

# **TUGAS AKHIR**

**PEMBUATAN APLIKASI MAPVIEW STAND ALONE MENGGUNAKAN  
DOTSPATIAL OPEN SOURCE LIBRARY CONTROL 1.4 MENGGUNAKAN  
BAHASA PEMROGRAMAN VISUAL C#**



**Disusun Oleh  
ALBERTO DOMINIKSOY MEO  
08.25.034**

**PROGRAM STUDI TEKNIK GEODESI S-1  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
INSTITUSI TEKNOLOGI NASIONAL MALANG  
2014**

**LEMBARAN PERSETUJUAN**

**PEMBUATAN APLIKASI MAPVIEW STAND ALONE MENGGUNAKAN  
DOTSPATIAL OPEN SOURCE LIBRARY CONTROL 1.4 MENGGUNAKAN BAHASA  
PEMROGRAMAN VISUAL C#**

**SKRIPSI**

**Diajukan untuk memenuhi persyaratan dalam mencapai  
Gelar Sarjana Teknik (ST) Strata Satu (S-1) Teknik Geodesi S-1  
Institut Teknologi Nasional Malang**

Oleh :

**ALBERTO DOMINIKSOY MEO**

**0825034**

Menyetujui :

**Dosen Pembimbing I**

  
(D.K. Sunaryo, ST, MT)

**Dosen Pembimbing II**

  
(M. Edwin Tjahjadi, ST., MGeomSc., PhD.)

**Mengetahui,**

**Ketua Jurusan Teknik Geodesi S-1**

  
(Ir. Agus Darpono, MT)



PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

LEMBARAN PENGESAHAN  
SKRIPSI

PEMBUATAN APLIKASI MAPVIEW STAND ALONE MENGGUNAKAN  
DOTSPATIAL OPEN SOURCE LIBRARY CONTROL 1.4 MENGGUNAKAN  
BAHASA PEMROGRAMAN VISUAL C#

Telah Dipertahankan di Hadapan Team Penguji Ujian Skripsi Jenjang Strata 1 (S-1)

Pada Hari : Jumat

Tanggal : 22 Agustus 2014

Dan diterima untuk memenuhi salah satu persyaratan guna memperoleh gelar Sarjana Teknik (ST).

Oleh:

Alberto Dominiksoy Meo


08.25.034

Panitia Ujian Skripsi

Ketua


  
( Ir. Agus Darpono, MT )

Sekretaris

  
(Silvester Sari Sai, ST., MT)

Anggota Penguji

Penguji I

  
( D.K. Sunaryo, ST., MT )

Penguji II

  
(Ir. Agus Darpono, MT)

Penguji III

  
(Ir. Jasmani, M.Kom)

**PEMBUATAN APLIKASI MAPVIEW STAND ALONE MENGGUNAKAN  
DOTSPATIAL OPEN SOURCE LIBRARY CONTROL 1.4 MENGGUNAKAN  
BAHASA PEMROGRAMAN VISUAL C#**

Alberto Dominiksoy Meo 08.25.034

Dosen Pembimbing I : D.K. Sunaryo, ST ., MT

Dosen Pembimbing II : M. Edwin Tjahjadi, ST ., MGeomSc., PhD.

**Abstraksi**

*Pada era globalisasi kini telah berkembang perubahan yang signifikansi terhadap komputer dengan lingkungan yang berbasis Windows, munculnya berbagai lingkungan pemrograman yang dirancang untuk meningkatkan keuntungan yang terkait dengan Windows. Visual C#, adalah contoh lingkungan pemrograman yang memungkinkan pengguna untuk dengan cepat dan efisien membuat program yang mudah digunakan.*

*Ada beberapa alasan seorang programmer dapat memilih untuk mengembangkan aplikasi dengan Dotspatial. Dalam beberapa situasi, Dotspatial menawarkan banyak manfaat bagi pengembang dengan menyediakan fleksibilitas yang besar yang memungkinkan pengembang untuk membuat aplikasi yang disesuaikan. Semakin kecil footprint memori, atau persyaratan RAM, memungkinkan untuk aplikasi yang efisien dan efektif.*

*Membangun aplikasi Mapview dengan menggunakan Dotspatial 1.4 dan Microsoft Visual C# 2010 diharapkan dapat memberikan kemudahan bagi seorang user untuk mengetahui segala informasi yang ingin digunakan dalam pengolahan dan penggunaan data berbasis Sistem Informasi Geografis.*

**Kata kunci :** *Dotspatial, pemrograman, Visual C#, aplikasi, Mapview, Sistem Informasi Geografis*

## SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertandatangan dibawah ini :

Nama : Alberto Dominiksoy Meo  
NIM : 0825034  
Program Studi : Teknik Geodesi S-1  
Fakultas : Teknik Sipil dan Perencanaan

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul

**“PEMBUATAN APLIKASI MAPVIEW STAND ALONE MENGGUNAKAN  
DOTSPATIAL OPEN SOURCE LIBRARY CONTROL 1.4 MENGGUNAKAN  
BAHASA PEMROGRAMAN VISUAL C#”**

Adalah hasil karya saya sendiri dan bukan menjiplak atau menduplikat serta tidak mengutip atau menyadur dari hasil karya oranglain kecuali disebutkan sumbernya.

Malang, 29 September 2014

Yang membuat pernyataan



Alberto Dominiksoy Meo

NIM : 0825034

## LEMBARAN PERSEMBAHAN

"Start Now Take the First Step. Together We Can Change the World"

### **KUPERSEMBAHKAN SKRIPSI INI KEPADA :**

Tuhan Yesus Kristus yang senantiasa mendampingi  
memberikan terang, kasih, karunia, serta rahmat -Nya  
sehingga aku dapat menjalani segalanya sesuai rencana -Nya  
karena aku yakin didalam DIA dan bersama DIA  
segala yang mustahil bagi manusia menjadi nyata bagi -Nya.

Bapakku tersayang Hendrikus Mere, Sp yang selalu menjadi motivator saya dalam belajar  
dan mencapai cita-cita

Ibuku tercinta Rosalia Roswinda Kedang yang tak pernah lelah memberikan kasih sayang  
dan nasehat-nasehat bijak agar saya menjadi manusia yang lebih baik

Opa-Oma Tersayang,

Lambertus Laba Kedang

Fransiska Moong

Alm. Dominikus Dhae Meo

Alm. Yosefina Ngatu

Om-Tanta Tersayang,

Alm. Karolus Kia Kedang, SH

Vinsensia Bunga

Kristianus Molang, SHut

Damasus Sogan, Skom

Kristianus Molang

1950

1951

1952

1953

1954

1955

1956

1957

1958

1959

1960

1961

1962

1963

1964

1965

1966

1967

1968

1969

1970

1971

1972

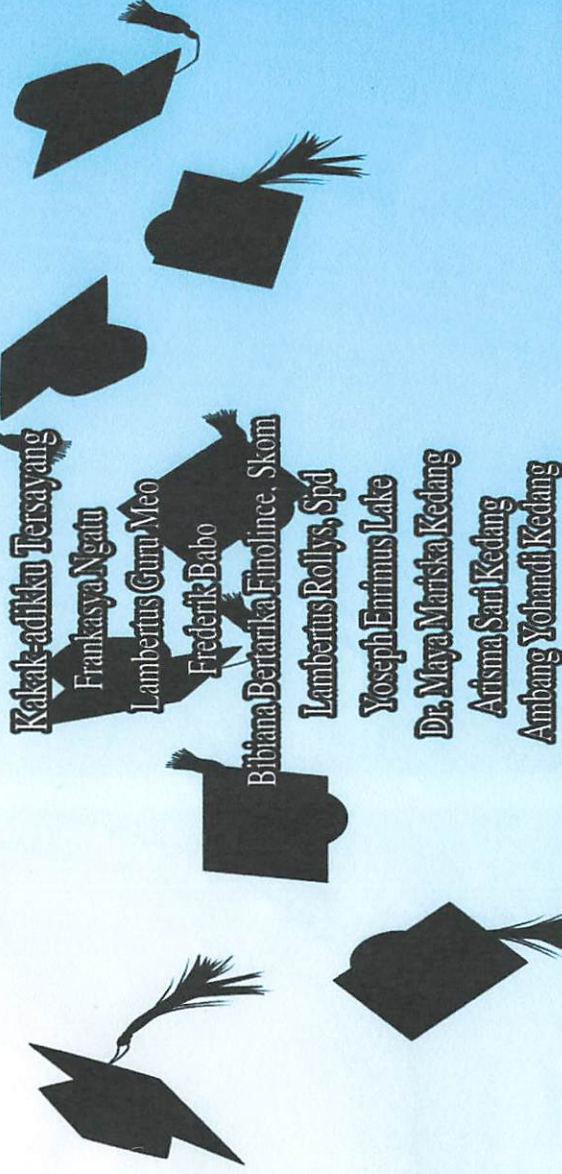
1973

1974

1975

1976

**"Start Now Take the First Step. Together We Can Change the World"**



**Seluruh Keluarga Besar, Om dan Tanta yang senantiasa memberikan doa dan dorongan, serta perhatian kepada saya hingga menyelesaikan skripsi ini**

**Ucapan terima kasih juga kucampaikan kepada,**

**-Bapak Ir. Agus Darpono, MT, selaku Ketua Jurusan Teknik Geodesi ITN Malang  
-Bapak D.K. Sumawo, ST., MT, dan Bapak M. Edwin Tjahjadi, ST., MGeomSc., PhD,  
yang telah membimbing saya dalam menyelesaikan skripsi ini  
-Sahabat-Sahabat Geo 08: Kent, Rio, Kristo, Gancang, Charlan, Obeth, Ady, Ernest, Ino, Yuston  
Dedy, Dewa, Wawan, Irfan, Indra, Dian, Reza, Rizal, Eka, Rony, Mulyawan, Adith,  
Gunawan, Eki, Hata, Agung, Ayu.**

**Kita telah melewati banyak hal, baik susah dan senang bersama.....**

**-Teman-teman Teknik Geodesi ITN Malang. 2004, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014 + transferan.**

**-Terima Kasih banyak juga buat teman-teman kontrakan yang sudah memberikan dukungan dan masukan sehingga dapat menyelesaikan laporan tugas akhir ini.**



CONFIDENTIAL - SECURITY INFORMATION

1. The purpose of this document is to provide information regarding the security of the system. This document is intended for use by personnel who are responsible for the security of the system. It is not to be distributed outside of the organization.

2. The information contained in this document is classified as CONFIDENTIAL - SECURITY INFORMATION. It is to be controlled in accordance with the policies and procedures of the organization.

3. This document contains information that is essential to the security of the system. It is to be protected from unauthorized disclosure. Any unauthorized disclosure of this information could result in the compromise of the system and the security of the organization.

**"Start Now Take the First Step. Together We Can Change the World"**

Terima Kasih kuucapkan kepada klaseman Crew,

- Kent

Teman SEJATI selalu berbagi  
andai kamu bunga,  
aku mau jadi kumbangnya  
andai kamu Juliet,  
aku siap jadi Romeo nya  
andai aku Tarzan.

sudikah kamu jadi monyetnya?

-Rio

Ketika tiba saat perpisahan  
janganlah kalian berduka,  
sebab apa yang paling kalian kasihi darinya  
mungkin akan nampak lebih nyata dari kejauhan  
seperti gunung yang nampak lebih agung  
terlihat dari padang dan dataran.

-Yuston

Banyak orang ingin naik dengan  
Anda dalam limusin,  
tapi apa yang Anda inginkan adalah  
seseorang yang akan  
naik bus dengan Anda  
ketika limusin rusak

- Sandro

Kalau ada yang bilang kamu jelek, sabar aja.  
Kalau ada yang bilang kamu bego, Cuekin aja.  
Kalau ada yang bilang kamu dungu, Cool aja.  
Tapi kalau ada yang berani bilang kamu cakep,  
Tampar aja karena itu FITNAH!

-Gogo

Jangan berjalan di belakangku,  
aku mungkin tidak memimpin.  
Jangan berjalan di depanku,  
aku mungkin tidak mengikuti.  
Hanya berjalan di sampingku  
dan tetap menjadi Sahabatku.

-Opa Efen

Persahabatan itu seperti kencing di celana Anda,  
setiap orang bisa melihatnya,  
tapi hanya kamu yang dapat merasakan kehangatannya

-Jems, Gina, Sonny, Even, Apri, Elfish, Pipoz, Rusly,  
Angky, Doni

dunia ini luas  
hingga langit ke tujuh  
dan di tiap-tiap langit itu  
kau akan temukan atmosfer yang lain  
yang mungkin lebih baik dari yang ini

*Ikan hiu makan permen. I miss you, man!*



## **KATA PENGANTAR**

Dengan memanjatkan puji syukur kehadiran Allah yang Maha Esa, atas limpahan Rahmat dan Karunia-Nya sehingga penyusun dapat menyelesaikan laporan tugas akhir dengan judul **“Pembuatan Aplikasi Mapview Stand Alone Menggunakan Dotspatial Open Source Library Control 1.4 Menggunakan Bahasa Pemrograman Visual C#”**

Tugas akhir ini disusun untuk memenuhi salah satu persyaratan untuk menyelesaikan studi S-1 Teknik Geodesi ITN Malang

Oleh karena itu pada kesempatan ini penyusun menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Agus Darpono, MT selaku Ketua Jurusan Teknik Geodesi ITN Malang
2. Seluruh Rekan-rekan Geodesi ITN Malang yang membantu penulis dalam penyusunan laporan Tugas Akhir.
3. Semua pihak yang telah memberikan masukan dan arahan baik secara langsung maupun tidak langsung sehingga terwujud Laporan Tugas Akhir

Penulis menyadari masih banyak kekurangan dalam penyusunan Laporan Tugas Akhir ini, untuk itu kami mengharapkan adanya saran dan kritik yang bersifat membangun demi sempurnanya Laporan Tugas Akhir ini.

Akhir kata penyusun berharap semoga Tugas Akhir ini dapat bermanfaat bagi pembaca semua.

Malang, September 2014

(Penyusun)

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>LEMBAR PERSETUJUAN</b> .....	<b>ii</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>iii</b>
<b>ABSTRAKSI</b> .....	<b>iv</b>
<b>PERNYATAAN KEASLIAN SKRIPSI</b> .....	<b>v</b>
<b>LEMBAR PERSEMBAHAN</b> .....	<b>vi</b>
<b>KATA PENGANTAR</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>ix</b>
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	2
1.4. Batasan Masalah .....	2
1.5. Tinjauan Pustaka .....	3
1.6. Sistematika Penelitian .....	3
<b>BAB II Landasan Teori</b>	
2.1. Konsep Informasi Geografi .....	5
2.1.1. Pengertian Sistem .....	5
2.1.2. Karakteristik Sistem .....	5
2.2. Konsep Informasi Geografi .....	7
2.3. Pengertian Sistem Informasi .....	8
2.4. Sistem Informasi Geografis .....	8
2.4.1. Konsep Dasar Sistem Informasi Geografis .....	8
2.4.2. Pengertian Sistem Informasi Geografis .....	9

2.4.3. Komponen Sistem Informasi Geografis .....	10
2.4.4. Subsistem Sistem Informasi Geografis .....	12
2.4.5. Basis Data .....	12
2.5. Peta .....	13
2.5.1. Pengertian Peta .....	13
2.5.2. Fungsi dan Tujuan Peta.....	13
2.5.3. Jenis dan Unsur-Unsur Peta.....	14
2.5.3.1. Jenis Peta .....	14
2.5.3.2. Unsur-Unsur .....	15
2.6. <i>C# (C Sharp)</i> .....	19
2.7. <i>.Net</i> .....	23
2.8. <i>Dotspatial</i> .....	27

### **BAB III METODE PERANCANGAN APLIKASI**

3.1. Persiapan .....	28
3.1.1. Alat dan Bahan Perancangan Aplikasi.....	28
3.2. Langkah Perancangan Aplikasi .....	29
3.2.1. Flowchart Perancangan Aplikasi .....	29
3.2.2. Flowchart Input Data .....	31
3.3. Tahapan Perancangan Aplikasi .....	32
3.3.1. Persiapan.....	32
3.3.2. Studi Literatur/Pengumpulan Data .....	32
3.3.3. Desain Aplikasi Mapview.....	32
3.3.4. Pembuatan Aplikasi Mapview.....	32
3.3.5. Analisa dan Simulasi Aplikasi.....	32
3.3.6. Validasi .....	33
3.3.6. Hasil .....	33
3.4. Tahapan Persiapan dan Pengumpulan Data .....	33
3.4.1. Persiapan dan Pengumpulan Data.....	33
3.4.2. Editing Data Spasial.....	33

3.4.3. Membuat Topologi .....	35
3.4.4. Export Data .....	39
3.4.5. Pembuatan Basis Data .....	40
3.4.6. Memulai Operasi ArcGIS .....	41
3.4.7. Penggabungan Data ( <i>Join</i> ).....	43
3.4.8. Penyimpanan Hasil Olahan Data .....	44
3.4. Tahapan Pembuatan Aplikasi Mapview .....	44
3.5.1. Memulai Operasi Microsoft Visual Studio 2010.....	45

#### **BAB IV HASIL DAN PEMBAHASAN**

4.1. Aplikasi Mapview .....	55
4.2. Form Utama .....	55
4.3. Menu Bar .....	56
4.4. Menu Toolbar .....	57
4.5. Informasi Koordinat.....	58
4.6. Form Layer Peta.....	59
4.7. Form Legend Peta.....	59
4.8. Form Labeling.....	60
4.9. Form Atribut Tabel .....	60
4.9.1. Form Query.....	61
4.9.2. Form Layer Properties .....	61
4.9.3. Form Measure .....	62
4.9.4. Form Identify .....	62
4.9.5. Form Print Layout.....	63
4.9.6. Form Help .....	64
4.9.7. Form About.....	64
4.9.8. Kelebihan Aplikasi Mapview .....	65
4.9.9. Kelemahan Aplikasi Mapview.....	65

#### **BAB V PENUTUP**

3.4.3. Membuat Topologi .....	35
3.4.4. Export Data .....	39
3.4.5. Pembuatan Basis Data .....	40
3.4.6. Memulai Operasi ArcGIS .....	41
3.4.7. Penggabungan Data ( <i>Join</i> ).....	43
3.4.8. Penyimpanan Hasil Olahan Data .....	44
3.4. Tahapan Pembuatan Aplikasi Mapview .....	44
3.5.1. Memulai Operasi Microsoft Visual Studio 2010.....	45

#### **BAB IV HASIL DAN PEMBAHASAN**

4.1. Aplikasi Mapview .....	55
4.2. Form Utama .....	55
4.3. Menu Bar .....	56
4.4. Menu Toolbar .....	57
4.5. Informasi Koordinat.....	58
4.6. Form Layer Peta.....	59
4.7. Form Legend Peta.....	59
4.8. Form Labeling.....	60
4.9. Form Atribut Tabel .....	60
4.9.1. Form Query.....	61
4.9.2. Form Layer Properties .....	61
4.9.3. Form Measure .....	62
4.9.4. Form Identify .....	62
4.9.5. Form Print Layout.....	63
4.9.6. Form Help .....	64
4.9.7. Form About.....	64
4.9.8. Kelebihan Aplikasi Mapview .....	65
4.9.9. Kelemahan Aplikasi Mapview.....	65

#### **BAB V PENUTUP**



5.1. Kesimpulan ..... 66

5.2. Saran ..... 66

**DAFTAR PUSTAKA**

**LAMPIRAN**

## DAFTAR GAMBAR

Gambar 2.1. Komponen SIG(Prahasta, 2005) .....	11
Gambar 3.2.1 Flowchart Perancangan Aplikasi.....	30
Gambar 3.2.2 Flowchart Sampel Data .....	31
Gambar 3.2.3 Contoh Penggunaan Trim .....	34
Gambar 3.2.4 Contoh Penggunaan Extend .....	35
Gambar 3.2.5 Langkah Kerja Drawing Cleanup.....	35
Gambar 3.2.6 Langkah Kerja Kotak Drawing Cleanup .....	36
Gambar 3.2.7 Langkah Kerja Kotak Cleanup Actions .....	36
Gambar 3.2.8 Kotak Dialog Cleanup Methods.....	36
Gambar 3.2.9 Proses Create Topology .....	37
Gambar 3.3.0 Tampilan Kotak Dialog Created Topology.....	37
Gambar 3.3.1 Proses Create Closed Polylines.....	38
Gambar 3.3.2 Create Closed Polylines .....	38
Gambar 3.3.3 Proses Export Data Hasil Topology .....	39
Gambar 3.3.4 Kotak Dialok Export Options.....	40
Gambar 3.3.5 Memulai ArcMap .....	41
Gambar 3.3.6 Kotak Dialog Untuk Memulai Aplikasi ArcMap.....	42
Gambar 3.3.7 Kotak Dialog Add Data.....	42
Gambar 3.3.8 Langkah Mengaktifkan Perintah Join.....	43
Gambar 3.3.9 Kotak Dialog Join Data.....	43
Gambar 3.4.0 Langkah Membuka Atribut .....	44
Gambar 3.4.1 Memulai ArcMap .....	45
Gambar 3.4.2 Kotak Dialog Untuk Memulai Aplikasi Microsoft Visual Studio 2010...45	
Gambar 3.4.3 Pemilihan Templates Aplikasi yang Akan Dibuat.....	46
Gambar 3.4.4 Penentuan Nama dan Lokasi Penyimpanan Project.....	46
Gambar 3.4.5 Pemilihan Target Framework.....	47
Gambar 3.4.6 Menambahkan Pustaka Dotspatial Control.....	47
Gambar 3.4.7 Penambahan referensi Dotspatial .....	48

Gambar 3.4.8 Tampilan Desain Form Main .....	49
Gambar 3.4.9 Jendela Kode (kiri) dan Jendela property (kanan) .....	50
Gambar 3.5.0 Penambahan Source Code Attribute Table .....	50
Gambar 3.5.1 Penambahan Source Code Print Preview .....	51
Gambar 3.5.2 Penambahan Source Code Labeling.....	51
Gambar 3.5.3 Penambahan Source Code Export To Excel .....	52
Gambar 3.5.4 Penambahan Source Code Symbology .....	52
Gambar 3.5.5 Penambahan Source Code Layer Search Kelurahan.....	53
Gambar 3.5.6 Penambahan Form Baru .....	53
Gambar 3.5.7 Proses Debuging.....	54
Gambar 4.1. Form Tampilan Aplikasi .....	55
Gambar 4.2. Form Koordinat.....	58
Gambar 4.3 Form Layer Peta .....	59
Gambar 4.4 Form Legend Peta .....	59
Gambar 4.5. Form Labeling.....	60
Gambar 4.6. Form Atribut Tabel .....	60
Gambar 4.7. Form Query .....	61
Gambar 4.8. Form Layer Properties.....	61
Gambar 4.9. Form Measure (measure distance dan measure area) .....	62
Gambar 4.9.1. Form Identify .....	63
Gambar 4.9.2. Form Print Layout.....	63
Gambar 4.9.3. Form Help .....	64
Gambar 4.9.4. Form About.....	64
Gambar 4.9.5. Kelemahan Menu Identify.....	65

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Pada era globalisasi kini telah berkembang perubahan yang signifikan terhadap komputer dengan lingkungan yang berbasis *Windows*, munculnya berbagai lingkungan pemrograman yang dirancang untuk meningkatkan keuntungan yang terkait dengan *Windows*. *Visual C#*, adalah contoh lingkungan pemrograman yang memungkinkan pengguna untuk dengan cepat dan efisien membuat program yang mudah digunakan.

*Dotspatial* adalah sebuah pustaka/library GIS yang ditulis untuk *.NET Framework 4* (Mogikanin, 2014). Hal ini memungkinkan pengembang untuk menggabungkan data spasial, analisis dan fungsi pemetaan ke dalam aplikasi mereka atau berkontribusi menghasilkan ekstensi GIS untuk digunakan secara umum. Kemampuan pustaka *Dotspatial* antara lain :

1. Menampilkan peta pada *.NET Windows Forms* atau aplikasi *Web*.
2. Membuka *shapefile*, *grid*, *raster* dan gambar.
3. Melakukan *symbolology* dan pelabelan.
4. Manipulasi dan menampilkan atribut data.
5. Analisis saintis.
6. Membaca data *GPS*.

Ada beberapa alasan seorang programmer dapat memilih untuk mengembangkan aplikasi dengan *Dotspatial*. Dalam beberapa situasi, *Dotspatial* menawarkan banyak manfaat bagi pengembang dengan menyediakan *fleksibilitas*

yang besar yang memungkinkan pengembang untuk membuat aplikasi yang disesuaikan. Semakin kecil *footprint memori*, atau persyaratan *RAM*, memungkinkan untuk aplikasi yang efisien dan efektif.

## 1.2. Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat disusun sebagai berikut:

- a. Bagaimana membuat aplikasi desktop Mapview dengan menggunakan *Dotspatial 1.4* dan *Microsoft Visual C# 2010*
- b. Bagaimana membangun sebuah aplikasi Mapview untuk memberikan informasi tentang peta yang mudah digunakan (*user friendly*)

## 1.3. Tujuan Penelitian

Membangun aplikasi *Mapview* dengan menggunakan *Dotspatial 1.4* dan *Microsoft Visual C# 2010* yang diharapkan dapat memberikan kemudahan bagi seorang user untuk mengetahui segala informasi yang ingin digunakan dalam pengolahan dan penggunaan data berbasis Sistem Informasi Geografis.

## 1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi pada hal-hal sebagai berikut:

1. Aplikasi dibuat berbasis *desktop* dengan menggunakan *Dotspatial 1.4* dan *Microsoft Visual C# 2010*
2. Sampel data menggunakan Peta Kecamatan Blimbing Skala 1:5000

### 1.5. Tinjauan Pustaka

Peta merupakan gambaran dari permukaan bumi dalam skala tertentu dan di gambarkan diatas bidang datar melalui sistem proyeksi. Peta mengandung arti komunikasi, artinya merupakan suatu *signal/chanel* antara pengirim pesan (pembuat peta) dari si pengirim pesan kepada pemakai (*user*). Dengan demikian peta digunakan untuk mengirim pesan, yang berupa tentang informasi tentang realita (Prinhandito, 1989).

*Dotspatial* adalah sebuah pustaka/*Library GIS* yang ditujukan untuk *.NET Framework 4*. Hal ini memungkinkan pengembang untuk menggabungkan data spasial, analisis dan fungsi pemetaan ke dalam aplikasi mereka atau berkontribusi menghasilkan ekstensi *GIS* untuk digunakan secara umum (Mogikanin, 2014). *Dotspatial* adalah proyek *open-source* yang berisi kontrol yang dapat digunakan untuk memanipulasi dan menampilkan informasi geografis.

*Dotspatial* bertujuan untuk menyediakan satu set perpustakaan gratis, *open-source platform* yang konsisten dan dapat diandalkan. *.NET, Mono* dan *Silverlight*, memungkinkan pengembang untuk dengan mudah menggabungkan analisis data spasial dan pemetaan kedalam aplikasi mereka

### 1.6. Sistematika Penelitian

Agar tercapainya sebuah penulisan yang baik dan teratur, maka pada sub ini akan dijelaskan beberapa pembahasan yang merupakan bagian penting dalam penyusunan laporan Tugas Akhir. Sistematika penulisan dalam laporan ini adalah sebagai berikut:

## **BAB I Pendahuluan**

Pada bab ini menguraikan tentang latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, batasan masalah dan Tinjauan Pustaka.

## **BAB II Landasan Teori**

Pada bab ini menguraikan tentang teori-teori yang berkaitan dalam penyelesaian laporan, antara lain : sistem informasi geografis, basis data , teori tentang *C#*, *.Net* dan *dotspatial*.

## **BAB III Metodologi Perancangan Aplikasi**

Pada bab ini berisi tentang deskripsi aplikasi Mapview yang direncanakan dan uraian data-data yang diperlukan dalam pembangunan aplikasi Mapview serta tahap-tahap Pembuatan aplikasi Mapview.

## **BAB IV Hasil Dan Pembahasan**

Pada bab ini berisi tentang analisa hasil yang dicapai setelah prosedur perancangan yang telah dilakukan, dengan landasan teori yang ada dan didasarkan pada parameter-parameter yang sudah ditentukan. Evaluasi tentang Aplikasi Mapview dengan menggunakan Sistem Informasi Geografis.

## **BAB V Penutup**

Pada bab ini berisi kesimpulan dari penelitian, saran-saran dan perbaikan maupun peningkatan laporan penelitian.

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Sistem Informasi

Adapun pengertian sistem dari beberapa literatur dapat dijelaskan sebagai berikut:

##### 2.1.1. Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Jogiyanto, 2001).

Sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling keterhubungannya (inter-relasi) dalam mencapai tujuan atau sasaran bersama (Prahasta, 2005). Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan (McLeod, 2001).

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolahan (*process*) dan sasaran (*objective*) atau tujuan (*goal*).

##### 2.1.2. Karakteristik Sistem

###### 1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai



sifat-sifat dari subsistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

## 2. Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

## 3. Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

## 4. Penghubung Sistem

Penghubung sistem (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran (*output*) dari satu subsistem akan menjadi satu masukan (*input*) bagi subsistem yang lain dan akan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

## 5. Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan peralatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang diproses agar didapat keluaran. Sebagai contoh didalam sistem Komputer, program adalah *maintenance input* yang digunakan

untuk mengoperasikan komputernya sedangkan data adalah *signal input* untuk diolah menjadi informasi.

#### 6. Keluaran Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

#### 7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran

#### 8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali, masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Berdasarkan teori-teori yang dikemukakan diatas, maka dapat disimpulkan bahwa sistem adalah sebagai suatu kumpulan objek atau elemen-elemen yang terangkai dalam interaksi dan saling ketergantungan yang teratur serta saling berinteraksi membentuk kesatuan dalam interaksi yang kuat maupun lemah dengan pembatas yang jelas (Jogianto, 2001).

## 2.2. Konsep Dasar Informasi

Jogianto (2005) menjelaskan bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Sumber informasi adalah data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu.

Data merupakan bentuk yang masih mentah dan perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data tersebut ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh Jhon Burch (1986) disebut dengan siklus informasi

(*information cycle*) atau ada yang menyebutnya sebagai istilah siklus pengolahan data (*data processing cycle*).

Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya berakhir. Robert N. Anthony dan Dearden menyebut keadaan dari sistem dalam hubungannya dengan keberakhirannya dengan istilah *entropy*. Informasi yang berguna bagi sistem akan menghindari proses *entropy* yang disebut dengan *negative entropy* atau *negentropy*.

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timeliness*) dan relevan (*relevance*).

### **2.3. Pengertian Sistem Informasi**

Menurut (Budihar, 1995), sistem informasi adalah suatu sistem manusia-mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi, manajemen, dan pengambilan keputusan dalam organisasi.

Leitch dan Davis (1983) dalam Jogianto (2005) mendefinisikan Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.

Dari beberapa pengertian sistem informasi di atas, maka dapat diambil kesimpulan bahwa sistem informasi merupakan suatu sistem berbasis komputer yang berguna untuk menghasilkan suatu informasi yang berasal dari data yang tersedia yang digunakan untuk mendukung suatu kegiatan dan dalam pengambilan suatu keputusan.

### **2.4. Sistem Informasi Geografis**

#### **2.4.1. Konsep Dasar Sistem Informasi Geografis**

Data SIG terdiri dari dua data yaitu data tabular (*table*) dan data grafis (gambar). Data tersebut dapat diperoleh dengan berbagai cara misalnya dari hasil *survey* dan dari bidang penginderaan jauh (*remote sensing*) sehingga akan didapatkan bentuk dan data dari objek, kejadian (fenomena), atau area yang diamati di permukaan bumi.

Tidak semua data SIG pada awalnya dalam bentuk digital, tetapi adapula yang berupa analog. Contoh data analog misalnya peta hasil penggambaran dan peta hasil cetakan (*print map*). Data analog tidak dapat langsung digunakan dalam analisis GIS secara komputerisasi sehingga harus diubah terlebih dahulu ke dalam bentuk digital. Cara yang dapat digunakan untuk mengubah data analog tersebut misalnya dengan cara digitasi (*digitizing*) baik dengan cara digitasi *on screen* (setelah data analog melalui proses pemindahan melalui mesin pemindai/*scanner*) dan digitasi dengan menggunakan meja digitasi (*digitizer table*).

Data SIG merupakan data yang mewakili kenampakan atau objek yang ada di permukaan bumi dengan bentuk yang lebih sederhana. Oleh sebab itu, untuk lebih mendekati keadaan di bumi maka sistem koordinat menjadi hal yang penting.

Hal yang membedakan objek gambar dengan objek pada data SIG adalah adanya koordinat. Dengan demikian, koordinat harus mutlak diikatkan terlebih dahulu kepada gambar *scanned* sebelum memulai proses digitasi. Digitasi merupakan proses pengubahan data menjadi digital sehingga mudah untuk dilakukan analisis. Baik buruknya data SIG sangat bergantung pada hasil digitasi. Digitasi yang kurang teliti akan mengakibatkan data SIG menjadi kurang bermutu dan tidak layak untuk selanjutnya dilakukan analisis. Dengan demikian, diperlukan proses lanjutan yaitu proses *editing* pada hasil digitasi. Proses *editing* ini akan memberikan hasil digitasi yang lebih baik lagi sehingga layak menjadi data untuk analisis

#### **2.4.2. Pengertian Sistem Informasi Geografis**

Sistem Informasi Geografis adalah suatu komponen yang terdiri dari perangkat lunak, perangkat keras, data geografis dan sumber daya manusia yang bekerja bersama secara efektif untuk menangkap, menyimpan, memperbaiki, memperbarui, mengelola, memanipulasi, mengintegrasikan, menganalisa, dan menampilkan data dalam suatu informasi berbasis geografis (Budiyanto, 2002). Banyak lagi pengertian-pengertian tentang SIG yang dikemukakan oleh para

ahli namun pada prinsipnya mempunyai kesamaan unsur yaitu berupa komponen perangkat keras, perangkat lunak, data geografis, data personel yang saling berkaitan dalam suatu sistem yang memungkinkan untuk perekaman, penyimpanan, analisis dan penayangan dari data geografis secara penuh.

Data input SIG terdiri atas data spasial yang berupa data vektor, raster dan data non spasial yang berupa tabular alfanumerik.

➤ Data spasial

Data yang berisi informasi tentang lokasi dan bentuk-bentuk dari unsur-unsur geografi serta hubungannya yang dibuat dalam bentuk peta. Ada dua macam format data spasial yaitu format vektor dan raster.

1. Format Data Raster

Struktur data dalam bentuk sel yang terbentuk atas baris dan kolom, setiap sel mempunyai satu nilai dan terisi satu informasi, grup dari sel mewakili unsur-unsur.

2. Format Data Vektor

Merupakan tipe data yang menggunakan luasan, garis dan titik untuk menampilkan obyek.

➤ Data Non Spasial

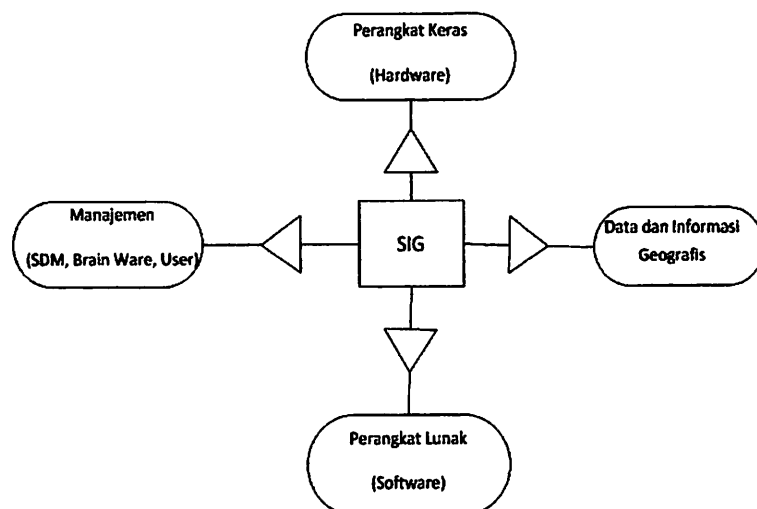
Yaitu data yang berupa angka atau teks yang bersumber dari catatan statistik atau sumber lainnya seperti hasil *survey*, data non spasial ini merupakan pelengkap bagi data spasial karena berfungsi sebagai deskripsi tambahan pada titik, garis, poligon atau batas wilayah.

### **2.4.3. Komponen Sistem Informasi Geografis**

SIG merupakan sistem kompleks yang biasanya terintegrasi dengan lingkungan sistem-sistem komputer yang lain di tingkat fungsional dan jaringan. Sistem SIG menurut Gistut dalam (Prahasta, 2005) terdiri dari beberapa komponen, antara lain:

1. Perangkat Keras :Adapun perangkat keras yang sering digunakan untuk SIG adalah komputer (*PC*), *mouse*, *digitizer*, *printer*, *plotter*, dan *scanner*.
2. Perangkat Lunak : Perangkat lunak SIG menyediakan fungsi untuk masukan, menyimpan, menganalisis dan menampilkan data dalam bentuk geografis. Perangkat SIG yang umum digunakan adalah Arcview, Map Info, ArcGis
3. Data dan Informasi Geografi : SIG dapat mengumpulkan dan menyimpan data yang diperlukan baik secara tidak langsung maupun mengimpornya dari perangkat lunak SIG lainnya maupun secara langsung dengan cara digitasi data spasial dari peta dan masukan data atributnya dari tabel dan laporan dengan menggunakan *keyboard*. Data geografis juga dapat diperoleh dengan membelinya dari penyedia jasa peta seperti Bakosurtanal.
4. Manajemen (*SDM*) : Suatu proyek SIG akan berhasil jika dikelola dengan baik dan dikerjakan oleh orang-orang yang memiliki keahlian yang tepat pada semua tingkatan.

Dari komponen SIG yang telah disebutkan di atas, SIG dapat mempresentasikan *real world* (dunia nyata) di atas monitor komputer sebagaimana lembaran peta dapat mempresentasikan dunia nyata di atas kertas. Adapun Proses untuk mempresentasikannya adalah:



Gambar 2.1 Komponen SIG (Prahasta, 2005)

#### 2.4.4. Subsistem Sistem Informasi Geografis

SIG memiliki beberapa subsistem, diantaranya adalah:

1. Data Input : Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini pula yang bertanggung jawab dalam mengkonversi atau mentransformasikan format-format data-data aslinya ke dalam format yang dapat digunakan oleh SIG.
2. Data Output : Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti table, grafik, peta dan lain-lain.
3. Data Management : Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

#### 2.4.5. Basis Data

Basis data adalah kumpulan data-data (*file*) yang *non-redundant* yang saling terkait satu sama lainnya (dinyatakan oleh atribut-atribut kunci dari table-tabelnya atau struktur data dan relasi-relasi) di dalam usaha membentuk bangunan informasi yang penting (*enterprise*) (Prahasta, 2005).

Kehadiran basis data mengimplementasikan adanya pengertian keterpisahan antara penyimpanan (*storage*) fisik data yang digunakan dengan program-program aplikasi yang mengaksesnya untuk mencegah saling ketergantungan (*dependence*) antara data dan program-program yang mengaksesnya.

Adapun Keuntungan basis data adalah:

1. Reduksi duplikasi data (*minimum redundancy* data yang pada gilirannya akan mencegah inkonsistensi dan isolasi data).
2. Kemudahan, kecepatan dan efisiensi (*data sharing* dan *availability*) akses (pemanggilan) data.
3. Penjagaan Integritas data
4. Menyebabkan data menjadi *self-documented* dan *self-descriptive*.
5. Mereduksi biaya pengembangan perangkat lunak.
6. Meningkatkan faktor keamanan data (*security*)

## 2.5. Peta

### 2.5.1. Pengertian Peta

Peta merupakan gambaran dari permukaan bumi dalam skala tertentu dan digambarkan di atas bidang datar melalui sistem proyeksi. Peta mengandung arti komunikasi, artinya merupakan suatu *signal/chanel* antara si pengirim pesan (pembuat peta) dari si pengirim pesan kepada pemakai (*user*). Dengan demikian peta digunakan untuk mengirim pesan, yang berupa tentang informasi tentang realita. (*Prinhandito.1989*).

### 2.5.2. Fungsi dan Tujuan Peta

Fungsi pembuatan peta adalah:

- a) Menunjukkan posisi atau lokasi relatif (letak suatu tempat dalam hubungan dengan tempat lain di permukaan bumi).
- b) Memperlihatkan ukuran (peta yang dapat diukur luas daerah dan jarak-jarak diatas permukaan bumi).
- c) Memperhatikan bentuk (misalnya: bentuk Benua, Negara, Gunung dan lainnya), sehingga dimensinya dapat terlihat dalam peta.
- d) Mengumpulkan dan menyeleksi data-data dari suatu daerah dan menyajikannya diatas peta. Dalam hal ini dipakai simbol-simbol



“wakil” dari data-data tersebut, sehingga dapat dimengerti oleh pemakai peta.

Tujuan pembuatan peta adalah:

- Untuk komunikasi informasi ruang.
- Untuk menyimpan informasi.
- Digunakan untuk membantu suatu pekerjaan, misalnya: untuk konstruksi jalan, navigasi, perencanaan dan lain-lain.
- Digunakan untuk membantu suatu desain, misalnya: desain jalan dan sebagainya. Untuk analisis data spasial, misalnya: perhitungan volume dan sebagainya.

### **2.5.3. Jenis dan Unsur-Unsur Peta**

#### **2.5.3.1. Jenis Peta**

Peta sangat beragam, berdasarkan kegunaannya peta dibedakan menjadi dua, yakni:

##### **1. Peta Umum**

Peta umum disebut juga dengan peta topografi. Peta umum merupakan peta yang menggambarkan keadaan umum dari suatu wilayah. Keadaan Umum yang digambarkan meliputi objek atau kenampakan alam dan buatan. Objek alam misalnya gunung, sungai, dataran rendah, dataran tinggi, dan laut. Objek buatan misalnya kota, jalan, dan rel kereta api. Peta Indonesia yang sering dipajang di dinding kantor atau sekolah-sekolah merupakan contoh peta umum. Peta Indonesia pada contoh di atas juga termasuk peta umum. Peta umum biasa digunakan untuk belajar di sekolah, untuk kepentingan kantor dan wisata.

## 2. Peta Khusus

Peta khusus merupakan peta yang menggambarkan data-data tertentu di suatu wilayah. Peta khusus disebut juga dengan peta tematik.

Contoh peta khusus adalah:

- a. Peta Persebaran Fauna di Indonesia.
- b. Peta Hasil Tambang di Indonesia.

### 2.5.3.2 Unsur-unsur peta

#### 1. Judul peta

Judul peta merupakan identitas atau nama untuk menjelaskan isi atau gambar peta. Judul peta biasanya terletak dibagian atas peta. Judul peta merupakan komponen yang penting. Biasanya sebelum memperhatikan isi peta, pasti seseorang terlebih dahulu membaca judulnya. Judul peta ditulis dibagian atas dengan huruf yang menonjol. Misalnya, PETA JAWA BARAT, PETA KALIMANTAN, PETA INDONESIA, dan sebagainya.

#### 2. Legenda

Legenda merupakan keterangan yang berisi gambar-gambar atau simbol-simbol beserta artinya. Legenda biasanya terletak dibagian pojok kiri bawah peta

#### 3. Skala

Skala merupakan perbandingan jarak antara dua titik pada peta dengan jarak sebenarnya di permukaan bumi. Misalnya skala 1: 200.000. Skala ini artinya 1 cm jarak pada peta sama dengan 200.000 cm atau 2 km jarak sebenarnya. Sebuah peta selalu dibuat jauh lebih kecil dari aslinya. Akan tetapi, letak, jarak, dan arti keadaan yang sebenarnya. Ada dua macam yaitu skala angka dan skala garis.

1. Petas Khas  
Petas khas merupakan petas yang menggunakan  
material khusus di dalam petas. Petas khas dibuat juga  
dari petas tematik.  
Contoh petas khas adalah:  
a. Petas Petas Khas di Indonesia  
b. Petas Petas Khas di Indonesia

2.3.2 Struktur petas

1. Label petas  
Label petas merupakan identitas atau nama merek  
petas yang terdapat pada label petas biasanya  
dibuat dengan menggunakan petas yang merupakan  
komponen yang penting. Biasanya sebelum menggunakan  
petas pasti seseorang terlebih dahulu membaca label yang  
ada pada label. Label petas dengan huruf yang  
terdapat pada label petas Jawa Barat Petas  
Khas dan Petas Khas dan sebagainya.

2. Struktur petas  
Struktur petas merupakan bagian yang berlainan  
dari petas yang terdapat di bagian atas petas.  
Struktur petas terdapat pada bagian atas petas  
yang terdapat pada bagian atas petas dan bagian  
atas petas. Struktur petas terdapat pada bagian  
atas petas yang terdapat pada bagian atas petas  
dan bagian atas petas. Struktur petas terdapat  
pada bagian atas petas yang terdapat pada bagian  
atas petas dan bagian atas petas. Struktur petas  
terdapat pada bagian atas petas yang terdapat  
pada bagian atas petas dan bagian atas petas.

#### 4. Simbol

Simbol merupakan lambang-lambang atau gambar yang menunjukkan objek alam atau buatan. Simbol peta harus memenuhi tiga syarat yakni sederhana, mudah dimengerti, dan bersifat umum. Berdasarkan kenampakan lingkungannya, simbol dibedakan menjadi 2, yaitu:

##### ❖ Simbol Budaya

Simbol budaya adalah simbol yang mewakili kenampakan budaya, misalnya jalan, rel, kota dan lain-lain.

##### ❖ Simbol Alam

Simbol Alam adalah simbol yang mewakili kenampakan alam misalnya sungai, gunung, danau dan lainnya.

#### 5. Garis astronomis

Garis astronomis merupakan garis khayal di atas permukaan bumi. Garis astronomis terdiri dari garis lintang dan garis bujur. Garis lintang merupakan garis dari timur ke barat. Sedangkan garis bujur merupakan garis dari utara ke selatan. Dalam peta itu terdapat garis-garis tegak (*vertikal*) dan mendatar (*horizontal*). Garis-garis itu disebut garis astronomis. Garis-garis yang tegak disebut garis bujur, sementara garis-garis yang mendatar disebut garis lintang. Garis astronomis berguna untuk menentukan letak suatu tempat atau wilayah.

#### 6. Mata angin

Mata angin merupakan pedoman atau penunjuk arah mata angin. Mata angin pada peta biasanya berupa tanda panah yang menunjukkan ke atas (utara). Mata angin sangatlah penting keberadaannya supaya tidak terjadi kekeliruan arah. Mata angin juga berarti arah, jurusan, atau kiblat suatu tempat. Penunjuk arah mata angin dalam peta

sangatlah penting. Penunjuk mata angin membantu kita bisa menjelaskan posisi suatu tempat.

#### 7. Garis Tepi

Garis Tepi merupakan garis yang dibuat mengelilingi gambar peta untuk menunjukkan batas peta tersebut. Garis tepi peta adalah batas-batas pinggir gambar peta. Fungsi garis tepi untuk menulis angka-angka derajat astronomis (lintang dan bujur).

#### 8. Tahun pembuatan peta

Tahun pembuatan peta menunjukkan kapan peta tersebut dibuat. Dari tahun pembuatan kita dapat mengetahui peta tersebut masih sesuai atau tidak untuk digunakan saat ini.

#### 9. *Insert* peta

*Insert* peta merupakan gambar peta yang ingin diperjelas atau karena letaknya di luar garis batas peta. *Insert* peta digambarkan bila diperlukan. *Insert* peta disebut juga peta sisipan.

#### 10. Tata warna

Tata warna merupakan pewarnaan pada peta untuk membedakan objek satu dengan yang lainnya. Berikut adalah arti warna-warna dalam peta. Warna pada peta memiliki makna sendiri. Penggunaan warna hijau identik dengan dataran rendah dan tutupan vegetasi. Biru untuk perairan, dan cokelat untuk daratan. Lima warna yang umum digunakan pada peta adalah sebagai berikut:

- ❖ Warna merah dan hitam umumnya digunakan untuk mewakili hasil budaya manusia, meliputi jalan, batas daerah, kota, dan lain sebagainya. Merah dan hitam juga digunakan untuk mewakili gunung api aktif (warna merah) dan yang tidak aktif (hitam)

- ❖ Warna hijau untuk mewakili kenampakan vegetasi dan biasanya juga digunakan untuk mewakili dataran rendah.
- ❖ Warna biru untuk mewakili perairan seperti danau, sungai, dan laut. Semakin tua warnah biru pada peta maka semakin dalam suatu perairan.
- ❖ Warna kuning dan cokelat untuk mewakili dataran tinggi dan pegunungan. Makin tua warna cokelat di suatu wilayah pada peta, makin tinggi *relief* wilayah tersebut.
- ❖ Warna putih untuk mewakili kenampakkan *gletser* di muka bumi. Misalnya, untuk mewakili daerah kutub dan *gletser* di atas pegunungan tinggi.

#### 11. Tata Penulisan (*Lettering*)

Pada peta terdapat aturan-aturan penulisan objek-objek geografis. Setidak-tidaknya terdapat empat aturan penulisan dalam peta yang harus dipatuhi

- ❖ Nama-nama ibu kota, Negara, benua, dan pegunungan harus ditulis dengan huruf kapital tegak.
- ❖ Nama-nama samudera, teluk yang luas, laut, dan selat yang luas harus ditulis dengan huruf kapital miring.
- ❖ Nama-nama kota kecil dan gunung harus ditulis dengan huruf kecil tegak. Awal nama kota dan gunung ditulis dengan huruf besar.
- ❖ Nama-nama sungai, danau, selat yang sempit, dan teluk yang sempit harus ditulis dengan huruf kecil miring.

## 2.6. C# (C Sharp)

C# (*C sharp* atau *see sharp*) sering di anggap sebagai bahasa penerus C++ atau versi canggih dari C++, karena ada anggapan bahwa tanda # adalah panduan dari 4 buah tanda tambah yang di susun sedemikian rupa sehingga membentuk tanda pagar. Akan tetapi terlepas dari benar tidaknya anggapan tersebut, C# adalah sebuah bahasa pemrograman yang sangat menjanjikan. C# adalah sebuah bahasa pemrograman yang berorientasi pada object yang dikembangkan oleh Microsoft dan menjadi salah satu bahasa pemrograman yang mendukung .NET programming melalui Visual Studio.

C# didasarkan pada bahasa pemrograman C++, C# juga memiliki kemiripan dengan beberapa bahasa pemrograman seperti Visual Basic, Java, Delphi, dan tentu saja C++. C# memiliki kemudahan *syntax* (cara penulisan) seperti Visual Basic dan tentu saja ketangguhan seperti java dan C++. Kemiripan-kemiripan ini tentunya memudahkan programmer dari berbagai latar belakang bahasa pemrograman tidak perlu waktu yang lama untuk menguasainya, karena bagaimanapun juga C# lebih sederhana dibandingkan bahasa-bahasa pemrograman seperti C++ dan Java.

C# didesain oleh program designer dari Microsoft, Anders Hajisberg. Sebelum bekerja pada Microsoft, Anders bekerja di Borland, tempat dia menulis Pascal *compiler*. Sebelum mengembangkan C# Anders pernah mengembangkan j++ untuk Microsoft. Setelah itu Anders mengembangkan C# dan *Common Language Runtime* yang merupakan mesin virtual dan *runtime library* yang merupakan salah satu tiang utama teknologi .NET. Anders mengetahui berbagai macam kekurangan pada C++, Delphi, Java, dan Smaltalk, karena itu Anders menciptakan bahasa C# yang lebih tangguh. Hal ini juga menjelaskan mengapa C# memiliki kemiripan dengan beberapa bahasa tertentu.

C# bergantung pada CLR yang juga merupakan sumber *library* dari program *Net* lain. Semua program C# memerlukan CLR (berarti juga memerlukan *Net Framework*) untuk dapat dijalankan. Sama halnya dengan Visual Basic 6 yang memerlukan *runtime library* tertentu untuk dapat dijalankan. Bahasa C# dapat

digunakan untuk menciptakan aplikasi Windows, *console class* yang dapat digunakan kembali, dan aplikasi web. (Wahana, 2008 hal 6-7)

Ada beberapa alasan kenapa memilih C#, yaitu :

❖ **Sederhana (simple)**

C# menghilangkan beberapa hal yang bersifat kompleks yang terdapat dalam beberapa macam bahasa pemrograman seperti Java dan C++, termasuk diantaranya menghilangkan *macro*, *templates*, *multiple inheritance* dan *virtual base classes*. Hal-hal tersebut yang dapat menyebabkan kebingungan pada saat menggunakannya, dan juga berpotensi dapat menjadi masalah bagi para programmer C++.

C# bersifat sederhana, karena bahasa ini didasarkan kepada bahasa C dan C++. Jika anda familiar dengan C dan C++ atau bahkan Java, anda akan menemukan aspek-aspek yang begitu familiar, seperti *statements*, *expression*, *operators*, dan beberapa fungsi yang diadopsi langsung dari C dan C++, tetapi dengan berbagai perbaikan yang membuat bahasanya menjadi lebih sederhana.

❖ **Modern**

Apa yang membuat C# menjadi suatu bahasa pemrograman yang modern? Jawabannya adalah adanya beberapa fitur seperti *exception handling*, *garbage collection*, *extensible data types*, dan *code security* (keamanan kode/bahasa pemrograman). Dengan adanya fitur-fitur tersebut, menjadikan bahasa C# sebagai bahasa pemrograman yang modern.

❖ **Object-Oriented Language**

Kunci dari bahasa pemrograman yang bersifat *Object Oriented* adalah *encapsulation*, *inheritance*, dan *polymorphism*. Secara sederhana, istilah-istilah tersebut bisa didefinisikan sebagai berikut :



- ✓ *encapsulation*, dimana semua fungsi ditempatkan dalam satu paket (*single package*).
- ✓ *inheritance*, adalah suatu cara yang terstruktur dari suatu kode-kode pemrograman dan fungsi untuk menjadi sebuah program baru dan berbentuk suatu paket.
- ✓ *polymorphism*, adalah kemampuan untuk mengadaptasi apa yang diperlukan untuk dikerjakan.

Sifat-sifat tersebut di atas, telah dimiliki oleh C# sehingga bahasa C# merupakan bahasa yang bersifat *Object Oriented*.

❖ **Powerfull dan fleksibel**

C# bisa digunakan untuk membuat berbagai macam aplikasi, seperti aplikasi pengolah kata, grafik, *spreadsheets*, atau bahkan membuat *compiler* untuk sebuah bahasa pemrograman.

❖ **Efisien**

C# adalah bahasa pemrograman yang menggunakan jumlah kata-kata yang tidak terlalu banyak. C# hanya berisi kata-kata yang biasa disebut dengan *keywords*. *Keyword* ini digunakan untuk menjelaskan berbagai macam informasi. Jika anda berpikiran bahwa bahasa pemrograman yang menggunakan sangat banyak kata-kata (*keywords*) akan lebih *powerfull*, maka jawabannya adalah “pemikiran itu tidak selalu benar”, karena hal itu justru bisa menambah kerumitan para *developer* pada saat membuat suatu aplikasi.

Berikut daftar *keywords* yang ada dalam bahasa C#:

abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	int
goto	if	implicit	in	long
interface	internal	is	lock	operator
namespace	new	null	object	protected
out	override	params	private	sbyte
public	readonly	ref	return	this
sealed	short	sizeof	stackalloc	uint
static	string	struct	switch	using
throw	true	try	typeof	
ulong	unchecked	unsafe	ushort	
virtual	void	while		

Tabel 2.1 Daftar keywords pada bahasa C#

#### ❖ Modular

Kode C# ditulis dengan pembagian masing Class-Class (*classes*) yang terdiri dari beberapa *routines* yang disebut sebagai *member methods*. Class-Class dan metode-metode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh Class dan metode yang dimaksud, maka kita akan dapat membuat suata kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (*reusable code*)

#### ❖ C# akan menjadi populer

Dengan dukungan penuh dari Microsoft yang akan mengeluarkan produk-produk utamanya dengan dukungan *Framework*

*.NET*, maka masa depan bahasa C# sebagai salah satu bahasa pemrograman yang ada di dalam lingkungan *Framework .NET* akan lebih baik. (Agus Kurniawan, et.al, 2004 hal 14-19.)

## 2.7. *.Net*

*.Net* adalah sebuah teknologi yang diciptakan oleh Microsoft untuk pengembangan program-program yang berorientasi objek. Keistimewaan teknologi ini adalah para *developer* tidak hanya dapat mengembangkan program-program aplikasi biasa, tetapi juga dapat mengembangkan aplikasi-aplikasi internet. Kesimpulannya teknologi *.Net* ini memungkinkan para *developer* mengembangkan software yang berupa aplikasi Windows biasa maupun aplikasi internet yang sangat tangguh dan dapat dijalankan di semua jenis *hardware* dan semua sistem operasi yang memiliki *.Net platform*. Dulu para *developer* VB 6 akan menyertakan *runtime* VB 6 sewaktu akan mendistribusikan programnya supaya program tetap dapat berjalan pada komputer yang tidak mempunyai VB 6. Hal itu tidak diperlukan lagi pada teknologi *.Net* karena program akan langsung dapat dijalankan pada komputer yang sudah memiliki *.Net framework*.

Teknologi *.Net* menyediakan berbagai macam *library*, modul-modul yang sangat memanjakan para *developer* karena dapat mempersingkat waktu pembuatan program. Dan modul-modul serta *library* yang tersedia di dalam teknologi *.Net* ini tidak bergantung pada bahasa pemrograman yang digunakan. (Wahana, 2008 hal 2).

*Framework.NET* adalah suatu komponen windows yang terintegrasi yang dibuat dengan tujuan untuk mensupport pengembangan berbagai macam jenis aplikasi serta untuk dapat menjalankan berbagai macam aplikasi generasi mendatang termasuk pengembangan aplikasi Web Services XML.

*Framework .NET* di design untuk dapat memenuhi beberapa tujuan berikut ini :

- a) Untuk menyediakan *environment* kerja yang konsisten bagi bahasa pemrograman yang berorientasi objek (*object-oriented programming - OOP*) baik kode objek itu di simpan dan di eksekusi secara lokal, atau

dieksekusi secara lokal tapi didistribusikan melalui internet atau di eksekusi secara *remote*.

- b) Untuk menyediakan *environment* kerja di dalam mengeksekusi kode yang dapat meminimaliasi proses *software deployment* dan menghindari konflik penggunaan versi software yang di buat.
- c) Untuk menyediakan *environment* kerja yang aman dalam hal pengeksekusian kode, termasuk kode yang dibuat oleh pihak ketiga (*third party*).
- d) Untuk menyediakan *environment* kerja yang dapat mengurangi masalah pada persoalan performa dari kode atau dari lingkungan *interpreter* nya.
- e) Membuat para *developer* lebih mudah mengembangkan berbagai macam jenis aplikasi yang lebih bervariasi, seperti aplikasi berbasis windows dan aplikasi berbasis web.
- f) Membangun semua komunikasi yang ada di dalam standar industri untuk memastikan bahwa semua kode aplikasi yang berbasis *Framework .NET* dapat berintegrasi dengan berbagai macam kode aplikasi lain.

Sebagai salah satu sarana untuk dapat memenuhi tujuan di atas, maka dibuatlah berbagai macam bahasa pemrograman yang dapat digunakan dan dapat berjalan di atas *platform Framework .NET* seperti bahasa C#, VB .NET, J#, Perl.NET dan lain-lain. Masing-masing bahasa tersebut mempunyai kelebihan dan kekurangannya masing-masing, namun yang pasti, apapun bahasa pemrograman yang digunakan, semuanya akan dapat saling berkomunikasi dan saling *compatible* satu dengan yang lainnya dengan bantuan *Framework .NET*.

*Framework .NET* terdiri dari dua buah komponen utama, yaitu *Common Language Runtime (CLR)* dan *.NET Framework Class Library* atau kadang juga sering disebut dengan *Base Class Library (BCL)*.

*Common Language Runtime (CLR)* adalah pondasi utama dari *Framework .NET*. *CLR* merupakan komponen yang bertanggung jawab terhadap berbagai macam hal, seperti bertanggung jawab untuk melakukan *managemen memory*, melakukan eksekusi kode, melakukan verifikasi terhadap keamanan kode, menentukan hak akses

dari kode, melakukan kompilasi kode, dan berbagai layanan sistem lainnya. Dengan adanya fungsi *CLR* ini, maka aplikasi berbasis *.NET* biasa juga disebut dengan *managed code*, sedangkan aplikasi di luar itu biasa disebut dengan *un-managed code*.

Berikut ini beberapa hal yang disediakan *CLR* bagi para *developer*:

1. Dapat lebih menyederhakan proses pengembangan aplikasi.
2. Memungkinkan adanya variasi dan integrasi dari berbagai bahasa pemrograman yang ada di lingkungan *Framework .NET*
3. Keamanan dengan melakukan indenting pada kode aplikasi.
4. Bersifat *Assembly* pada saat proses *deployment* / kompilasi
5. Melakukan versioning sebuah komponen yang bisa di daur ulang.
6. Memungkinkan penggunaan kembali kode, dengan adanya sifat *inheritance*.
7. Melakukan pengaturan / manajemen tentang *life time* sebuah objek.
8. Melakukan penganalisaan objek-objek secara otomatis.

*CLR* akan melakukan kompilasi kode-kode aplikasi kita menjadi bahasa *assembly MSIL (Microsoft Intermediate Language)*. Proses kompilasi ini sendiri dilakukan oleh komponen yang bernama *Just In Time (JIT)*. *JIT* hanya akan mengkompilasi metode-metode yang memang digunakan dalam aplikasi, dan hasil kompilasi ini sendiri di *chace* di dalam mesin dan akan dikompilasi kembali jika memang ada perubahan pada kode aplikasi kita.

*.NET Framework Class Library* atau sering juga disebut *Base Case Library (BCL)* adalah koleksi dari *reusable types* yang sangat terintegrasi secara melekat dengan *CLR*. *Class library* bersifat berorientasi terhadap objek yang akan menyediakan *types* dari fungsi-fungsi *managed code*. Hal ini tidak hanya berpengaruh kepada kemudahan dalam hal penggunaan, tetapi juga dapat mengurangi waktu yang diperlukan pada saat eksekusi. Dengan sifat tersebut, maka komponen pihak ketiga akan dengan mudah diaplikasikan ke dalam aplikasi yang dibuat.

Dengan adanya *BCL* ini, maka kita bisa menggunakan *Framework .NET* untuk membuat berbagai macam aplikasi, seperti :

- Aplikasi console
- Aplikasi berbasis windows (*Windows Form*)
- Aplikasi ASP.NET (berbasis web)
- Aplikasi Web Services XML
- Aplikasi berbasis *Windows Services*

Jika kita membuat sekumpulan *Class* untuk membuat aplikasi berbasis windows, maka *Class-Class* itu bisa kita gunakan untuk jenis aplikasi lain, seperti aplikasi berbasis web (*ASP.NET*).

Berikut beberapa keuntungan dari *Framework .NET* antara lain:

❖ **Mudah**

Kemudahan di sini lebih ke arah pada kemudahan bagi para *developer* untuk membuat aplikasi yang dijalankan pada lingkungan *Framework.NET*. Beberapa hal yang merepotkan *developer* pada saat membuat aplikasi, telah di hilangkan atau di ambil alih kemampuannya oleh *Framework .NET*, misalnya masalah *life time* sebuah objek yang biasanya luput dari perhatian *developer* pada saat proses pembuatan aplikasi. Masalah ini telah ditangani dan diatur secara otomatis oleh *Framework .NET* melalui komponen yang bernama *Garbage Collector* yang bertanggung jawab untuk mencari dan membuang objek yang sudah tidak terpakai secara otomatis.

❖ **Efisien**

Kemudahan pada saat proses pembuatan aplikasi, akan berimplikasi terhadap efisiensi dari suatu proses produktivitas, baik efisien dalam hal waktu pembuatan aplikasi atau juga efisien dalam hal lain, seperti biaya (*cost*).

## BAB III

### METODE PERANCANGAN APLIKASI

#### 3.1. Persiapan

Sebelum melakukan sebuah pembuatan aplikasi diperlukan persiapan yang cukup untuk mendapatkan hasil yang optimal dalam penelitian yaitu:

##### 3.1.1. Alat dan Bahan Perancangan Aplikasi

Adapun alat dan bahan yang dibutuhkan dalam proses perancangan aplikasi, antara lain:

##### 1. Perangkat keras (*hardware*)

- ❖ *Intel(R) Core™ 2 Duo cpu e7500 @2.93 Ghz (2 CPUs)*
- ❖ *Hard disk 500 GB*
- ❖ *VGA NVIDIA GeForce 9500 GT 1 GB*
- ❖ *RAM 2 GB*
- ❖ *Mouse*
- ❖ *Keyboard*
- ❖ *Monitor Samsung Sync Master 753DFX*

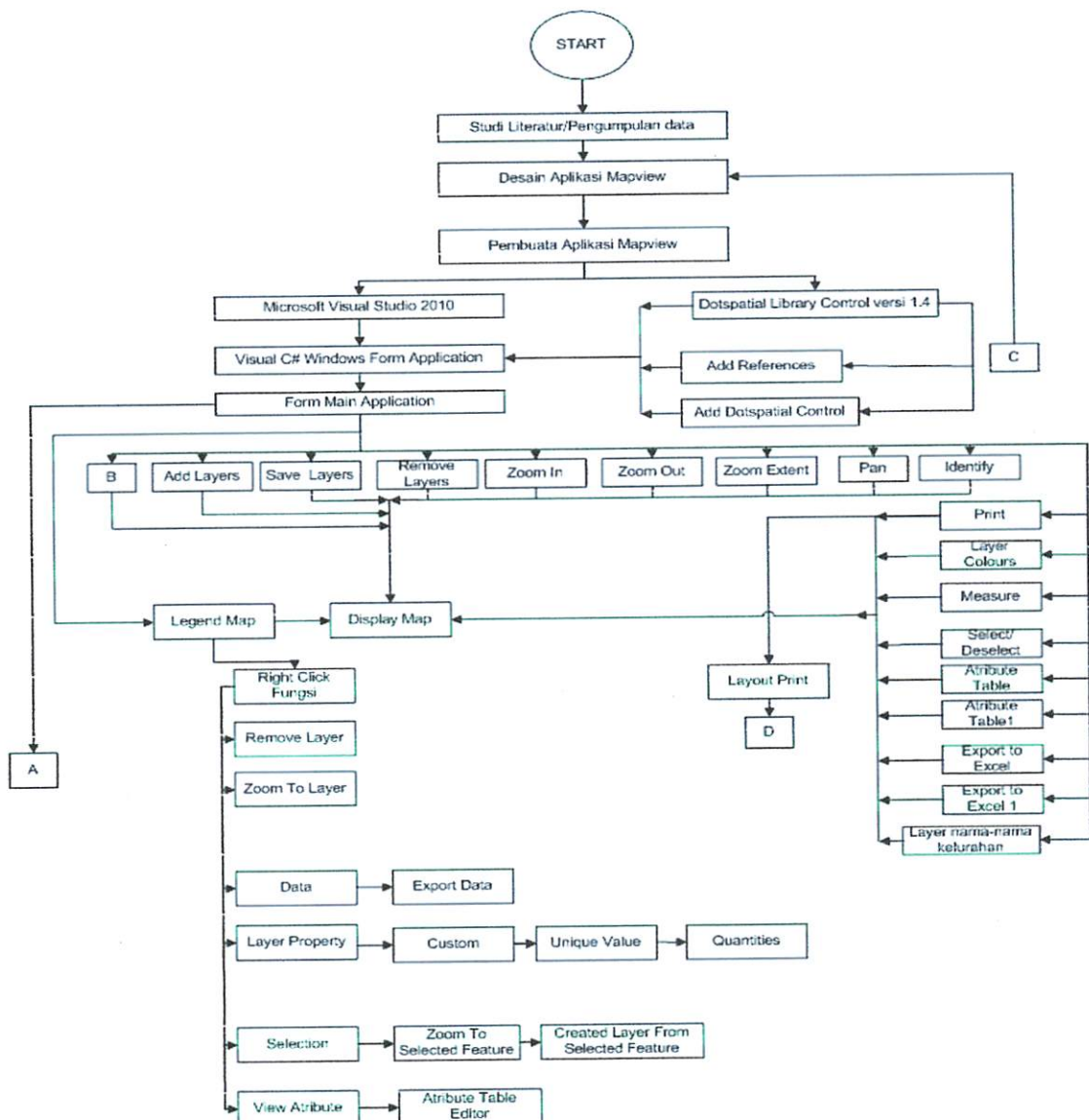
##### 2. Perangkat Lunak (*software*)

- ❖ *Operation Sistem (OS) windows XP SP3*
- ❖ *Dotspatial Versi 1.4 Library Control*
- ❖ *Microsoft Visual Studio 2010*
- ❖ *Notepad++*
- ❖ *ArcGIS 9.3*
- ❖ *Autodesk Map 2004*
- ❖ *Microsoft Office 2007*

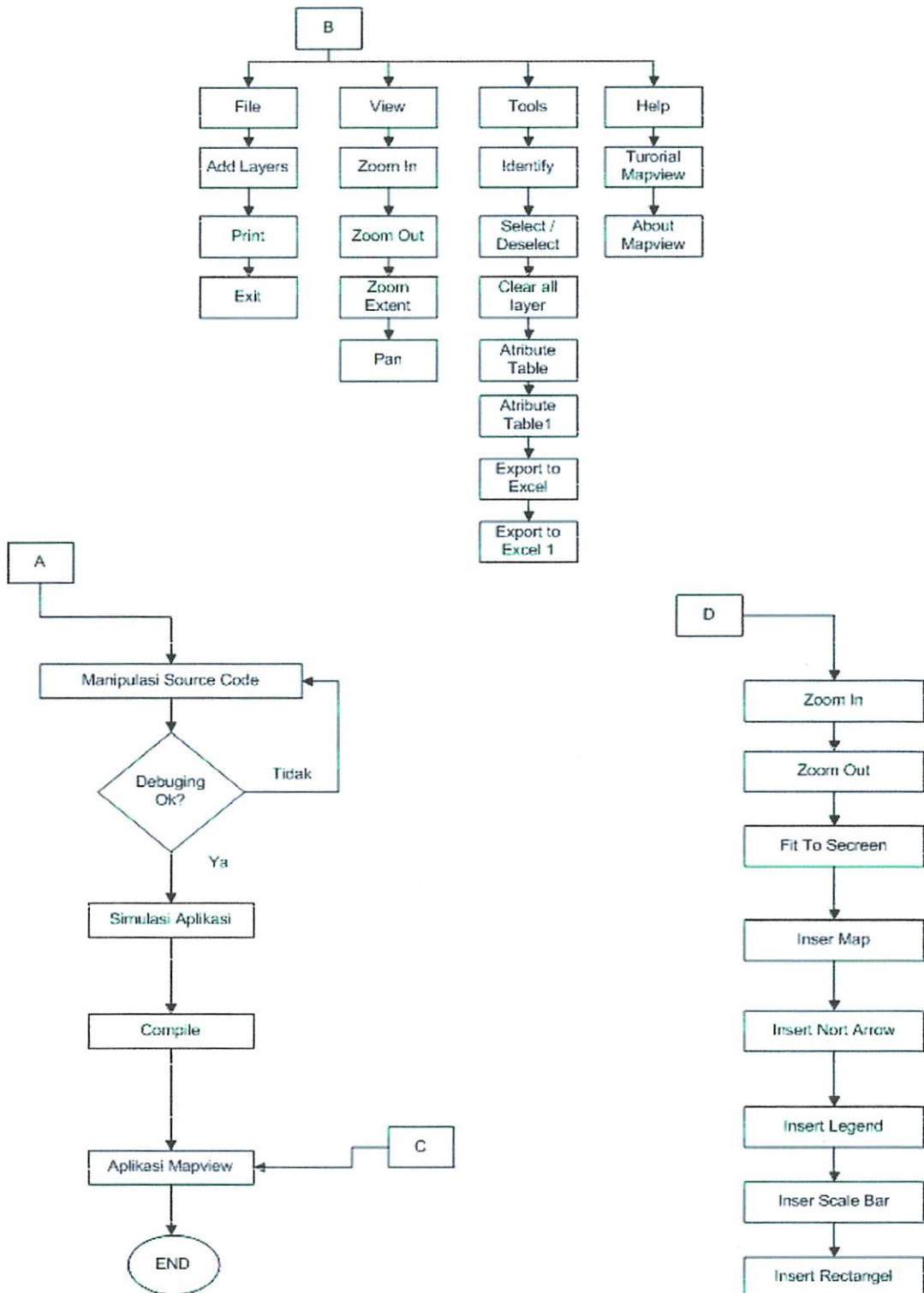
### 3.2. Langkah Perancangan Aplikasi

Dalam proses penelitian haruslah dibuat suatu kerangka pekerjaan yang sistematis agar mudah dipahami dan mempermudah dalam penelitian. Adapun langkah atau alur penelitian yang akan dilakukan sebagai berikut:

#### 3.2.1. Flowchart Perancangan Aplikasi

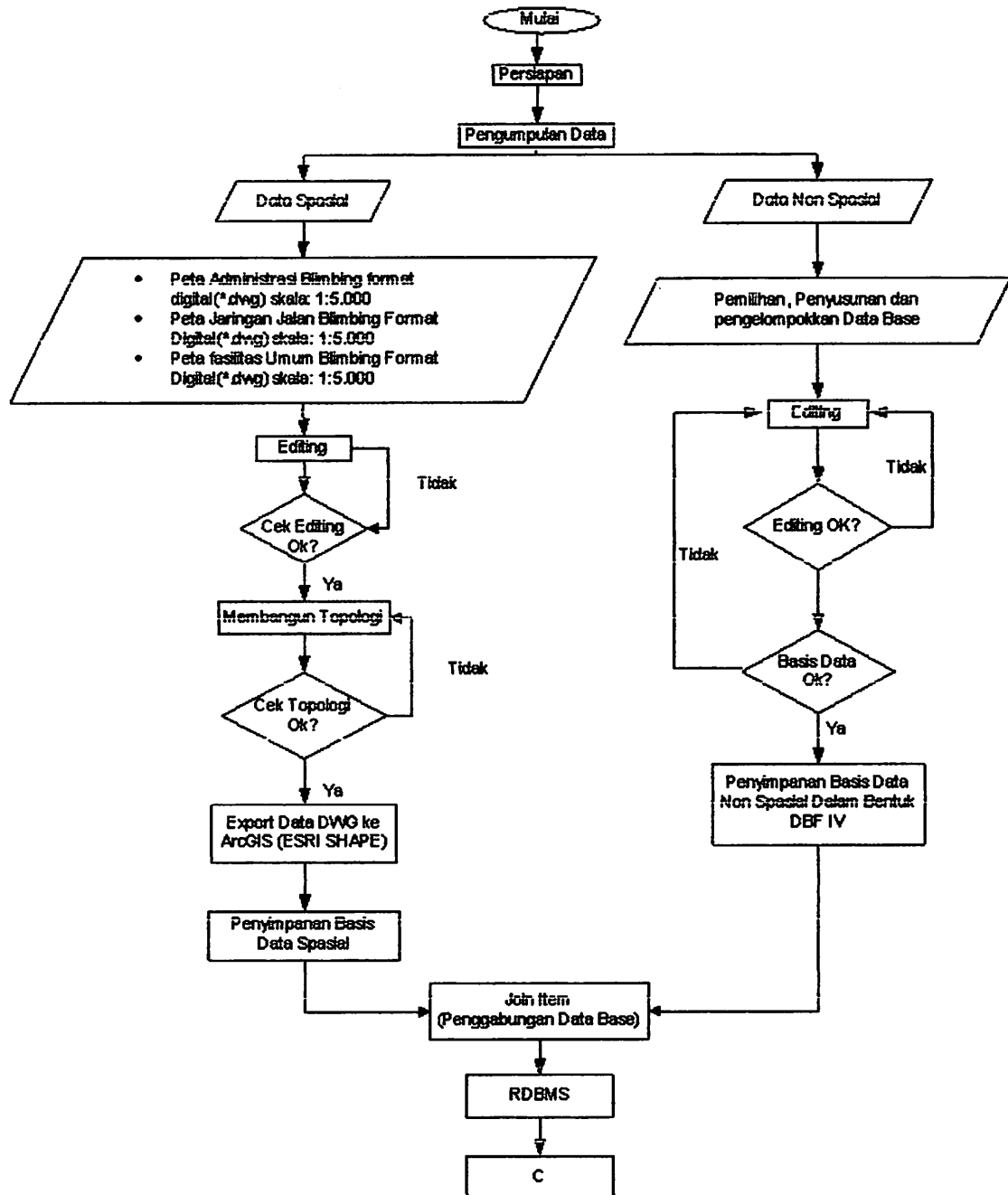






Gambar 3.2.1 Flowchart Perancangan Aplikasi

3.2.2 Flowchart Sampel Data



Gambar 3.2.2 Flowchart Sampel Data

### 3.3. Tahapan Perancangan Aplikasi

#### 3.3.1. Persiapan

Persiapan penelitian yaitu pengumpulan peralatan dan bahan yang dibutuhkan dalam pelaksanaan penelitian meliputi *hardware*, *software*, dan data penunjang lainnya.

#### 3.3.2. Studi Literatur/Pengumpulan Data

Pada tahapan ini dilakukan pengumpulan data berupa *source code* untuk membangun aplikasi yang telah direncanakan. Selain itu dilakukan pengumpulan data spasial dan data non-spasial yang digunakan sebagai data input perlengkapan dalam proses *running* program yang telah dibuat.

#### 3.3.3. Desain Aplikasi Mapview

Pada tahapan ini dilakukan pendesainan aplikasi yang akan dibuat, sesuai dengan tujuan yang telah dipaparkan diatas.

#### 3.3.4. Pembuatan Aplikasi Mapview

Pada tahapan ini dilakukan pembuatan aplikasim *Mapview* Sesuai dengan desain yang telah dibuat. Dalam Pembuatan aplikasi *Mapview* dilakukan manipulasi data *sourcecode* menggunakan *Microsoft Visual Studio 2010* dan *Dotspatial 4.1*

#### 3.3.5. Analisa dan Simulasi Aplikasi

Pada tahapan ini dilakukan analisa terhadap aplikasi yang telah dibuat apakah telah memenuhi tujuan yang telah direncanakan. Analisa didasarkan pada simulasi program pada saat dilakukan *debug* menggunakan *Microsoft Visual Studio 2010*.

### 3.3.6. Validasi

Pada proses validasi dilakukan pengecekan *error* yang terjadi pada saat program dijalankan (*running*). Proses Validasi aplikasi/*debug* di Microsoft *Visual Studio 2010* sehingga *file package sourcecode* aplikasi tersebut dapat *dicompile* menjadi aplikasi *Mapview*.

### 3.3.7. Hasil

Hasil dari *Compile* tadi akan menjadi sebuah aplikasi *Mapview* yang memberikan informasi peta yang bisa dijalankan di *Operation Sistem Windows (OS)*.

## 3.4 . Tahap Persiapan dan Pengumpulan Data

### 3.4.1. Persiapan dan Pengumpulan Data


Pada tahap persiapan ini, meliputi persiapan peralatan yang akan digunakan dalam penelitian baik perangkat keras maupun perangkat lunak, sedangkan pengumpulan data merupakan proses mengumpulkan data-data yang akan digunakan dalam penelitian ini baik yang berupa data spasial berupa peta digital dan data non spasial berupa tabel.

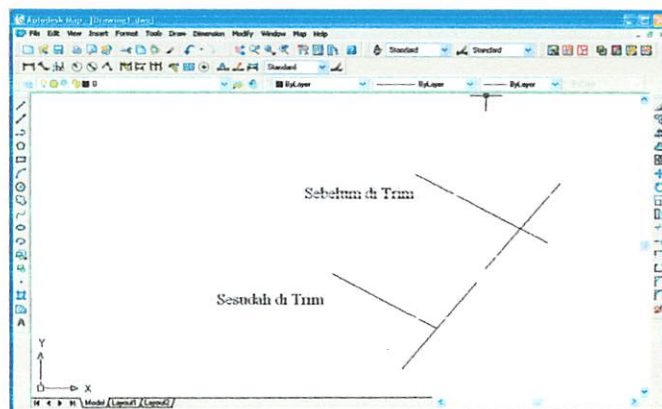
### 3.4.2. Editing Data Spasial

Editing dilakukan untuk memeriksa kembali dan memperbaiki peta digital yang digunakan dari berbagai kesalahan pada proses digitasi yang kurang sempurna. Adapun proses editing ini biasanya menggunakan perintah-perintah yang ada pada software Auto Cad, yaitu :

a. Perintah *trim*

Perintah *trim* digunakan untuk memotong garis yang melebihi batas yang telah kita inginkan. Adapun langkah menggunakan perintah *trim* yaitu :


1. Ketik perintah *trim* pada *command* atau memilih *toolbar trim* .
2. Klik garis yang menjadi batas sebagai acuan memotong garis yang berlebih.
3. Tekan *enter* atau klik kanan pada *mouse*.
4. Kemudian klik garis yang akan dipotong sehingga garis yang melebihi batas tersebut terpotong.

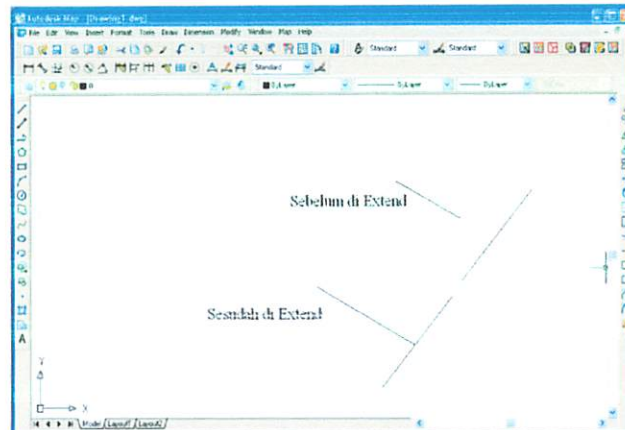


Gambar 3.2.3 Contoh Penggunaan Trim

b. Perintah *extend*

Perintah *extend* digunakan untuk menghubungkan garis yang belum tersambung. Adapun langkahnya sebagai berikut :

1. Ketik perintah *extend* pada *command* atau memilih perintah  *extend*.
2. Klik batas garis yang akan dihubungkan.
3. Tekan *enter* atau klik kanan pada *mouse*.
4. Klik garis yang akan disambungkan sehingga garis yang belum terhubung tersebut sudah benar-benar terhubung.

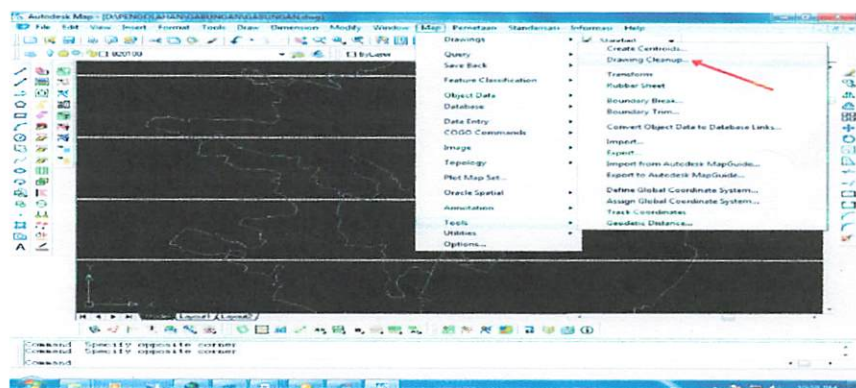


Gambar 3.2.4 Contoh Penggunaan Extend

### 3.4.3. Membuat Topologi

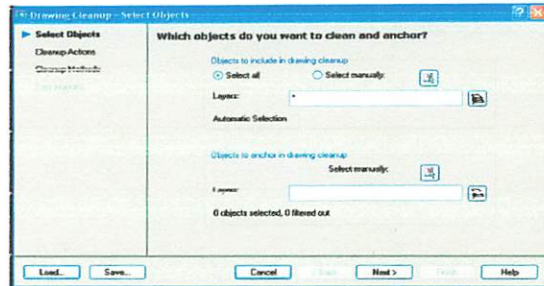
Topologi data merupakan tahap akhir pekerjaan yang dilakukan di AutoCAD. Pembuatan topologi berfungsi untuk membentuk hubungan eksplisit diantara feature geografi pada coverage meliputi connectivity, contiguity, dan definisi area. Proses pembuatan topologi ini membantu untuk mengidentifikasi kesalahan yang terdapat pada data, misalnya Arc yang tidak berhubungan dengan arc lainnya dan polygon yang tidak tertutup. Adapun langkah-langkahnya sebagai berikut :

1. Peta hasil digitasi di Cleanup dengan memilih menu *Map* pada Menu Bar, kemudian pilih *Tools* dan pilih *Drawing cleanup*.



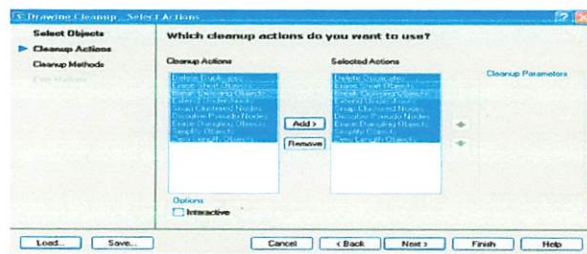
Gambar 3.2.5 Langkah Kerja Drawing Cleanup

2. Pada kotak *Objects to Include Drawing In Cleanup* – Dipilih *Select Manually* setelah itu pilih *Icon* kemudian blok semua peta digitasi lalu pilih *Kanan* dan *Next*.



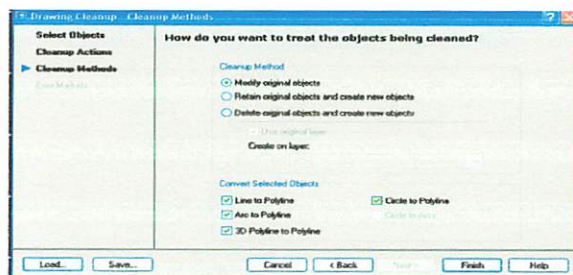
Gambar 3.2.6 Langkah Kerja Kotak Drawing Cleanup

3. Pada kotak *Cleanup Actions* pindahkan semua perintah – perintah ke kotak *Selected Actions* setelah itu pilih *Next*.



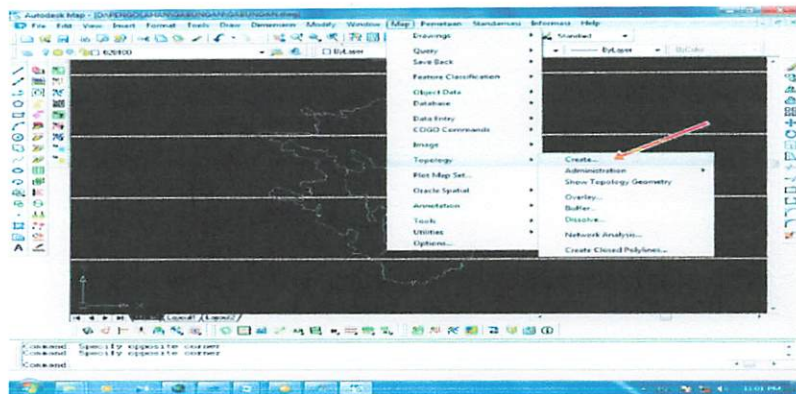
Gambar 3.2.7 Langkah Kerja Kotak Cleanup Actions

4. Pilih semua *Objects* pada kotak *Convert Selected Objects*.



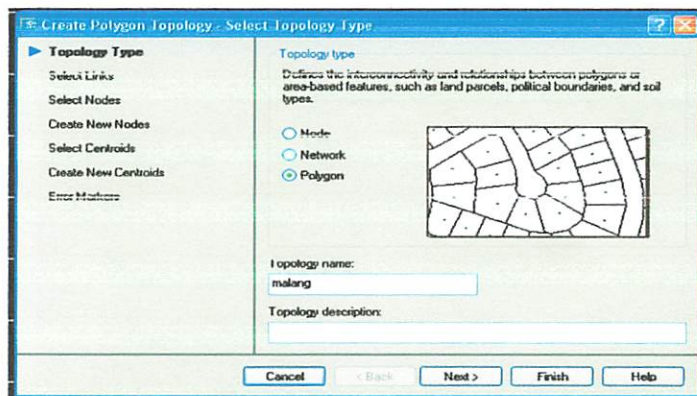
Gambar 3.2.8 Kotak Dialog Cleanup Methods

5. Pilih *Finish*.
6. Pilih menu *Map*, kemudian pilih menu *Topology* dan pilih *Create Topologi*.



Gambar 3.2.9 Proses Create Topology

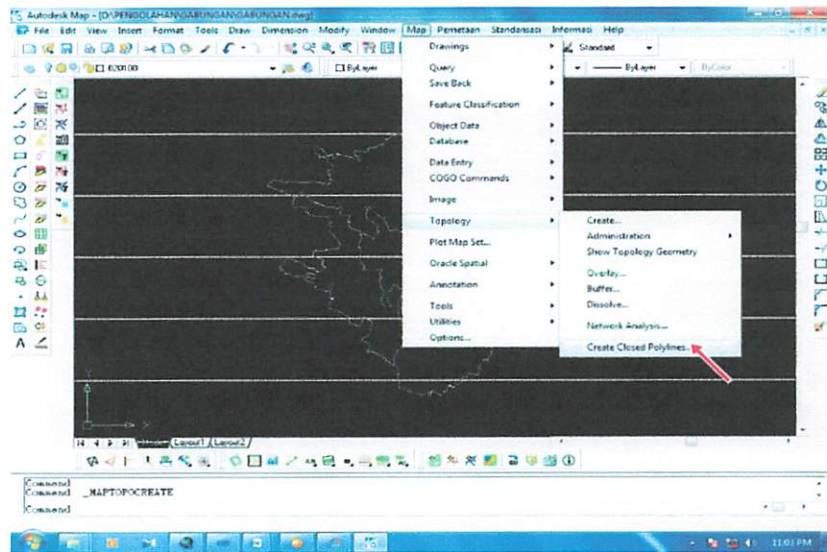
7. Isikan *topology name*. Pada proses topologi terdapat tujuh tahap, dimana ketujuhnya harus dilakukan secara berurutan. Urutan-urutannya ialah memilih ketujuh langkah tersebut secara bertahap yang setiap tahap disertai *select manually* atau bias juga langsung dengan *select all*, untuk mengakhiri proses topologi pilih *finish*.



Gambar 3.3.0 Tampilan Kotak Dialog Created Topology

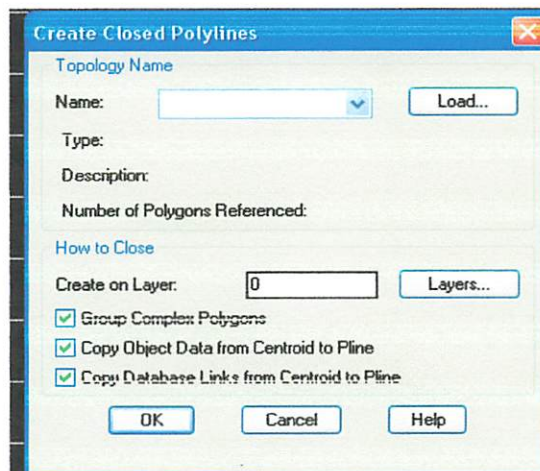
8. Langkah selanjutnya adalah pilih kembali *Map*, kemudian pilih menu *Topology* dan pilih *Create Closed Polylines*.





Gambar 3.3.1 Proses Create Closed Polylines

9. Kemudian pilih semua pada *How To Close* dan setelah itu pilih *Ok*.



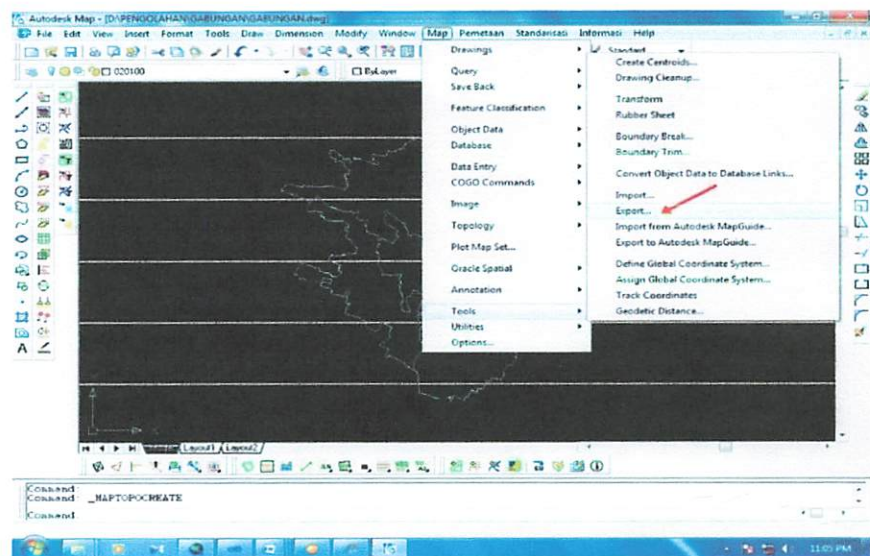
Gambar 3.3.2 Create Closed Polylines

10. Maka peta hasil Digitasi telah menjadi Data Polygon.

### 3.4.4. Export Data

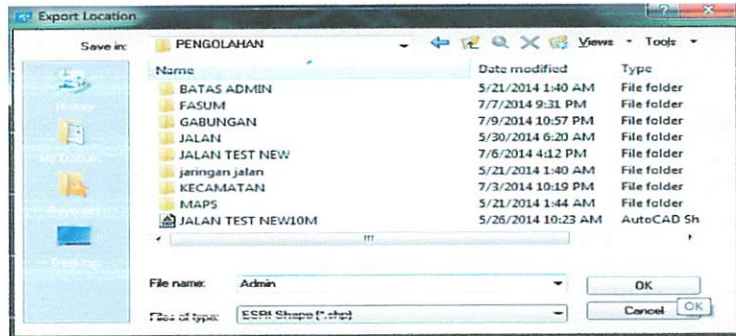
Export data yaitu mengubah format data dari tipe data DWG menjadi ESRI SHAPE agar data bias diproses di software ArcGIS. Adapun langkahnya adalah sebagai berikut :

1. Aktifkan peta administrasi pada AutoCad Land Desktop 2004 hasil proses topologi.
2. Arahkan kursor ke menu *Map* kemudian pilih *Tools* dan pilih *Export*.



Gambar 3.3.3 Proses Export Data Hasil Topology

3. Muncul kotak dialog *Export Location*, merubah export type menjadi *ESRI Shape*. Karena ESRI shape tipe yang dapat dibaca oleh software ArcGIS.
4. Membuat folder data spasial dan data yang akan di export disimpan dengan *FILE NAME* (administrasi). Ganti file of type menjadi *ESRI Shape (.shp)*. Pilih Ok.



Gambar 3.3.4 Kotak Dialog Export Options

### 3.4.5. Pembuatan Basis Data

Pembuatan desain basis data dalam Sistem Informasi Geografis untuk informasi fasilitas umum Kecamatan Belimbing, harus ditentukan entitas terlebih dahulu. Adapun entitas yang digunakan dalam penelitian ini yaitu peta administrasi dan peta jaringan jalan. Masing-masing entitas memiliki Id yang dapat dilihat pada tabel berikut :

Tabel 3.1 Id Kelurahan

ID_Kelurahan	NAMA KELURAHAN
21	KELURAHAN JODIPAN
22	KELURAHAN KSATRIAN
23	KELURAHAN POLEHAN
24	KELURAHAN BUNULREJO
25	KELURAHAN PURWANTORO
26	KELURAHAN BLIMBING
27	KELURAHAN PURWODADI
28	KELURAHAN POLOWIJEN
29	KELURAHAN PANDANWANGI
30	KELURAHAN BALEARJOSARI
31	KELURAHAN ARJOSARI


Tabel 3.2 Id Kecamatan

ID_KECAMATAN	KECAMATAN
9	KECAMATAN LOWOKWARU
10	KECAMATAN SUKUN
11	KECAMATAN KEDUNG KANDANG
13	KECAMATAN KLOJEN

### 3.4.6. Memulai Operasi ArcGIS

Adapun langkah-langkah untuk memulai operasi ArcGIS yaitu :

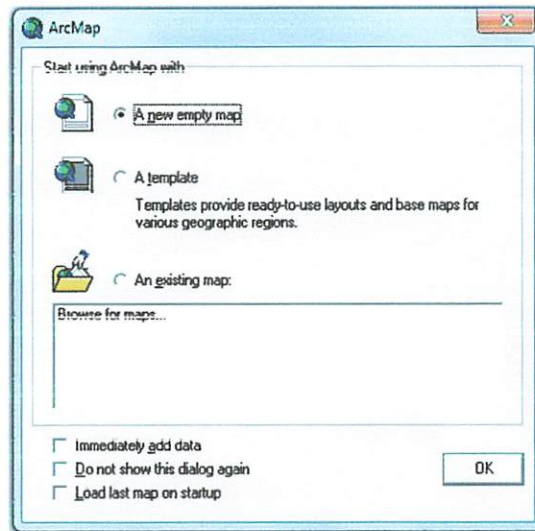
#### a. Memulai aplikasi ArcMap

Pilih icon ArcMap  dari menu Start – Programs – ArcGIS – ArcMap.




Gambar 3.3.5 Memulai ArcMap

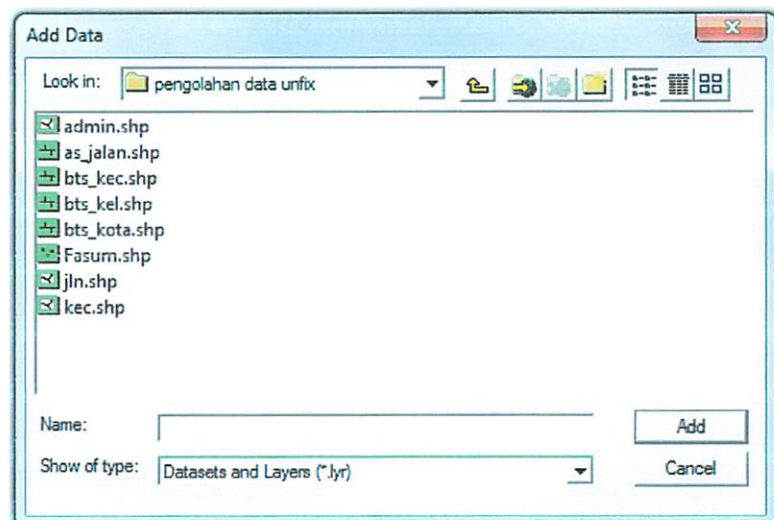
Ketika aplikasi ArcMap terbuka, kita diberikan tiga pilihan untuk memulai aplikasi ini, yaitu memulai dengan map kosong, memulai dengan template yang telah tersedia, atau membuka map yang telah ada. Pilih *A new empty map*.



Gambar 3.3.6 Kotak Dialog Untuk Memulai Aplikasi ArcMap

b. Menampilkan Data Spasial

Pilih pada *Icon Add Data*  , buka directori tempat kita menyimpan hasil ekspor, pilih data yang akan di buka pada ArcMAP klik *Add*.

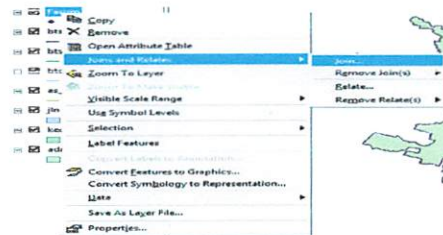


Gambar 3.3.7 Kotak Dialog Add Data

### 3.4.7. Penggabungan Data (*Join*)

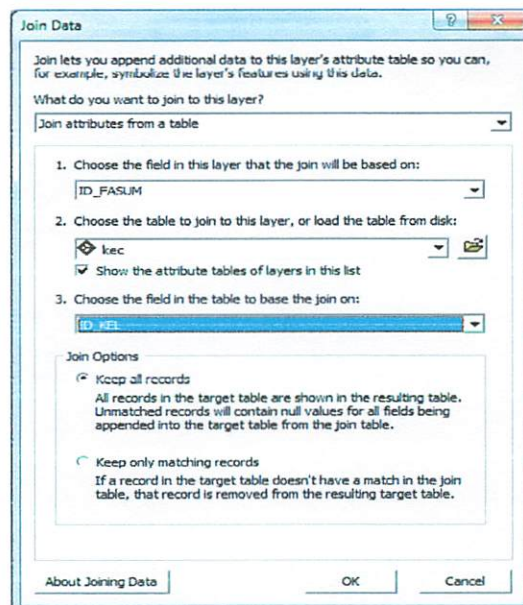
Penggabungan data dilakukan pada perangkat lunak ArcGIS. Maksud dari penggabungan data yaitu menggabungkan data atribut dengan data spasial, sebagai *post identifiernya* adalah ID dari masing-masing data. Adapun tahapannya adalah :

- a. Pilih kanan *layer* yang akan dijoin – *join and relates – join*



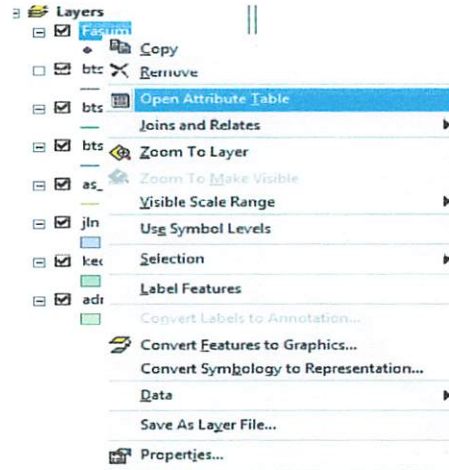
Gambar 3.3.8 Langkah Mengaktifkan Perintah Join

- b. Pada kotak dialog join data isikan ID pada kolom 1 dan 3, sedangkan pada kolom 2 diisi data atribut yang akan digabungkan dengan data spasial.



Gambar 3.3.9 Kotak Dialog Join Data

- c. Edit tabel hasil join dengan cara klik kanan pada layer lalu *open atribut table*.



Gambar 3.4.0 Langkah Membuka Atribut

- d. Pilih *editor* menu dan pilih *start editing*  
 e. Pilih baris yang akan diedit.

Lakukan proses penggabungan data diatas pada data spasial dan data non spasial lainnya secara berurutan.

#### 3.4.8. Penyimpanan Hasil Olah Data

Penyimpanan hasil akhir dari proses pengolahan data SIG dalam format *shp file* yang nantinya diperlukan pada saat proses pengujian/ prosesing runing aplikasi Mapview. Sehingga data tersebut dapat ditampilkan pada aplikasi yang telah dibuat, guna memberikan informasi tentang fasilitas umum khususnya fasilitas umum kecamatan Blimbing.

### 3.5. Tahap Pembuatan Aplikasi Mapview

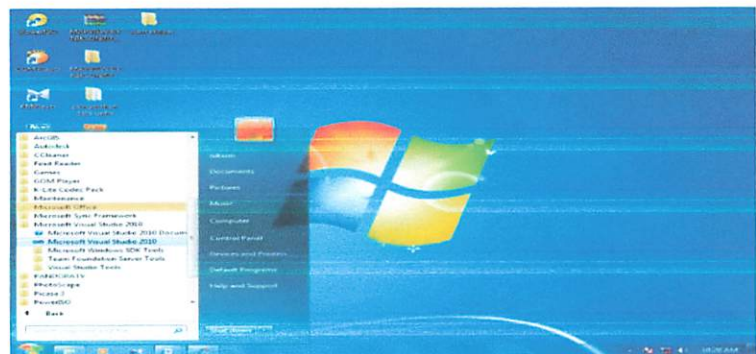
Pada tahapan ini meliputi pembuatan aplikasi mapview yang telah didesain dan dirancang sebelumnya. Pembuatan Aplikasi Mapview menggunakan Software Microsoft Visual Studio 2010 ,dengan pengkhususan pada bahasa pemrograman C#. Adapun tahap-tahapan pembuatan aplikasi mapview antara lain sebagai berikut:

### 3.5.1. Memulai Operasi Microsoft Visual Studio 2010

Adapun langkah-langkah untuk memulai operasi Microsoft Visual Studio 2010 yaitu :

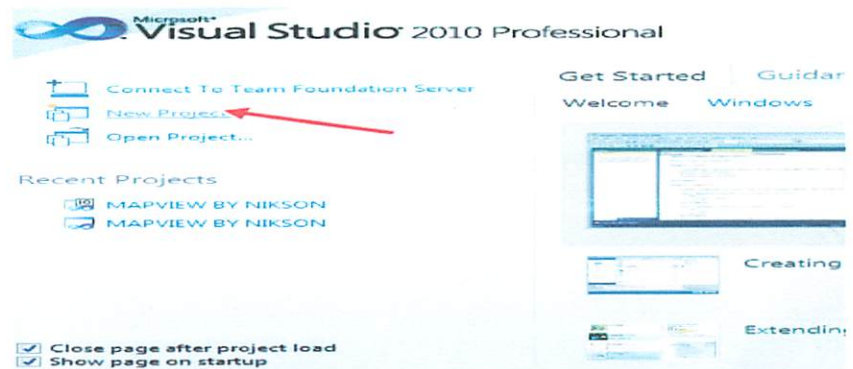
a. Memulai aplikasi Microsoft Visual Studio 2010

Pilih icon Visual Studio 2010  dari menu Start – Programs – Microsoft Visual Studio 2010 – Microsoft Visual Studio 2010.



Gambar 3.4.1 Memulai ArcMap

Ketika aplikasi Microsoft Visual Studio terbuka, kita diberikan tiga pilihan untuk memulai aplikasi ini, yaitu *Connect To Team Foundation Server*, *New Project*, atau *Open Project*. Pilih *New Project*.

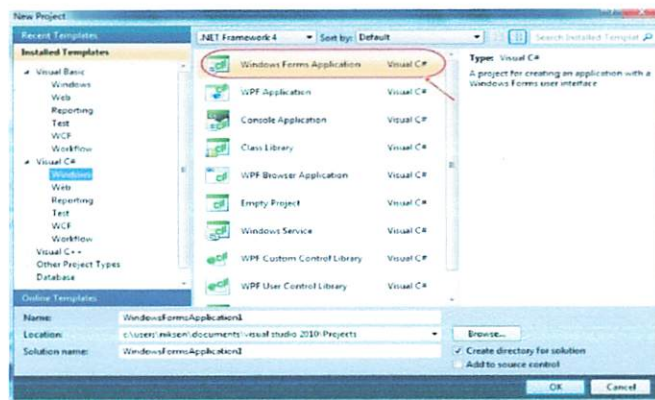


Gambar 3.4.2 Kotak Dialog Untuk Memulai Aplikasi Microsoft Visual Studio 2010



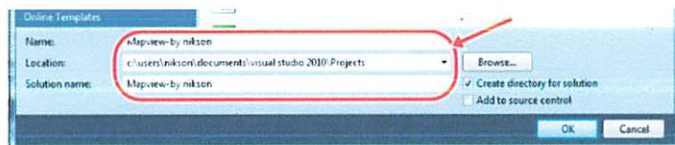
## b. Pemilihan Templates

Disini kita diberikan beberapa pilihan *templates* untuk desain aplikasi yang akan kita buat. Pemilihan *templates* disesuaikan dengan bahasa pemrograman yang kita pilih. Pilih Visual C# – Windows – Windows Forms Application



Gambar 3.4.3 Pemilihan Templates Aplikasi yang Akan Dibuat

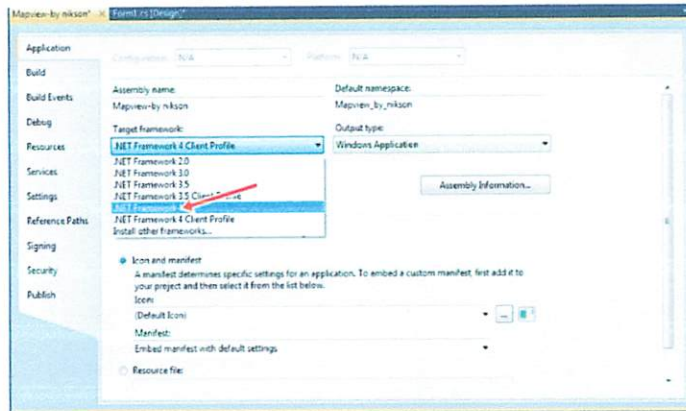
Setelah memilih *Templates* langkah Menentukan nama dan lokasi dari penyimpanan *project* yang akan di buat.



Gambar 3.4.4 Penentuan Nama dan Lokasi Penyimpanan Project

## c. Konfigurasi Project

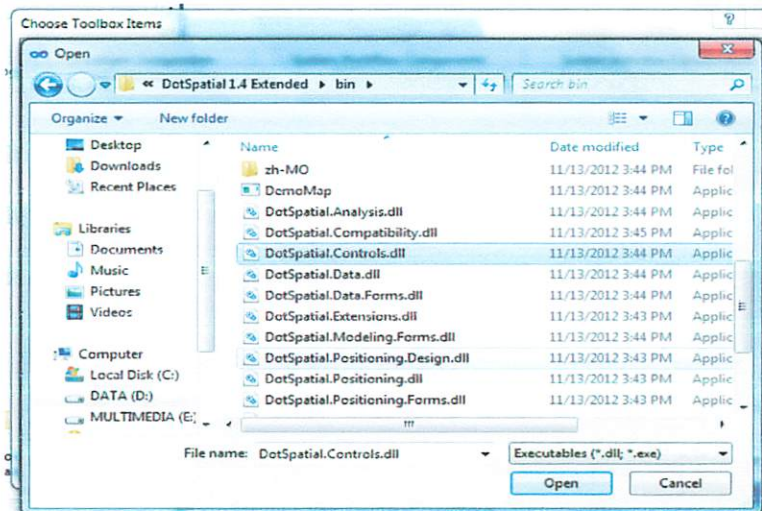
Selanjutnya adalah melakukan konfigurasi awal sebelum *Project* disimpan. Konfigurasi yang dilakukan antara lain pengaturan Target *Framework* dengan memilih “.NET Framework 4”. Hal ini dilakukan karena pustaka *dotspatial* yang akan digunakan hanya dapat bekerja pada lingkaran .NET Framework 4



Gambar 3.4.5 Pemilihan Target Framework

#### d. Menambahkan Pustaka/ Library Dotspatial

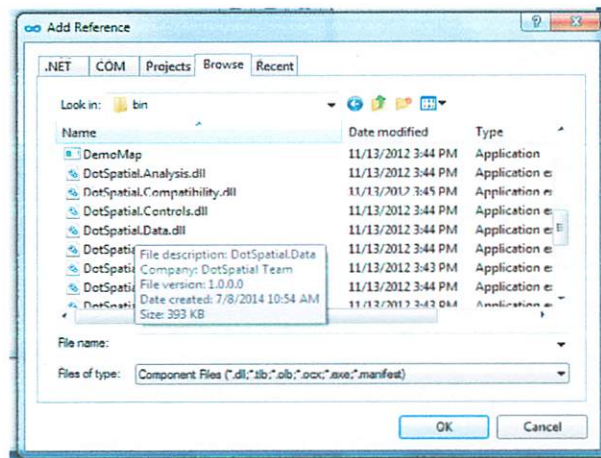
Pada tahapan ini dilakukan penambahan pustaka/ library dotspatial yang bertujuan untuk memunculkan object /komponen dotspatial pada *Toolbox* Microsoft Visual Studio 2010, salah satu komponen yang sangat dibutuhkan adalah *Map* dan *Legend* yang mampu membaca dan menampilkan data-data spasial vektor maupun raster.



Gambar 3.4.6 Menambahkan Pustaka Dotspatial Control

#### e. Menambahkan Referensi Dotspatial

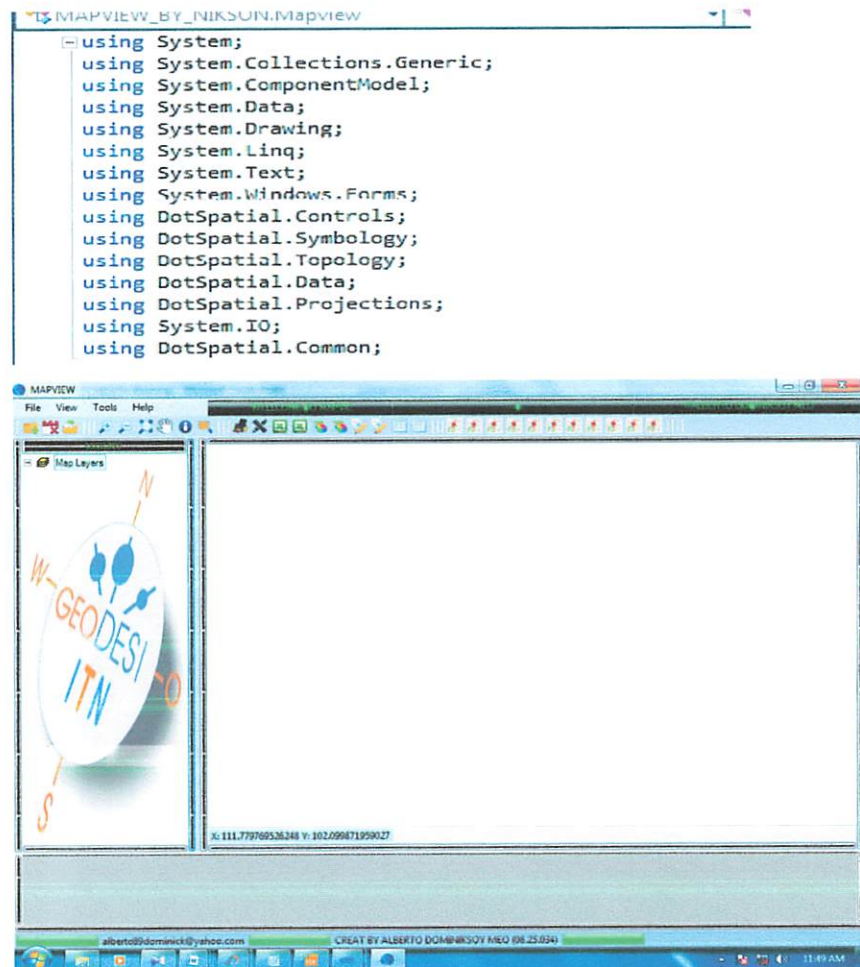
Selanjutnya adalah menambahkan beberapa referensi yang merupakan pustaka dari *Dotspatial*, antara lain : *Dotspatial.Controls*, *Dotspatial.Common*, *Dotspatial.Data*, *Dotspatial.Projections*, *Dotspatial.Symbology*, *Dotspatial.Tools*, *Dotspatial.Topology*, *Dotspatial.Modeling*, *Dotspatial.Positioning*, dan *Dotspatial.Serialization*,



Gambar 3.4.7 Penambahan referensi Dotspatial

#### f. Merancang Desain Antar Muka Aplikasi

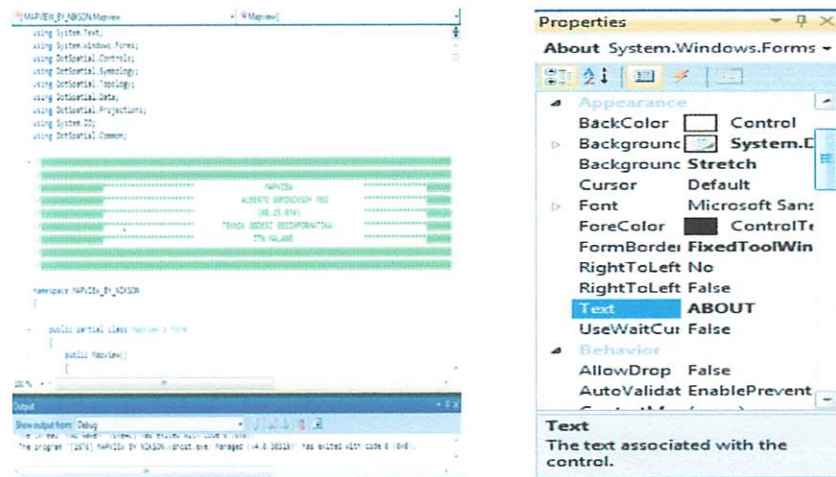
Setelah konfigurasi dan penambahan pustaka selesai maka tahap selanjutnya adalah merancang desain antar muka aplikasi dengan menambahkan berbagai komponen yang dibutuhkan seperti : *Container*, *StatusStrip*, *ToolStrip*, *Map*, *Legend*, *GroupBox*, dan *PictureBox*. *Form* utama aplikasi yang merupakan pusat dari segala aktifitas program dinamakan "*Form Main*", Tampilan dapat dilihat pada gambar 3.4.8 . Pada jendela kode setiap *Form* harus ditambahkan perintah untuk memanggil referensi yang telah ditambahkan dengan perintah sebagai berikut ,



Gambar 3.4.8 Tampilan Desain Form Main

g. Penambahan Properti dan *Source Code* Aplikasi

Tahap selanjutnya adalah mengatur *property* dan menambahkan *source code* agar setiap komponen aplikasi dapat bekerja sebagai mana mestinya . Pada "*Form Main*" , ditambahkan *source code* untuk menambahkan *layer/data* , perangkat *Zoom*, Perangkat *Selection*, Perangkat *Identify*, Perangkat *Measure*, dan memunculkan *Toolbar*. Agar aplikasi terlihat lebih menarik , perlu ditambahkan beberapa icon untuk tombol-tombol yang terdapat pada aplikasi.

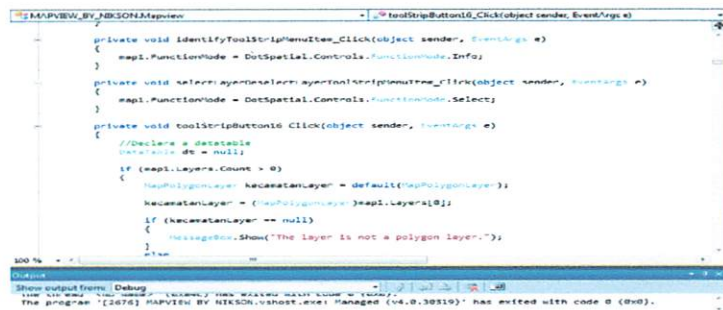


Gambar 3.4.9 Jendela Kode (kiri) dan Jendela property (kanan)

Berikut ini beberapa contoh penambahan *source code* pada *Form Main* Aplikasi:

- Penambahan *Source code* Untuk menampilkan *Atribute Table* Pada Aplikasi

Pada tahap ini dilakukan penambahn *source code* untuk menampilkan *atribut table* dari data *GIS* yang akan ditampilkan pada aplikasi, sehingga *User* dapat mengetahui data-data yang terkandung dalam sebuah *layer* serta informasi yang disampaikan oleh si pembuat peta.



Gambar 3.5.0 Penambahan Source Code Atribute Table



- Penambahan *Source Code Export To Excel*

```

MAPVIEW_BY_NIKSON.Mapview | toolStripButton14_Click(object sender, EventArgs e)
private void toolStripButton14_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecasatanLayer = default(MapPolygonLayer);
        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first layer from the MapControl
        kecasatanLayer = (MapPolygonLayer)map1.Layers[0];

        //Check whether stateLayer is polygon layer or not
        if (kecasatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //add StateName as labels on the stateLayer
            //kecasatanLayer is an attribute from the given example US States shape file.
            map1.AddLabels(kecasatanLayer, "[Nama_kel]", new Font("Tahoma", (float)8.0), Color.DkL
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

```

Output  
Show output from: Debug  
The program '[3564] MAPVIEW BY NIKSON.vshost.exe: Managed (v4.0.30319)' has exited with code 0 (0x0).

Gambar 3.5.3 Penambahan Source Code Export To Excel

- Penambahan *Source Code Symbology*

```

MAPVIEW_BY_NIKSON.Mapview | toolStripButton13_Click(object sender, EventArgs e)
private void toolStripButton13_Click(object sender, EventArgs e)
{
    //check the number of layers from map control
    if (map1.Layers.Count > 0)
    {
        //Declacre a MapPolygonLayer
        MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);

        //Type cast the FirstLayer of MapControl to MapPolygonLayer
        kelurahanLayer = (MapPolygonLayer)map1.Layers[0];

        //Check the MapPolygonLayer( Make sure that it has a polygon layer)
        if (kelurahanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Create a new PolygonScheme
            PolygonScheme scheme = new PolygonScheme();

            //Set the ClassificationType for the PolygonScheme via EditorSettings
            scheme.EditorSettings.ClassificationType = ClassificationType.UniqueValues;

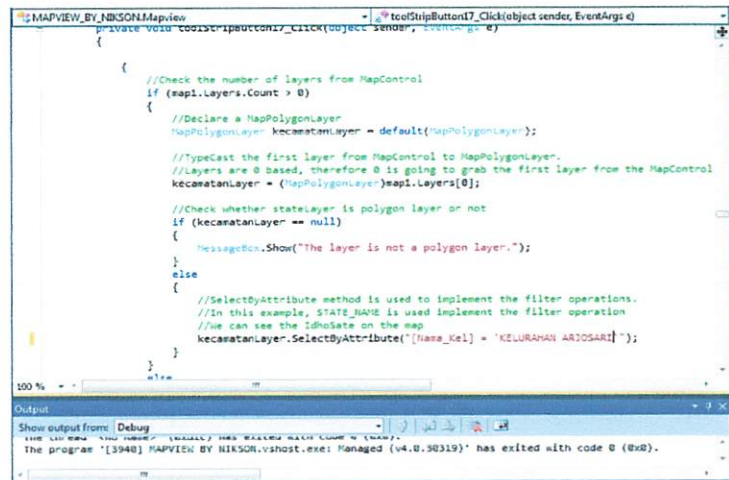
            //Keep the InformalName field name

```

Output  
Show output from: Debug  
The program '[3940] MAPVIEW BY NIKSON.vshost.exe: Managed (v4.0.30319)' has exited with code 0 (0x0).

Gambar 3.5.4 Penambahan Source Code Symbology

- Penambahan *Source Code Shortcut Search Layer*  
Kelurahan



```

private void toolStripButton7_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[0];

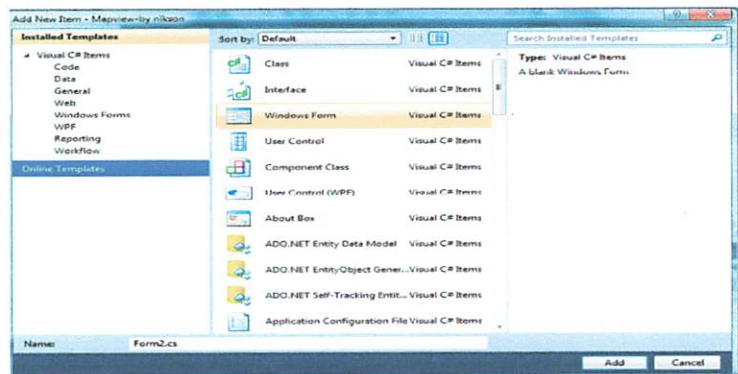
        //Check whether stateLayer is polygon layer or not
        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter operations.
            //In this example, STATE_NAME is used to implement the filter operation
            //You can see the IdState on the map
            kecamatanLayer.SelectByAttribute("[Name_Kel] = 'KELURAHAN ARJOSARI'");
        }
    }
}

```

Gambar 3.5.5 Penambahan Source Code Layer Search Kelurahan

#### h. Pembuatan Form Baru

Setelah pengaturan property dan penambahan *source code* pada aplikasi yang telah didesain tahap selanjutnya adalah pembuatan form baru. Form baru dimaksudkan untuk menampilkan informasi tambahan baik itu informasi tentang orang yang membuat aplikasi atau informasi tentang tutorial menjalankan aplikasi yang telah dibuat.



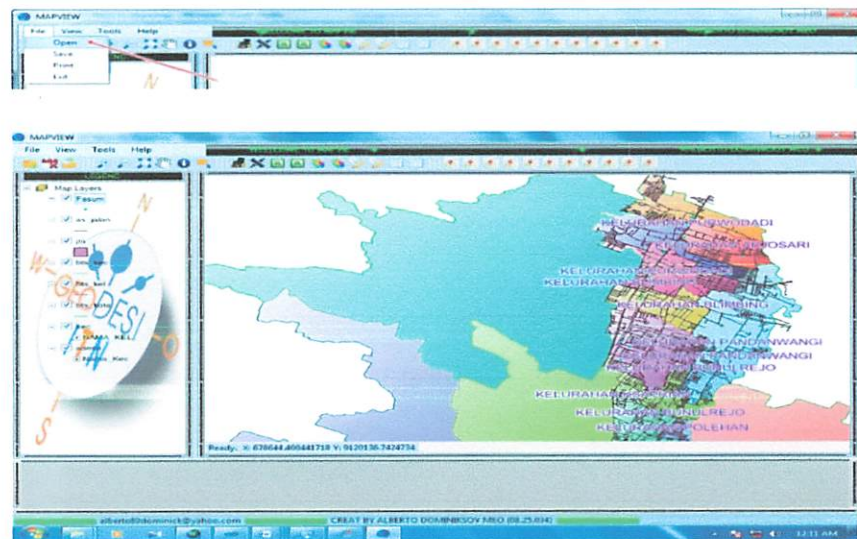
Gambar 3.5.6 Penambahan Form Baru



## i. Validasi

Pada proses validasi dilakukan pengecekan *error* yang terjadi pada saat program dijalankan (*running*). Proses Validasi aplikasi/*debug* di Microsoft *Visual Studio 2010* sehingga *file package sourcecode* aplikasi tersebut dapat *dicompile* menjadi aplikasi *MapView*.

Berikut ini tampilan aplikasi saat dilakukan proses *debugging*:



Gambar 3.5.7 Proses *Debugging*

## BAB IV

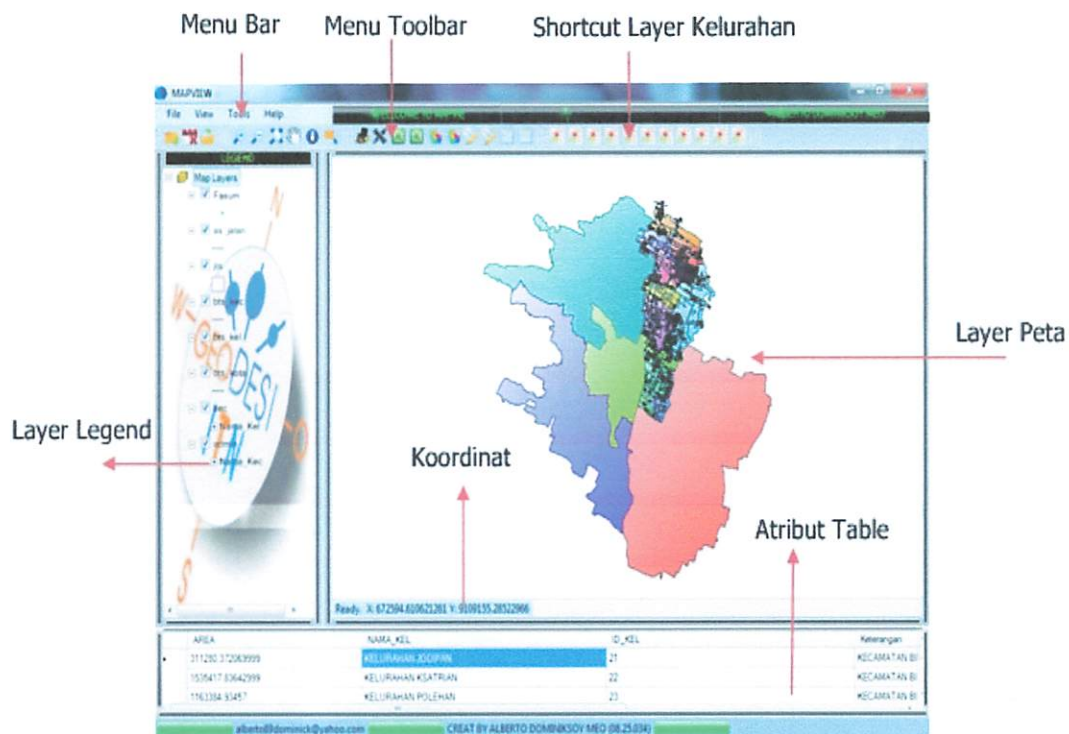
### HASIL DAN PEMBAHASAN

#### 4.1. Aplikasi Mapview

Aplikasi mapview disajikan dalam satu *main form* yang didesain secara sederhana agar dapat mempermudah pengguna untuk mencari informasi yang terkandung dalam sebuah peta, disini penulis menggunakan peta informasi fasilitas umum kecamatan Blimbing sebagai sampel data pada saat aplikasi dijalankan.

#### 4.2. Form Utama

Pada *form* utama ini terdapat menu *bar*, menu *toolbar*, *layer* peta, *layer legend*, koordinat, dan *layer* atribut table, *shortcut* layer kelurahan.



Gambar 4.1. Form Tampilan Aplikasi

### 4.3. Menu Bar








Menu *bar* terdiri dari menu *bar File*, menu *bar View*, menu *bar Tools*, menu *bar Help*. Tiap menu bar memiliki sub menu bar antara lain dapat dilihat pada tabel berikut ini:










Menu bar	Sub menu bar	Fungsi
File	Open	Untuk proses pemanggilan/load data
	Save	Untuk proses Penyimpanan Data
	Print	Untuk proses print informasi peta yang di tampilkan pada aplikasi
	Exit	Untuk mengakhiri program
View	Zoom In	Untuk memperbesar tampilan/display dari Informasi peta yang ditampilkan pada aplikasi
	Zoom Out	Untuk memperkecil tampilan/display dari Informasi peta yang ditampilkan pada aplikasi
	Pan	Untuk menggeser posisi display peta sesuai dengan posisi yang diinginkan
	Zoom Extent	Untuk menampilkan peta secara keseluruhan ,sehingga semua bagian yang dipetakan dapat terlihat oleh user
Tools	Identify	Untuk memberikan informasi pada layer peta
	Select Layer/Deselect Layer	Untuk proses pemilihan/select pada layer yang diinginkan oleh user atau sebaliknya.
	Clear All Layer	Untuk membersihkan semua data pada tampilan layer peta
	Atribut tabel	Untuk menampilkan atribut table layer kecamatan

	Atribut Tabel I	Untuk menampilkan atribut layer kelurahan
	Export To Excel	Untuk proses pengeksporan atribut table layer kecamatan ke Microsoft Excel
	Export To Excel I	Untuk proses pengeksporan atribut table layer kelurahan ke Microsoft Excel
Help	Tutorial Mapview	Untuk memberikan tutorial tentang penggunaan dari aplikasi mapview
	About Mapview	Berisikan informasi singkat tentang si pembuat aplikasi

#### 4.4. Menu Toolbar

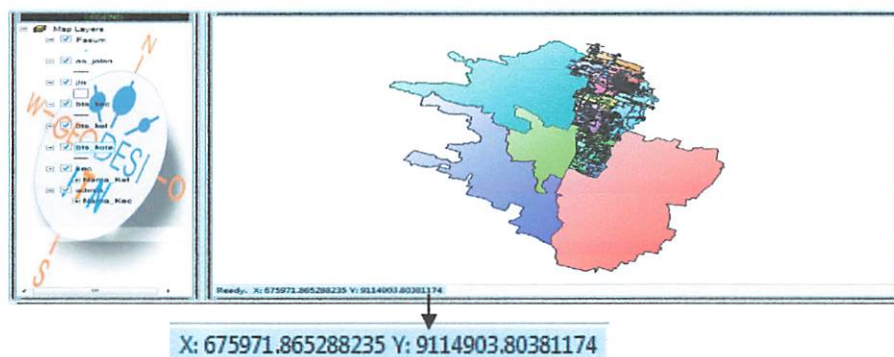
Menu *Toolbar* berguna untuk membantu pengguna dalam melakukan pengoperasian program ini

Toolbar	Fungsi
	Untuk load data
	Untuk menghapus data
	Untuk Menyimpan data
	Untuk memperbesar objek
	Untuk memperkecil object
	Untuk menampilkan object secara keseluruhan
	Untuk menggeserkan peta

	Untuk pemberian informasi pada layer peta
	Untuk pemilihan/select layer yang diinginkan
	Untuk print peta
	Untuk mengukur luasan dan jarak suatu object pada peta
	Untuk export data ke Microsoft Excel
	Untuk symbology peta
	Untuk menampilkan label pada peta
	Untuk menampilkan atribut table data
	Untuk menampilkan layer-layer kelurahan pada peta

#### 4.5. Informasi Koordinat

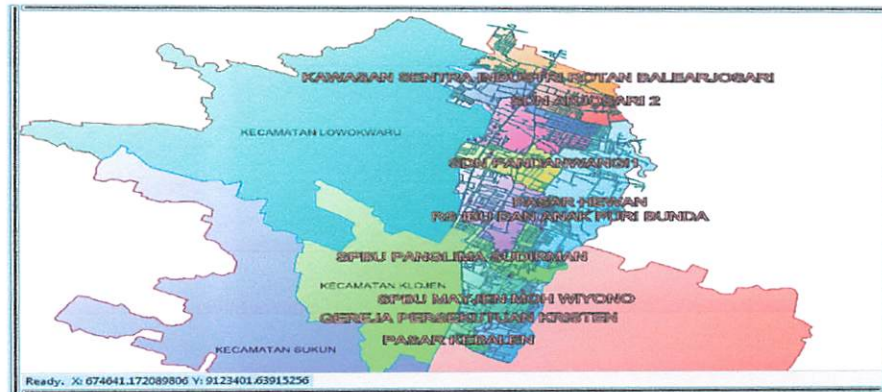
Informasi koordinat disini agar dapat mempermudah pengguna untuk mengetahui koordinat dari masing-masing *object* yang ditampilkan pada peta.



Gambar 4.2. Form Koordinat

#### 4.6. Form Layer Peta

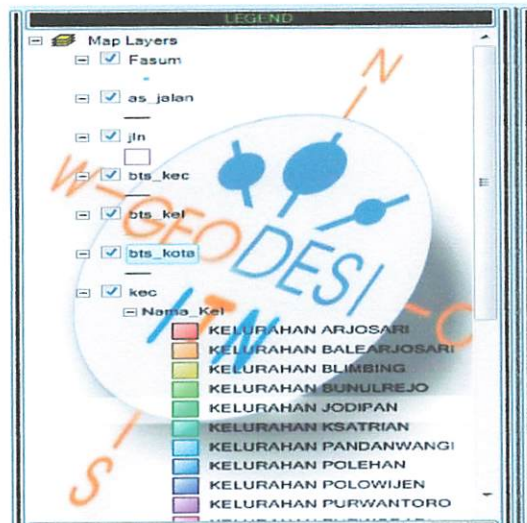
*Form layer* peta memiliki fungsi yang sangat penting dalam program ini dikarenakan semua informasi yang ada langsung terhubung dengan *form layer* peta lengkap dengan koordinat peta. Untuk lebih jelasnya dapat dilihat pada gambar 4.3



Gambar 4.3 Form Layer Peta

#### 4.7. Form Legend Peta

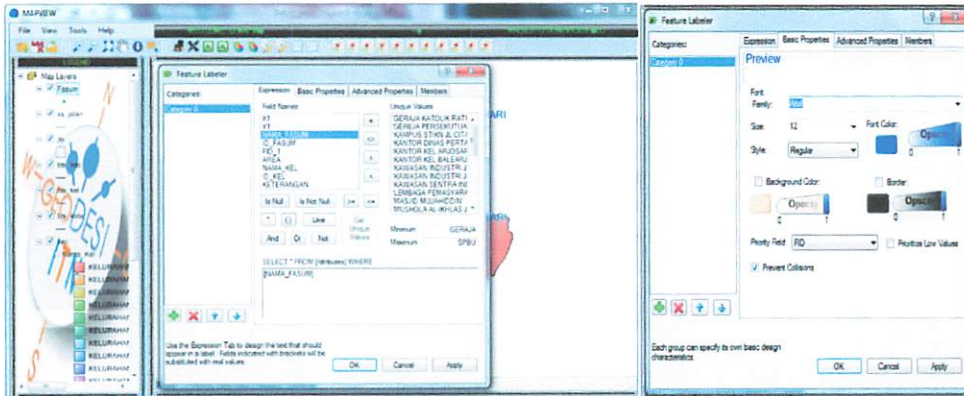
*Form legend* peta memiliki fungsi untuk menampilkan legenda dari *layer* peta yang *diload* serta tipe *object* tersebut ( *point*, *line*, atau *polygon*).



Gambar 4.4 Form Legend Peta

#### 4.8. Form Labeling

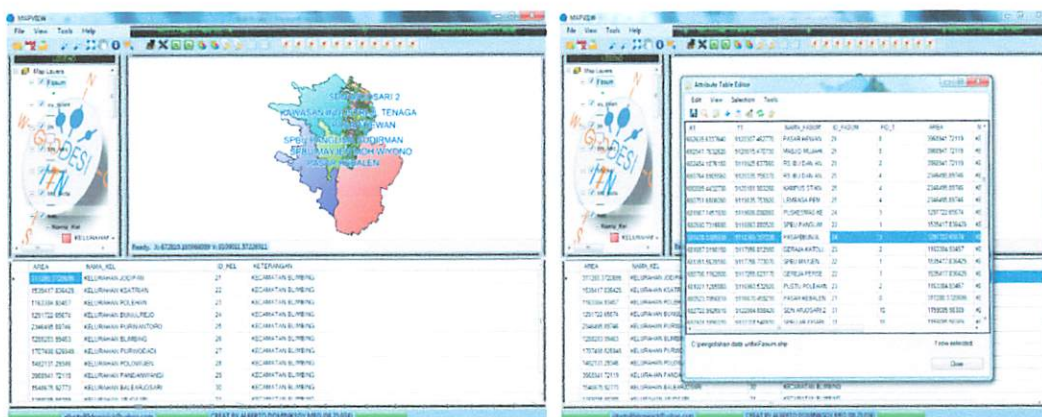
*Form Labeling* memiliki fungsi untuk menampilkan label dari tiap-tiap *object* yang terdapat dalam sebuah peta. Pada proses *labelling* terdapat beberapa pilihan baik itu informasi-informasi apa yang mau ditampilkan sampai pada customasi pewarnaan, *size*/ukuran huruf serta jenis huruf dari label yang akan diberikan.



Gambar 4.5. Form Labeling

#### 4.9. Form Atribut Tabel

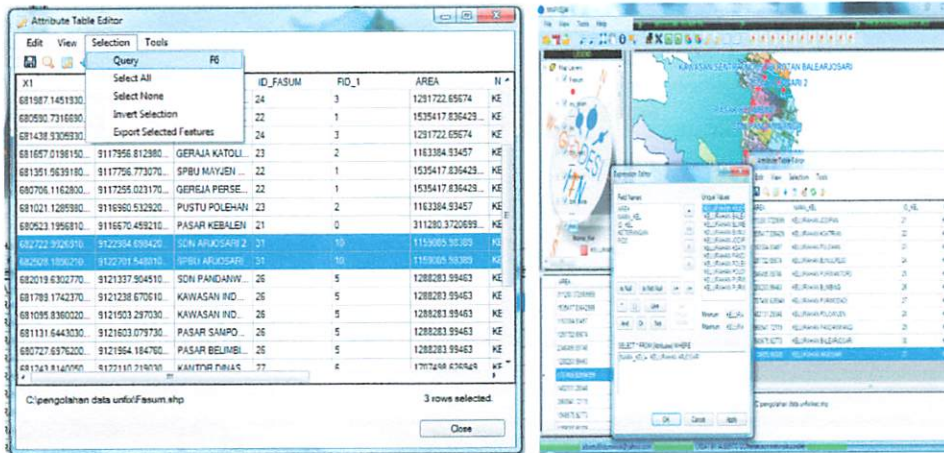
*Form* atribut tabel berfungsi untuk menampilkan atribut tabel dari data/informasi yang ditampilkan pada peta. Pada *form* atribut tabel kita dapat melakukan editing apabila terdapat data yang tidak sesuai yaitu dengan cara *double click* pada kolom dan baris pada atribut table yang akan dilakukan editing.



Gambar 4.6. Form Atribut Tabel

#### 4.9.1. Form Query

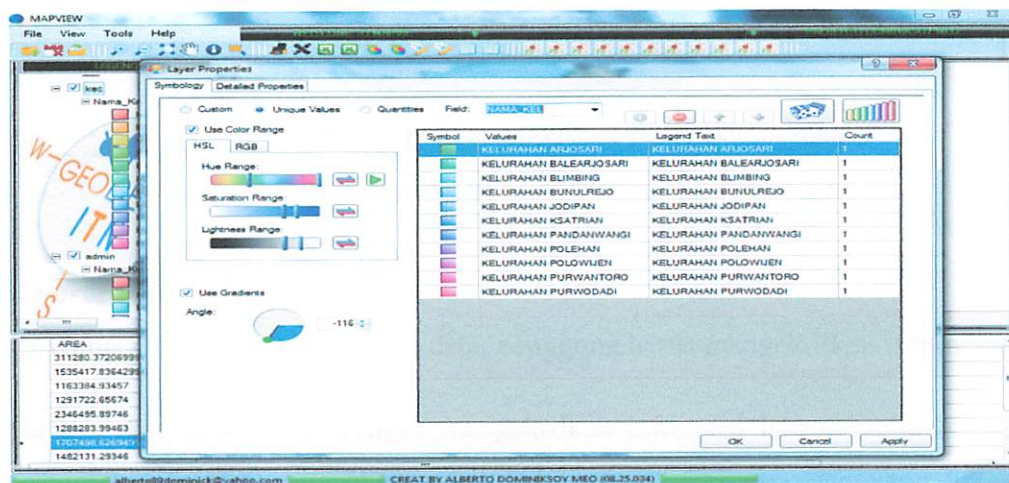
Sebuah program *otomatisasi* dibuat untuk mempermudah pengguna dalam melakukan pencarian data-data sehingga waktu dan tenaga yang dikeluarkan lebih efisien. Untuk kebutuhan tersebut, program ini dilengkapi dengan proses pencarian dengan metode *Structure Query Language (SQL)*.



Gambar 4.7. Form Query

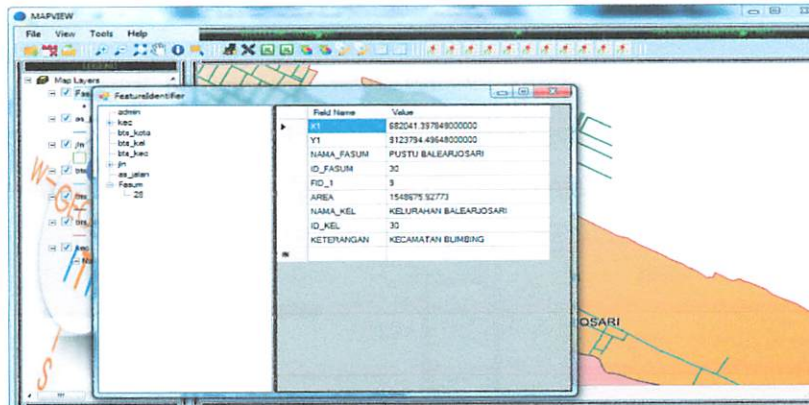
#### 4.9.2. Form Layer Properties

*Form layer properties* berfungsi untuk memberikan *simbology* pada *layer* peta yang ditampilkan. Didalam *form layer properties* terdapat beberapa pilihan untuk proses *simbology* antara lain *custom*, *unique value*, *quantities*.




Gambar 4.8. Form Layer Properties

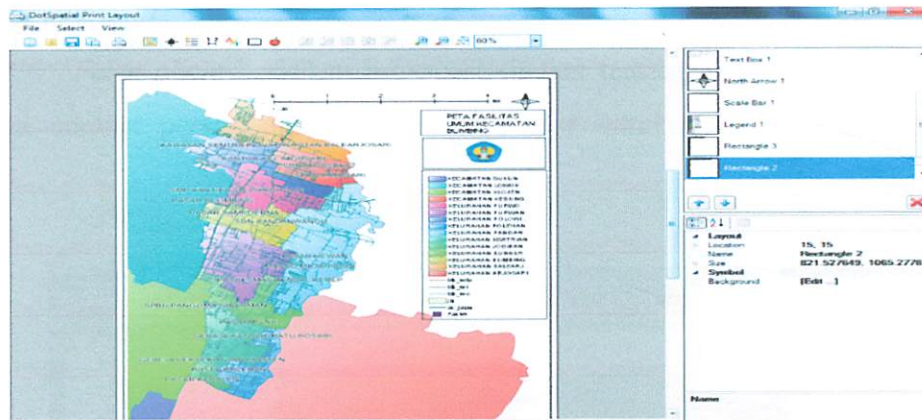




Gambar 4.9.1. Form Identify

#### 4.9.5. Form Print Layout

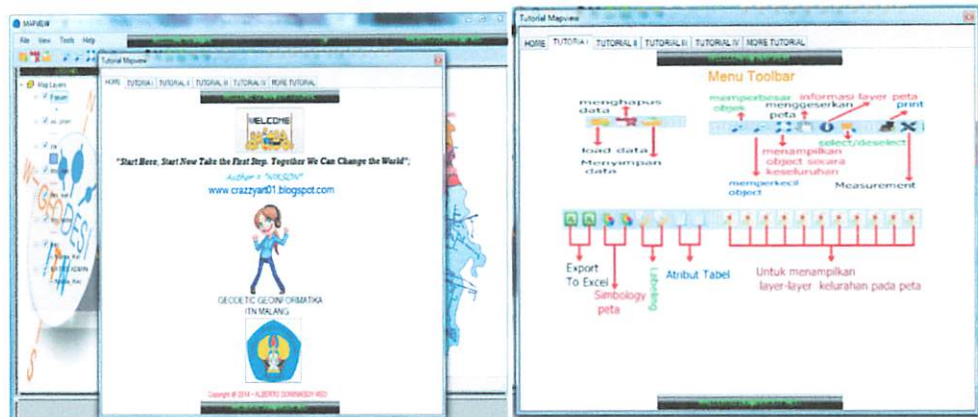
Salah satu kebutuhan penting dari proses otomatisasi adalah kebutuhan untuk menyimpan dan mencetak data yang berbentuk digital menjadi bentuk *hard copy*. Kebutuhan untuk mencetak peta difasilitasi dalam program ini pada menu Print dengan mengklik icon .



Gambar 4.9.2. Form Print Layout

#### 4.9.6. Form Help

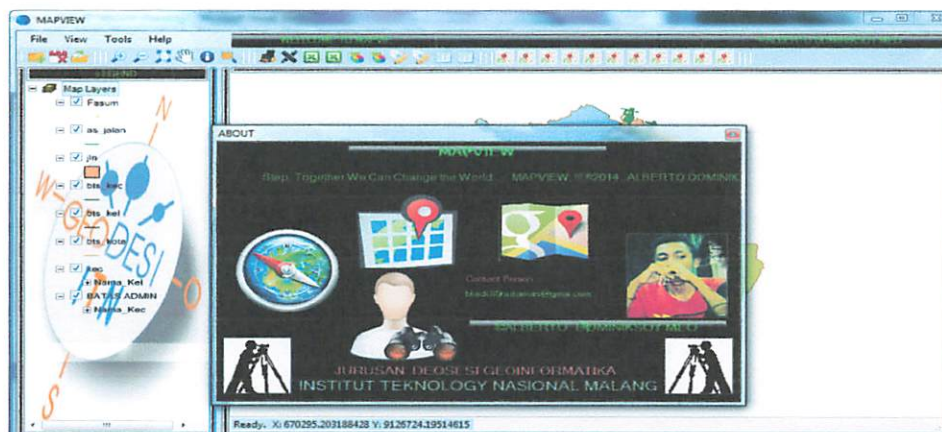
*Form Help* berisikan tentang tutorial singkat untuk menjalankan aplikasi mapview sehingga pengguna dapat memahami setiap menu yang tersedia dalam aplikasi dengan tujuan mempermudah *user/pengguna* dalam menjalankan aplikasi.



Gambar 4.9.3. Form Help

#### 4.9.7. Form About

*Form about* berisikan informasi singkat tentang si pembuat aplikasi serta berisikan contact person agar memudahkan *user* untuk menghubungi si pembuat aplikasi.



Gambar 4.9.4. Form About

#### 4.9.8. Kelebihan Aplikasi Mapview

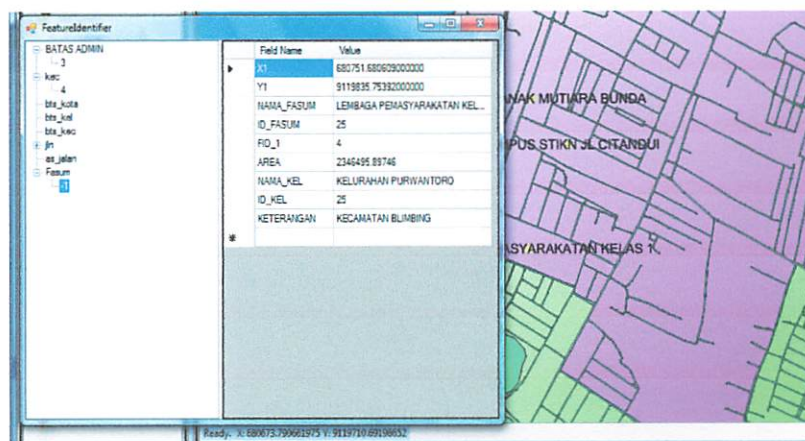
Kelebihan program mapview antara lain sebagai berikut ;

- 1) Hasil Data pada program ini dapat diperbaharui dengan penambahan data dan meng-edit atribut apabila terjadi perubahan data sewaktu-waktu, maka pengguna dapat meng-update informasi terbaru.
- 2) Didesain secara sederhana agar dapat mempermudah pengguna untuk mencari informasi yang dibutuhkan.

#### 4.9.9. Kelemahan Aplikasi Mapview

Kelemahan program mapview adalah:

- 1) Pada pilihan Menu Identify tidak bersifat select pada layer yang di klik tetapi mampu memberikan informasi detail dari layer yang di klik beserta informasi layer lainnya



Gambar 4.9.5. Kelemahan Menu Identify

- 2) Hanya dapat digunakan untuk menampilkan informasi dalam peta, tidak dapat digunakan dalam proses analisis data.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Setelah melakukan penelitian serta tinjauan dari berbagai pustaka dalam pembuatan program menggunakan *dotspatial library*, peneliti mempunyai beberapa kesimpulan yang didapat, yaitu :

- a. Telah dihasilkan aplikasi “*Mapview*” yang berbasis teknologi Sistem Informasi Geografis (*GIS*) dengan bahasa pemrograman C# dan pustaka *Dotspatial*.
- b. *Dotspatial library* memiliki keuntungan dalam pengembangan aplikasi berbasis *GIS* karena bersifat *free*, sehingga memudahkan bagi programmer dalam mengembangkan aplikasinya tanpa harus mengeluarkan biaya yang banyak.
- c. *Dotspatial library* memiliki fasilitas yang cukup lengkap dalam mengembang aplikasi berbasis *GIS*, hal ini memudahkan bagi para programmer yang berkelut dibidang *GIS* dalam mengembangkan aplikasinya.

#### 5.2. Saran

Sebagai penutup dalam Laporan Tugas Akhir ini, peneliti mempunyai saran dalam penelitian ini, yaitu :

- a. *Dotspatial library* belum begitu familiar bagi sebagian pengguna hal ini dikarenakan kurangnya referensi dan artikel-artikel yang diterbitkan mengenai *dotspatial*.
- b. Kurangnya programmer yang mengembangkan aplikasinya menggunakan *dotspatial library* menggunakan *Microsoft visual studio 2010* khususnya menggunakan bahasa pemrograman C#.

## DAFTAR PUSTAKA

- Agus Kurniawan, et.al, 2004. Project Otak : Seri Buku Komunitas , *Pengenalan Bahasa C#*, CSH101, 14-19.
- Budiyanto, Eko, 2002. *Sistem Informasi Geografis Menggunakan ArcView GIS*. Yogyakarta: Andi Publisher .
- Jogiyanto, 2001. *Analisa & Desain Sistem Informasi*. Yogyakarta: Andi Publisher.
- Jogiyanto, 2005. *Analisis dan Desain Sistem Informasi, Edisi III*. Yogyakarta: Andi Publisher.
- John Burch, Gary Grudnitski, 1986. *Information Systems Theory and Practice*, Edisi keempat. New York: John Wiley & Sons.
- Komputer Wahana, 2008. *Belajar Pemrograman C#*. Yogyakarta: Andi Publisher.
- Mogikanin, 2014. Project Hosting for Open Source Software.[online]. <http://dotspatial.codeplex.com>[10 April 2014].
- Pressman, R. S, 2002. *Rekayasa Perangkat Lunak, Pendekatan Praktis* (Buku Dua). Yogyakarta: Andi Publisher.
- Prahasta, E, 2005. *Konsep-konsep Dasar Sistem Informasi Geografis*. Bandung: Informatika.
- Prihandito, A, 1998. *Kartografi* . Yogyakarta: PT.Mitra Gama Widya.

## **LAMPIRAN**

**Lampiran I. Source Code Aplikasi Mapview**

## SOURCE CODE FORM APLIKASI MAPVIEW

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using DotSpatial.Controls;
using DotSpatial.Symbology;
using DotSpatial.Topology;
using DotSpatial.Data;
using DotSpatial.Projections;
using System.IO;
using DotSpatial.Common;

namespace MAPVIEW_BY_NIKSON
{
    public partial class Mapview : Form
    {
        public Mapview()
        {
            InitializeComponent();

            map1.AddLayer("C:\\pengolahan data unfix\\BATAS ADMIN.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\kec_blimbing.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\bts_kota.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\bts_kel.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\bts_kec.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\jln.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\as_jalan.shp");
            map1.AddLayer("C:\\pengolahan data unfix\\Fasum.shp");

            if (map1.Layers.Count > 0)
            {
                MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);

                kelurahanLayer = (MapPolygonLayer)map1.Layers[0];

                if (kelurahanLayer == null)
                {
                    MessageBox.Show("The layer is not a polygon layer.");
                }
                else
                {
                    PolygonScheme scheme = new PolygonScheme();
                }
            }
        }
    }
}
```

```

//Set the ClassificationType for the PolygonScheme via
EditorSettings
scheme.EditorSettings.ClassificationType =
ClassificationType.UniqueValues;

scheme.EditorSettings.FieldName = "Nama_Kec";
scheme.CreateCategories(kelurahanLayer.DataSet.DataTable);
kelurahanLayer.Symbology = scheme;
}
}
else
{
MessageBox.Show("Please add a layer to the map.");
}

//check the number of layers from map control
if (map1.Layers.Count > 0)
{
//Delacre a MapPolygonLayer
MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

//Type cast the FirstLayer of MapControl to MapPolygonLayer
kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

//Check the MapPolygonLayer( Make sure that it has a polygon
layer)
if (kecamatanLayer == null)
{
MessageBox.Show("The layer is not a polygon layer.");
}
else
{
//Create a new PolygonScheme
PolygonScheme scheme = new PolygonScheme();

//Set the ClassificationType for the PolygonScheme via
EditorSettings
scheme.EditorSettings.ClassificationType =
ClassificationType.UniqueValues;

scheme.EditorSettings.FieldName = "Nama_Kel";

//create categories on the scheme based on the attributes
table and field name
//In this case field name is Nama_kel
scheme.CreateCategories(kecamatanLayer.DataSet.DataTable);

//Set the scheme to kecamatanLayer's symbology
kecamatanLayer.Symbology = scheme;
}
}

```



```
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton1_Click(object sender, EventArgs e)
{
    map1.AddLayer();
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    map1.Layers.Clear();
}

private void toolStripButton3_Click(object sender, EventArgs e)
{
    map1.SaveLayer();
}

private void toolStripButton4_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.ZoomIn;
}

private void toolStripButton5_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.ZoomOut;
}

private void toolStripButton6_Click(object sender, EventArgs e)
{
    map1.ZoomToMaxExtent();
}

private void toolStripButton7_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.Pan;
}

private void toolStripButton8_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.Info;
}

private void toolStripButton9_Click(object sender, EventArgs e)
```

```

    {
        map1.FunctionMode = DotSpatial.Controls.FunctionMode.Select;
    }

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    map1.AddLayers();
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    map1.SaveLayer();
}

private void printToolStripMenuItem_Click(object sender, EventArgs e)
{
    DotSpatial.Controls.LayoutForm frm = new
DotSpatial.Controls.LayoutForm();
    frm.MapControl = map1;
    frm.Show();
}

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Close();
}

private void zoomInToolStripMenuItem_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.ZoomIn;
}

private void zoomOutToolStripMenuItem_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.ZoomOut;
}

private void panToolStripMenuItem_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.Pan;
}

e) private void zoomExtentToolStripMenuItem_Click(object sender, EventArgs
    {
        map1.ZoomToMaxExtent();
    }

e) private void identifyToolStripMenuItem_Click(object sender, EventArgs
    {
        map1.FunctionMode = DotSpatial.Controls.FunctionMode.Info;
    }

```

```

private void selectLayerDeselectLayerToolStripMenuItem_Click(object
sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.Select;
}

private void toolStripButton16_Click(object sender, EventArgs e)
{
    //Declare a datatable
    DataTable dt = null;

    if (map1.Layers.Count > 0)
    {
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        kecamatanLayer = (MapPolygonLayer)map1.Layers[0];

        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Get the shapefile's attribute table to our datatabledt
            dt = kecamatanLayer.DataSet.DataTable;

            //Set the datagridviewdatasource from datatabledt
            dgvAttributeTable.DataSource = dt;
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void dgvAttributeTable_SelectionChanged(object sender,
EventArgs e)
{
    foreach (DataGridViewRow row in dgvAttributeTable.SelectedRows)
    {
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        kecamatanLayer = (MapPolygonLayer)map1.Layers[0];

        if (kecamatanLayer == null)
        { MessageBox.Show("The layer is not a polygon layer."); }
        else
        {
            kecamatanLayer.SelectByAttribute("[Nama_Kec] =" + "" +
row.Cells["Nama_Kec"].Value + "");
        }
    }
}

```

```

    }
}

private void toolStripButton14_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);
        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[0];

        //Check whether stateLayer is polygon layer or not
        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //add kecamatanName as labels on the kecamatanLayer
            //[kecamatanLayer] is an attribute from the given example
Kecamatan Blimbing shape file.
            map1.AddLabels(kecamatanLayer, "[Nama_Kec]", new
Font("Tahoma", (float)8.0), Color.Black);
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton15_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);
        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        //Check whether kecamatanLayer is polygon layer or not
        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
    }
}

```

```

    }
    else
    {
        //add kecamatanName as labels on the stateLayer
        //[kecamatanLayer] is an attribute from the given example
        Kecamatan Blimbing shape file.
        map1.AddLabels(kecamatanLayer, "[NAMA_KEL]", new
        Font("Tahoma", (float)8.0), Color.Black);
    }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton10_Click(object sender, EventArgs e)
{
    DotSpatial.Controls.LayoutForm frm = new
    DotSpatial.Controls.LayoutForm();
    frm.MapControl = map1;
    frm.Show();
}

private void toolStripButton12_Click(object sender, EventArgs e)
{
    {
        //Declare a datatable
        System.Data.DataTable dt = null;

        if (map1.Layers.Count > 0)
        {
            MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);
            kecamatanLayer = map1.Layers[0] as MapPolygonLayer;
            if (kecamatanLayer == null)
            {
                MessageBox.Show("The layer is not a polygon layer.");
            }
            else
            {
                //Get the shapefile's attribute table to our datatable
                dt = kecamatanLayer.DataSet.DataTable;
                //Call the sub ExportToExcel
                //This sub procedure expects a datatable as an input
                ExportToExcel(dt);
            }
        }
        }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

```

```

    }
}

private void ExportToExcel(System.Data.DataTable objDT)
{
    //excel = new Excel.Application();
    Microsoft.Office.Interop.Excel.Application xlApp = new
Microsoft.Office.Interop.Excel.Application();

    string strFilename = null;
    int intCol = 0;
    int intRow = 0;
    //path for storing excel datasheet
    string strPath = "C:\\MAPVIEW BY NIKSON\\";

    if (xlApp == null)
    {
        MessageBox.Show("It appears that Excel is not installed on this
machine. This operation requires MS Excel to be installed on this machine.",
"", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        return;
    }
    try
    {
        //var _with1 = Microsoft.Office.Interop.Excel.Application();
        xlApp.SheetsInNewWorkbook = 1;
        xlApp.Workbooks.Add();
        xlApp.Worksheets[1].Select();
        xlApp.Cells[1, 1].value = "KECAMATAN";
        //Heading of the excel file
        xlApp.Cells[1, 1].EntireRow.Font.Bold = true;

        //Add the column names from the attribute table to excel
worksheet
        int intI = 1;
        for (intCol = 0; intCol <= objDT.Columns.Count - 1; intCol++)
        {
            xlApp.Cells[2, intI].value =
objDT.Columns[intCol].ColumnName;
            xlApp.Cells[2, intI].EntireRow.Font.Bold = true;
            intI += 1;
        }

        //Add the data row values from the attribute table to excel
worksheet
        intI = 3;
        int intK = 1;
        for (intCol = 0; intCol <= objDT.Columns.Count - 1; intCol++)
        {
            intI = 3;

```

```

        for (intRow = 0; intRow <= objDT.Rows.Count - 1; intRow++)
        {
            xlApp.Cells[intI, intK].Value =
objDT.Rows[intRow].ItemArray[intCol];
            intI += 1;
        }
        intK += 1;
    }

    if (strPath.Substring(strPath.Length - 1, 1) != "\\")
    {
        strPath = strPath + "\\";
    }

    strFilename = strPath + "KECAMATAN.xls";

    xlApp.ActiveCell.Worksheet.SaveAs(strFilename);

    System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);

    xlApp = null;
    MessageBox.Show("Data's are exported to Excel Succesfully in '"
+ strFilename + "'", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    // The excel is created and opened for insert value. We most close
this excel using this system
    System.Diagnostics.Process[] pro =
(System.Diagnostics.Process[])System.Diagnostics.Process.GetProcessesByName("EX
CEL");
    foreach (System.Diagnostics.Process i in pro)
    {
        i.Kill();
    }
}

private void atributeTableToolStripMenuItem_Click(object sender,
EventArgs e)
{
    //Declare a datatable
    DataTable dt = null;

    if (map1.Layers.Count > 0)
    {
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        kecamatanLayer = (MapPolygonLayer)map1.Layers[0];

        if (kecamatanLayer == null)

```

```

        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Get the shapefile's attribute table to our datatabledt
            dt = kecamatanLayer.DataSet.DataTable;

            //Set the datagridviewdatasource from datatabledt
            dgvAttributeTable.DataSource = dt;
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton16_Click_1(object sender, EventArgs e)
{
    //Declare a datatable
    DataTable dt = null;

    if (map1.Layers.Count > 0)
    {
        MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);

        kelurahanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kelurahanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Get the shapefile's attribute table to our datatabledt
            dt = kelurahanLayer.DataSet.DataTable;

            //Set the datagridviewdatasource from datatabledt
            dgvAttributeTable1.DataSource = dt;
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void dgvAttributeTable_SelectionChanged1(object sender,
EventArgs e)
{
    foreach (DataGridViewRow row in dgvAttributeTable1.SelectedRows)

```



```

    {
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kecamatanLayer == null)
        { MessageBox.Show("The layer is not a polygon layer."); }
        else
        {
            kecamatanLayer.SelectByAttribute("[NAMA_KEL] =" + "'" +
row.Cells["NAMA_KEL"].Value + "'");
        }
    }
}

private void toolStripButton37_Click(object sender, EventArgs e)
{
    {
        //Declare a datatable
        System.Data.DataTable dt = null;

        if (map1.Layers.Count > 0)
        {
            MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);
            kelurahanLayer = map1.Layers[1] as MapPolygonLayer;
            if (kelurahanLayer == null)
            {
                MessageBox.Show("The layer is not a polygon layer.");
            }
            else
            {
                //Get the shapefile's attribute table to our datatable

                dt = kelurahanLayer.DataSet.DataTable;
                //Call the sub ExportToExcel
                //This sub procedure expects a datatable as an input
                ExportToExcel1(dt);
            }
        }
        else
        {
            MessageBox.Show("Please add a layer to the map.");
        }
    }
}

private void ExportToExcel1(System.Data.DataTable objDT)
{
    //excel = new Excel.Application();
}

```

```

Microsoft.Office.Interop.Excel.Application xlApp = new
Microsoft.Office.Interop.Excel.Application();

string strFilename = null;
int intCol = 0;
int intRow = 0;
//path for storing excel datasheet
string strPath = "C:\\MAPVIEW BY NIKSON\\";

if (xlApp == null)
{
    MessageBox.Show("It appears that Excel is not installed on this
machine. This operation requires MS Excel to be installed on this machine.",
"", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    return;
}
try
{
    //var _with1 = Microsoft.Office.Interop.Excel.Application();
    xlApp.SheetsInNewWorkbook = 1;
    xlApp.Workbooks.Add();
    xlApp.Worksheets[1].Select();
    xlApp.Cells[1, 1].value = "KELURAHAN";
    //Heading of the excel file
    xlApp.Cells[1, 1].EntireRow.Font.Bold = true;

    //Add the column names from the attribute table to excel
worksheet
    int intI = 1;
    for (intCol = 0; intCol <= objDT.Columns.Count - 1; intCol++)
    {
        xlApp.Cells[2, intI].value =
objDT.Columns[intCol].ColumnName;
        xlApp.Cells[2, intI].EntireRow.Font.Bold = true;
        intI += 1;
    }

    //Add the data row values from the attribute table to excel
worksheet
    intI = 3;
    int intK = 1;
    for (intCol = 0; intCol <= objDT.Columns.Count - 1; intCol++)
    {
        intI = 3;
        for (intRow = 0; intRow <= objDT.Rows.Count - 1; intRow++)
        {
            xlApp.Cells[intI, intK].Value =
objDT.Rows[intRow].ItemArray[intCol];
            intI += 1;
        }
        intK += 1;
    }
}

```

```

        if (strPath.Substring(strPath.Length - 1, 1) != "\\")
        {
            strPath = strPath + "\\";
        }

        strFilename = strPath + "KELURAHAN.xls";

        xlApp.ActiveCell.Worksheet.SaveAs(strFilename);

        System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);

        xlApp = null;
        MessageBox.Show("Data's are exported to Excel Succesfully in '"
+ strFilename + "'", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    // The excel is created and opened for insert value. We most close
    this excel using this system
    System.Diagnostics.Process[] pro =
    (System.Diagnostics.Process[])System.Diagnostics.Process.GetProcessesByName("EX
    CEL");
    foreach (System.Diagnostics.Process i in pro)
    {
        i.Kill();
    }
}

private void toolStripButton13_Click(object sender, EventArgs e)
{
    //check the number of layers from map control
    if (map1.Layers.Count > 0)
    {
        //Delacre a MapPolygonLayer
        MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);

        //Type cast the FirstLayer of MapControl to MapPolygonLayer
        kelurahanLayer = (MapPolygonLayer)map1.Layers[0];

        //Check the MapPolygonLayer( Make sure that it has a polygon
layer)
        if (kelurahanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Create a new PolygonScheme
            PolygonScheme scheme = new PolygonScheme();

```

```

        //Set the ClassificationType for the PolygonScheme via
EditorSettings
        scheme.EditorSettings.ClassificationType =
ClassificationType.UniqueValues;

        scheme.EditorSettings.FieldName = "Nama_Kec";
scheme.CreateCategories(kelurahanLayer.DataSet.DataTable);

        kelurahanLayer.Symbology = scheme;
    }
}
else
{
    MessageBox.Show("Please add a layer to the map.");
}

}

private void toolStripButton38_Click(object sender, EventArgs e)
{
    //check the number of layers from map control
    if (map1.Layers.Count > 0)
    {
        //Delacre a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //Type cast the FirstLayer of MapControl to MapPolygonLayer
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        //Check the MapPolygonLayer( Make sure that it has a polygon
layer)
        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Create a new PolygonScheme
            PolygonScheme scheme = new PolygonScheme();

            //Set the ClassificationType for the PolygonScheme via
EditorSettings
            scheme.EditorSettings.ClassificationType =
ClassificationType.UniqueValues;

            scheme.EditorSettings.FieldName = "Nama_Kel";

            //create categories on the scheme based on the attributes
table and field name

```

```

        scheme.CreateCategories(kecamatanLayer.DataSet.DataTable);

        kecamatanLayer.Symbology = scheme;
    }
}
else
{
    MessageBox.Show("Please add a layer to the map.");
}
}

private void toolStripButton17_Click(object sender, EventArgs e)
{
    {
        //Check the number of layers from MapControl
        if (map1.Layers.Count > 0)
        {
            //Declare a MapPolygonLayer
            MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

            //TypeCast the first layer from MapControl to
            MapPolygonLayer.
            //Layers are 0 based, therefore 0 is going to grab the
            first layer from the MapControl
            kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

            //Check whether kecamatanLayer is polygon layer or not
            if (kecamatanLayer == null)
            {
                MessageBox.Show("The layer is not a polygon layer.");
            }
            else
            {
                //SelectByAttribute method is used to implement the
                filter operations.
                kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN JODIPAN'");
            }
        }
        else
        {
            MessageBox.Show("Please add a layer to the map.");
        }
    }
}

private void toolStripButton18_Click(object sender, EventArgs e)
{

```

```

//Check the number of layers from MapControl
if (map1.Layers.Count > 0)
{
    //Declare a MapPolygonLayer
    MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

    //TypeCast the first layer from MapControl to MapPolygonLayer.
    //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
    kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

    //Check whether kecamatanLayer is polygon layer or not
    if (kecamatanLayer == null)
    {
        MessageBox.Show("The layer is not a polygon layer.");
    }
    else
    {
        //SelectByAttribute method is used to implement the filter
operations.
        kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
KSATRIAN'");
    }
}
else
{
    MessageBox.Show("Please add a layer to the map.");
}
}

private void toolStripButton19_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {

```

```

        //SelectByAttribute method is used to implement the filter
operations.
        kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
POLEHAN'");
    }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton20_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter
operations.
            kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN BUNULREJO'");
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton21_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

```

```

//TypeCast the first layer from MapControl to MapPolygonLayer.
//Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

if (kecamatanLayer == null)
{
    MessageBox.Show("The layer is not a polygon layer.");
}
else
{
    //SelectByAttribute method is used to implement the filter
operations.

    kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
PURWANTORO'");
}
else
{
    MessageBox.Show("Please add a layer to the map.");
}
}

private void toolStripButton22_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter
operations.

            kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
BLIMBING'");
        }
    }
}

```



```

else
{
    MessageBox.Show("Please add a layer to the map.");
}
}

private void toolStripButton23_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter
operations.
            kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
PURWODADI'");
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton24_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kecamatanLayer == null)

```

```

        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter
operations.
            kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
POLOWIJEN'");
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton25_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter
operations.
            kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
PANDANWANGI'");
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton26_Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl

```

```

if (map1.Layers.Count > 0)
{
    //Declare a MapPolygonLayer
    MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

    //TypeCast the first layer from MapControl to MapPolygonLayer.
    //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
    kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

    if (kecamatanLayer == null)
    {
        MessageBox.Show("The layer is not a polygon layer.");
    }
    else
    {
        //SelectByAttribute method is used to implement the filter
operations.

        kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
BALEARJOSARI'");
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void toolStripButton27 Click(object sender, EventArgs e)
{
    //Check the number of layers from MapControl
    if (map1.Layers.Count > 0)
    {
        //Declare a MapPolygonLayer
        MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);

        //TypeCast the first layer from MapControl to MapPolygonLayer.
        //Layers are 0 based, therefore 0 is going to grab the first
layer from the MapControl
        kecamatanLayer = (MapPolygonLayer)map1.Layers[1];

        //Check whether kecamatanLayer is polygon layer or not
        if (kecamatanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //SelectByAttribute method is used to implement the filter
operations.

            kecamatanLayer.SelectByAttribute("[Nama_Kel] = 'KELURAHAN
ARJOSARI'");

```

```

    }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void aboutMapViewToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form dlg = new About();
    if (DialogResult.OK == dlg.ShowDialog())
    {
    }
}

private void toolStripButton36_Click(object sender, EventArgs e)
{
}

private void toolStripButton11_Click(object sender, EventArgs e)
{
    {
        //Declare a datatable
        System.Data.DataTable dt = null;

        if (map1.Layers.Count > 0)
        {
            MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);
            kecamatanLayer = map1.Layers[0] as MapPolygonLayer;
            if (kecamatanLayer == null)
            {
                MessageBox.Show("The layer is not a polygon layer.");
            }
            else
            {
                //Get the shapefile's attribute table to our datatable
                dt = kecamatanLayer.DataSet.DataTable;
                //Call the sub ExportToExcel
                //This sub procedure expects a datatable as an input
                ExportToExcel(dt);
            }
        }
        else
        {
            MessageBox.Show("Please add a layer to the map.");
        }
    }
}

```

```

    }
}

private void clearAllLayerToolStripMenuItem_Click(object sender,
EventArgs e)
{
    map1.Layers.Clear();
}

private void atributeTable1ToolStripMenuItem Click(object sender,
EventArgs e)
{
    //Declare a datatable
    DataTable dt = null;

    if (map1.Layers.Count > 0)
    {
        MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);

        kelurahanLayer = (MapPolygonLayer)map1.Layers[1];

        if (kelurahanLayer == null)
        {
            MessageBox.Show("The layer is not a polygon layer.");
        }
        else
        {
            //Get the shapefile's attribute table to our datatabledt
            dt = kelurahanLayer.DataSet.DataTable;

            //Set the datagridviewdatasource from datatabledt
            dgvAttributeTable.DataSource = dt;
        }
    }
    else
    {
        MessageBox.Show("Please add a layer to the map.");
    }
}

private void exportToExcelToolStripMenuItem_Click(object sender,
EventArgs e)
{
    //Declare a datatable
    System.Data.DataTable dt = null;

    if (map1.Layers.Count > 0)
    {

```

```

MapPolygonLayer kecamatanLayer = default(MapPolygonLayer);
kecamatanLayer = map1.Layers[0] as MapPolygonLayer;
if (kecamatanLayer == null)
{
    MessageBox.Show("The layer is not a polygon layer.");
}
else
{
    //Get the shapefile's attribute table to our datatable dt
    dt = kecamatanLayer.DataSet.DataTable;
    //Call the sub ExportToExcel
    //This sub procedure expects a datatable as an input
    ExportToExcel(dt);
}
}
else
{
    MessageBox.Show("Please add a layer to the map.");
}
}

private void exportToExcel1ToolStripMenuItem_Click(object sender,
EventArgs e)
{
    {
        //Declare a datatable
        System.Data.DataTable dt = null;

        if (map1.Layers.Count > 0)
        {
            MapPolygonLayer kelurahanLayer = default(MapPolygonLayer);
            kelurahanLayer = map1.Layers[1] as MapPolygonLayer;
            if (kelurahanLayer == null)
            {
                MessageBox.Show("The layer is not a polygon layer.");
            }
            else
            {
                //Get the shapefile's attribute table to our datatable
                dt = kelurahanLayer.DataSet.DataTable;
                //Call the sub ExportToExcel
                //This sub procedure expects a datatable as an input
                ExportToExcel1(dt);
            }
        }
        else
        {
            MessageBox.Show("Please add a layer to the map.");
        }
    }
}
}

```

```
private void tutorialMapviewToolStripMenuItem Click(object sender,
EventArgs e)
{
    Form dlg = new Form1();
    if (DialogResult.OK == dlg.ShowDialog())
    {
    }
}

private void toolStripButton47_Click(object sender, EventArgs e)
{
    map1.FunctionMode = DotSpatial.Controls.FunctionMode.Measure;
}
}
}
```

## SOURCE CODE FORM TUTORIAL APLIKASI MAPVIEW

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MAPVIEW_BY_NIKSON
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void label19_Click(object sender, EventArgs e)
            {
                System.Diagnostics.Process.Start("http://crazyart01.blogspot.com/2014/08/mapview-use-dotspatial-open-source.html");
            }

            private void label3_Click(object sender, EventArgs e)
            {
                System.Diagnostics.Process.Start("http://crazyart01.blogspot.com/");
            }

            private void label23_Click(object sender, EventArgs e)
            {
                System.Diagnostics.Process.Start("www.black89rastaman@gmail.com");
            }

            private void button15_Click(object sender, EventArgs e)
            {
                System.Diagnostics.Process.Start("www.black89rastaman@gmail.com");
            }
        }
    }
}
```



## SOURCE CODE FORM ABOUT APLIKASI MAPVIEW

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MAPVIEW_BY_NIKSON
{
    public partial class About : Form
    {
        //deklarasi variable string dan menampung kalimat didalamnya
        string kata = "Start Now Take the First Step. Together We Can Change
the World.....MAPVIEW..!!! @2014...ALBERTO DOMINIKSOY MEO...(0825034)";
        //deklarasi nilai awal untuk lable1
        int nilai_awal = 0;
        int panjang = 0;
        public About()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            //menghitung panjang krakter pada variable kata
            panjang = Convert.ToInt16(kata.Length);
            //perulangan ketika nilai panjang sudah sama dengan nilai awal
            if (nilai_awal == panjang)
            {
                nilai_awal = 0;
            }
            //tesk dibuat seolah2 berjalan, dengan memanfaatkan pengurangan
            (panjang - nilai_awal)
            label1.Text = kata.Substring(nilai_awal, panjang - nilai_awal);
            nilai_awal = nilai_awal + 1;
        }

        private void label4_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start("www.black89rastaman@gmail.com");
        }
    }
}
```