

# 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

## **Pengenalan DAPPER (Micro ORM) Pada VB 2013**

Object Relational Mapping (ORM) adalah sebuah framework yang dapat menjembatani perbedaan sistembasis data yang bersifat relational dengan paradigma pengembangan aplikasi yang berorientasi objek. Setiap objek yang akan memetakan menjadi tabel-tabel pada basis data relasional dibungkus oleh suatu interface dengan menerapkan konsep design pattern. Hal tersebut bertujuan untuk memudahkan lapisan aplikasi (controller) mengakses data tersebut.

Untuk para developer yang telah menggunakan ORM seperti Linq to Sql, EF maupun NHibernate pasti akan merasakan benefitnya seperti *productivity* maupun *maintability* pada saat membangun aplikasi. Yang menjadi pertanyaan bagaimanakah

performa aplikasi yang menggunakan ORM ini? Salah satu isu yang masih “menghantui” ORM adalah masalah performa dalam pengaksesan data pada Database.

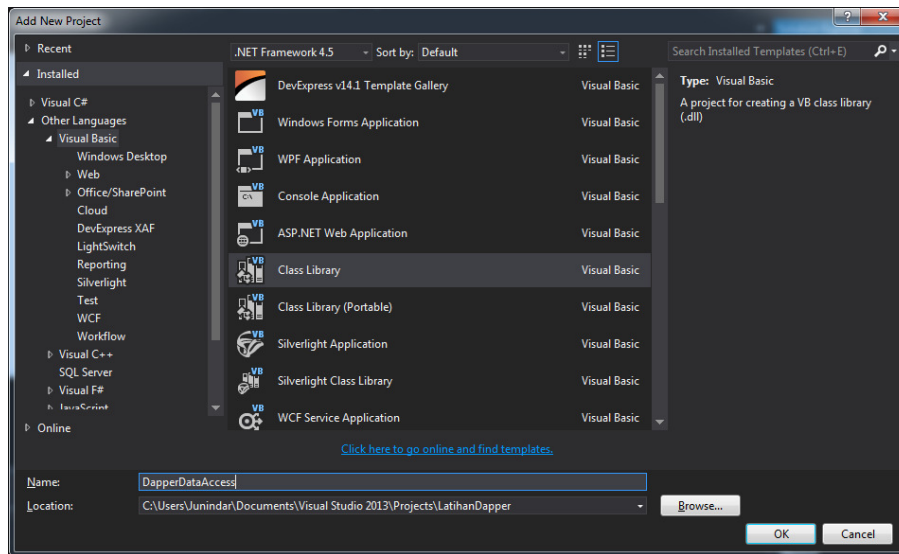
Bagi para developer yang ingin menggunakan ORM dengan performa yang lebih baik dapat menggunakan Micro ORM. Terdapat banyak Micro ORM pada .Net seperti Dapper, Peta POCO, FluentData dan lain-lain. Kebanyakan Micro ORM menggunakan *single file* dan sangat mudah untuk menggunakannya pada suatu project. Pada artikel ini akan dijelaskan bagaimana cara menggunakan salah satu Micro ORM yaitu Dapper.

Untuk mengetahui lebih lanjut, saya akan menjelaskan bagaimana membuat aplikasi dengan menggunakan Dapper sebagai Micro ORM. Ikuti langkah-langkah dibawah ini.

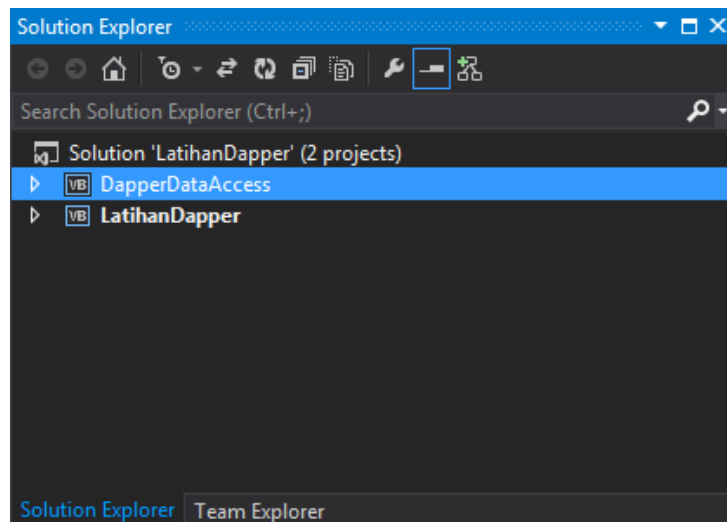
- Membuat database dengan nama Latihan pada SQL Server. Selanjutnya buat satu buah table dengan nama tblSiswa, dengan field-field sebagai berikut.

<b>Nama Field</b>	<b>Field Type</b>
Nim	Varchar (15)
Nama	Varchar (50)
Alamat	Varchar (255)
TempatLahir	Varchar (50)
TglLahir	Date
Umur	Integer

- Membuat Class Library
  - Buat sebuah project dengan nama “LatihanDapper”.
  - Lalu pada project tersebut, pilih File > Add Project
  - Pilih Class Library dan ganti nama project tersebut menjadi “DapperDataAccess”. Pada aplikasi ini terdapat beberapa layer, dimana untuk koneksi dan manipulasi data akan dilakukan pada Class Library (DapperDataAccess). 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.

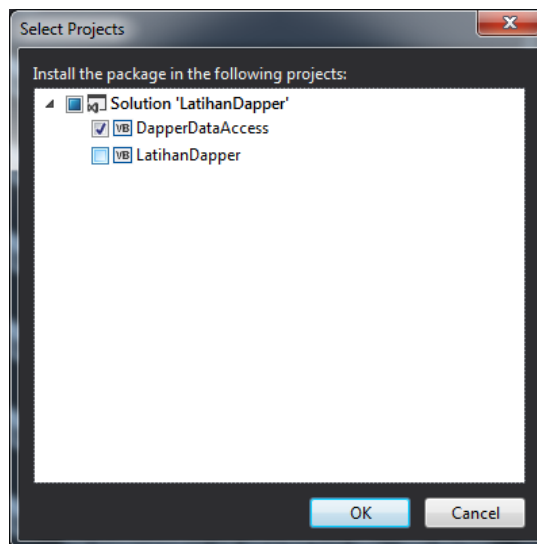
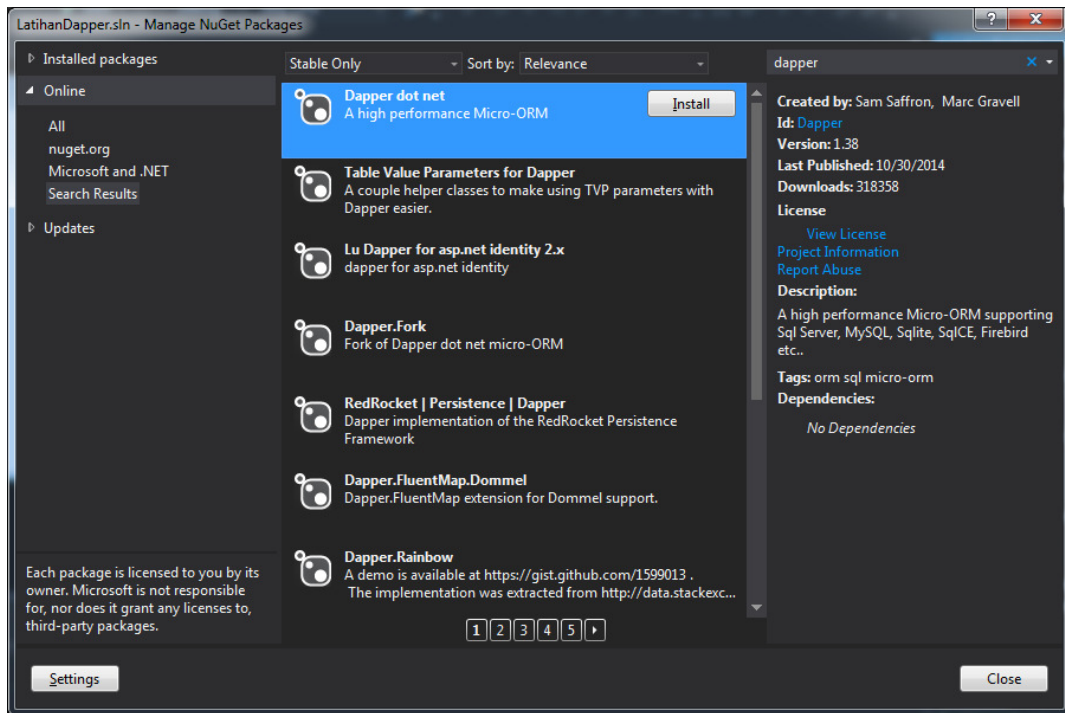


- Sehingga pada Solution akan terdapat dua project seperti pada gambar.

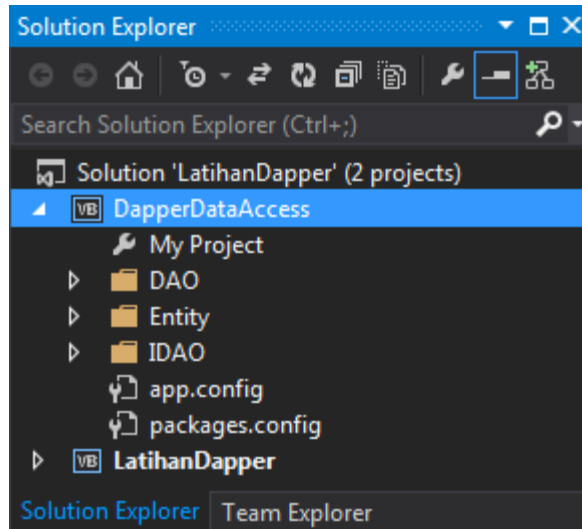


- Pada DapperDataAccess, klik Tools > NuGet Package Manager > Manage NuGet Packages for Solution

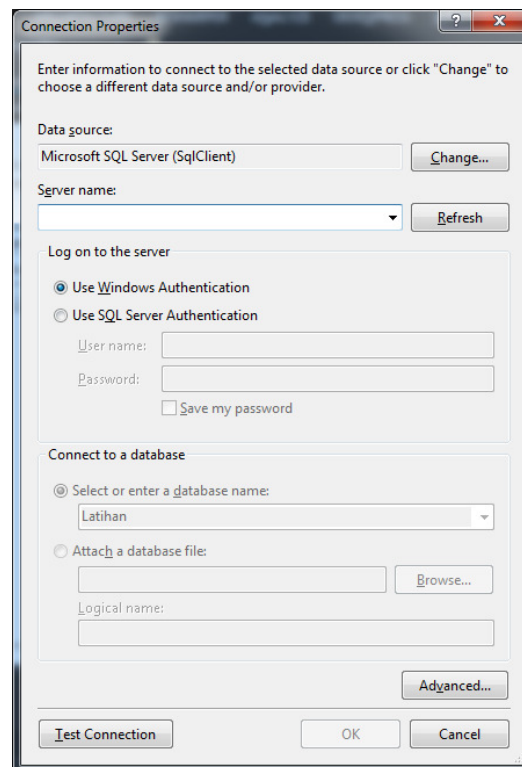
- Pada jendela Manage NuGet Package, search “Dapper”, lalu install Dapper.



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



- Selanjutnya adalah membuat connection ke Database yang akan digunakan. Klik kanan project “DapperDataAccess”, selanjutnya pilih Tab “Setting”. Pada kolom Name ketikkan “myConDapper”, kolom Type pilih (Connection string), selanjutnya klik browse pada kolom value.



Pada Connection Properties, pilih Server name dan Database dan klik Test Connection dan OK button.

- 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.

- Selanjutnya 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 import SqlClient dan Dapper seperti dibawah.

```
Imports System.Data.SqlClient
Imports Dapper
```

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

```
Public Sub DeleteSiswa(Nim As String) Implements ISiswaDAO.DeleteSiswa
    Using mycon As SqlConnection = New
        SqlConnection(My.Settings.myConDapper)
        Const strQuery As String = "Delete From tblSiswa Where Nim=@Nim"
        mycon.Open()
        mycon.Query(strQuery, New With {Nim})
        mycon.Close()
    End Using
End Sub

Public Sub InsertSiswa(Entity As Siswa) Implements ISiswaDAO.InsertSiswa
    Using mycon As SqlConnection = New
        SqlConnection(My.Settings.myConDapper)
        Const strQuery As String = "Insert Into tblSiswa
(Nim>Nama>Alamat>TempatLahir>TglLahir>Umur) " &
        " Values (@Nim,@Nama,@Alamat,@TempatLahir,@TglLahir,@Umur)"
        mycon.Open()
        mycon.Query(strQuery, Entity)
    End Using
End Sub
```

```
Public Sub Updatesiswa(Entity As Siswa) Implements ISiswaDAO.Updatesiswa
    Using mycon As SqlConnection = New
        SqlConnection(My.Settings.myConDapper)
        Const strQuery As String = "Update tblSiswa Set
        Nama=@Nama,Alamat=@Alamat, " &
        " TempatLahir=@TempatLahir,TglLahir=@TglLahir,Umur=@Umur Where
        Nim=@Nim "
        mycon.Open()
        mycon.Query(strQuery, Entity)
    End Using
End Sub

Public Function SelectSiswa() As List(Of Siswa) Implements
ISiswaDAO.SelectSiswa
    Using mycon As SqlConnection = New
        SqlConnection(My.Settings.myConDapper)
        Const strQuery As String = "SELECT * FROM tblSiswa Order By Nim"
        mycon.Open()
        Return mycon.Query(Of Siswa)(strQuery)
    End Using
End Function

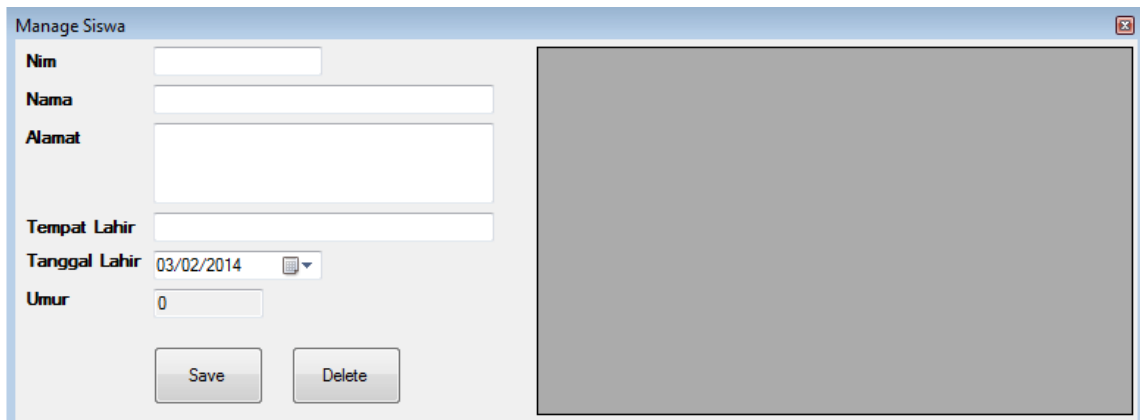
Public Function SelectSiswaByNim(Nim As String) As Siswa Implements
ISiswaDAO.SelectSiswaByNim
    Using mycon As SqlConnection = New
        SqlConnection(My.Settings.myConDapper)
        Const strQuery As String = "SELECT * FROM tblSiswa Where Nim=@Nim"
        mycon.Open()
        Return mycon.Query(Of Siswa)(strQuery, New With
        {Nim}).FirstOrDefault()
    End Using
End Function
```

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

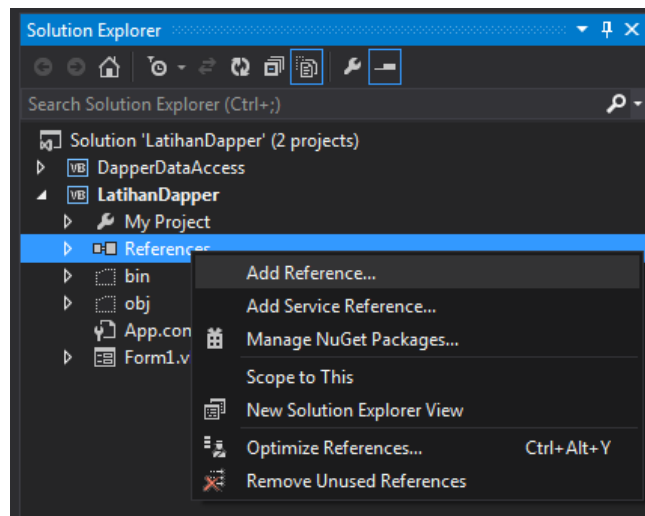
- Membuat User Interface pada aplikasi. (ProjectDynamicDatabase)
  - Tambahkan 5 buah textbox pada Form1. Dengan properties masing-masing sebagai berikut.
    1. Name = txtNim
    2. Name = txtNama
    3. Name = txtAlamat, Multiline = True
    4. Name = txtTempat
    5. Name = txtUmur, ReadOnly = True



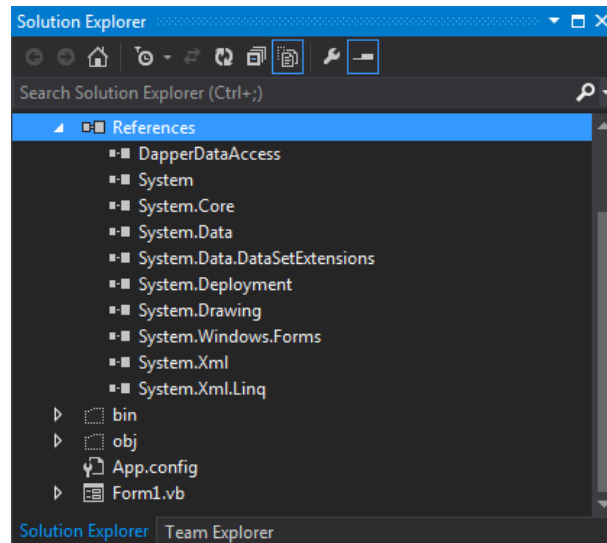
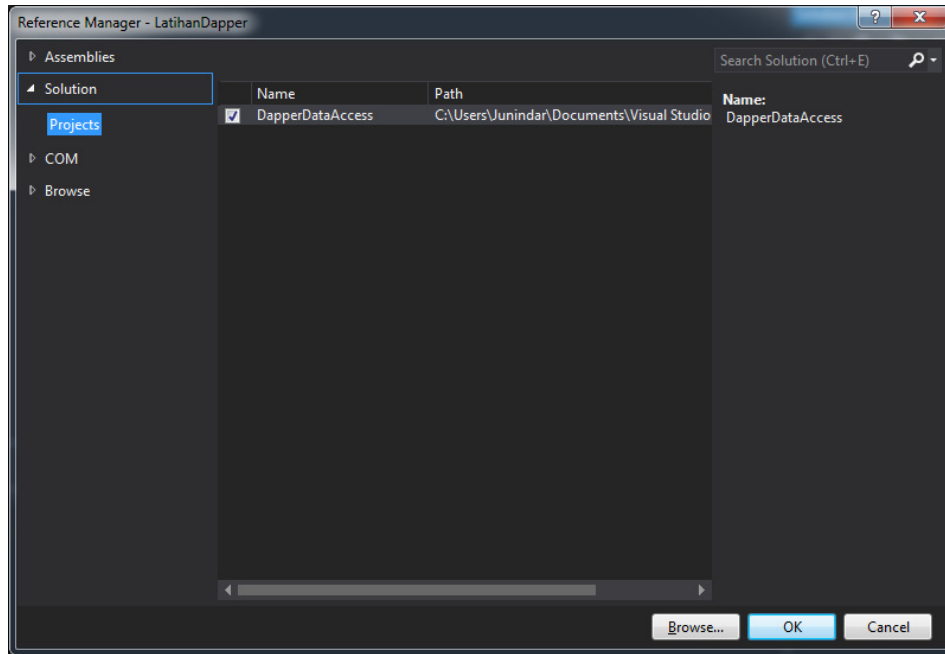
- Tambahkan sebuah DateTimePicker dan ganti properties Name menjadi “dtLahir”.
- Selanjutnya tambahkan 6 buah Label dengan Properties Text masing-masing “Nim, Nama, Alamat, Tempat Lahir, Tanggal Lahir dan Umur”.
- Lalu tambahkan 2 buah button dan ganti properties seperti dibawah.
  1. Name = btnSave, 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 (DapperDataAccess) 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 “DapperDataAccess” seperti gambar dibawah.



- Import namespace GenericDataAccess.

```
Imports DapperDataAccess
```

- 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
```

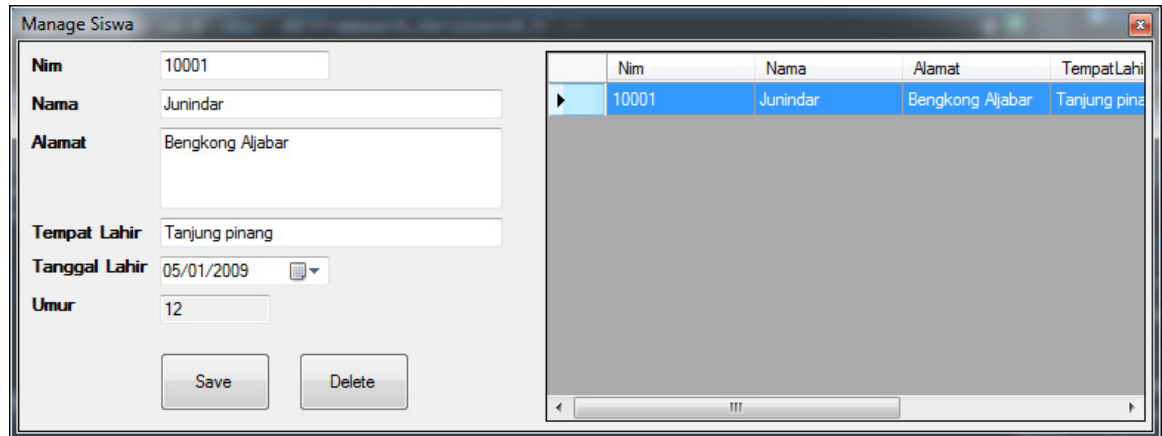
Button ini melakukan operasi penghapusan data siswa berdasarkan Nim yang dicari. Seperti pada sintaks sebelumnya, system akan mencari data siswa berdasarkan Nim, jika tidak ditemukan maka sebuah message box akan tampil untuk menginformasikan bahwa Nim yang dimaksud tidak ada pada database. Jika Nim yang 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.To
oString())
        txtNim.Text = mysiswa.Nim
        txtNama.Text = mysiswa>Nama
        txtAlamat.Text = mysiswa.Alamat
        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.

Jalankan aplikasi dan coba semua fungsi yang terdapat pada form.



The screenshot shows a window titled "Manage Siswa". On the left is a form with the following fields:

- Nim: 10001
- Nama: Junindar
- Alamat: Bengkong Aljabar
- Tempat Lahir: Tanjung pinang
- Tanggal Lahir: 05/01/2009
- Umur: 12

Below the form are "Save" and "Delete" buttons. On the right is a table with the following data:

Nim	Nama	Alamat	TempatLahir
10001	Junindar	Bengkong Aljabar	Tanjung pina

Untuk Source Code Project dapat di download disini

<http://junindar.blogspot.com/2014/12/pengenalan-dapper-micro-orm-pada-vb-2013.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**”