
Public Const EFFECT_SUNKEN = 2
Public Const EFFECT_FLOATING = 3
Public Const EFFECT_HALO = 4
Public Const EFFECT_RAISEDHALO = 5
Public Const EFFECT_SUNKENHALO = 6
Public Const EFFECT_FLOATINGHALO = 7

' Special effects

Public Const SPECIAL_EFFECT_NONE = 0
Public Const SPECIAL_EFFECT_RAISED = 1
Public Const SPECIAL_EFFECT_SUNKEN = 2
Public Const SPECIAL_EFFECT_SUNKEN_SIMPLE = 3

' Special effects factors

Public Const SPECEFF_VERYHIGH = 0
Public Const SPECEFF_HIGH = 1
Public Const SPECEFF_MEDIUM = 2

Public Const SURFACE_NRON_SHAPED = 11
Public Const SURFACE_AQUA_SHAPED = 12

' Text positions

Public Const TEXTPOS_CAPTION = 0 ' Center-Middle
Public Const TEXTPOS_LT = 1 ' Left-Top
Public Const TEXTPOS_CT = 2 ' Center-Top
Public Const TEXTPOS_RT = 3 ' Right-Top
Public Const TEXTPOS_LM = 4 ' Left-Middle
Public Const TEXTPOS_RM = 5 ' Right-Middle
Public Const TEXTPOS_LB = 6 ' Left-Bottom
Public Const TEXTPOS_CB = 7 ' Center-Bottom
Public Const TEXTPOS_RB = 8 ' Right-Bottom

' Picture positions

Public Const PICTPOS_CM = 0 ' Center-Middle
Public Const PICTPOS_LT = 1 ' Left-Top
Public Const PICTPOS_CT = 2 ' Center-Top
Public Const PICTPOS_RT = 3 ' Right-Top
Public Const PICTPOS_LM = 4 ' Left-Middle
Public Const PICTPOS_RM = 5 ' Right-Middle
Public Const PICTPOS_LB = 6 ' Left-Bottom
Public Const PICTPOS_CB = 7 ' Center-Bottom
Public Const PICTPOS_RB = 8 ' Right-Bottom
Public Const PICTPOS_TILE = 9 ' Tile
Public Const PICTPOS_STRETCH = 10 ' Stretch
Public Const PICTPOS_SIZETOCONTENT = 11 ' Size to content

' Transparent modes

Public Const TRANSPMODE_USRCOLOR = 0
Public Const TRANSPMODE_USELEFTTOP_PIXEL = 1

' Text effects

Public Const EFFECT_NONE = 0
Public Const EFFECT_RAISED = 1
Public Const EFFECT_SUNKEN = 2
Public Const EFFECT_FLOATING = 3
Public Const EFFECT_HALO = 4
Public Const EFFECT_RAISEDHALO = 5
Public Const EFFECT_SUNKENHALO = 6
Public Const EFFECT_FLOATINGHALO = 7

' Special effects

Public Const SPECIAL_EFFECT_NONE = 0
Public Const SPECIAL_EFFECT_RAISED = 1
Public Const SPECIAL_EFFECT_SUNKEN = 2
Public Const SPECIAL_EFFECT_SUNKEN_SIMPLE = 3

' Special effects factors

Public Const SPECIEFF_VERYHIGH = 0
Public Const SPECIEFF_HIGH = 1
Public Const SPECIEFF_MEDIUM = 2

Public Const SPECEFF_LOW = 3
Public Const SPECEFF_VERYLOW = 4

' Pictures

Public Const PICTURE_NORMAL = 0
Public Const PICTURE_PRESSED = 1
Public Const PICTURE_MOUSEOVER = 2
Public Const PICTURE_CUSTOM = 3
Public Const PICTURE_TEXTURE = 4
Public Const PICTURE_DISABLED = 5

' Sounds

Public Const SOUND_DOWN = 0
Public Const SOUND_UP = 1
Public Const SOUND_MOUSEOVER = 2
Public Const SOUND_MOUSEOUT = 3

' Cursors

Public Const CURSOR_NORMAL = 0

' Download types

Public Const DLL_TYPE_PICTURE = 0
Public Const DLL_TYPE_ANIM = 1
Public Const DLL_TYPE_SOUND = 2

' Smooth

Public Const SMOOTH_NONE = 0
Public Const SMOOTH_LOW = 1
Public Const SMOOTH_HIGH = 2

' Focus mode

Public Const FOCUS_NONE = 0
Public Const FOCUS_BORDER = 1
Public Const FOCUS_COLOR = 2

' Surface color elements

Public Const SURFACE_BACK = 0
Public Const SURFACE_MO = 1
Public Const SURFACE_FOCUS = 2
Public Const SURFACE_INTERIOR = 3
Public Const SURFACE_DISABLED = 4
Public Const SURFACE_PRESSED = 5

' 3D Render Type

Public Const COLOR_RENDER_BEST_MATCH = 0
Public Const COLOR_RENDER_BEST_3D = 1

' 3D Gradient Type

Public Const COLOR_GRADIENT_CLASSIC = 0
Public Const COLOR_GRADIENT_HORIZONTAL = 1
Public Const COLOR_GRADIENT_DIAGONAL = 2
Public Const COLOR_GRADIENT_VERTICAL = 3

```
Public Const COLOR_GRADIENT_VERTICAL = 3
Public Const COLOR_GRADIENT_DIAGONAL = 2
Public Const COLOR_GRADIENT_HORIZONTAL = 1
Public Const COLOR_GRADIENT_CLASSIC = 0

' 3D Gradient Type
Public Const COLOR_RENDERER_BEST_3D = 1
Public Const COLOR_RENDERER_BEST_MATCH = 0

' 3D Render Type
Public Const SURFACE_PRESSED = 5
Public Const SURFACE_DISABLED = 4
Public Const SURFACE_INTERIOR = 3
Public Const SURFACE_FOCUS = 2
Public Const SURFACE_MO = 1
Public Const SURFACE_BACK = 0

' Surface color elements
Public Const FOCUS_COLOR = 2
Public Const FOCUS_BORDER = 1
Public Const FOCUS_NONE = 0

' Focus mode
Public Const SMOOTH_HIGH = 2
Public Const SMOOTH_LOW = 1
Public Const SMOOTH_NONE = 0

' Smooth
Public Const DLL_TYPE_SOUND = 2
Public Const DLL_TYPE_ANIM = 1
Public Const DLL_TYPE_PICTURE = 0

' Download type
Public Const CURSOR_NORMAL = 0

' Cursors
Public Const SOUND_MOUSEOUT = 3
Public Const SOUND_MOUSEOVER = 2
Public Const SOUND_UP = 1
Public Const SOUND_DOWN = 0

' Sounds
Public Const PICTURE_DISABLED = 5
Public Const PICTURE_TEXTURE = 4
Public Const PICTURE_CUSTOM = 3
Public Const PICTURE_MOUSEOVER = 2
Public Const PICTURE_PRESSED = 1
Public Const PICTURE_NORMAL = 0

' Pictures
Public Const SPEECH_VERYLOW = 4
Public Const SPEECH_LOW = 3
```

' Texture modes

Public Const TEXTURE_NONE = 0
Public Const TEXTURE_WHOLE = 1
Public Const TEXTURE_OUTSIDE = 2
Public Const TEXTURE_INSIDE = 3

' Export images types

Public Const EXPORT_IMAGE_NORMAL = 0
Public Const EXPORT_IMAGE_MOUSEOVER = 1
Public Const EXPORT_IMAGE_PRESSED = 2
Public Const EXPORT_IMAGE_FOCUS = 3
Public Const EXPORT_IMAGE_DISABLED = 4

' Export images formats {

Public Const EXPORT_FORMAT_BMP = 0
Public Const EXPORT_FORMAT_PNG = 1
Public Const EXPORT_FORMAT_JPG = 2
Public Const EXPORT_FORMAT_CLIPBOARD = 3
Public Const EXPORT_FORMAT_GIF = 4

' Pillow factors

Public Const PILLOW_VERYLOW = 0
Public Const PILLOW_LOW = 1
Public Const PILLOW_MEDIUM = 2
Public Const PILLOW_HIGH = 3
Public Const PILLOW_VERYHIGH = 4

' Shadow modes

Public Const SHADOW_NONE = 0
Public Const SHADOW_SOLID = 1
Public Const SHADOW_HALFBLURRED = 2
Public Const SHADOW_BLURRED = 3

' Orientations

Public Const DIR_ALL = 0
Public Const DIR_WEST = 1
Public Const DIR_NORTH = 2
Public Const DIR_EAST = 3
Public Const DIR_SOUTH = 4
Public Const DIR_NW = 5
Public Const DIR_NE = 6
Public Const DIR_SE = 7
Public Const DIR_SW = 8

' Caption Alignment

Public Const ALIGN_LEFT = 0
Public Const ALIGN_CENTER = 1
Public Const ALIGN_RIGHT = 2

Public Type LOGFONT

Public Type LOCATION

Public Const ALIGN_RIGHT = 2
Public Const ALIGN_CENTER = 1
Public Const ALIGN_LEFT = 0

' Caption Alignment

Public Const DIR_SW = 8

Public Const DIR_SE = 7

Public Const DIR_NE = 6

Public Const DIR_NW = 5

Public Const DIR_SOUTH = 4

Public Const DIR_EAST = 3

Public Const DIR_NORTH = 2

Public Const DIR_WEST = 1

Public Const DIR_ALL = 0

' Orientations

Public Const SHADOW_BLURRED = 3

Public Const SHADOW_HAIFBLURRED = 2

Public Const SHADOW_SOLID = 1

Public Const SHADOW_NONE = 0

' Shadow modes

Public Const PILLOW_VERYHIGH = 4

Public Const PILLOW_HIGH = 3

Public Const PILLOW_MEDIUM = 2

Public Const PILLOW_LOW = 1

Public Const PILLOW_VERYLOW = 0

' Pillow factors

Public Const EXPORT_FORMAT_GIF = 4

Public Const EXPORT_FORMAT_CLIPBOARD = 3

Public Const EXPORT_FORMAT_PNG = 2

Public Const EXPORT_FORMAT_PIC = 1

Public Const EXPORT_FORMAT_BMP = 0

' Export images formats

Public Const EXPORT_IMAGE_DISABLED = 4

Public Const EXPORT_IMAGE_FOCUS = 3

Public Const EXPORT_IMAGE_PRESSED = 2

Public Const EXPORT_IMAGE_MOUSEOVER = 1

Public Const EXPORT_IMAGE_NORMAL = 0

' Export images types

Public Const TEXTURE_INSIDE = 3

Public Const TEXTURE_OUTSIDE = 2

Public Const TEXTURE_WHOLE = 1

Public Const TEXTURE_NONE = 0

' Texture modes

```
IfHeight As Long
IfWidth As Long
IfEscapement As Long
IfOrientation As Long
IfWeight As Long
IfItalic As Byte
IfUnderline As Byte
IfStrikeOut As Byte
IfCharSet As Byte
IfOutPrecision As Byte
IfClipPrecision As Byte
IfQuality As Byte
IfPitchAndFamily As Byte
IfFaceName As String * 32
End Type
```

```
Public Type TEXT_DESC
    colorNormal As Long
    colorMouseOver As Long
    colorPressed As Long
    xOffset As Integer
    yOffset As Integer
    iEffect As Integer
    nFactor As Integer
    nAngle As Integer
    colorSolidBack As Long
    colorBorder As Long
    nTransparentFactor As Integer
    bAntialias As Long
    bSolidBack As Long
    bVisible As Long
    bBorder As Long
    bDoubleSpace As Long
    bSolidBackShadow As Long
    colorHighlight As Long
    colorShadow As Long
    colorHalo As Long
    colorSolidBackShadow As Long
End Type
```

```
Public Type SURFACE_COLOR_DESC
    Color As Long
    nGradientFactor As Integer
    nRender3D As Integer
    nGradientType As Integer
End Type
```

' Generic operations

```
Declare Sub abmUseUnicodeStrings Lib "3dbapi.dll" (ByVal bUseUnicode As Integer)
```

```
Declare Sub abmClearButtoncache Lib "3dbapi.dll" ()
```

' Container Level Operations


```

Declare Function abmContainerSubclass Lib "3dbapi.dll" (ByVal hwndContainer As Long) As Integer
Declare Function abmContainerUnsubclass Lib "3dbapi.dll" (ByVal hwndContainer As Long) As Integer
Declare Function abmContainerSetBuggy Lib "3dbapi.dll" (ByVal hwndContainer As Long) As Integer
Declare Function abmContainerSetAutoRefresh Lib "3dbapi.dll" (ByVal hwndContainer As Long,
    ByVal bAuto As Integer) As Integer
Declare Function abmContainerSetting Lib "3dbapi.dll" (ByVal hwndContainer As Long, _
    ByVal nSetting As Integer, ByVal newValue As Long) As Integer
Declare Function abmContainerTextSetting Lib "3dbapi.dll" (ByVal hwndContainer As Long, _
    ByVal nPosition As Integer, descText As TEXT_DESC, ByVal nFlags As Integer) As Integer
Declare Function abmContainerFontSetting Lib "3dbapi.dll" (ByVal hwndContainer As Long, _
    ByVal nPosition As Integer, If As LOGFONT) As Integer
Declare Function abmContainerColorSetting Lib "3dbapi.dll" (ByVal hwndContainer As Long, _
    ByVal nElement As Integer, scd As SURFACE_COLOR_DESC) As Integer

' Button Level Operations
Declare Function abmButtonSubclassEx Lib "3dbapi.dll" (ByVal hwndContainer As Long, _
    ByVal hwnd As Long, ByVal style As Integer) As Integer
Declare Function abmButtonRender Lib "3dbapi.dll" (ByVal hwnd As Integer, _
    ByVal hDC As Integer) As Integer
Declare Function abmSetTextVisible Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nText As Long, ByVal bVisible As Integer)
Declare Sub abmSetTextOnPos Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nPosition As Integer, ByVal strText As String)
Declare Function abmGetTextOnPos Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nPosition As Integer) As String
Declare Sub abmSetTextSettingOnPos Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal pos As Integer, desc As TEXT_DESC, ByVal flags As Integer)
Declare Function abmGetTextSettingOnPos Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal pos As Integer, desc As TEXT_DESC) As Integer
Declare Sub abmEnableSound Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nSound As Integer, ByVal bEnable As Integer)
Declare Sub abmSetPictureFromHandle Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nPicture As Long, ByVal hBmp As Long)
Declare Function abmSetPictureFromMemory Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nPicture As Long, pData As Any, ByVal sizeData As Long) As Long
Declare Function abmSetAnimFromMemory Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nAnim As Long, pData As Any, ByVal sizeData As Long) As Long
Declare Function abmSetCursorFromHandle Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal hCursor As Long)
Declare Function abmSetSoundFromMemory Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nSound As Long, pData As Any, ByVal sizeData As Long) As Long
Declare Sub abmSetSoundSync Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nSound As Long, ByVal bSync As Integer)
Declare Function abmSetPictureFromNet Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nPicture As Long, ByVal strAddress As String, ByVal strUsername As String, _
    ByVal strPassword As String) As Long
Declare Function abmSetSoundFromNet Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nSound As Long, ByVal strAddress As String, ByVal strUsername As String, _

```



```

Declare Function abmContainerSetAutoRefresh Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
Declare Function abmContainerUnsubscribe Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
Declare Function abmContainerSubscribe Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
Declare Function abmContainerSetAutoRefresh Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
...
ByVal hwnd As Integer) As Integer
Declare Function abmContainerSetting Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
ByVal hwnd As Integer, ByVal newValue As Long) As Integer
Declare Function abmContainerSetting Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
ByVal position As Integer, desc As TEXT_DESC, ByVal flags As Integer) As Integer
Declare Function abmContainerSetting Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
ByVal position As Integer, h As LOGFONT) As Integer
Declare Function abmContainerColorSetting Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
ByVal element As Integer, set As SURFACE_COLOR_DESC) As Integer

' Button Level Operations
Declare Function abmButtonSubscribe Lib "abapi.dll" (ByVal hwndContainer As Long) As Integer
ByVal hwnd As Long, ByVal style As Integer) As Integer
Declare Function abmButtonRender Lib "abapi.dll" (ByVal hwnd As Integer) As Integer
ByVal hDC As Integer) As Integer
Declare Function abmSetVisible Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal text As Long, ByVal visible As Integer)
Declare Sub abmSetText Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal position As Integer, ByVal text As String)
Declare Function abmSetText Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal position As Integer) As String
Declare Sub abmSetText Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal pos As Integer, desc As TEXT_DESC, ByVal flags As Integer)
Declare Function abmSetText Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal pos As Integer, desc As TEXT_DESC) As Integer
Declare Sub abmEnable Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal hwnd As Integer, ByVal enable As Integer)
Declare Sub abmPictureFromHandle Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal picture As Long, ByVal hBmp As Long)
Declare Function abmPictureFromMemory Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal picture As Long, pData As Any, ByVal sizeData As Long) As Long
Declare Function abmPictureFromMemory Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal name As Long, pData As Any, ByVal sizeData As Long) As Long
Declare Function abmPictureFromHandle Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal picture As Long) As Integer
Declare Function abmPictureFromMemory Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal picture As Long, ByVal sizeData As Long) As Long
Declare Sub abmSound Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal sound As Long, pData As Any, ByVal sizeData As Long) As Long
Declare Sub abmSound Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal sound As Long) As Integer
Declare Function abmPictureCommit Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal picture As Long, ByVal address As String, ByVal streamName As String) As Integer
ByVal password As String) As Long
Declare Function abmSoundFromNet Lib "abapi.dll" (ByVal hwnd As Long) As Integer
ByVal sound As Long, ByVal address As String, ByVal streamName As String) As Integer

```

```

ByVal strPassword As String) As Long
Declare Function abmSetAnimFromNet Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nAnim As Long, ByVal strAddress As String, ByVal strUsername As String, _
    ByVal strPassword As String) As Long
Declare Sub abmSetSurfaceColorDesc Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nElement As Integer, desc As SURFACE_COLOR_DESC)
Declare Sub abmGetSurfaceColorDesc Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nElement As Integer, desc As SURFACE_COLOR_DESC)
Declare Function abmExportBtnImage Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal nImage As Integer, ByVal strFilename As String, ByVal nFormat As Long)
Declare Function abmLoadTemplate Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal strFilename As String) As Integer
Declare Function abmSetTooltipTexts Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal strNormal As String, ByVal strChecked As String) As Integer
Declare Sub abmGetFontText Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal pos As Integer, lpLogFont As LOGFONT)
Declare Sub abmSetFontText Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal pos As Integer, lpLogFont As LOGFONT)
Declare Sub abmAboutBox Lib "3dbapi.dll" ()

```

' Properties

```

Declare Function abmGetCaption Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetCaption Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal lpszCaption As String)
Declare Function abmGetShape Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetShape Lib "3dbapi.dll" (ByVal hwnd As Long, ByVal shape As Integer)
Declare Function abmGetRoundness Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetRoundness Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal roundness As Integer)
Declare Function abmGetCornerFactor Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetCornerFactor Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal factor As Integer)
Declare Function abmGetSlant Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetSlant Lib "3dbapi.dll" (ByVal hwnd As Long, ByVal slant As Integer)
Declare Function abmGetSurface Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetSurface Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal surface As Integer)
Declare Function abmGetPictureTranspColor Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
Declare Sub abmSetPictureTranspColor Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal Color As Long)
Declare Function abmGetSoundDown Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetSoundDown Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal lpszSound As String)
Declare Function abmGetSoundUp Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetSoundUp Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal lpszSound As String)
Declare Function abmGetCursor Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetCursor Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal lpszCursor As String)
Declare Function abmGetSoundMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetSoundMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long, _
    ByVal lpszSound As String)

```


Declare Function abmGetSoundMouseOut Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetSoundMouseOut Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszSound As String)
Declare Function abmGetPicture Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetPicture Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszPicture As String)
Declare Function abmGetPicturePressed Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetPicturePressed Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszPicture As String)
Declare Function abmGetPictureMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetPictureMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszPicture As String)
Declare Function abmGetPictureCustom Lib "3dbapi.dll" (ByVal hwnd As Long) As String
Declare Sub abmSetPictureCustom Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszPicture As String)
Declare Function abmGetPicturePosition Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetPicturePosition Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal pos As Integer)
Declare Function abmGetPictureXOffset Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetPictureXOffset Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal x As Integer)
Declare Function abmGetPictureYOffset Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetPictureYOffset Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal y As Integer)
Declare Function abmGetEnabled Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetEnabled Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal bEnabled As Integer)
Declare Function abmGetPictureTranspMode Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetPictureTranspMode Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal mode As Integer)
Declare Function abmGetButtonCustomRaised Lib "3dbapi.dll" (ByVal hwnd As Long) As
Integer
Declare Sub abmSetButtonCustomRaised Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal bRaised As Integer)
Declare Function abmGetBackColorContainer Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
Declare Sub abmSetBackColorContainer Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
Declare Function abmGetMovePictureOnPress Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetMovePictureOnPress Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal bMove As Integer)
Declare Function abmGetButtonRaiseFactor Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
Declare Sub abmSetButtonRaiseFactor Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal factor As Integer)
Declare Function abmGetHighlightColor Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
Declare Sub abmSetHighlightColor Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
Declare Function abmGetShadowColor Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
Declare Sub abmSetShadowColor Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
Declare Function abmGetButtonCustomFitPicture Lib "3dbapi.dll" (ByVal hwnd As Long) As
Integer
Declare Sub abmSetButtonCustomFitPicture Lib "3dbapi.dll" (ByVal hwnd As Long, _

ByVal bFit As Integer)
 Declare Function abmGetHyperlink Lib "3dbapi.dll" (ByVal hwnd As Long) As String
 Declare Sub abmSetHyperlink Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszHyperlink As String)
 Declare Function abmGetSmoothEdges Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetSmoothEdges Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal smooth As Integer)
 Declare Function abmGetPictureKeepRatio Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetPictureKeepRatio Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal bKeepRatio As Integer)
 Declare Function abmGetFocusMode Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetFocusMode Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal mode As Integer)
 Declare Function abmGetSpecialEffect Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetSpecialEffect Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal effect As Integer)
 Declare Function abmGetCaptionWordWrapPerc Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetCaptionWordWrapPerc Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal perc As Integer)
 Declare Function abmGetClickable Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetClickable Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal bClickable As Integer)
 Declare Function abmGetBackColor Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetBackColor Lib "3dbapi.dll" (ByVal hwnd As Long, ByVal Color As Long)
 Declare Function abmGetForeColor Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetForeColor Lib "3dbapi.dll" (ByVal hwnd As Long, ByVal Color As Long)
 Declare Function abmGetSpecialEffectFactor Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetSpecialEffectFactor Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal factor As Integer)
 Declare Function abmGetTexture Lib "3dbapi.dll" (ByVal hwnd As Long) As String
 Declare Sub abmSetTexture Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal lpszTexture As String)
 Declare Function abmGetTextureMode Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetTextureMode Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal mode As Integer)
 Declare Function abmGetTextureLightIntensity Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetTextureLightIntensity Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal light As Integer)
 Declare Function abmGetTextureColorMix Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetTextureColorMix Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetTextureColorMixPerc Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetTextureColorMixPerc Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal perc As Integer)
 Declare Function abmGetTextureBevelFactor Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetTextureBevelFactor Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal factor As Integer)
 Declare Function abmGetForeColorDisabled Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetForeColorDisabled Lib "3dbapi.dll" (ByVal hwnd As Long, _

ByVal Color As Long)
 Declare Function abmGetBackColorDisabled Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetBackColorDisabled Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetBackColorPressed Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetBackColorPressed Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetBackColorMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetBackColorMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetBackColorFocus Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetBackColorFocus Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetBackColorInterior Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetBackColorInterior Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetForeColorMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetForeColorMouseOver Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetFlatPillowFactor Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetFlatPillowFactor Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal factor As Integer)
 Declare Function abmGetShadowMode Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetShadowMode Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal mode As Integer)
 Declare Function abmGetForeColorPressed Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetForeColorPressed Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal Color As Long)
 Declare Function abmGetValue Lib "3dbapi.dll" (ByVal hwnd As Long) As Long
 Declare Sub abmSetValue Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal value As Long)
 Declare Function abmGetPictureVisible Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetPictureVisible Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal bVisible As Integer)
 Declare Function abmGetOrientation Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetOrientation Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal nOrientation As Integer)
 Declare Function abmGetSpotlightOffsetX Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetSpotlightOffsetX Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal nOffset As Integer)
 Declare Function abmGetSpotlightOffsetY Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer
 Declare Sub abmSetSpotlightOffsetY Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal nOffset As Integer)
 Declare Function abmGetSpotlightResizeWidth Lib "3dbapi.dll" (ByVal hwnd As Long) As
 Integer
 Declare Sub abmSetSpotlightResizeWidth Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal nWidth As Integer)
 Declare Function abmGetSpotlightResizeHeight Lib "3dbapi.dll" (ByVal hwnd As Long) As
 Integer
 Declare Sub abmSetSpotlightResizeHeight Lib "3dbapi.dll" (ByVal hwnd As Long, _
 ByVal nHeight As Integer)
 Declare Function abmGetUseAntialiasing Lib "3dbapi.dll" (ByVal hwnd As Long) As Integer


```

Declare Function amdGpuBackColorDisabled Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorDisabled Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuBackColorRefresh Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorRefresh Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuBackColorRefreshOver Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorRefreshOver Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuBackColorFocus Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorFocus Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuBackColorInterior Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorInterior Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuBackColorRefreshOver Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorRefreshOver Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuBackColorRefreshFactor Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBackColorRefreshFactor Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal factor As Integer)
Declare Function amdGpuBackColorRefreshMode Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBackColorRefreshMode Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal mode As Integer)
Declare Function amdGpuBackColorRefresh Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuBackColorRefresh Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Color As Long)
Declare Function amdGpuValue Lib "amdgpudll" (ByVal hwnd As Long) As Long
Declare Sub amdGpuValue Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal value As Long)
Declare Function amdGpuPictureVisible Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuPictureVisible Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Visible As Integer)
Declare Function amdGpuOrientation Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuOrientation Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Orientation As Integer)
Declare Function amdGpuBrightness Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBrightness Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Brightness As Integer)
Declare Function amdGpuBrightnessY Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBrightnessY Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Brightness As Integer)
Declare Function amdGpuBrightnessX Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBrightnessX Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Brightness As Integer)
Declare Function amdGpuBrightnessWith Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBrightnessWith Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Width As Integer)
Declare Function amdGpuBrightnessHeight Lib "amdgpudll" (ByVal hwnd As Long) As Integer
Declare Sub amdGpuBrightnessHeight Lib "amdgpudll" (ByVal hwnd As Long, ...
ByVal Height As Integer)
Declare Function amdGpuBrightnessing Lib "amdgpudll" (ByVal hwnd As Long) As Integer

```

```
Declare Sub abmSetUseAntialiasing Lib "3dbapi.dll" (ByVal hwnd As Long, _  
    ByVal bUseAntialias As Integer)
```

```
' WinAPI function declarations and constants
```

```
Private Declare Function GdiRectangle Lib "gdi32" Alias "Rectangle" (ByVal hdc As Long, ByVal  
X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
```

```
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
```

```
Private Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, ByVal hdc As Long) As  
Long
```

```
Private Declare Function SetROP2 Lib "gdi32" (ByVal hdc As Long, ByVal nDrawMode As  
Long) As Long
```

```
Private Const R2_NOTXORPEN = 10
```

```
' global map
```

```
Dim m_map As MapObjects2.Map
```

```
' variables that keep track of moving the indicator
```

```
Dim m_hDC As Long ' a DC to draw into
```

```
Dim m_hWnd As Long ' window handle
```

```
Dim m_xMin As Integer, m_yMin As Integer ' drag indicator
```

```
Dim m_xMax As Integer, m_yMax As Integer ' drag indicator
```

```
Dim m_xPrev As Integer ' click location
```

```
Dim m_yPrev As Integer ' click location
```

```
Function DragFinish(x As Single, y As Single) As MapObjects2.Rectangle
```

```
GdiRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
```

```
ReleaseDC m_hWnd, m_hDC
```

```
' return the rectangle
```

```
Dim r As New MapObjects2.Rectangle
```

```
PixelsRectToMap m_xMin, m_yMin, m_xMax, m_yMax, r
```

```
Set DragFinish = r
```

```
End Function
```

```
Sub DragMove(x As Single, y As Single)
```

```
' convert to pixels
```

```
xNext = m_map.Parent.ScaleX(x, vbTwips, vbPixels)
```

```
yNext = m_map.Parent.ScaleY(y, vbTwips, vbPixels)
```

```
GdiRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
```

```
m_xMin = m_xMin + (xNext - m_xPrev)
```

```
m_xMax = m_xMax + (xNext - m_xPrev)
```

```
m_yMin = m_yMin + (yNext - m_yPrev)
```

```
m_yMax = m_yMax + (yNext - m_yPrev)
```

```
GdiRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
```

```
m_xPrev = xNext
```

```
m_yPrev = yNext
```

```
End Sub
```

```

' WinAPI function declarations and constants
Private Declare Function GetRectangle Lib "GDI32" Alias "Rectangle" (ByVal hdc As Long, ByVal
    XI As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
Private Declare Function GetDC Lib "USER32" (ByVal hwnd As Long) As Long
Private Declare Function ReleaseDC Lib "USER32" (ByVal hwnd As Long, ByVal hdc As Long) As
    Long
Private Declare Function SetROP2 Lib "GDI32" (ByVal hdc As Long, ByVal nDrawMode As
    Long) As Long
Private Const R2_NOTXORPEN = 10

' global map
Dim m_map As MapObjects.Map

' variables that keep track of moving the indicator
Dim m_hDC As Long ' a DC to draw into
Dim m_hWnd As Long ' window handle
Dim m_xMin As Integer, m_yMin As Integer ' drag indicator
Dim m_xMax As Integer, m_yMax As Integer ' drag indicator
Dim m_xPrev As Integer ' click location
Dim m_yPrev As Integer ' click location

Function DragFinish(x As Single, y As Single) As MapObjects.Rectangle
    GetRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
    ReleaseDC m_hWnd, m_hDC

    ' return the rectangle
    Dim r As New MapObjects.Rectangle
    PixelRectToMap m_xMin, m_yMin, m_xMax, m_yMax, r
    Set DragFinish = r
End Function

Sub DragMove(x As Single, y As Single)
    ' convert to pixels
    xNext = m_map.ParentScaleX(x, vbTwips, vbPixels)
    yNext = m_map.ParentScaleY(y, vbTwips, vbPixels)

    GetRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
    m_xMin = m_xMin + (xNext - m_xPrev)
    m_xMax = m_xMax + (xNext - m_xPrev)
    m_yMin = m_yMin + (yNext - m_yPrev)
    m_yMax = m_yMax + (yNext - m_yPrev)

    GetRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
    m_xPrev = xNext
    m_yPrev = yNext
End Sub

```

Declare Sub absntUseAnimating Lib "GDI32" (ByVal hwnd As Long,
 ByVal bUseAnim As Integer)

```
Sub DragStart(rect As MapObjects2.Rectangle, Map As MapObjects2.Map, x As Single, y As Single)
```

```
Set m_map = Map  
' initialize the hwnd and hdc variables  
m_hWnd = m_map.hwnd  
m_hDC = GetDC(m_hWnd)  
SetROP2 m_hDC, R2_NOTXORPEN ' raster op for inverting
```

```
MapRectToPixels rect, m_xMin, m_yMin, m_xMax, m_yMax
```

```
' draw the rectangle  
GdiRectangle m_hDC, m_xMin, m_yMin, m_xMax, m_yMax
```

```
' remember the click position  
' convert to pixels  
m_xPrev = m_map.Parent.ScaleX(x, vbTwips, vbPixels)  
m_yPrev = m_map.Parent.ScaleY(y, vbTwips, vbPixels)  
End Sub
```

```
Private Sub MapRectToPixels(r As MapObjects2.Rectangle, xMin As Integer, yMin As Integer, xMax As Integer, yMax As Integer)
```

```
Dim p As New Point  
Dim xc As Single, yc As Single
```

```
p.x = r.Left  
p.y = r.Top  
m_map.FromMapPoint p, xc, yc
```

```
' convert to pixels  
xMin = m_map.Parent.ScaleX(xc, vbTwips, vbPixels)  
yMin = m_map.Parent.ScaleY(yc, vbTwips, vbPixels)
```

```
p.x = r.Right  
p.y = r.Bottom  
m_map.FromMapPoint p, xc, yc
```

```
' convert to pixels  
xMax = m_map.Parent.ScaleX(xc, vbTwips, vbPixels)  
yMax = m_map.Parent.ScaleY(yc, vbTwips, vbPixels)  
End Sub
```

```
Sub PixelsRectToMap(xMin As Integer, yMin As Integer, xMax As Integer, yMax As Integer, r As MapObjects2.Rectangle)
```

```
Dim xc As Single, yc As Single
```

```
' convert to twips  
xc = m_map.Parent.ScaleX(xMin, vbPixels, vbTwips)  
yc = m_map.Parent.ScaleY(yMin, vbPixels, vbTwips)
```

```
Set p = m_map.ToMapPoint(xc, yc)  
r.Left = p.x  
r.Top = p.y
```

```

Sub DrawStart( rect As MapObjects.Rectangle, Map As MapObjects.Map, x As Single, y As Single)
    Set m_map = Map
    ' initialize the bound and hitc variables
    m_hitnd = m_map.hitnd
    m_hitc = GetDC(m_hitnd)
    SetROP2 m_hitc, RS_NOTXORPEN ' raster op for inverting

    MapRectToPixels rect, m_xmin, m_xmax, m_ymin, m_ymax

    ' draw the rectangle
    GDIRectangle m_hitc, m_xmin, m_xmax, m_ymin, m_ymax

    ' remember the click position
    ' convert to pixels
    m_xprev = m_map.Parent.ScaleX(x, vbTwips, vbPixels)
    m_yprev = m_map.Parent.ScaleY(y, vbTwips, vbPixels)
End Sub

Private Sub MapRectToPixels( r As MapObjects.Rectangle, xmin As Integer, xmax As Integer, ymin As Integer, ymax As Integer)
    Dim p As New Point
    Dim xc As Single, yc As Single

    p.x = r.Left
    p.y = r.Top
    m_map.FromMapPoint p, xc, yc

    ' convert to pixels
    xmin = m_map.Parent.ScaleX(xc, vbTwips, vbPixels)
    ymin = m_map.Parent.ScaleY(yc, vbTwips, vbPixels)

    p.x = r.Right
    p.y = r.Bottom
    m_map.FromMapPoint p, xc, yc

    ' convert to pixels
    xmax = m_map.Parent.ScaleX(xc, vbTwips, vbPixels)
    ymax = m_map.Parent.ScaleY(yc, vbTwips, vbPixels)
End Sub

Sub PixelsRectToMap(xmin As Integer, xmax As Integer, ymin As Integer, ymax As Integer, r As MapObjects.Rectangle)
    Dim xc As Single, yc As Single

    ' convert to twips
    xc = m_map.Parent.ScaleX(xmin, vbPixels, vbTwips)
    yc = m_map.Parent.ScaleY(ymin, vbPixels, vbTwips)

    Set p = m_map.ToMapPoint(xc, yc)
    r.Left = p.x
    r.Top = p.y

```

```
' convert to twips
xc = m_map.Parent.ScaleX(xMax, vbPixels, vbTwips)
yc = m_map.Parent.ScaleY(yMax, vbPixels, vbTwips)
Set p = m_map.ToMapPoint(xc, yc)
r.Right = p.x
r.Bottom = p.y
End Sub
```

```
Public PolyColor As Long
Dim m_grid As SnappingGrid
Dim m_map As MapObjects2.Map
```

```
' polygon array
Dim m_polys() As MapObjects2.Polygon
```

```
' selection
Dim m_selPoly As Integer
Dim m_selVertex As Integer
Dim Tol As Double
```

```
Sub DeleteSelection()
' delete the selected vertex, if any
If m_selVertex >= 0 Then
Set pts = m_polys(m_selPoly).Parts(0)
If pts.Count > 4 Then
pts.Remove m_selVertex
m_selVertex = m_selVertex - 1
Else
m_selVertex = -1
End If
```

```
Refresh
Exit Sub
End If
```

```
' delete selected polygon
If m_selPoly >= 0 Then
Set m_polys(m_selPoly) = Nothing
numPolys = UBound(m_polys) - 1
Set Last = m_polys(numPolys)
ReDim Preserve m_polys(numPolys)
If Not Last Is Nothing Then
Set m_polys(m_selPoly) = Last
End If
```

```
If m_selPoly >= numPolys Then ' select the next
m_selPoly = m_selPoly - 1
End If
```

```
Refresh
End If
```

```
End Sub
r.Bottom = p.y
r.Right = p.x
Set p = m_map.ToMapPoint(xc, yc)
yc = m_map.ParentScaleY(Max vbPixels, vbTwips)
xc = m_map.ParentScaleX(Max vbPixels, vbTwips)
' convert to twips
```

```
Dim m_map As MapObjects2.Map
Dim m_grid As snappingGrid
Public PolyColor As Long
```

```
Dim m_poly() As MapObjects2.Polygon
' polygon array
```

```
Dim Tol As Double
Dim m_selVertex As Integer
Dim m_selfPoly As Integer
' selection
```

```
End If
m_selVertex = -1
Else
m_selVertex = m_selVertex - 1
pts.Remove m_selVertex
If pts.Count > 4 Then
Set pts = m_poly(m_selfPoly).Parts()
Set pts = m_poly(m_selfPoly).Parts()
If m_selVertex >= 0 Then
' delete the selected vertex, if any
Sub Deselection()
Dim Tol As Double
Dim m_selVertex As Integer
Dim m_selfPoly As Integer
' selection
```

```
End If
Refresh
Exit Sub
End If
```

```
End If
Set m_poly(m_selfPoly) = Last
If Not Last Is Nothing Then
ReDim Preserve m_poly(numPolys)
Set Last = m_poly(numPolys)
numPolys = UBound(m_poly) + 1
Set m_poly(m_selfPoly) = Nothing
If m_selfPoly >= 0 Then
' delete selected polygon
```

```
End If
m_selfPoly = m_selfPoly - 1
If m_selfPoly >= numPolys Then ' select the next
```

```
End If
Return
```

End Sub

Sub Draw()

```
Dim sym As New MapObjects2.Symbol
sym.Color = PolyColor
sym.style = moGrayFill
For i = 0 To UBound(m_polys) - 1
    m_map.DrawShape m_polys(i), sym
Next i
```

' draw the selection

```
sym.SymbolType = moPointSymbol
sym.Size = 7
sym.Color = m_grid.Color
```

If m_selPoly >= 0 Then

```
Set pts = m_polys(m_selPoly).Parts(0)
```

For Each p In pts

```
    m_map.DrawShape p, sym
```

Next p

```
sym.Color = moYellow
```

```
If m_selVertex >= 0 Then m_map.DrawShape pts(m_selVertex), sym
```

End If

End Sub

Sub ExportToShapefile(pathName As String)

```
Dim filename As String, dirName As String
```

```
'SplitPath pathName, dirName, fileName
```

```
Dim dc As New MapObjects2.DataConnection
```

```
dc.Database = dirName
```

```
If Not dc.connect Then Exit Sub 'bogus dataConnection
```

```
Dim tDesc As New MapObjects2.TableDesc
```

```
Dim gs As GeoDataset
```

```
Set gs = dc.AddGeoDataset(filename, moPolygon, tDesc)
```

```
If gs Is Nothing Then Exit Sub 'invalid file
```

```
Dim layer As New MapObjects2.MapLayer
```

```
Set layer.GeoDataset = gs
```

```
Set Recs = layer.Records
```

```
Set shpFld = Recs.Fields("Shape")
```

' write out the polygons

```
For i = 0 To UBound(m_polys) - 1
```

```
    Recs.AddNew
```

```
    shpFld.value = m_polys(i)
```

```
    Recs.Update
```

Next i

End Sub

Sub Initialize(Map As MapObjects2.Map, grid As SnappingGrid)

```
Set m_map = Map
```



```

Set m_map = Map
Sub Initialize(Map As MapObject2.Map And As SnappingGrid)
    End Sub

    Next i
    Recs.Update
    shpFld.Value = m_poly(i)
    Recs.AddNew
    For i = 0 To Ubound(m_poly) - 1
        ' write out the polygons
    Set shpFld = Recs.Fields("Shape")
    Set Recs = Layer.Records
    Set Layer.Geodataset = gs
    Dim Layer As New MapObject2.MapLayer
    If gs Is Nothing Then Exit Sub ' invalid file
    Set gs = dc.AddGeodataset(filename, mofPolygon, Hsec)
    Dim gs As Geodataset
    Dim Hsec As MapObject2.TableDesc
    If Not dc.Connect Then Exit Sub ' bogus data connection
    dc.Database = filename
    Dim dc As New MapObject2.DataConnection
    SplitPath pathName, filename, filename
    Dim filename As String, dirName As String
    Sub ExportShapefile(pathName As String)
    End Sub

    End Sub

    If m_selfIndex >= 0 Then m_map.DrawShape pts(m_selfIndex) sym
    sym.Color = mofYellow
    Next p
    m_map.DrawShape p, sym
    For Each p In pts
        Set pts = m_poly(m_selfIndex).Parts(0)
    If m_selfPoly >= 0 Then
        sym.Color = m_grid.Color
        sym.Size = 7
        sym.SymbolType = mofPointSymbol
        ' draw the selection
    Next i
    m_map.DrawShape m_poly(i), sym
    For i = 0 To Ubound(m_poly) - 1
        sym.Style = mofGrayFill
        sym.Color = PolyColor
    Dim sym As New MapObject2.Symbol
    Sub Draw()
    End Sub
End Sub

```

```
Set m_grid = grid
End Sub
```

```
Sub Refresh()
m_map.TrackingLayer.Refresh True
End Sub
```

```
Function SelectVertex(pt As MapObjects2.Point) As Boolean
Tol = m_map.ToMapDistance(100)
If m_selPoly >= 0 Then
m_selVertex = -1
Set pts = m_polys(m_selPoly).Parts(0)
For Each v In pts
m_selVertex = m_selVertex + 1
If v.DistanceTo(pt) < Tol Then
Refresh
SelectVertex = True
Exit Function
End If
Next v
End If
```

```
SelectVertex = False
End Function
```

```
Sub SnapPolygons()
For i = 0 To UBound(m_polys) - 1
Set m_polys(i) = m_grid.SnapPolygon(m_polys(i), m_map)
Next i
Refresh
End Sub
```

```
Sub SplitPolygon(pt As MapObjects2.Point)
If m_selPoly = -1 Then Exit Sub
```

```
Dim pts As MapObjects2.Points
Set pts = m_polys(m_selPoly).Parts(0)
Tol = m_map.ToMapDistance(100)
```

```
pts.Add pts(0)
For i = 1 To pts.Count - 1
dist = pt.DistanceToSegment(pts(i - 1), pts(i))
If dist <= Tol Then
m_grid.SnapPoint pt, m_map
pts.Insert i, pt ' insert the point
m_selVertex = i ' select the vertex
Refresh
Exit Sub
End If
Next i
End Sub
```

```

End Sub
Next i
End If
Exit Sub
Return
m_SelectVertex = i, select the vertex
pt = next i in pt, next the point
m_MinDistancePoint = pt, m_min
If dist <= 101 Then
dist = ByDistanceToSegment(pt(i-1), pt(i))
For i = 1 To pt.Count - 1
pt = pt(0)

```

```

101 = m_min.Distance(100)
Set pt = m_pt(m_SelectVertex).Parent(0)
Dim pt As MapObject.Point

```

```

If m_SelectVertex = -1 Then Exit Sub
Sub SearchPoint(pt As MapObject.Point)

```

```

End Sub
Return
Next i
Set m_pt(i) = m_SearchPoint(m_pt(i), m_min)
For i = 0 To UBound(m_pt) - 1
Sub SearchPoint()

```

```

End Function
SelectVertex = False

```

```

End If
Next v
End If
Exit Function
SelectVertex = True
Return
If DistanceTo(pt) < 101 Then
m_SelectVertex = m_SelectVertex + 1
For Each v In pt
Set pt = m_pt(m_SelectVertex).Parent(0)
m_SelectVertex = -1
If m_SelectVertex >= 0 Then
101 = m_min.Distance(100)
Function SelectVertex(pt As MapObject.Point) As Boolean

```

```

End Sub
m_min.TrackByLayer: Return True
Sub Return()

```

```

End Sub
Set m_min = min

```

```

Function SelectPolygon(pt As MapObjects2.Point) As Integer
' returns 1 if a vertex is selected
' first see if a vertex is selected
If SelectVertex(pt) Then
    SelectPolygon = 1
    Exit Function
End If

' otherwise select a polygon
m_selPoly = -1
m_selVertex = -1
For i = 0 To UBound(m_polys) - 1
    If m_polys(i).IsPointIn(pt) Then
        m_selPoly = i
        Exit For
    End If
Next i

Refresh
SelectPolygon = 0
End Function

Sub MoveVertex(pt As MapObjects2.Point)
' moves the selected vertex
If m_selPoly >= 0 And m_selVertex >= 0 Then
    m_grid.SnapPoint pt, m_map
    Set Poly = m_polys(m_selPoly)
    Dim pts As Points
    Set pts = Poly.Parts(0)
    pts.Set m_selVertex, pt

    ' if it's the first vertex, then move the last too
    If m_selVertex = 0 Then pts.Set pts.Count - 1, pt
    ' if it's the last vertex, then move the first too
    If m_selVertex = pts.Count - 1 Then pts.Set 0, pt
    Refresh
End If
End Sub

Sub AddPolygon()
Dim Poly As MapObjects2.Polygon
Set Poly = m_map.TrackPolygon

' snap the polygon and add it to the array
If Not Poly Is Nothing Then
    numPolys = UBound(m_polys)
    ReDim Preserve m_polys(numPolys + 1)
    Set m_polys(numPolys) = m_grid.SnapPolygon(Poly, m_map)

' select the new poly
m_selPoly = numPolys
m_selVertex = -1

```

```
Refresh
End If
End Sub
```

```
Function VertexHandle() As MapObjects2.Rectangle
If m_selVertex = -1 Then Exit Function
```

```
Set pt = m_polys(m_selPoly).Parts(0).Item(m_selVertex)
r = m_map.ToMapDistance(60)
Dim rect As New MapObjects2.Rectangle
rect.Left = pt.x - r
rect.Right = pt.x + r
rect.Top = pt.y + r
rect.Bottom = pt.y - r
```

```
Set VertexHandle = rect
End Function
```

```
Private Sub Class_Initialize()
ReDim m_polys(0)
PolyColor = moCyan
m_selPoly = -1
m_selVertex = -1
End Sub
```

```
Private Declare Function SetPixel Lib "gdi32" (ByVal hDC As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
```

```
Public Spacing As Double ' spacing between grid points
Public Color As Long ' color of grid points
```

```
Sub Draw(Map As MapObjects2.Map, hDC As Long)
If Spacing = 0 Then Exit Sub
```

```
' if the grid points are going to be to dense, just bail
spacingC = Map.FromMapDistance(Spacing) ' control
spacingP = Map.Parent.ScaleX(spacingC, vbTwips, vbPixels) ' pixels
```

```
If spacingP < 4 Then Exit Sub
```

```
Set ext = Map.Extent
Set fext = Map.FullExtent
```

```
Dim xCount As Integer, yCount As Integer
Dim xFirst As Integer, yFirst As Integer
```

```
' calculate the index of the first visible grid point
xFirst = (ext.Left - fext.Left) / Spacing
yFirst = (ext.Bottom - fext.Bottom) / Spacing
```

```
' calculate the number of visible grid points
```

```
xCount = ext.Width / Spacing
yCount = ext.Height / Spacing
```

```
' calculate the map coordinates of the first visible
' grid point (lower left) and the last visible grid point
' (upper right)
```

```
Dim p1 As New MapObjects2.Point
Dim p2 As New MapObjects2.Point
p1.x = fext.Left + (Spacing * xFirst)
p1.y = fext.Bottom + (Spacing * yFirst)
p2.x = fext.Left + (Spacing * (xFirst + xCount))
p2.y = fext.Bottom + (Spacing * (yFirst + yCount))
```

```
' convert the first and last visible grid point
' into control coordinates, then pixels
```

```
Dim xc1 As Single, xc2 As Single, yc1 As Single, yc2 As Single
Dim xp1 As Integer, xp2 As Integer, yp1 As Integer, yp2 As Integer
Map.FromMapPoint p1, xc1, yc1
Map.FromMapPoint p2, xc2, yc2
```

```
xp1 = Map.Parent.ScaleX(xc1, vbTwips, vbPixels) ' pixels
yp1 = Map.Parent.ScaleY(yc1, vbTwips, vbPixels) ' pixels
xp2 = Map.Parent.ScaleX(xc2, vbTwips, vbPixels) ' pixels
yp2 = Map.Parent.ScaleY(yc2, vbTwips, vbPixels) ' pixels
```

```
' calculate the distance between grid points in pixels
```

```
Dim xFact As Double, yFact As Double
xFact = CDbl(xp2 - xp1) / xCount
yFact = CDbl(yp2 - yp1) / yCount
```

```
For x = 0 To xCount - 1
  For y = 0 To yCount - 1
    SetPixel hDC, xp1 + x * xFact, yp1 + y * yFact, Color
  Next y
Next x
End Sub
```

```
Sub SnapPoint(pt As MapObjects2.Point, Map As MapObjects2.Map)
  If Spacing = 0 Then Exit Sub
```

```
  Set fext = Map.FullExtent
```

```
' calculate the index of the grid point to the left
' and below
```

```
Dim xGrid As Integer, yGrid As Integer
xGrid = (pt.x - fext.Left) / Spacing
yGrid = (pt.y - fext.Bottom) / Spacing
```

```
' calculate the map coordinates of this grid point
```

```
Dim xM As Double, yM As Double
xM = fext.Left + Spacing * xGrid
yM = fext.Bottom + Spacing * yGrid
```

```
' if the distance between the point and the
' grid point is less then have the spacing size, then
' snap to the grid point, otherwise snap to the next one
If (pt.x - xM) < (Spacing / 2) Then
    pt.x = xM
Else
    pt.x = xM + Spacing
End If
```

```
If (pt.y - yM) < (Spacing / 2) Then
    pt.y = yM
Else
    pt.y = yM + Spacing
End If
End Sub
```

```
Function SnapPolygon(Poly As MapObjects2.Polygon, Map As MapObjects2.Map) As
MapObjects2.Polygon
    ' snap each point of the polygon
    Dim SnapPoly As New MapObjects2.Polygon
    Dim SnapPts As New MapObjects2.Points

    Dim pts As MapObjects2.Points
    Set pts = Poly.Parts(0)

    Dim p As MapObjects2.Point
    For Each p In pts
        SnapPoint p, Map
        SnapPts.Add p
    Next p

    SnapPoly.Parts.Add SnapPts
    Set SnapPolygon = SnapPoly
End Function
```

```
Private Sub Class_Initialize()
    Spacing = 0
End Sub
```

```
Option Explicit
Dim strMarkerStyle(4) As String
Dim strLineStyle(4) As String
Dim strFillStyle(10) As String
Dim strPanelDesc(5) As String
Public lyr As MapObjects2.MapLayer
Public Recs As MapObjects2.Recordset
Public tDesc As MapObjects2.TableDesc
Public flds As MapObjects2.Fields
Dim a As Integer
Dim vmr As MapObjects2.ValueMapRenderer
Dim cbr As MapObjects2.ClassBreaksRenderer
```

```
Dim lr As MapObjects2.LabelRenderer
Dim lp As MapObjects2.LabelPlacer
Dim zRend As MapObjects2.ZRenderer
Dim colorMask, colorText As Long
Dim justOpened As Boolean
```

```
Private Sub Form_Load()
'tampil didepan
SetWindowPos frmLayerSymbol.hwnd, conHwndTopmost, 0, 0, 0, 0, flags
```

```
Dim i As Integer
Dim fnt As New StdFont
```

```
'Position this form into the top right
'corner of the screen
Me.Top = 0
Me.Left = Screen.Width - Me.Width
```

```
colorText = moBlack
colorMask = moWhite
```

```
Set lyr = g_ActiveLayer
```

```
Set Recs = lyr.Records
Set tDesc = Recs.TableDesc
Set flds = Recs.Fields
```

```
frmLayerSymbol.Caption = "Symbol properties for the " & UCase(lyr.Name) & " layer."
strPanelDesc(0) = "The Single Symbol classification displays all the features in a layer with the
same symbol."
strPanelDesc(1) = "The Unique Values classification displays features by applying a symbol to
each unique value for a specified field."
strPanelDesc(2) = "The Class Breaks classification applies symbols to a set of discrete values."
strPanelDesc(3) = "The Standard Labels classification draws text for a specified field."
strPanelDesc(4) = "The No Overlapping Labels classification draws text for a specified field and
attempts to resolve overlapping and crowding of labels."
strPanelDesc(5) = "The Elevation classification draws features according to their Z values, if the
data supports it."
txtLayerName = lyr.Name
strMarkerStyle(0) = "Circle marker"
strMarkerStyle(1) = "Square marker"
strMarkerStyle(2) = "Triangle marker"
strMarkerStyle(3) = "Cross marker"
strMarkerStyle(4) = "TrueType marker"
strLineStyle(0) = "Solid line"
strLineStyle(1) = "Dash line"
strLineStyle(2) = "Dot line"
strLineStyle(3) = "Dash dot line"
strLineStyle(4) = "Dash dot dot line"
strFillStyle(0) = "Solid fill"
strFillStyle(1) = "Transparent fill"
```



```
strFillStyle(2) = "Horizontal fill"  
strFillStyle(3) = "Vertical fill"  
strFillStyle(4) = "Upward diagonal"  
strFillStyle(5) = "Downward diagonal"  
strFillStyle(6) = "Cross fill"  
strFillStyle(7) = "Diagonal cross fill"  
strFillStyle(8) = "Light gray fill"  
strFillStyle(9) = "Gray fill"  
strFillStyle(10) = "Dark gray fill"
```

```
justOpened = True
```

```
'Read the contents of the active layer's Renderer property.  
'Run one of the "Load..." procedures to populate the appropriate  
'tab with the current renderer's properties.
```

```
Select Case True
```

```
Case lyr.Renderer Is Nothing
```

```
    sstLayerProp.Tab = 0
```

```
    Call LoadSingleSymbol
```

```
Case TypeOf lyr.Renderer Is MapObjects2.ValueMapRenderer
```

```
    sstLayerProp.Tab = 1
```

```
    Call LoadUniqueValues
```

```
Case TypeOf lyr.Renderer Is MapObjects2.ClassBreaksRenderer
```

```
    sstLayerProp.Tab = 2
```

```
    Call LoadClassBreaks
```

```
Case TypeOf lyr.Renderer Is MapObjects2.LabelRenderer
```

```
    sstLayerProp.Tab = 3
```

```
    Call LoadStandardLabels
```

```
Case TypeOf lyr.Renderer Is MapObjects2.LabelPlacer
```

```
    sstLayerProp.Tab = 4
```

```
    colorMask = lyr.Renderer.MaskColor
```

```
    Call LoadNoOverlapLabels
```

```
Case TypeOf lyr.Renderer Is MapObjects2.ZRenderer
```

```
    sstLayerProp.Tab = 5
```

```
    Call LoadZRenderer
```

```
Case Else
```

```
    sstLayerProp.Tab = 0
```

```
    Call LoadSingleSymbol
```

```
End Select
```

```
'If the active layer does not support Z shapes, disable  
'the "Elevation" rendering tab.
```

```
If (InStr(lyr.Tag, "[SHAPEFILZ]") = 0) And (InStr(lyr.Tag, "[SDEZ]") = 0) Then
```

```
    sstLayerProp.TabEnabled(5) = False
```

```
End If
```

```
End Sub
```

```
Private Sub cmdApply_Click()
```

```
'When the user hits the Apply button (or the OK button)
```

```
'read the option controls on the active tab, and use
```

```
'them to build a new renderer with which to draw the layer.
Select Case sstLayerProp.Tab
  Case 0: Call ApplySingleSymbol
  Case 1: Call ApplyUniqueValues
  Case 2: Call ApplyClassBreaks
  Case 3: Call ApplyStandardLabels
  Case 4: Call ApplyNoOverlapLabels
  Case 5: Call ApplyZRenderer
End Select
```

```
'Rename the layer with the contents of txtLayerName.Text
lyr.Name = txtLayerName.Text
End Sub
```

```
Private Sub cmdCancel_Click()
```

```
'Throw away all work on this form.
'Layer retains original rendering information.
Unload frmLayerSymbol
```

```
End Sub
```

```
Private Sub cmdOK_Click()
```

```
'Use Apply, then unload
Call cmdApply_Click
Unload frmLayerSymbol
```

```
End Sub
```

```
Private Sub cmdNOL_Click()
```

```
'Choose font properties for the LabelPlacer
cdlgLayerProp.Color = colorText
cdlgLayerProp.flags = cdICFEffects Or cdICFBoth
cdlgLayerProp.ShowFont
txtNOL.ForeColor = cdlgLayerProp.Color
txtNOL.Text = cdlgLayerProp.FontName
colorText = cdlgLayerProp.Color
```

```
End Sub
```

```
Private Sub cmdSL_Click()
```

```
'Choose font properties for the LabelRenderer
cdlgLayerProp.flags = cdICFEffects Or cdICFBoth
cdlgLayerProp.ShowFont
txtSL.ForeColor = cdlgLayerProp.Color
txtSL.Text = cdlgLayerProp.FontName
```

```
End Sub
```

```
Private Sub cmdCB_Click()
```

```
'Build a new tentative class breaks legend
Call PopulateNewCBlegend(cboCB(0).Text)
cmdApply.Enabled = True
cmdOK.Enabled = True
```

```
End Sub
```

```
Private Sub cmdUV_Click()
```

```
'Build a new tentative unique values legend
Call PopulateNewUVlegend(cboUV.Text)
cmdApply.Enabled = True
cmdOK.Enabled = True
```

```
End Sub
```

```
Private Sub cmdZR_Click()
```

```
'Build a new tentative Z elevation breaks legend
Call PopulateNewZRLegend
cmdApply.Enabled = True
cmdOK.Enabled = True
```

```
End Sub
```

```
Private Sub cboSSP_Click(index As Integer)
```

```
Dim fnt As New StdFont
```

```
Select Case index
```

```
Case 0 'choose single symbol style
```

```
  If index = 0 Then
```

```
    If cboSSP(0).Text = "TrueType marker" Then
```

```
      Dim i As Integer
```

```
      cboSSP(1).Enabled = True
```

```
      cboSSP(2).Enabled = True
```

```
      lblSSP(3).Enabled = True
```

```
      lblSSP(4).Enabled = True
```

```
      lblSSP(5).Enabled = True
```

```
      lblSSP(7).Enabled = True
```

```
      hsbSSP.Enabled = True
```

```
      For i = 0 To Screen.FontCount - 1
```

```
        cboSSP(1).AddItem Screen.Fonts(i)
```

```
      Next i
```

```
      cboSSP(1).ListIndex = 0
```

```
      fnt.Name = cboSSP(1).Text
```

```
      Set cboSSP(2).Font = fnt
```

```
      cboSSP(2).Clear
```

```
      For i = 0 To 255
```

```
        cboSSP(2).AddItem Chr(i)
```

Next

Else 'if not TT font, then disable controls specific to TT fonts

cboSSP(1).Clear

cboSSP(2).Clear

cboSSP(1).Enabled = False

cboSSP(2).Enabled = False

lblSSP(3).Enabled = False

lblSSP(4).Enabled = False

lblSSP(5).Enabled = False

lblSSP(7).Enabled = False

hsbSSP.Enabled = False

End If

End If

Case 1 'populate combobox list of TT font glyphs in the chosen font

cboSSP(2).Clear

fnt.Name = cboSSP(1).Text

Set cboSSP(2).Font = fnt

For i = 0 To 255

cboSSP(2).AddItem Chr(i)

Next

End Select

End Sub

Private Sub hsbSSP_Change()

'Sets the rotation on a single symbol point marker

'that is using a TT font

lblSSP(7).Caption = hsbSSP.value

End Sub

Private Sub hsbSL_Scroll()

'Sets the rotation on standard label text

lblSL(7).Caption = hsbSL.value

End Sub

Private Sub hsbSL_Change()

'Sets the rotation on standard label text

lblSL(7).Caption = hsbSL.value

End Sub

Private Sub picCBramp_Click(index As Integer)

```
'Sets start and stop ramp colors for class breaks renderer
cdlgLayerProp.ShowColor
picCBramp(index).BackColor = cdlgLayerProp.Color
```

```
End Sub
```

```
Private Sub picNOL_Click()
```

```
'User changes the MaskColor of the LabelPlacer
If colorMask <> moWhite Then
    cdlgLayerProp.Color = colorMask
End If
cdlgLayerProp.ShowColor
picNOL.BackColor = cdlgLayerProp.Color
colorMask = cdlgLayerProp.Color
```

```
'Turn on the MaskLabels check box
chkNOL(2).value = 1
```

```
End Sub
```

```
Private Sub picSSP_Click(index As Integer)
```

```
'Sets the color for single symbol rendering
cdlgLayerProp.ShowColor
picSSP(index).BackColor = cdlgLayerProp.Color
```

```
End Sub
```

```
Private Sub vsbUV_Change()
```

```
'If the legend preview is too large for its
'frame container, use the VSB to move it up
'and down on demand.
fraUVinner.Top = 200 - (vsbUV.value * 200)
```

```
End Sub
```

```
Private Sub sstLayerProp_Click(PreviousTab As Integer)
```

```
'When the user clicks one of the renderer tabs, determine
'whether the active layer is already using that type of
'renderer. Then...
' ...if it is, then read the properties of that layer and
' use those properties to load the various option
' controls on that tab. Use one of the "Load..." procedures.
' ...or if it is not, then load the various option controls
' controls on that tab with some default values. Use
' one of the "Init..." procedures.
'
```

```
Dim lyrRend As Object
```

```
'lblPanelDesc.Caption = strPanelDesc(sstLayerProp.Tab)
```

```
If lyr.Renderer Is Nothing Then    'Use of the Point class here is arbitrary  
  Set lyrRend = New MapObjects2.Point 'and only temporary in use. VB's "TypeOf"  
Else  
  'keyword has trouble with "Nothing", which  
  Set lyrRend = lyr.Renderer    'is what the Layer's Renderer property holds  
End If  
    'when drawing the layer with a single symbol.
```

```
If PreviousTab = 2 Then  
  If TypeOf lyrRend Is MapObjects2.ValueMapRenderer Then  
    Exit Sub  
  End If  
End If
```

```
Select Case sstLayerProp.Tab  
  Case 0  
    If TypeOf lyrRend Is MapObjects2.Point Then  
      Call LoadSingleSymbol  
    Else  
      Call InitSingleSymbol  
    End If  
  Case 1  
    If TypeOf lyrRend Is MapObjects2.ValueMapRenderer Then  
      Call LoadUniqueValues  
    Else  
      Call InitUniqueValues  
    End If  
  Case 2  
    If TypeOf lyrRend Is MapObjects2.ClassBreaksRenderer Then  
      Call LoadClassBreaks  
    Else  
      Call InitClassBreaks  
    End If  
  Case 3  
    If TypeOf lyrRend Is MapObjects2.LabelRenderer Then  
      Call LoadStandardLabels  
    Else  
      Call InitStandardLabels  
    End If  
  Case 4  
    If TypeOf lyrRend Is MapObjects2.LabelPlacer Then  
      Call LoadNoOverlapLabels  
    Else  
      Call InitNoOverlapLabels  
    End If  
  Case 5  
    If TypeOf lyrRend Is MapObjects2.ZRenderer Then  
      Call LoadZRenderer  
    Else  
      Call InitZRenderer  
    End If  
End Select
```

End Sub

'
'
'

'THE SIX PROCEDURES THAT FOLLOW, THAT BEGIN WITH THE WORD "INIT..."
'ARE THOSE THAT RUN WHEN A RENDERER IS CHOSEN WHICH DOES NOT
'COINCIDE WITH THE ACTIVE LAYER'S CURRENT RENDERER. THE OPTION
'CONTROLS ON THAT TAB ARE LOADED WITH DEFAULT VALUES THAT THE USER
'CAN CHANGE.

' InitSingleSymbol()
' InitUniqueValues()
' InitClassBreaks()
' InitStandardLabels()
' InitNoOverlapLabels()
' InitZRenderer()
'
'
'

Private Sub InitSingleSymbol()

Dim i As Integer

Dim fnt As New StdFont

cboSSP(0).Clear

Select Case lyr.shapeType

Case moShapeTypePoint

'set control visibility

cboSSP(1).Visible = True

cboSSP(2).Visible = True

chkSSP.Visible = False

hsbSSP.Visible = True

picSSP(1).Visible = False

lblSSP(3).Visible = True

lblSSP(4).Visible = True

lblSSP(5).Visible = True

lblSSP(6).Visible = False

lblSSP(7).Visible = True

'retrieve and display current values

txtSSP(0).Text = 5

lblSSP(0).Caption = "Marker Color:"

lblSSP(2).Caption = "Size:"

For i = 0 To 4

 cboSSP(0).AddItem strMarkerStyle(i)

Next

picSSP(0).BackColor = moGreen

cboSSP(0).Text = strMarkerStyle(moSquareMarker)

cboSSP(0).ListIndex = 1

hsbSSP.value = 0

lblSSP(7).Caption = "0"

 cboSSP(1).Enabled = False

 cboSSP(2).Enabled = False

```
lblSSP(3).Enabled = False
lblSSP(4).Enabled = False
lblSSP(5).Enabled = False
lblSSP(7).Enabled = False
hsbSSP.Enabled = False
```

Case moShapeTypeMultipoint

```
'set control visibility
cboSSP(1).Visible = True
cboSSP(2).Visible = True
chkSSP.Visible = False
hsbSSP.Visible = True
picSSP(1).Visible = False
lblSSP(3).Visible = True
lblSSP(4).Visible = True
lblSSP(5).Visible = True
lblSSP(6).Visible = False
lblSSP(7).Visible = True
'retrieve and display current values
txtSSP(0).Text = 5
lblSSP(0).Caption = "Marker Color:"
lblSSP(2).Caption = "Size:"
For i = 0 To 4
  cboSSP(0).AddItem strMarkerStyle(i)
Next
picSSP(0).BackColor = moGreen
cboSSP(0).Text = strMarkerStyle(moSqaureMarker)
cboSSP(0).ListIndex = 1
hsbSSP.value = 0
lblSSP(7).Caption = "0"
cboSSP(1).Enabled = False
cboSSP(2).Enabled = False
lblSSP(3).Enabled = False
lblSSP(4).Enabled = False
lblSSP(5).Enabled = False
lblSSP(7).Enabled = False
hsbSSP.Enabled = False
```

Case moLine

```
'set visibility
cboSSP(1).Visible = False
cboSSP(2).Visible = False
chkSSP.Visible = False
hsbSSP.Visible = False
picSSP(1).Visible = False
lblSSP(3).Visible = False
lblSSP(4).Visible = False
lblSSP(5).Visible = False
lblSSP(6).Visible = False
lblSSP(7).Visible = False
'retrieve and display current values
txtSSP(0).Text = 1
```



```

lblSSP(0).Caption = "Line Color:"
lblSSP(2).Caption = "Line width:"
For i = 0 To 4
    cboSSP(0).AddItem strLineStyle(i)
Next
picSSP(0).BackColor = moBlue
cboSSP(0).Text = strLineStyle(0)
cboSSP(0).ListIndex = 0
Case moPolygon
    'set visibility
    cboSSP(1).Visible = False
    cboSSP(2).Visible = False
    chkSSP.Visible = True
    hsbSSP.Visible = False
    picSSP(1).Visible = True
    lblSSP(3).Visible = False
    lblSSP(4).Visible = False
    lblSSP(5).Visible = False
    lblSSP(6).Visible = True
    lblSSP(7).Visible = False
    'retrieve and display current values
    txtSSP(0).Text = "1"
    lblSSP(0).Caption = "Fill Color:"
    lblSSP(2).Caption = "Outline width:"
    For i = 0 To 10
        cboSSP(0).AddItem strFillStyle(i)
    Next
    picSSP(0).BackColor = moLightGray
    picSSP(1).BackColor = moBlack
    cboSSP(0).Text = strFillStyle(0)
    cboSSP(0).ListIndex = 0
    chkSSP.value = 1
End Select
cmdApply.Enabled = True
cmdOK.Enabled = True

End Sub

Private Sub InitUniqueValues()

Dim i As Integer

'Load ComboBox with layer field names
cboUV.Clear
For i = 0 To tDesc.FieldCount - 1
    cboUV.AddItem tDesc.FieldName(i)
Next
cboUV.ListIndex = 0

fraUVouter.Caption = "Legend Preview"

'If a legend already exists, unload it

```

```

If picUV.Count > 1 Then
  For i = (picUV.Count - 1) To 1 Step -1
    Unload picUV(i)
    Unload lblUV(i)
  Next
End If
picUV(0).Visible = False
lblUV(0).Visible = False

cmdApply.Enabled = False
cmdOK.Enabled = False
chkUV.Visible = (lyr.shapeType = moShapeTypePolygon)

End Sub

```

```

Private Sub InitClassBreaks()

```

```

  Dim i As Integer
  Dim fld As MapObjects2.Field

  'Clear and reload ComboBoxes
  cboCB(0).Clear
  cboCB(0).AddItem "FeatureID"
  cboCB(1).ListIndex = 3
  For i = 0 To tDesc.FieldCount - 1
    Set fld = flds(tDesc.FieldName(i))
    If fld.Type = moDouble Or fld.Type = moLong Then
      cboCB(0).AddItem fld.Name
    End If
  Next
  cboCB(0).ListIndex = 0

```

```

  'If a legend already exists, unload it
  If picCBlegend.Count > 1 Then
    For i = (picCBlegend.Count - 1) To 1 Step -1
      Unload picCBlegend(i)
      Unload lblCBlegend(i)
    Next
  End If
  picCBlegend(0).Visible = False
  lblCBlegend(0).Visible = False

```

```

  cmdApply.Enabled = False
  cmdOK.Enabled = False
  chkCB.Visible = (lyr.shapeType = moShapeTypePolygon)

```

```

End Sub

```

```

Private Sub InitStandardLabels()

```

```

  Dim i As Integer
  For i = 0 To tDesc.FieldCount - 1

```

```
cboSL(0).AddItem tDesc.FieldName(i)
If tDesc.FieldType(i) = moLong Or _
  tDesc.FieldType(i) = moDouble Then
  cboSL(3).AddItem tDesc.FieldName(i)
  cboSL(4).AddItem tDesc.FieldName(i)
End If
Next
cboSL(0).ListIndex = 0
cboSL(1).ListIndex = 1
cboSL(2).ListIndex = 1
cdlgLayerProp.FontName = "MS Sans Serif"
cdlgLayerProp.FontSize = 10
cdlgLayerProp.FontBold = False
cdlgLayerProp.FontItalic = False
cdlgLayerProp.FontStrikethru = False
cdlgLayerProp.FontUnderline = False
cdlgLayerProp.Color = moBlack
cmdApply.Enabled = True
cmdOK.Enabled = True
```

End Sub

Private Sub InitNoOverlapLabels()

```
Dim i As Integer
Dim scaleHeightUnit As Double
```

```
cboNOL.Clear
For i = 0 To tDesc.FieldCount - 1
  cboNOL.AddItem tDesc.FieldName(i)
Next
cboNOL.ListIndex = 0
```

```
cdlgLayerProp.FontName = "MS Sans Serif"
cdlgLayerProp.FontSize = 10
cdlgLayerProp.FontBold = False
cdlgLayerProp.FontItalic = False
cdlgLayerProp.FontStrikethru = False
cdlgLayerProp.FontUnderline = False
cdlgLayerProp.Color = moBlack
cmdApply.Enabled = True
cmdOK.Enabled = True
```

```
fraNOL(0).Enabled = (lyr.shapeType <> moShapeTypePolygon)
For i = 0 To optNOL.Count - 1
  optNOL(i).Enabled = (lyr.shapeType <> moShapeTypePolygon)
Next
```

End Sub

Public Sub InitZRenderer()

Dim i As Integer

If picZRlegend.Count > 1 Then

For i = (picZRlegend.Count - 1) To 1 Step -1

Unload picZRlegend(i)

Unload lblZRlegend(i)

Next

End If

picZRlegend(0).Visible = False

lblZRlegend(0).Visible = False

cmdApply.Enabled = False

cmdOK.Enabled = False

End Sub

'
'
'

'THE SIX PROCEDURES THAT FOLLOW, THAT BEGIN WITH THE WORD "LOAD..."
'ARE THOSE THAT RUN WHEN A RENDERER IS CHOSEN WHICH COINCIDES
'WITH THE ACTIVE LAYER'S CURRENT RENDERER. THE PROPERTIES OF
'THAT RENDERER ARE LOADED INTO THE OPTION CONTROLS ON THAT
'RENDERER'S FORM. THE USER CAN CHANGE THEM AT THAT POINT.

' LoadSingleSymbol()

' LoadUniqueValues()

' LoadClassBreaks()

' LoadStandardLabels()

' LoadNoOverlapLabels()

' LoadZRenderer()

'
'
'

Private Sub LoadSingleSymbol()

Dim i As Integer

Dim fnt As New StdFont

cboSSP(0).Clear

Select Case lyr.shapeType

Case moShapeTypePoint

'set control visibility

cboSSP(1).Visible = True

cboSSP(2).Visible = True

chkSSP.Visible = False

hsbSSP.Visible = True

picSSP(1).Visible = False

lblSSP(3).Visible = True

lblSSP(4).Visible = True

lblSSP(5).Visible = True

lblSSP(6).Visible = False

lblSSP(7).Visible = True

```

'retrieve and display current values
txtSSP(0).Text = lyr.Symbol.Size
lblSSP(0).Caption = "Marker Color:"
lblSSP(2).Caption = "Size:"
For i = 0 To 4
  cboSSP(0).AddItem strMarkerStyle(i)
Next
picSSP(0).BackColor = lyr.Symbol.Color
cboSSP(0).Text = strMarkerStyle(lyr.Symbol.style)
cboSSP(0).ListIndex = lyr.Symbol.style
hsbSSP.value = lyr.Symbol.Rotation
lblSSP(7).Caption = lyr.Symbol.Rotation
If lyr.Symbol.style = moTrueTypeMarker Then
  cboSSP(1).Enabled = True
  cboSSP(2).Enabled = True
  lblSSP(3).Enabled = True
  lblSSP(4).Enabled = True
  lblSSP(5).Enabled = True
  lblSSP(7).Enabled = True
  hsbSSP.Enabled = True
  For i = 0 To Screen.FontCount - 1
    cboSSP(1).AddItem Screen.Fonts(i)
  Next i
  cboSSP(1).Text = lyr.Symbol.Font.Name
  For i = 0 To cboSSP(1).ListCount - 1
    If cboSSP(1).List(cboSSP(1).ListIndex) = cboSSP(1).Text Then
      cboSSP(1).ListIndex = 1
      Exit For
    End If
  Next
  fnt.Name = cboSSP(1).Text
  Set cboSSP(2).Font = fnt
  cboSSP(2).Clear
  For i = 0 To 255
    cboSSP(2).AddItem Chr(i)
  Next
  cboSSP(2).Text = lyr.Symbol.CharacterIndex
  cboSSP(2).ListIndex = lyr.Symbol.CharacterIndex
End If

```

Case moShapeTypeMultipoint

```

'set control visibility
cboSSP(1).Visible = True
cboSSP(2).Visible = True
chkSSP.Visible = False
hsbSSP.Visible = True
picSSP(1).Visible = False
lblSSP(3).Visible = True
lblSSP(4).Visible = True
lblSSP(5).Visible = True
lblSSP(6).Visible = False
lblSSP(7).Visible = True

```

```

'retrieve and display current values
txtSSP(0).Text = lyr.Symbol.Size
lblSSP(0).Caption = "Marker Color:"
lblSSP(2).Caption = "Size:"
For i = 0 To 4
  cboSSP(0).AddItem strMarkerStyle(i)
Next
picSSP(0).BackColor = lyr.Symbol.Color
cboSSP(0).Text = strMarkerStyle(lyr.Symbol.style)
cboSSP(0).ListIndex = lyr.Symbol.style
hsbSSP.value = lyr.Symbol.Rotation
lblSSP(7).Caption = lyr.Symbol.Rotation
If lyr.Symbol.style = moTrueTypeMarker Then
  cboSSP(1).Enabled = True
  cboSSP(2).Enabled = True
  lblSSP(3).Enabled = True
  lblSSP(4).Enabled = True
  lblSSP(5).Enabled = True
  lblSSP(7).Enabled = True
  hsbSSP.Enabled = True
  For i = 0 To Screen.FontCount - 1
    cboSSP(1).AddItem Screen.Fonts(i)
  Next i
  cboSSP(1).Text = lyr.Symbol.Font.Name
  For i = 0 To cboSSP(1).ListCount - 1
    If cboSSP(1).List(cboSSP(1).ListIndex) = cboSSP(1).Text Then
      cboSSP(1).ListIndex = 1
    Exit For
  End If
Next
fnt.Name = cboSSP(1).Text
Set cboSSP(2).Font = fnt
cboSSP(2).Clear
For i = 0 To 255
  cboSSP(2).AddItem Chr(i)
Next
cboSSP(2).Text = lyr.Symbol.CharacterIndex
cboSSP(2).ListIndex = lyr.Symbol.CharacterIndex
End If

```

```

Case moShapeTypeLine
'set visibility
cboSSP(1).Visible = False
cboSSP(2).Visible = False
chkSSP.Visible = False
hsbSSP.Visible = False
picSSP(1).Visible = False
lblSSP(3).Visible = False
lblSSP(4).Visible = False
lblSSP(5).Visible = False
lblSSP(6).Visible = False
lblSSP(7).Visible = False

```

```

'retrieve and display current values
txtSSP(0).Text = lyr.Symbol.Size
lblSSP(0).Caption = "Line Color:"
lblSSP(2).Caption = "Line width:"
For i = 0 To 4
  cboSSP(0).AddItem strLineStyle(i)
Next
picSSP(0).BackColor = lyr.Symbol.Color
cboSSP(0).Text = strLineStyle(lyr.Symbol.style)
cboSSP(0).ListIndex = lyr.Symbol.style
Case moShapeTypePolygon
'set visibility
cboSSP(1).Visible = False
cboSSP(2).Visible = False
chkSSP.Visible = True
hsbSSP.Visible = False
picSSP(1).Visible = True
lblSSP(3).Visible = False
lblSSP(4).Visible = False
lblSSP(5).Visible = False
lblSSP(6).Visible = True
lblSSP(7).Visible = False
'retrieve and display current values
If lyr.Symbol.Size = 0 Then
  txtSSP(0).Text = 1
Else
  txtSSP(0).Text = lyr.Symbol.Size
End If
lblSSP(0).Caption = "Fill Color:"
lblSSP(2).Caption = "Outline width:"
For i = 0 To 10
  cboSSP(0).AddItem strFillStyle(i)
Next
picSSP(0).BackColor = lyr.Symbol.Color
picSSP(1).BackColor = lyr.Symbol.OutlineColor
cboSSP(0).Text = strFillStyle(lyr.Symbol.style)
cboSSP(0).ListIndex = lyr.Symbol.style
Select Case lyr.Symbol.Outline
  Case True: chkSSP.value = 1
  Case False: chkSSP.value = 0
End Select
End Select

End Sub

Private Sub LoadUniqueValues()

Dim i As Integer

Set vmr = lyr.Renderer

cboUV.Clear

```

```

cboUV.Text = vmr.Field
For i = 0 To tDesc.FieldCount - 1
  cboUV.AddItem tDesc.FieldName(i)
  If tDesc.FieldName(i) = vmr.Field Then
    cboUV.ListIndex = i
  End If
Next
Select Case vmr.Symbol(0).Outline
  Case True: chkUV.value = 1
  Case False: chkUV.value = 0
End Select
chkUV.Visible = (lyr.shapeType = moShapeTypePolygon)

```

Call PopulateExistingUVlegend

End Sub

Private Sub LoadClassBreaks()

```

Dim i, j As Integer
Set cbr = lyr.Renderer
j = -1
cboCB(0).Clear
cboCB(0).Text = cbr.Field
For i = 0 To tDesc.FieldCount - 1
  If tDesc.FieldType(i) = moDouble Or _
    tDesc.FieldType(i) = moLong Then
    j = j + 1
    cboCB(0).AddItem tDesc.FieldName(i)
    If tDesc.FieldName(i) = cbr.Field Then
      cboCB(0).ListIndex = j
    End If
  End If
Next
cboCB(1).Text = cbr.BreakCount + 1
cboCB(1).ListIndex = cbr.BreakCount - 1
Select Case cbr.Symbol(0).Outline
  Case True
    chkCB.value = 1
  Case False
    chkCB.value = 0
End Select
picCBbramp(0).BackColor = cbr.Symbol(0).Color
picCBbramp(1).BackColor = cbr.Symbol(cbr.BreakCount).Color
chkCB.Visible = (lyr.shapeType = moShapeTypePolygon)

```

Call PopulateExistingCBlegend

End Sub

Private Sub LoadStandardLabels()


```
Dim i As Integer
Dim strFN As String
Set lr = lyr.Renderer
```

```
For i = 0 To tDesc.FieldCount - 1
  strFN = tDesc.FieldName(i)
  cboSL(0).AddItem strFN
  If lr.Field = strFN Then
    cboSL(0).ListIndex = i
  End If
  If tDesc.FieldType(i) = moLong Or _
    tDesc.FieldType(i) = moDouble Then
    cboSL(3).AddItem strFN
    If lr.XOffsetField = strFN Then
      cboSL(3).ListIndex = i
    End If
    cboSL(4).AddItem strFN
    If lr.YOffsetField = strFN Then
      cboSL(4).ListIndex = i
    End If
  End If
End If
Next
```

```
Select Case lr.Symbol(0).HorizontalAlignment
  Case moAlignLeft
    cboSL(1).ListIndex = 0
  Case moAlignCenter
    cboSL(1).ListIndex = 1
  Case moAlignRight
    cboSL(1).ListIndex = 2
End Select
```

```
Select Case lr.Symbol(0).VerticalAlignment
  Case moAlignTop
    cboSL(2).ListIndex = 0
  Case moAlignCenter
    cboSL(2).ListIndex = 1
  Case moAlignBottom
    cboSL(2).ListIndex = 2
End Select
```

```
hsbSL.value = lr.Symbol(0).Rotation
lblSL(7).Caption = lr.Symbol(0).Rotation
```

```
For i = 0 To 3
  chkSL(i).value = 0
Next
If lr.DrawBackground Then
  chkSL(0).value = 1
End If
If lr.AllowDuplicates Then
  chkSL(1).value = 1
```

```
End If
If lr.SplinedText Then
    chkSL(2).value = 1
End If
If lr.Flip Then
    chkSL(3).value = 1
End If

txtSL.Text = lr.Symbol(0).Font.Name
txtSL.ForeColor = lr.Symbol(0).Color
cdlgLayerProp.FontName = lr.Symbol(0).Font.Name
cdlgLayerProp.FontSize = lr.Symbol(0).Font.Size
cdlgLayerProp.FontBold = lr.Symbol(0).Font.Bold
cdlgLayerProp.FontItalic = lr.Symbol(0).Font.Italic
cdlgLayerProp.FontStrikethru = lr.Symbol(0).Font.Strikethrough
cdlgLayerProp.FontUnderline = lr.Symbol(0).Font.Underline
cdlgLayerProp.Color = lr.Symbol(0).Color
```

End Sub

Private Sub LoadNoOverlapLabels()

```
Dim i As Integer
Dim scaleHeightUnit As Double
Dim strFN As String
```

```
Set lp = lyr.Renderer
colorMask = lp.MaskColor
```

```
cboNOL.Clear
For i = 0 To tDesc.FieldCount - 1
    strFN = tDesc.FieldName(i)
    cboNOL.AddItem strFN
    If lp.Field = strFN Then
        cboNOL.ListIndex = i
    End If
Next
```

```
cdlgLayerProp.FontName = lp.DefaultSymbol.Font.Name
cdlgLayerProp.FontSize = lp.DefaultSymbol.Font.Size
cdlgLayerProp.FontBold = lp.DefaultSymbol.Font.Bold
cdlgLayerProp.FontItalic = lp.DefaultSymbol.Font.Italic
cdlgLayerProp.FontStrikethru = lp.DefaultSymbol.Font.Strikethrough
cdlgLayerProp.FontUnderline = lp.DefaultSymbol.Font.Underline
colorText = lp.DefaultSymbol.Color
txtNOL.Text = lp.DefaultSymbol.Font.Name
txtNOL.ForeColor = colorText
```

```
optNOL(0).value = lp.PlaceOn
optNOL(1).value = lp.PlaceAbove
optNOL(2).value = lp.PlaceBelow
```

```
Select Case Ip.DrawBackground
  Case False: chkNOL(0).value = 0
  Case True:  chkNOL(0).value = 1
End Select
```

```
Select Case Ip.AllowDuplicates
  Case False: chkNOL(1).value = 0
  Case True:  chkNOL(1).value = 1
End Select
```

```
Select Case Ip.MaskLabels
  Case False: chkNOL(2).value = 0
  Case True
    chkNOL(2).value = 1
    picNOL.BackColor = Ip.MaskColor
End Select
```

```
scaleHeightUnit = lyr.Extent.Width / 10000
hsbNOL.value = 1000 - (Ip.DefaultSymbol.Height / scaleHeightUnit)
```

```
End Sub
```

```
Private Sub LoadZRenderer()
```

```
  Dim i, j As Integer
  Set zRend = lyr.Renderer
  j = -1
  cboZRclasses.Clear
  cboZRclasses.Text = zRend.BreakCount + 1
  cboZRType.ListIndex = zRend.ValueCalculation
  picZRramp(0).BackColor = zRend.Symbol(0).Color
  picZRramp(1).BackColor = zRend.Symbol(zRend.BreakCount).Color
```

```
  Call PopulateExistingZRlegend
```

```
End Sub
```