

# STEP BY STEP MENJADI PROGRAMMER HANDAL DENGAN VB.NET

## ***Lisensi Dokumen:***

*Copyright © 2003 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Junindar, ST, MCPD, MOS, MCT, MVP VB.NET

## Membuat Aplikasi Dynamic Provider Database Menggunakan VB 11

ADO.NET adalah NET Library yang digunakan untuk pengaksesan data dengan menyediakan kumpulan class-class yang tergabung dalam beberapa namespace. Namespace adalah pengelompokan secara logic class-class kedalam nama tertentu. Dan setiap database memiliki namespace tersendiri. Contoh untuk MS Access menggunakan “System.Data.OleDb” dan untuk SQL Server menggunakan “System.Data.SqlClient”. Terdapat beberapa class didalam namespace tersebut.

<b>System.Data.OleDb</b>	<b>System.Data.SqlClient</b>
OleDbConnection	SqlConnection
OleDbCommand	SqlCommand
OleDbDataReader	SqlDataReader
OleDbDataAdapter	SqlDataAdapter

Untuk memanggil namespace tersebut menggunakan keyword “Imports”. Seperti “Imports System.Data.OleDb” atau “Imports System.Data.SqlClient”. Jadi yang harus diperhatikan disini adalah, setiap mengubah database kita harus mengubah namespace dan class-class yang terdapat pada namespace tersebut.

Ini akan menjadi masalah dimana pada kemudian hari, aplikasi yang kita buat terjadi pergantian DBMS karena alasan tertentu. Jika aplikasi yang dibuat berskala besar ini merupakan pekerjaan besar bagi developer.

Lalu sebagai developer yang handal, kita harus sudah memikirkan hal ini sebelum aplikasi dibuat. Yaitu dengan membuat aplikasi yang “tahan banting” jika hal yang saya sebutkan diatas terjadi. Lalu bagaimanakah solusi untuk hal ini, tanpa harus mengganti code yang telah kita buat. Solusi nya adalah dengan membuat generic data access agar aplikasi tidak bergantung hanya dengan satu DBMS. Pada .NET Framework kita dapat menggunakan namespace “System.Data.Common” untuk membuat generic data access. Untuk mengetahui lebih lanjut, saya akan menjelaskan bagaimana membuat aplikasi dengan menggunakan generic data access. Ikuti langkah-langkah dibawah ini.

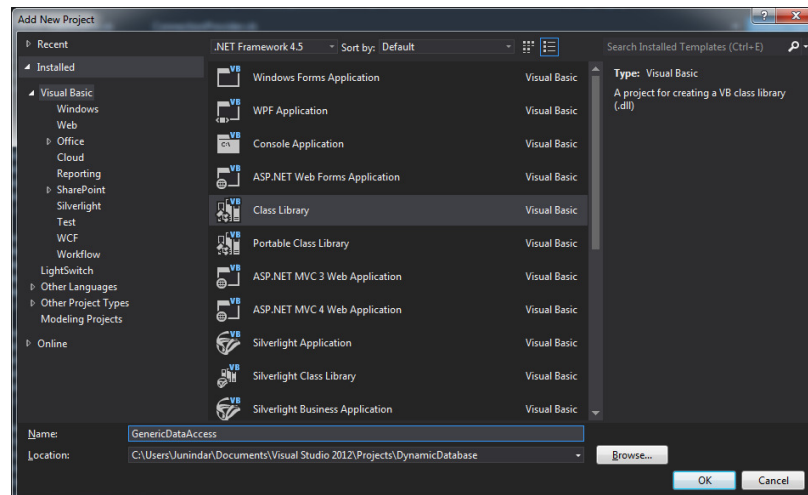
- Membuat database dengan nama Latihan pada SQL Server dan MS Access. Selanjutnya buat satu buah table dengan nama tblSiswa, dengan field-field sebagai berikut.

<b>Nama Field</b>	<b>Field Type</b>	
	<b>SQL Server</b>	<b>MS Access</b>
Nim	Text (15)	Varchar (15)
Nama	Text (50)	Varchar (50)
Alamat	Text (255)	Varchar (255)
TempatLahir	Text (50)	Varchar (50)

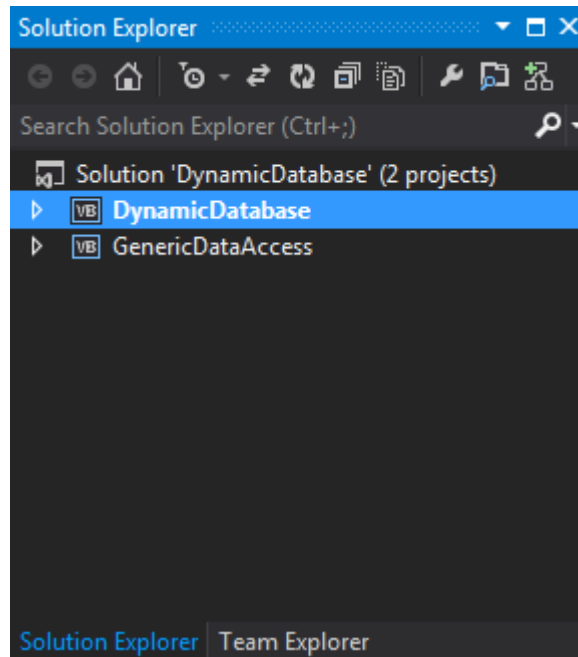
TglLahir	Date/Time	Date
Umur	Number	Integer

- Membuat Class Library

- Buat sebuah project dengan nama “DynamicDatabase”.
  - Lalu pada project tersebut, pilih File > Add Project
  - Pilih Class Library dan ganti nama project tersebut menjadi “GenericDataAccess”.
- Pada aplikasi ini terdapat beberapa layer, dimana untuk koneksi dan manipulasi data akan dilakukan pada Class Library (GenericDataAccess). Salah satu keuntungan dengan memisah-misahkan layer seperti ini adalah tidak ada ketergantungan pada jenis User Interface yang akan kita buat. Artinya jika ada perubahan User Interface dari Windows ke Web tidak perlu untuk mengubah code pada Class Library, sehingga tidak ada ketergantungan antara Layer Data Access dengan User Interface.

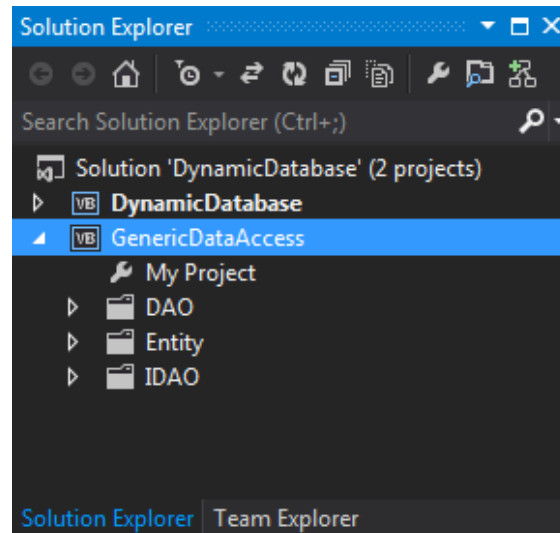


- Sehingga pada Solution akan terdapat dua project seperti pada gambar.



- Pada GenericDataAccess, tambahkan 3 buah folder dengan nama masing-masing DAO, IDAO dan Entity.

-



- Pada folder Entity tambahkan sebuah class dengan nama Siswa. Dan ketikkan code seperti dibawah ini.

```
Public Class Siswa
    Public Property Nim() As String
    Public Property Nama() As String
    Public Property Alamat() As String
    Public Property TempatLahir() As String
    Public Property TglLahir() As Date
    Public Property Umur() As Integer
End Class
```

Class Siswa merupakan representasi dari table yang telah kita buat sebelum nya.

- Pada folder DAO, buat sebuah class dengan nama ConnectionProvider. Dan ketikkan sintaks dibawah ini.

```
Imports System
Imports System.Data.Common
Imports System.Configuration
Public Class ConnectionProvider
    Private Shared conn As DbConnection

    Public Shared Function BuildConnection() As DbConnection
        Try
            Dim factory As DbProviderFactory =
                DbProviderFactories.GetFactory(ConfigurationManager.AppSettings("Driver").ToString())
            conn = factory.CreateConnection()
            conn.ConnectionString =
                ConfigurationManager.AppSettings("ConnectionString").ToString()
            conn.Open()
        Catch ex As DbException
            Throw New Exception(ex.Message.ToString())
        End Try
        Return conn
    End Function
End Class
```

Jangan lupa untuk Add Reference “System.Configuration”. Namespace ini digunakan untuk mengambil value pada AppSettings.

Seperti yang telah di jelaskan diatas, untuk membuat generic data access, kita akan menggunakan namespace “System.Data.Common”. Sedangkan DbProviderFactory digunakan untuk pembuatan instance provider dengan menggunakan AppSettings dengan value nya adalah “Driver”. Di ikuti dengan

connection string dan open connection. Class ini berfungsi untuk koneksi dengan DBMS yang akan digunakan.

- Pada folder IDAO, tambahkan sebuah Interface dengan nama ISiswaDAO, selanjutnya ketikkan sintaks dibawah ini.

```
Public Interface ISiswaDAO

    Sub InsertSiswa(ByVal Entity As Siswa)
    Sub Updatesiswa(ByVal Entity As Siswa)
    Sub DeleteSiswa(ByVal Nim As String)
    Function SelectSiswa() As List(Of Siswa)
    Function SelectSiswaByNim(ByVal Nim As String) As Siswa

End Interface
```

Interface merupakan kontrak yang semua method atau property nya harus diikuti dengan oleh class yang mengimplementasi nya. Dan interface hanya mendefinisikan kerangka method/property sedangkan isi dari proses nya terdapat pada class.

- Dan langkah terakhir dari pembuatan Class Library ini adalah dengan membuat sebuah class SiswaDAO. Class ini akan mengimplemetasikan interface yang baru kita buat di atas. Pada code “Class SiswaDAO”, ketikkan “Implements ISiswaDAO” dan enter. Maka secara otomatis akan dibuat method sesuai dengan Intercafe diatas. Seperti gambar dibawah.

```
Public Class SiswaDAO
    Implements ISiswaDAO

    Public Sub DeleteSiswa(Nim As String) Implements ISiswaDAO.DeleteSiswa
    End Sub

    Public Sub InsertSiswa(Entity As Siswa) Implements ISiswaDAO.InsertSiswa
    End Sub

    Public Function SelectSiswa() As List(Of Siswa) Implements ISiswaDAO.SelectSiswa
    End Function

    Public Function SelectSiswaByNim(Nim As String) As Siswa Implements ISiswaDAO.SelectSiswaByNim
    End Function

    Public Sub Updatesiswa(Entity As Siswa) Implements ISiswaDAO.Updatesiswa
    End Sub
End Class
```

- Masih pada class yang sama buat 3 variable seperti dibawah.

```
Private mycmd As DbCommand  
Private mydr As DbDataReader
```

- Selanjutnya ketikkan code dibawah, pada masing-masing method pada class SiswaDAO.

```
Public Sub DeleteSiswa(Nim As String) Implements ISiswaDAO.DeleteSiswa  
    Try  
        Const strQuery As String = "Delete From tblSiswa Where Nim=@Nim"  
        Using conn As DbConnection = ConnectionProvider.BuildConnection()  
            mycmd = conn.CreateCommand()  
            mycmd.Connection = conn  
            mycmd.CommandText = strQuery  
  
            Dim ParamNim As DbParameter = mycmd.CreateParameter()  
            ParamNim.ParameterName = "@Nim"  
            ParamNim.Value = Nim  
            ParamNim.DbType = DbType.String  
            ParamNim.Size = 15  
            mycmd.Parameters.Add(ParamNim)  
  
            mycmd.ExecuteNonQuery()  
        End Using  
    Catch ex As DbException  
        Throw New Exception(ex.Message.ToString())  
    End Try  
End Sub  
  
Public Function SelectSiswa() As List(Of Siswa) Implements  
ISiswaDAO.SelectSiswa  
    Dim list As New List(Of Siswa)()  
    Try  
        Using conn As DbConnection = ConnectionProvider.BuildConnection()  
            mycmd = conn.CreateCommand()  
            mycmd.Connection = conn  
            mycmd.CommandText = "SELECT * FROM tblSiswa Order By Nim"  
            mydr = mycmd.ExecuteReader()  
            While mydr.Read()  
                Dim mysiswa As New Siswa With {.Nim =  
mydr("Nim").ToString(),  
                .Nama = mydr("Nama").ToString(), .Alamat =  
mydr("Alamat").ToString(),  
                .TempatLahir = mydr("TempatLahir").ToString(),  
                .TglLahir = mydr("TglLahir").ToString(), .Umur =  
mydr("Umur")}  
  
                list.Add(mysiswa)  
            End While  
            mydr.Close()  
        End Using  
    Catch ex As DbException  
        Throw New Exception(ex.Message.ToString())  
    End Try  
    Return list  
End Function
```

```
Public Sub InsertSiswa(Entity As Siswa) Implements ISiswaDAO.InsertSiswa
    Try
        Const strQuery As String = "Insert Into tblSiswa
(Nim,Nama,Alamat,TempatLahir,TglLahir,Umur) " &
        " Values (@Nim,@Nama,@Alamat,@TempatLahir,@TglLahir,@Umur)"
        Using conn As DbConnection = ConnectionProvider.BuildConnection()
            mycmd = conn.CreateCommand()
            mycmd.Connection = conn
            mycmd.CommandText = strQuery

            Dim ParamNim As DbParameter = mycmd.CreateParameter()
            ParamNim.ParameterName = "@Nim"
            ParamNim.Value = Entity.Nim
            ParamNim.DbType = DbType.String
            ParamNim.Size = 15
            mycmd.Parameters.Add(ParamNim)

            Dim ParamNama As DbParameter = mycmd.CreateParameter()
            ParamNama.ParameterName = "@Nama"
            ParamNama.Value = Entity.Nama
            ParamNama.DbType = DbType.String
            ParamNama.Size = 50
            mycmd.Parameters.Add(ParamNama)

            Dim ParamAlamat As DbParameter = mycmd.CreateParameter()
            ParamAlamat.ParameterName = "@Alamat"
            ParamAlamat.Value = Entity.Alatat
            ParamAlamat.DbType = DbType.String
            ParamAlamat.Size = 255
            mycmd.Parameters.Add(ParamAlamat)

            Dim ParamTempat As DbParameter = mycmd.CreateParameter()
            ParamTempat.ParameterName = "@TempatLahir"
            ParamTempat.Value = Entity.TempatLahir
            ParamTempat.DbType = DbType.String
            ParamTempat.Size = 50
            mycmd.Parameters.Add(ParamTempat)

            Dim ParamTanggal As DbParameter = mycmd.CreateParameter()
            ParamTanggal.ParameterName = "@TglLahir"
            ParamTanggal.Value = Entity.TglLahir
            ParamTanggal.DbType = DbType.Date
            mycmd.Parameters.Add(ParamTanggal)

            Dim ParamUmur As DbParameter = mycmd.CreateParameter()
            ParamUmur.ParameterName = "@Umur"
            ParamUmur.Value = Entity.Umur
            ParamUmur.DbType = DbType.Int16
            mycmd.Parameters.Add(ParamUmur)

            mycmd.ExecuteNonQuery()
        End Using
        Catch ex As DbException
            Throw New Exception(ex.Message.ToString())
        End Try
    End Sub
```



```
Public Sub Updatesiswa(Entity As Siswa) Implements ISiswaDAO.Updatesiswa
    Try
        Const strQuery As String = "Update tblSiswa Set
        Nama=@Nama,Alamat=@Alamat, " &
        " TempatLahir=@TempatLahir,TglLahir=@TglLahir,Umur=@Umur Where
        Nim=@Nim "
        Using conn As DbConnection = ConnectionProvider.BuildConnection()
            mycmd = mycon.CreateCommand()
            mycmd.Connection = mycon
            mycmd.CommandText = strQuery

            Dim ParamNama As DbParameter = mycmd.CreateParameter()
            ParamNama.ParameterName = "@Nama"
            ParamNama.Value = Entity.Nama
            ParamNama.DbType = DbType.String
            ParamNama.Size = 50
            mycmd.Parameters.Add(ParamNama)

            Dim ParamAlamat As DbParameter = mycmd.CreateParameter()
            ParamAlamat.ParameterName = "@Alamat"
            ParamAlamat.Value = Entity.Alat
            ParamAlamat.DbType = DbType.String
            ParamAlamat.Size = 255
            mycmd.Parameters.Add(ParamAlamat)

            Dim ParamTempat As DbParameter = mycmd.CreateParameter()
            ParamTempat.ParameterName = "@TempatLahir"
            ParamTempat.Value = Entity.TempatLahir
            ParamTempat.DbType = DbType.String
            ParamTempat.Size = 50
            mycmd.Parameters.Add(ParamTempat)

            Dim ParamTanggal As DbParameter = mycmd.CreateParameter()
            ParamTanggal.ParameterName = "@TglLahir"
            ParamTanggal.Value = Entity.TglLahir
            ParamTanggal.DbType = DbType.Date
            mycmd.Parameters.Add(ParamTanggal)

            Dim ParamUmur As DbParameter = mycmd.CreateParameter()
            ParamUmur.ParameterName = "@Umur"
            ParamUmur.Value = Entity.Umur
            ParamUmur.DbType = DbType.Int16
            mycmd.Parameters.Add(ParamUmur)

            Dim ParamNim As DbParameter = mycmd.CreateParameter()
            ParamNim.ParameterName = "@Nim"
            ParamNim.Value = Entity.Nim
            ParamNim.DbType = DbType.String
            ParamNim.Size = 15
            mycmd.Parameters.Add(ParamNim)
            mycmd.ExecuteNonQuery()
        End Using

        Catch ex As DbException
            Throw New Exception(ex.Message.ToString())
        End Try
    End Sub
```

```
Public Function SelectSiswaByNim(Nim As String) As Siswa Implements
ISiswaDAO.SelectSiswaByNim
    Dim mysiswa As Siswa = Nothing
    Try
        Using conn As DbConnection = ConnectionProvider.BuildConnection()
            mycmd = mycon.CreateCommand()
            mycmd.Connection = mycon
            mycmd.CommandText = "SELECT * FROM tblSiswa Where Nim=@Nim"

            Dim ParamNim As DbParameter = mycmd.CreateParameter()
            ParamNim.ParameterName = "@Nim"
            ParamNim.Value = Nim
            ParamNim.DbType = DbType.String
            ParamNim.Size = 15
            mycmd.Parameters.Add(ParamNim)

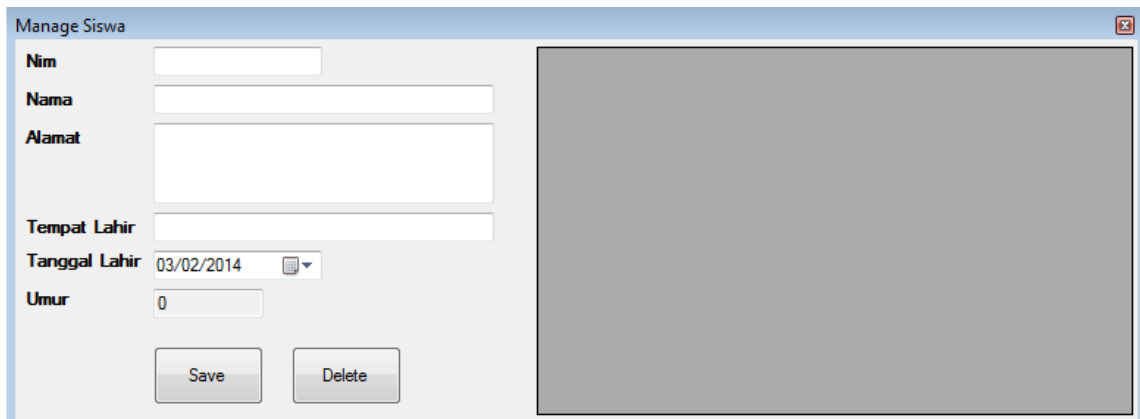
            myrdr = mycmd.ExecuteReader()
            If myrdr.Read() Then
                mysiswa = New Siswa() With {
                    .Nim = myrdr("Nim").ToString(),
                    .Nama = myrdr("Nama").ToString(),
                    .Alamat = myrdr("Alamat").ToString(),
                    .TempatLahir = myrdr("TempatLahir").ToString(),
                    .TglLahir = myrdr("TglLahir").ToString(),
                    .Umur = myrdr("Umur").ToString()}
            End If

            myrdr.Close()
        End Using
        Catch ex As DbException
            Throw New Exception(ex.Message.ToString())
        End Try
        Return mysiswa
    End Function
```

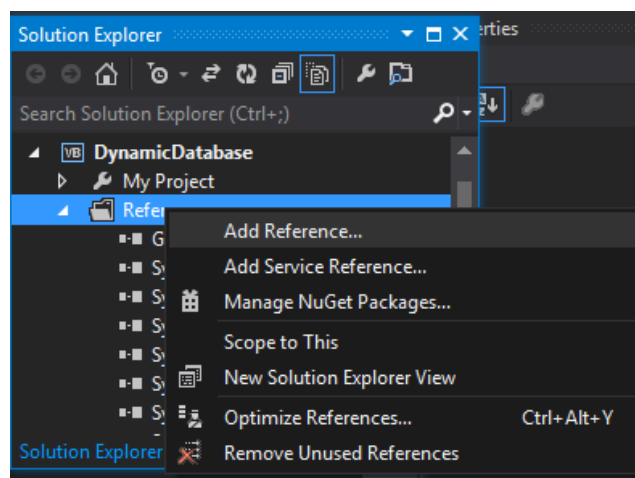
Sekarang kita telah selesai membuat Class Library (GenericDataAccess), dan langkah selanjutnya adalah bekerja pada layer User Interface.

- Membuat User Interface pada aplikasi. (ProjectDynamicDatabase)
  - Tambahkan 5 buah textbox pada Form1. Dengan properties masing-masing sebagai berikut.
    1. Name = txtNim
    2. Name = txtNama
    3. Name = txtAlamat, Multiline = True
    4. Name = txtTempat
    5. Name = txtUmur, ReadOnly = True
  - Tambahkan sebuah DateTimePicker dan ganti properties Name menjadi "dtLahir".

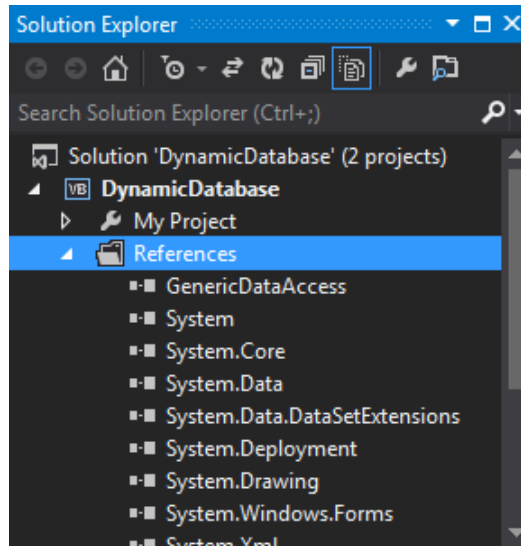
- Selanjutnya tambahkan 6 buah Label dengan Properties Text masing-masing “Nim, Nama, Alamat, Tempat Lahir, Tanggal Lahir dan Umur”.
- Lalu tambahkan 2 buah button dan ganti properties seperti dibawah.
  1. Name = btnSave, Text = Save
  2. Name = btnDelete, Text = Delete
- Dan terakhir tambahkan sebuah DataGridView dan ganti properties Name menjadi “DgvView”.
- Selanjutnya atur posisi masing-masing control seperti gambar dibawah ini.



- Add Reference Class Library (GenericDataAccess) yang telah kita buat sebelumnya.
  - a. Klik kanan pada Reference, dan klik Add Reference.



- b. Pada jendela Reference Manager Pilih Project, dan klik CheckBox pada nama project. Dan klik OK. Maka pada project reference akan terdapat “GenericDataAccess” seperti gambar dibawah.



- Import namespace GenericDataAccess.

```
Imports GenericDataAccess
```

- Selanjutnya ketikkan code seperti dibawah.

```
Dim MyAccessSiswa As ISiswaDAO = New SiswaDAO()  
  
Private Sub ClearForm()  
    txtNim.Text = ""  
    txtNama.Text = ""  
    txtTempat.Text = ""  
    txtAlamat.Text = ""  
    dtLahir.Text = Now.Date  
    txtUmur.Text = "0"  
End Sub  
  
Private Sub BindGrid()  
    DgvSiswa.DataSource = MyAccessSiswa.SelectSiswa()  
End Sub
```

- Dan pada event Form1\_Load ketikkan sintaks berikut.

```
Try  
    BindGrid()  
Catch ex As Exception  
    MsgBox(ex.Message)  
End Try
```

Method “BindGrid” berfungsi untuk mengambil data dan menampilkan pada GridView, sehingga pada setiap aplikasi dibuka akan menampilkan data-data yang ada pada database.

- Buat Event TextChanged pada dtLahir dan ketikkan sintaks dibawah.

```
Try
    txtUmur.Text = DateDiff(DateInterval.Year, CDate(dtLahir.Text), Now.Date)
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

- Klik ganda button Save, dan ketikkan sintaks berikut.

```
Try
    Dim siswa As New Siswa With {.Nim = txtNim.Text,
    .Nama = txtNama.Text, .Alamat = txtAlamat.Text, .TempatLahir =
    txtTempat.Text,
    .TglLahir = dtLahir.Text, .Umur = txtUmur.Text}

    If IsNothing(MyAccessSiswa.SelectSiswaByNim(txtNim.Text)) Then
        MyAccessSiswa.InsertSiswa(siswa)
    Else
        MyAccessSiswa.Updatesiswa(siswa)
    End If
    BindGrid()
    ClearForm()
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

Pada event ini, terdapat beberapa logic yang harus diperhatikan. Pertama-tama semua data akan disimpan kedalam Class Siswa yang telah kita buat sebelum nya pada Class Library. Selanjutnya system akan mengecek data siswa berdasarkan Nim. Jika tidak ada pada database maka data akan di tambah, jika ditemukan maka system akan menjalankan proses update data. Lalu data akan ditampilkan pada GridView dan yang terakhir form akan di refresh.

- Klik ganda button Delete dan ketikkan sintaks dibawah.

```
Try
    If IsNothing(MyAccessSiswa.SelectSiswaByNim(txtNim.Text)) Then
        Throw New Exception("Nim tidak ada pada table")
    End If
    MyAccessSiswa.DeleteSiswa(txtNim.Text)
    ClearForm()
    BindGrid()
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

Button ini melakukan operasi penghapusan data siswa berdasarkan Nim yang dicari. Seperti pada sintaks sebelumnya, system akan mencari data siswa berdasarkan Nim, jika tidak ditemukan maka sebuah message box akan tampil untuk menginformasikan bahwa Nim yang dimaksud tidak ada pada database. Jika Nim yang cari terdapat pada dabase maka proses hapus data akan dilakukan.

- Selanjutnya buat event CellClick untuk GgvView dan ketikkan sintaks dibawah ini.

```
Try
    If DgvSiswa.Rows(e.RowIndex).Cells("Nim").Value <> Nothing Then
        Dim mysiswa As Siswa =
MyAccessSiswa.SelectSiswaByNim(DgvSiswa.Rows(e.RowIndex).Cells("Nim").Value.ToString())
        txtNim.Text = mysiswa.Nim
        txtNama.Text = mysiswa>Nama
        txtAlamat.Text = mysiswa.Alatnat
        txtTempat.Text = mysiswa.TempatLahir
        dtLahir.Text = mysiswa.TglLahir
        txtUmur.Text = mysiswa.Umur
    End If
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

Sintaks ini berfungsi untuk mengambil data dari GridView dan menampilkannya kedalam textbox dan DateTimePicker berdasarkan baris yang dipilih.

- Langkah terakhir adalah dengan membuat Driver dan ConnectionString pada AppSettings. Value-value ini yang akan digunakan untuk mengakses ke DBMS yang di inginkan, sehingga jika pada masa yang akan datang akan terjadi perubahan pada jenis provider DBMS hal yang dilakukan adalah dengan mengubah value pada AppSettings.
- Buka file App.Config ketikkan XML format berikut.

```
<appSettings>
  <add key="Driver" value="System.Data.SqlClient"/>
  <add key="ConnectionString" value="Data Source=localhost;Initial
Catalog=Latihan;Integrated Security=True"/>

  <add key="Driver2" value="System.Data.OleDb"/>
  <add key="ConnectionString2" value="Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=Latihan.accdb;
Persist Security Info=False;"/>
</appSettings>
```

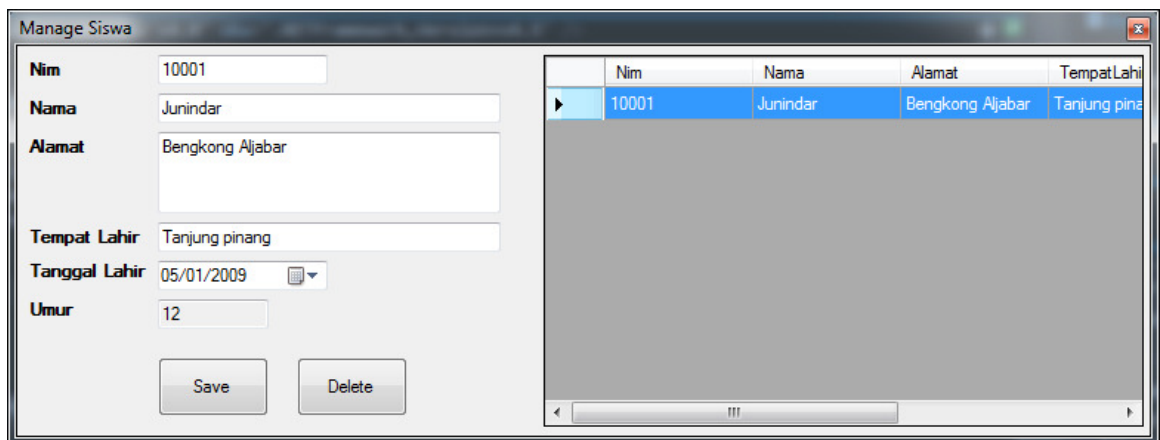
Sehingga seperti gambar dibawah.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <appSettings>
    <add key="Driver" value="System.Data.SqlClient" />
    <add key="ConnectionString" value="Data Source=localhost;Initial Catalog=Latihan;Integrated Security=True"/>
    <add key="Driver2" value="System.Data.OleDb" />
    <add key="ConnectionString2" value="Provider=Microsoft.ACE.OLEDB.12.0;Data Source=Latihan.accdb;Persist Security Info=False;" />
  </appSettings>
</configuration>
```

Jika diperhatikan terdapat 2 buah Driver dan ConnectionString pada code diatas.

Hal ini sengaja dilakukan penulis untuk memudahkan pengujian program.

Jalankan aplikasi dan coba semua fungsi yang terdapat pada form.



Untuk Source Code Project dapat di download disini

<http://junindar.blogspot.com/2014/02/membuat-aplikasi-dynamic-provider.html>

Semoga artikel ini dapat menambah wawasan kita semua khusus nya penulis sendiri.

Wassalam.



## Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Informatika di Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN-Bandung). Junindar mendapatkan Award Microsoft MVP VB pertanggal 1 oktober 2009 hingga saat ini. Senang mengutak-atik computer yang berkaitan dengan bahasa pemrograman. Keahlian, sedikit mengerti beberapa bahasa pemrograman seperti : VB.Net, C#, SharePoint, ASP.NET, VBA. Reporting: Crystal Report dan Report Builder. Database: MS Access, MY SQL dan SQL Server. Simulation / Modeling Packages: Visio Enterprise, Rational Rose dan Power Designer. Dan senang bermain gitar, karena untuk bisa menjadi pemain gitar dan seorang programmer sama-sama membutuhkan seni. Pada saat ini bekerja di salah satu Perusahaan Consulting dan Project Management di Malaysia sebagai Senior Consultant. Memiliki beberapa sertifikasi dari Microsoft yaitu Microsoft Certified Professional Developer (MCPD – SharePoint 2010), MOS (Microsoft Office Specialist) dan MCT (Microsoft Certified Trainer) Mempunyai moto hidup: **“Jauh lebih baik menjadi Orang Bodoh yang giat belajar, dari pada orang Pintar yang tidak pernah mengimplementasikan ilmunya”**.

*Kritik dan saran kirim ke : [junindar@gmail.com](mailto:junindar@gmail.com)*



## Referensi

1. [www.msdn.microsoft.com](http://www.msdn.microsoft.com)
2. [www.planetsourcecode.com](http://www.planetsourcecode.com)
3. [www.codeproject.com](http://www.codeproject.com)
4. [www.aspnet.com](http://www.aspnet.com)

Masih banyak lagi referensi yang ada di Intenet. Anda tinggal cari di [www.Google.com](http://www.Google.com).  
Dengan kata kunci “**tutorial VB.Net**”