

Membuat Aplikasi SIG dengan MapObjects Bagian I: Fungsi-fungsi Standar

Contributed by Administrator
Monday, 30 April 2007
Last Updated Thursday, 26 July 2007

[This article will translated in English after all sections complete]

Pengguna aplikasi-aplikasi SIG pasti cukup akrab dengan yang namanya Arcview, MapInfo, dan Autocad. Selain cukup populer (karena banyak bajakannya) aplikasi-aplikasi ini memang sangat handal untuk keperluan-keperluan pemetaan dan analisisnya. Bahkan Arcview dan Mapinfo bisa di customize tampilannya agar sesuai kebutuhan user.

Akan tetapi kadangkala ada suatu kebutuhan dimana sebuah aplikasi SIG terintegrasi dengan aplikasi lain, misalnya pada suatu aplikasi SPK (Sistem Penunjang Keputusan) dimana pemetaan (plus analisisnya) hanya merupakan salah satu tools disamping tools-tools yang lain. Kadang juga pengguna kurang suka dengan tampilan hasil customize dari arcview atau mapinfo yang memang terbatas. Untuk kebutuhan-kebutuhan inilah Esri, pembuat Arcview, mengeluarkan MapObjects (dan Mapinfo mengeluarkan MapX).

MapObjects adalah sebuah third party component yang bisa digunakan di Visual Basic, Visual C++, Delphi, dan lainnya, agar bisa menampilkan peta beserta navigasi, dan fungsi-fungsi pemetaan lainnya. Karena berupa komponen, maka aplikasi GIS yang dibuat dengan MapObjects ini dapat dibuat lebihflexible, dapat digabung dalam aplikasi yang lainnya, dan dapat bebas dalam membuat tampilan

Disini kita akan menggunakan MapObjects Versi 2. (2.0 atau 2.1) , dan akan dipakai dalam Visual Basic 6.0. Dalam penyampaian materi-nya, akan dibuat berdasar fungsi-fungsi pemetaan baru dari fungsi ini akan di breakdown kedalam object-object apa yang ikut, method dan property apa yang dipakai, dan langsung disertai contoh coding-nya.

I. Persiapan

Untuk membuat aplikasi SIG dengan VB dan MapObjects, tentu saja kita harus telah menyiapkan keduanya di komputer kita. Installer untuk MapObjects 2.1 versi evaluation dapat didownload di <http://if2.ubaya.ac.id/~daniel/uploads/mo21.exe>. Sedangkan untuk contoh data peta yang akan digunakan dalam tulisan ini adalah peta Mexico bawaan dari arcview 3.1, dapat didownload di <http://if2.ubaya.ac.id/~daniel/mexico.zip> . Untuk engine mapObject, cara instalasinya adalah dengan menjalankan file mo21.exe tadi. Sedangkan untuk datanya dapat diextract ke folder apapun.

Dalam tulisan ini, data shapefile ini diextract di folder c:\shapefile. Maka akan ada folder c:\shapefile\Mexico, dan didalamnya ada data-data peta bertipe shapefile. Sementara untuk mapobjects, setelah diinstall, kita dapat memakainya di VB dengan menambahkan komponen Esri MapObjects 2.1 lewat menu Project-Component seperti pada gambar 1 dibawah. Setelah ditambahkan komponen ini, maka akan ada tambahan object dalam toolbox, yaitu object “Map” seperti yang terlihat pada gambar 2.Object Map inilah yang nanti ditaruh dalam Form sebagai tempat dimana peta akan dimunculkan.

Selain ketersediaan file-file diatas, untuk dapat mengikuti materi dengan lancar, sebaiknya pembaca telah familiar dengan Visual Basic dan juga dengan konsep SIG.

II. Menampilkan Peta

Untuk menampilkan sebuah Peta dalam component Map, kita membutuhkan 2 object lagi selain component tersebut. Yang pertama adalah object DataConnection yang digunakan untuk mengkoneksikan database peta, dan yang kedua adalah object MapLayer yang digunakan untuk menyimpan Informasi untuk sebuah Layer Peta. Ketiga object ini nanti akan dipakai untuk membuat program pertama kita yang bertujuan untuk menampilkan peta Mexico.

Pertama, taruh componen Map ke dalam Form.

Berikutnya. Object DataConnection di deklarasikan sebagai berikut, ditaruh dalam variable private agar dapat diakses dari berbagai sub dalam form ini.

```
Dim dc as new MapObjects2.DataConnection.
```

Dalam object `DataConnection` terdapat property `Database` untuk menentukan di direktori mana data-data peta berada. Dan juga terdapat fungsi `Connect` untuk memulai koneksi data. Contoh penggunaannya adalah sebagai berikut:

```
dc.Database = "c:\shapefile\mexico"
If dc.Connect Then
    &lsquo;lakukan perintah2 jika data berhasil connect
Else
    &lsquo;lakukan perintah-perintah jika tidak berhasil connect
End If
```

Object `DataConnection` juga mempunyai fungsi `FindGeoDataset` yang bertujuan untuk mencari peta tertentu dan mengembalikannya sebagai class `Geodataset`, yaitu class yang menyimpan informasi-informasi spasial dan non-spasial dari data peta. Class `Geodataset` inilah yang merupakan data peta yang diperlukan oleh object `MapLayer`. Pastikan dalam direktori yang disetkan pada `dc.database` terdapat nama file (dengan extension `.shp`, `.shx`, dan `.dbf`) yang dimasukkan dalam fungsi `FindGeoDataset` ini. Contoh penggunaan dari `findgeodataset` sampai digunakan oleh `mapLayer` adalah sebagai berikut.

```
Dim L1 As mapobjects2.MapLayer
Set L1 = New Mapobejcts2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("states")
```

Dimana `‘States’` adalah peta propinsi-propinsi di Mexico yang akan kita tampilkan.

Setelah object `MapLayer` terbentuk, maka object ini dapat dimasukkan ke dalam component `Map` untuk ditampilkan ke layar. Component `Map` menyimpan layer-layer-nya dalam collection class `Layers`. Dimana dalam class ini terdapat method `Add` untuk menambahkan anggotanya. Peta akan disusun dengan layer yang ditambahkan terakhir berada di tumpukan paling atas. Jadi jika peta `Jalan` dimasukkan dalam `map.layers`, kemudian dilanjutkan dengan menambahkan peta `State`, maka peta `Jalan` tadi akan tertutup oleh peta `State`.

Contoh kode untuk menambahkan `maplayer` kedalam `Map.Layers` adalah sebagai berikut:

```
Map1.Layers.Add L1
```

Ini adalah perintah terakhir yang dibutuhkan dalam program pertama kita. Dengan asumsi peta akan di-load pada saat `Form_load`, kode lengkapnya adalah sebagai berikut:

```
Dim dc As New DataConnection
Private Sub Form_Load()
    dc.Database = "c:\shapefile\mexico"
    If dc.Connect Then
        Dim L1 As Mapobjects2.MapLayer
        Set L1 = New Mapobjects2.MapLayer
        L1.GeoDataset = dc.FindGeoDataset("states")
        Map1.Layers.Add L1
    Else
        MsgBox "direktori peta tidak ditemukan"
    End If
End Sub
```

Jadi, setelah menaruh component `Map` ke `Form` (namanya menjadi `Map1`) dan dengan menuliskan perintah-perintah diatas, kemudian menjalankan `Run`, kita akan mendapatkan sebuah form yang didalamnya terdapat peta Mexico yang terbagi dalam propinsi-propinsi.

Apabila anda belum berhasil atau peta tidak keluar, coba periksa lagi lokasi peta dan nama petanya. Program pertama ini harus berhasil dulu sebelum melangkah ke program-program selanjutnya.

Selanjutnya, dengan cara yang sama seperti saat menambahkan layer `‘States’`; tambahkan layer `‘Roads’`; dan `‘Cities’`; sehingga bagian didalam `If dc.Connect Then` adalah sebagai berikut :

```
If dc.Connect Then
    Dim L1 As Mapobjects2.MapLayer
    Set L1 = New Mapobjects2.MapLayer
    L1.GeoDataset = dc.FindGeoDataset("states")
    Map1.Layers.Add L1
```

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("roads")
Map1.Layers.Add L1
```

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("cities")
Map1.Layers.Add L1
```

```
Else &hellip;.
&hellip;.
```

Setelah ditambahkan layer ‘Roads’ dan ‘Cities’ maka saat di Run akan tampil peta yang berisi peta negara propinsi di Mexico, diatasnya terdapat Jalan, dan diatas jalan ada titik-titik yang menunjukkan kota-kota di Mexico. Contoh hasilnya dapat dilihat pada gambar 3 dibawah.

Jika dicermati, setiap kali aplikasi ini dijalankan, maka warna States, Roads, dan Cities akan berubah-ubah. Hal ini disebabkan karena kita belum melakukan setting tampilan pada tiap layer ini. Jika belum di set, default warnanya adalah random. Bab berikutnya akan menunjukkan bagaimana cara untuk memodifikasi tampilan Layer.

Gambar 3. Contoh tampilan program ‘menampilkan peta’

III. Mengatur tampilan peta

Tampilan atau bentuk penggambaran dari tiap layer peta dapat diatur sesuai keinginan user atau yang lebih sering lagi adalah sesuai dengan standar legenda peta pada umumnya. Beberapa contoh standar tampilan adalah seperti jalan berwarna merah, sungai berwarna hijau, bandara berbentuk tanda pesawat, dan sebagainya.

Pada layer bertipe polygon yang dapat diatur adalah warna bidang, bentuk arsiran bidang, warna garis tepi, bentuk dan tebal garis tepi. Untuk tipe Line, yang dapat diatur adalah warna, tebal, dan bentuk garis. Sedang untuk tipe Point, yang dapat diatur adalah warna simbol, ukuran simbol, dan bentuk simbol. Secara umum untuk mengatur tampilan pada masing-masing tipe ini dilakukan dengan mengubah-ubah nilai-nilai dalam property symbol pada object MapLayer.

Berikut adalah daftar property dalam maplayer.symbol, dan fungsinya:

color

- untuk pengaturan warna. Dapat diisi dengan nilai dari konstanta-konstanta warna, seperti moRed, moBlue, moYellow, moGrey, dan warna-warna yang punya ‘nama’ lainnya. Dapat juga diisi dengan nilai dari pencampuran warna merah, hijau, dan biru, pada fungsi RGB(nilai_merah,nilai_hijau,nilai_biru)

size

- ukuran simbol, hanya berlaku untuk tipe garis dan point

outline, outlinecolor

- ada tidaknya garis tepi, dan warna garis tepi. Khusus untuk tipe polygon. Outline diisi dengan true/false, sedang outlinecolor diisi warna seperti halnya color.

style

- untuk tipe polygon ini adalah bentuk arsiran, untuk tipe line ini adalah bentuk garis, dan untuk point adalah bentuk simbol. Dengan pilihan dibawah ini:

o untuk polygon

§ moSolidFill : bidang diwarnai penuh, tanpa arsiran

§ moTransparentFill : bidang tidak diwarnai dan transparent

§ moHorizontalFill : bidang diarsir dengan arsiran garis-garis mendatar

§ moVerticalFill : bidang diarsir dengan arsiran garis-garis tegak

§ moDownwardDiagonalFill : arsiran diagonal miring ke kiri

§ moUpwardDiagonalFill : arsiran diagonal miring ke kanan

§ moCrossFill : arsiran gabungan garis vertical dan horizontal

§ moDiagonalCrossFill : arsiran gabungan garis horizontal kiri dan orisontal kanan

§ moGrayFill : arsiran berupa titik-titik

§ moLightgrayFill : arsiran berupa titik-titik yang jarang

§ moDarkgrayFill : arsiran berupa titik-titik yang rapat

o untuk line

§ moSolidLine : berupa garis lurus
 § moDotLine : berupa titik-titik yang bersambung
 § moDashLine : berupa garis pendek-pendek yang bersambung
 § moDashDotLine : berupa garis–titik yang bersambung
 § moDashDotDotLine : berupa garis-titik-titik yang bersambung

o untuk point

§ moCircleMarker : berupa bentuk lingkaran
 § moCrossMarker : berupa bentuk cross
 § moSquareMarker : berupa bentuk kotak
 § moTriangleMarker : berupa bentuk segitiga
 § moTrueTypeMarker : berupa suatu karakter dari suatu font

Untuk type ini, diperlukan pengisian juga untuk property :

• symbol.Font : Nama font yang karakternya mau dipakai
 • symbol.CharacterIndex : nomor karakter yang ingin dipakai
 • symbol.Rotation : jika mau diputar (dalam satuan derajat)

Dengan mengisikan property symbol diatas, peta yang tampil akan dapat sangat bervariasi. Berikut disediakan dua contoh program perubahan tampilan.

Program pertama akan membuat warna state hijau (dari fungsi RGB), warna jalan merah dan besarnya 2. Sedangkan kota ditampilkan dengan bentuk lingkaran berwarna biru, dengan ukuran 8. Sedang program yang kedua membentuk state berarsir kotak-kotak berwarna kuning, Jalan berbentuk titik-garis-titik-garis berwarna biru, dan kota berbentuk simbol J (font=wingdings, nomor karakter=74) berwarna merah.

Cuplikan program pertama saat pengaturan tampilan adalah seperti berikut :

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("states")
L1.Symbol.Color = RGB(200, 250, 200)
Map1.Layers.Add L1
```

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("roads")
L1.Symbol.Color = moRed
L1.Symbol.Size = 3
Map1.Layers.Add L1
```

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("cities")
L1.Symbol.Style = moCircleMarker
L1.Symbol.Color = moBlue
L1.Symbol.Size = 8
Map1.Layers.Add L1
```

Sedang untuk program kedua adalah sebagai berikut :

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("states")
L1.Symbol.Color = moYellow
L1.Symbol.Style = moCrossFill
Map1.Layers.Add L1
```

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("roads")
L1.Symbol.Color = moBlue
L1.Symbol.Style = moDashDotLine
Map1.Layers.Add L1
```

```
Set L1 = New Mapobjects2.MapLayer
L1.GeoDataset = dc.FindGeoDataset("cities")
L1.Symbol.Style = moTrueTypeMarker
L1.Symbol.Color = moRed
L1.Symbol.Size = 12
L1.Symbol.Font = "wingdings"
L1.Symbol.CharacterIndex = 74
```

Map1.Layers.Add L1

Hasil untuk masing-masing program diatas adalah seperti gambar 4 dan 5 dibawah:

Gambar 4. Hasil untuk program perubahan tampilan 1

Gambar 5. Hasil untuk program perubahan tampilan 2

IV. Navigasi Peta

Salah satu fasilitas yang pasti ada dalam sebuah aplikasi SIG adalah adanya tombol-tombol navigasi peta. Untuk melihat detail peta secara mudah, biasanya sebuah form yang menampilkan peta akan punya tombol-tombol Zoom in, Zoom out, Full Extent dan Pan.

Zoom In dilakukan untuk membuat peta terasa lebih detail. Dengan zoom in, skala akan diperkecil. Saat dilakukan zoom in seakan-akan kita akan lebih didekatkan pada permukaan bumi. Fungsi zoom out adalah kebalikan dari zoom in, yaitu membuat peta lebih tidak detail. FullExtent adalah tombol yang membuat peta menjadi terlihat seluruhnya. Misal setelah melihat detail, pengguna ingin melihat peta keseluruhan dengan cepat, maka dia dapat menekan tombol full extent, daripada melakukan zoom out berkali-kali sampai batas full. Sementara Pan adalah fungsi untuk menggeser peta. Fungsi pan biasanya dilakukan setelah pengguna melakukan zoom in. Misal saat pengguna ingin melihat lebih detail, dia melakukan zoom in, ternyata dia juga ingin melihat detail yang disebelah kanan dari yang tampil sekarang, maka dia dapat melakukan penggeseran peta.

Untuk memberikan fasilitas navigasi diatas, tambahkan tombol-tombol pada Form. Biasanya ditaruh di atas peta, dan berbentuk tombol-tombol yang bergambar atau suatu toolbar yang bergambar. Gambar-gambar yang sering dipakai adalah seperti kaca pembesar, tangan, dan sebagainya. Untuk penyederhanaan masalah, disini akan ditambahkan command button biasa, ditaruh diatas peta, menjadi seperti gambar 6 dibawah.

Gambar 5. Bentuk form untuk contoh program navigasi peta

Proses Untuk Zoom In, Zoom Out, dan Pan, dilakukan dengan cara melakukan aktifitas mouse pada object Map. Ketiganya akan sama-sama dilakukan pada saat event MouseDown. Untuk itu, kita perlu memberikan sebuah tanda, katakanlah semacam mode, yang jika diklik tombol zoom in, berarti mode menjadi ‘zoomin’, menekan tombol zoom out, mode menjadi ‘zoomout’, dan pan menjadi mode ‘pan’. Agar kelihatan mode yang aktif sekarang adalah mode tertentu, maka saat men-set mode, sekalian dilakukan perubahan bentuk pointer Mouse. Kebetulan MapObjects menyediakan bentuk pointer mouse untuk mode-mode tadi, yaitu moZoomIn, moZoomOut, dan moPan.

Karena butuh menyimpan mode yang sedang aktif, dan bisa diakses oleh banyak sub di satu form, maka kita butuh variabel yang bersifat global yang disini akan kita namai dengan pmode.

```
dim Pmode as String
```

Dan, misalkan tombol-tombol tadi berturut-turut bernama command1, command2, command3, dan command4, maka kode untuk masing-masing tombol saat diklik adalah seperti dibawah.

```
Private Sub Command1_Click()
    pMode = "zoomin"
    Map1.MousePointer = moZoomIn
End Sub
Private Sub Command2_Click()
    pMode = "zoomout"
    Map1.MousePointer = moZoomOut
End Sub
Private Sub Command4_Click()
    pMode = "pan"
    Map1.MousePointer = moPan
End Sub
```

Setelah men-set mode, yang dilakukan selanjutnya adalah mengisikan kode untuk tiga macam mode diatas. Kode ini diisikan pada event MouseDown, sehingga kita butuh untuk mengecek dulu mode apa yang sedang aktif, dan kemudian

apa yang harus dilakukan pada saat mode ini. Disini akan digunakan pengecekan dengan Select Case seperti dibawah:

```
Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As _
    Single, Y As Single)
Select Case pmode
Case "zoomin"
    'kode untuk melakukan zoom in
Case "zoomout"
    'kode untuk melakukan zoom out
Case "pan"
    'kode untuk melakukan pan
End Select
End Sub
```

Property yang digunakan untuk pengaturan seberapa zoom peta yang ditampilkan adalah property Extent pada object map. Property ini menunjukkan seberapa besar “bingkai” pada peta yang sedang aktif saat ini. Untuk mendekatkan peta, bingkai ini diperkecil. Sebaliknya, untuk menjauhkan peta, bingkai ini diperbesar.

Proses pengguna melakukan zoom in adalah pengguna membuat suatu bingkai berbentuk rectangle yang kemudian rectangle tadi dijadikan bingkai yang aktif. Untuk membuat bingkai ini, pengguna ‘dipaksa’ untuk membuat rectangle pada map. Disini digunakan fungsi TrackRectangle dan bentuk rectanglenya disimpan dalam sebuah variabel, yang dalam program kita adalah variabel rect. Variabel rect ini kemudian dijadikan extent peta sekarang. Kode untuk melakukan ini adalah seperti dibawah :

```
Dim rect As MapObjects2.Rectangle
Set rect = Map1.TrackRectangle
Map1.Extent = rect
```

Berbeda dengan saat zoom in yang memerlukan pembuatan luasan (rectangle) dahulu sebelum dilakukan zoom in, pada zoom out pengguna tinggal melakukan klik pada suatu lokasi di peta. Setelah dilakukan klik, extent peta akan diambil, kemudian diperbesar, dan dikembalikan lagi ke extent aktif. Untuk memperbesar rectangle (dan juga untuk memperkecil) dapat digunakan fungsi ScaleRectangle dengan parameternya adalah berapa pengali skala yang diinginkan. Untuk menjauhkan peta, maka kita perlu untuk memperbesar rectangle, maka diberi skala yang lebih dari 1. Kode untuk melakukan ini adalah seperti dibawah :

```
Dim rect2 As MapObjects2.Rectangle
Set rect2 = Map1.Extent
rect2.ScaleRectangle 1.5
Map1.Extent = rect2
```

Untuk melakukan Pan, kita tinggal memanggil fungsi Pan pada object map.

```
Map1.Pan
```

Jadi kode lengkap yang ada pada event MouseDown adalah seperti berikut:

```
Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As _
    Single, Y As Single)
Select Case pMode
Case "zoomin"
    Dim rect As MapObjects2.Rectangle
    Set rect = Map1.TrackRectangle
    Map1.Extent = rect
Case "zoomout"
    Dim rect2 As MapObjects2.Rectangle
    Set rect2 = Map1.Extent
    rect2.ScaleRectangle 1.5
    Map1.Extent = rect2
Case "pan"
    Map1.Pan
End Select
End Sub
```

Kita masih menyisakan satu tombol yang belum berfungsi. Yaitu tombol FullExtent. MapObjects selalu menyimpan data besarnya bingkai yang diperlukan untuk menampilkan seluruh layer-layer peta dalam keadaan terlihat semua (full).

Property yang menyimpan rectangle ini adalah property FullExtent. Maka, untuk menjadikan peta terlihat full, tinggal dilakukan pengisian nilai Extent dengan nilai dari FullExtent. Dilakukan pada saat event click pada tombol full extent.

```
Private Sub Command3_Click()
    Map1.Extent = Map1.FullExtent
End Sub
```

Sampai disini, program SIG kita sudah mempunyai fungsi-fungsi navigasi standar, yaitu zoom in, zoom out, full extent, dan pan. Gambar 6 adalah gambar peta saat selesai melakukan zoom in.

Gambar 6. Contoh hasil proses zoom in

V. Legenda dan Skala

Pengaturan tampilan peta, seperti pada bab sebelumnya, membuat tampilan peta dapat tergantung pada pembuat aplikasi. Apabila peta mengandung banyak layer dan tiap layer dibentuk bermacam-macam, maka peta dapat membingungkan pembacanya, terutama bagi pengguna yang belum mengenal daerah yang digambarkan di peta. Nah, agar pengguna peta tidak merasa membingungkan, biasanya didalam aplikasi disediakan keterangan tentang simbol-simbol yang dipakai di peta. Misal, garis warna merah adalah jalan, garis warna biru adalah sungai, gambar pesawat adalah bandara, dan sebagainya.

Selain legenda, yang juga penting adalah keterangan berapa skala pada tampilan peta saat ini. Skala ini terupdate secara otomatis saat extent dari peta berubah, baik saat zoom in, zoom out, atau full extent.

MapObjects memberikan contoh bagaimana untuk membuat legenda dan skala dalam contoh program di folder “LegendComponentSample” dan “ScalebarComponentSample”. Di dalam contoh ini legend dan component dibuat dalam bentuk project OCX dan project untuk testernya. Untuk mempermudah, OCX ini sudah dipaket dan dapat didownload di <http://if2.ubaya.ac.id/~daniel/uploads/scalelegend.zip>

Setelah didownload, extractlah file scalelegend.zip ini di suatu tempat. Disitu akan terdapat file mo21samplelegend.ocx dan mo21sampleScalebar.ocx. Nah, tambahkan satu persatu file-file ocx ini dalam project kita lewat menu di vb project – component – browse. Sehingga componen terpilih dalam window project – component menjadi seperti gambar 7 dan toolbar menjadi seperti gambar 8 dibawah.

Gambar 7. Menambah ocx Legend dan Scalebar

Gambar 8. Tambahan ocx scalebar dan legend dalam toolbar

Setelah ada dalam toolbar, maka kita dapat menggunakan ocx legend dan toolbar ini dan dapat ditaruh di sembarang tempat untuk melengkapi fasilitas aplikasi kita. Disini, untuk contoh program berikutnya, ocx legend dan toolbar ditaruh di sebelah kanan peta dan warna background keduanya diganti menjadi putih. Sehingga desain form menjadi seperti gambar 9 berikut:

Gambar 9. Desain tampilan yang ditambah legend dan scalebar

Kedua Component tambahan ini cukup mudah cara penggunaannya. Untuk Legend, pertama harus di set, map mana yang “diikatkan” pada legend ini. Setelah itu secara otomatis legend akan mengikuti keterangan-keterangan dan setting-setting tampilan tiap-tiap layer. Ini dilakukan dengan fungsi SetMapSource. Dan jika selama runtime terdapat perubahan tampilan layer, dan legend perlu diupdate, maka tinggal dilakukan pemanggilan fungsi LoadLegend.

Nama layer yang ditampilkan dalam legend akan diambil dari property mapLayer.name yang belum kita set pada program-program sebelumnya. Nama layer ini bebas dan dapat berbeda dari nama file-nya. Sebagai contoh, layer states, roads, dan cities berturut-turut diberi nama propinsi, jalan, dan kota. Pemberian nama dan proses melekatkan legend ini ditaruh di even Form_Load. Sekalian saja, karena juga diisikan pada form_Load ini, diisikan pula kode untuk setting Scalebar.

Untuk mengupdate skala yang tercantum scalebar, pada sourcecode contoh penggunaan scalebar telah dibuatkan suatu fungsi yaitu fungsi UpdateScale . Fungsi ini perlu dituliskan kembali dalam form dan dipanggil pertama kali saat

form_load dan pada saat ada perubahan extent, yaitu di saat MouseDown. Sementara untuk setting awalnya, pada scalebar perlu di set ScaleBarUnit, dan MapUnit. Juga bentuk tampilan scalebar pada property ScaleText.

Berikut adalah code pada FormLoad yang sudah ditambahi, dan kode sub UpdateScale. Diikuti contoh tampilan yang sudah menerapkan legend dan scalebar pada gambar 10.

```
Private Sub Form_Load()
    dc.Database = "c:\shapefile\mexico"

    If dc.Connect Then
        Dim L1 As MapObjects2.MapLayer
        Set L1 = New MapObjects2.MapLayer
        L1.GeoDataset = dc.FindGeoDataset("states")
        L1.Symbol.Color = moYellow
        L1.Symbol.Style = moCrossFill
        Map1.Layers.Add L1
        L1.Name = "Propinsi"

        Set L1 = New MapObjects2.MapLayer
        L1.GeoDataset = dc.FindGeoDataset("roads")
        L1.Symbol.Color = moBlue
        L1.Symbol.Style = moDashDotLine
        Map1.Layers.Add L1
        L1.Name = "Jalan"

        Set L1 = New MapObjects2.MapLayer
        L1.GeoDataset = dc.FindGeoDataset("cities")
        L1.Symbol.Style = moTrueTypeMarker
        L1.Symbol.Color = moRed
        L1.Symbol.Size = 12
        L1.Symbol.Font = "wingdings"
        L1.Symbol.CharacterIndex = 74
        Map1.Layers.Add L1
        L1.Name = "Kota"

    Else
        MsgBox "direktori peta tidak ditemukan"
    End If

    legend1.setMapSource Map1
    legend1.LoadLegend

    ScaleBar1.ScaleBarUnits = suMeters
    ScaleBar1.ScreenUnits = suCentimeters
    ScaleBar1.ScaleText = RepresentativeFraction
    ScaleBar1.MapUnits = muDecimalDegrees
    RefreshScale ScaleBar1, Map1
End Sub

Public Sub RefreshScale(AEScaleBar As ScaleBar, MOMap As Map)
    Dim MapExt As sbExtent
    Dim Pageext As sbExtent
    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
    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
```

Gambar 10. Contoh tampilan yang sudah diberi legend dan scalebar

Terlihat pada gambar 10, dibagian scalebar terdapat keterangan kota, jalan, dan propinsi dengan bentuk simbolnya masing-masing. Pada tiap layer terlihat ada sebuah checkbox yang dapat digunakan untuk me-visible-kan dan unvisible-kan layer. Selain itu, tiap layer pada legend juga bisa di drag ke layer atasnya maupun ke layer bawahnya yang membuat urutan layer menjadi berubah. Untuk mengtur visible, harus ditambahkan dulu kode untuk me-refresh mapyang dipanggil pada event legend_AfterSetLayerVisible.

```
Private Sub legend1_AfterSetLayerVisible(Index As Integer,isVisible _
    As Boolean)
    Map1.Refresh
End Sub
```

VI. Posisi Lintang/Bujur

Sering pengguna ingin tahu posisi suatu object yang sesungguhnya di dunia ada di posisi bujur atau lintang berapa. Pada aplikasi Arcview, posisi ini berada pada bagian atas kanan dan nilainya berubah mengikuti posisi Mouse. Untuk memberikan keterangan ini, pada desain tampilan program berikutnya ditambah dengan Label di bagian atas legend seperti pada gambar 11.

Gambar 11. Desain tampilan yang telah ditambah Label untuk posisi lintang/bujur

MapObjects membuat fungsi ToMapPoint yang berfungsi men-convert suatu titik dalam layar menjadi lokasi aslinya di dunia. Lokasi asli ini juga berupa Point dan disini akan disimpan dalam variabel “P”. Dari variabel point inilah akan diambil X dan Y-nya. X akan dimasukkan pada Label3, dan Y dimasukkan pada label4. Karena posisi ini dibuat agar mengikuti gerakan mouse, maka kode untuk mengambil lokasi ini ditaruh di event MouseMove pada object map.

```
Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As _
    Single, Y As Single)
    Dim p As MapObjects2.Point
    Set p = Map1.ToMapPoint(X, Y)
    Label3 = p.X
    Label4 = p.y
End Sub
```

Setelah dijalankan. cek program ini dengan cara menggerak-gerakkan mouse di peta. Jika di sebelah kanan atas terdapat nilai yang berubah-ubah mengikuti gerak mouse, berarti program berjalan dengan baik. Contoh tampilan saat mouse diletakkan di suatu tempat di peta adalah seperti gambar 11 dibawah.

Gambar 12. Contoh tampilan yang dilengkapi Posisi Lintang dan Bujur

VII. Identify

Tool standar berikutnya adalah tool untuk melihat data Tabular dari object yang dipilih. Dalam Arcview, fungsi ini disebut dengan Identify, dan kita akan mengikuti standar penamaan ini. Proses Identify dilakukan dengan melakukan klik pada suatu posisi dalam peta. Dari posisi ini akan diambil keterangan dari object yang diklik. Dalam contoh berikutnya keterangan object yang akan ditampilkan adalah isi dari field Name pada Kota dan Propinsi, dan field Route pada layer Jalan. Untuk layer propinsi diambil juga luas dari propinsi yang terpilih, dan pada layer jalan diambil juga panjang jalan.

Dengan cara yang sama dengan Identify ini, kita dapat mengambil isi field apapun dari shapefile. Seringkali dalam shapefile hanya ada kode object saja, sedang keterangan-keterangan lainnya disimpan dalam database non spatial seperti MS Access, SQL Server, atau Oracle. Jika seperti ini bentuknya, maka saat Identify kita ambil kode object, kemudian kita query ke database non spatial tadi untuk mencari record yang field kode objectnya sama dengan yang terambil tadi. Jadi tidak masalah apabila data keterangan berada pada database diluar shapefile. Cara ini juga yang digunakan untuk mengkonekkan object tertentu di peta dengan data foto , video, atau yang lainnya.

Proses identify dilakukan saat eventMouseDown pada object map, seperti halnya zoom in, zoom out, dan pan. Jadi kita

akan memerlukan suatu mode yang menunjukkan bahwa yang akan berlaku saat ini saat klik di peta adalah untuk fungsi Identify. Dan agar tampilan lebih bagus, saat mengisi mode, kita juga mengubah bentuk pointer mouse dengan moldentify. Berikut adalah bentuk desain form untuk contoh identify.

Gambar 13. Desain tampilan untuk Identify

Kode pertama yang harus ditambahkan adalah pada saat tombol Identify diklik:

```
Private Sub Command5_Click()
    pMode = "identify"
    Map1.MousePointer = moldentify
End Sub
```

Selanjutnya dalam event MouseDown ditambah dengan case jika mode aktif adalah identify. Untuk proses pengambilan data atribut (non spatial) dari suatu object dapat dilakukan dengan beberapa cara. Jika akan mengambil object yang dekat dengan lokasi klik, dapat digunakan fungsi SearchByDistance milik object MapLayer. Fungsi ini akan mengembalikan recordset yang ada pada suatu shapefile yang dekat dengan suatu shape tertentu dan pada jarak tertentu. Shape disini adalah point yang dibentuk dengan fungsi ToMapPoint, sedang jarak diisi suatu nilai double yang besarnya akan menentukan seberapa jauh radius dari object2 dalam layer dari point tadi yang akan diambil datanya. Semakin besar jarak berarti semakin besar radius dan membuat hasil klik kurang spesifik karena mengembalikan banyak object. Sebaliknya jika terlalu kecil akan membuat kita sulit untuk mendapatkan informasi dari suatu object karena jaraknya harus tepat dengan klik kita.

Recordset yang dikembalikan dari hasil SearchByDistance memiliki field khusus yang digunakan untuk mengambil nilai spatial object. Nama field ini adalah field “Shape” yang mengeluarkan object point, line, atau polygon, tergantung dari tipe layernya. Untuk mengambil panjang jalan, dapat dilakukan dengan cara mengambil property length dari object Line yang ada dalam field shape ini, sedang untuk mengambil luas, dilakukan dengan cara memanggil property Area. Karena peta mexico yang kita pakai menggunakan satuan derajat, maka untuk panjang perlu dikalikan dengan suatu konstanta yang membuat nilai derajat menjadi satuan meter. Untuk daerah katulistiwa, satu derajat kurang lebih sama dengan 110 km.

Dalam contoh, akan dilakukan looping terlebih dahulu terhadap semua layer. Masing-masing layer akan disearch dan keterangan-keterangan objectnya akan ditampung dalam suatu string dan dikeluarkan dalam messagebox. Berikut adalah kode yang perlu ditambahkan dalam event MouseDown dalam object map, melengkapi kode-kode untuk zoom in, zoom out, dan pan.

```
Case "identify"
    Dim i As Long
    Dim L As MapObjects2.MapLayer
    Dim rec As MapObjects2.Recordset
    Dim s As String
    Dim P As MapObjects2.Point
    Dim FName As String
    Set P = Map1.ToMapPoint(X, Y)
    For i = 0 To 2
        '0=cities, 1=roads, 2=states
        If i = 0 Then
            s = s & "Kota:" & vbCrLf
            FName = "Name"
        End If
        If i = 1 Then
            s = s & "Jalan:" & vbCrLf
            FName = "Route"
        End If
        If i = 2 Then
            s = s & "Propinsi:" & vbCrLf
            FName = "Name"
        End If
        Set L = Map1.Layers(i)
        Set rec = L.SearchByDistance(P, 0.1, "")
        While Not rec.EOF
            s = s & vbTab & rec.Fields(FName).ValueAsString
            If i = 1 Then s = s & ", Panjang=" & _ Format(rec.Fields("Shape").Value.Length * 110, "0") & "Km"
            If i = 2 Then s = s & ", Luas=" & _
                Format(rec.Fields("Shape").Value.Area * 110 * 110, "0") & "Km2"
        End While
    End For
```

```
s = s & vbCrLf
Wend
```

```
Next
MsgBox s
```

Untuk mengetes program, klik-lah pada suatu titik dalam peta. Jika didekat titik klik tadi ada jalan, maka akan muncul rute-rute jalan tadi. Jika lokasi yang diklik dekat dengan. suatu kota, maka akan muncul nama-nama kota yang dekat dengan klik tadi. Contoh tampilan setelah melakukan identify ini dapat dilihat pada gambar 14.

Gambar 14. Contoh hasil saat melakukan Identify

VIII. Searching

Identify digunakan untuk melihat keterangan dari suatu lokasi yang ingin diketahui data attributnya. Atau bisa juga untuk melihat data foto atau video jika di-link-kan ke suatu object. Jadi pengguna melihat data non spatial dari data spatial. Tapi kadang juga pengguna yang tidak begitu hapal dengan peta yang dilihatnya lebih memilih melakukan pencarian data spatial dari data nonspatialnya. Untuk keperluan inilah disediakan tool Searching.

Proses searching dilakukan pada isi atribut-atribut dari layer. Pada contoh program berikutnya, layer yang akan dicari adalah layer propinsi atau layer kota. Keduanya akan dicari pada isi dari field Name-nya. Program akan mengambil record-record yang isi field name ini yang mengandung suatu string yang dimasukkan pengguna dan dimasukkan dalam sebuah listbox. Kemudian dari listbox tadi, pengguna dapat memilih salah satu nama propinsi (atau kota) dan melihat posisinya dalam peta. Pada layer propinsi, peta akan fokus ke propinsi yang ingin dilihat dan propinsi ini akan berkedap-kedip, sedang untuk kota hanya akan berkedap-kedip saja.

Berikut adalah modifikasi tampilan untuk menambahkan fungsi searching

Gambar14. Desain tampilan yang telah ditambah proses searching

Terdapat combobox di bagian kanan atas yang telah disiapkan pilihannya yaitu propinsi dan kota. Textbox dibawahnya digunakan untuk menerima input dari pengguna. Tombol dibawahnya untuk memulai pencarian yang hasilnya dimasukkan dalam listbox dibawahnya. Setelah terdapat hasil di listbox, isi listbox dapat digunakan untuk memilih kota / propinsi tertentu untuk dilihat dipeta lewat tombol Fokus dibawah list.

Untuk melakukan searching berdasar nilai non spatial dapat dilakukan dengan fungsi SearchExpression. Sama dengan SearchByDistance, fungsi ini juga mengembalikan sebuah recordset. Parameter untuk fungsi SearchExpression ini adalah sebuah kalimat query tapi hanya kalimat yang berada setelah “where”. Jadi semisal jika query yang biasanya untuk mencari propinsi yang nama propinsinya adalah ‘tabasco’ adalah “SELECT * FROM states WHERE name=’tabasco’”, dalam fungsi ini cukup ditulis “name=’tabasco’” saja.

Berikut adalah program saat pengguna melakukan pengisian pada textbox, dilanjutkan klik pada tombol Mulai.

```
Private Sub Command7_Click()
Dim L As MapLayer
Dim LName As String
Dim rec As mapobjects2.Recordset
List1.Clear
Set L = Map1.Layers(Combo1.Text)
Set rec = L.SearchExpression("Name Like '%" & Text1.Text & "%")
While Not rec.EOF
List1.AddItem rec.Fields("Name").ValueAsString
rec.MoveNext
Wend
End Sub
```

Berikutnya setelah isi list berisi nama propinsi/kota yang sesuai kriteria, pengguna dapat memilih salah satu dan menekan tombol fokus untuk melihat lokasinya di peta. Untuk melakukan ini data nonspatial pada layer di query lagi, hanya kali ini tidak menggunakan LIKE tetapi menggunakan “=” agar hasil yang terlihat dipeta adalah

benar-benar object yang bernama persis seperti yang dipilih. Query ini akan menghasilkan recordset juga dan akan diambil bentuk spatialnya untuk di fokuskan dan di kelap-kelipkan. Untuk membuat peta menjadi fokus ke object terpilih dapat dilakukan dengan cara mengubah extent peta menjadi extent object terpilih. Sedang untuk membuat kelap-kelip digunakan perintah FlashShape yang mempunyai parameter object apa yang akan dibuat kalp-kelip, dan berapa kali kelap-kelip akan terjadi. Berikut adalah kode pada tombol Mulai:

```
Private Sub Command6_Click()
Dim pilih As String
Dim rec As mapobjects2.Recordset
pilih = List1.List(List1.ListIndex)
Set L = Map1.Layers(Combo1.Text)
Set rec = L.SearchExpression("Name=" & pilih & "")
While Not rec.EOF
    If Combo1.Text = "propinsi" Then Map1.Extent = rec.Fields("Shape").Value.Extent
    Map1.Refresh
    Map1.FlashShape rec.Fields("Shape").Value, 3
    rec.MoveNext
Wend
End Sub
```

Contoh Hasil dari proses searching yang kemudian dilanjutkan dengan melihat lokasi object hasil searching dapat dilihat pada gambar 16 berikut:

Gambar 16. Contoh hasil proses searching

IX. Measuring

Salah satu keuntungan penggunaan data spatial adalah kita tidak perlu menyimpan nilai panjang dan luas, karena secara otomatis sudah dapat diambil dari data yang berbentuk vektor ini. Seperti sudah digunakan pada saat identify yang mengeluarkan panjang jalan dan luas propinsi dengan tinggal mengambil property length dan Area. Fungsi pengambilan panjang dan Luas ini dapat kita manfaatkan juga untuk pengukuran panjang suatu rute tertentu atau pengukuran luas suatu bidang berbentuk tertentu di peta yang dalam Arcview disebut dengan measure.

Untuk melakukan measure ini kita perlu menambahkan lagi tombol untuk pengukuran panjang dan pengukuran luas seperti pada gambar 17 dibawah.

Gambar 17. Desain tampilan yang sudah dilengkapi Pengukuran jarak dan Luas

Sama seperti Identify, proses pencarian jarak dan luas juga dilakukan saat event MouseDown pada object Peta. Jadi harus diberi mode dan bentuk pointer tertentu.

```
Private Sub Command8_Click()
    pMode = "jarak"
    Map1.MousePointer = moCross
End Sub

Private Sub Command9_Click()
    pMode = "luas"
    Map1.MousePointer = moCross
End Sub
```

Sebelum menambahkan kode pada event MouseDown, ada yang perlu disiapkan lagi. Yaitu bentuk dari rute dan bidang yang akan dibuat diatas peta. Membuat gambar diatas peta memerlukan object TrackingLayer. Object trackinglayer ini mempunyai koleksi simbol-simbol untuk menggambar bentuk tertentu yang disimpan dalam property Symbol dan jumlah simbolnya dalam property SymbolCount. Isi dari property symbol sama dengan isi property symbol pada object maplayer. Symbol-symbol ini harus didefinisikan terlebih dahulu, dan akan ditambahkan dalam event form_load. Disini symbol pertama digunakan untuk menggambar rute, sedang simbol kedua untuk menggambar bidang.

```

Map1.TrackingLayer.SymbolCount = 2
With Map1.TrackingLayer.Symbol(0)
    .SymbolType = moLineSymbol
    .Color = moRed
    .Size = 3
End With
With Map1.TrackingLayer.Symbol(1)
    .SymbolType = moFillSymbol
    .OutlineColor = moRed
    .Color = moCyan
End With

```

Setelah bentuk symbol siap, dilanjutkan dengan menambahkan case untuk jarak dan luas. Jika sebelumnya kita mengenal TrackRectangle saat zoom in untuk memaksa pengguna membuat rectangle, maka untuk pengukuran jarak dan luas ini kita akan memaksa pengguna untuk membuat polyline dan polygon. Masing-masing dengan fungsi TrackLine dan TrackPolygon. Setelah pengguna membuat garis/polygon, bentuk garis/polygon ini kemudian dimasukkan dalam trackinglayer dengan fungsi AddEvent dan digambar dengan simbol yang sudah kita set sebelumnya. Setelah digambar, keterangan panjang/luas ditampilkan dengan messagebox dengan terlebih dulu dikalikan dengan konstanta derajat ke km. Setelah messagebox ditutup, gambar garis/polygon ini akan dihapus dengan ClearEvent. Berikut kode yang harus ditambahkan pada event mouseMove.

```

Case "jarak"
    Dim Ln As mapobjects2.Line
    Set Ln = Map1.TrackLine
    Map1.TrackingLayer.AddEvent Ln, 0
    MsgBox "Panjang=" & Ln.Length * 110 & " Km"
    Map1.TrackingLayer.ClearEvents

```

```

Case "luas"
    Dim PI As mapobjects2.Polygon
    Set PI = Map1.TrackPolygon
    Map1.TrackingLayer.AddEvent PI, 1
    MsgBox "Luas=" & PI.Area * 110 * 110 & " Km2"
    Map1.TrackingLayer.ClearEvents

```

Gambar 18 dan 19 berturut-turut menunjukkan contoh hasil dari fungsi pengukuran panjang dan luas ini.

Gambar 18. Contoh tampilan saat melakukan pengukuran panjang

Gambar 19. Contoh hasil saat melakukan pengukuran luas

BERSAMBUNG KE BAGIAN II : FUNGSI-FUNGSI ADVANCE