

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

DAPPER (Micro ORM) dan Stored Procedure Pada VB 2013

Pada artikel ini akan dijelaskan bagaimana menggunakan *stored procedure* pada *Dapper*. Sebaiknya untuk lebih memahami artikel ini, disarankan membaca dan mempraktekkan pada artikel sebelumnya disini (<http://junindar.blogspot.com/2014/12/pengenalan-dapper-micro-orm-pada-vb-2013.html>). Pada artikel sebelumnya telah dijelaskan bagaimana penggunaan Dapper dengan Visual Basic dan disertai dengan sample source code (untuk source code juga disediakan Bahasa pemrograman C#). Disana dapat dilihat syntax-syntax SQL seperti insert, delete, update maupun select kita letakkan semua pada baris code (VB/C#).

Sedangkan pada artikel ini semua syntax SQL diatas akan kita letakkan pada stored procedure yang ada pada MS SQL Server. Stored procedure adalah sebuah kelompok

syntax SQL yang disimpan didalam katalog database dan dapat dipanggil kemudian baik dari Query Editor maupun dari program aplikasi lain. Banyak sekali keuntungan jika menggunakan stored procedure ini baik dari segi performance maupun security nya. Untuk lebih jelas mengenai stored procedure pada MS SQL Server dapat dilihat disini ([http://technet.microsoft.com/en-us/library/aa174792\(v=sql.80\).aspx](http://technet.microsoft.com/en-us/library/aa174792(v=sql.80).aspx)). Sedangkan pada artikel ini akan dijelaskan bagaimana membuat aplikasi dengan menggunakan Dapper dan stored procedure . Diasumsikan para pembaca telah mengerti dan menyelesaikan latihan pada artikel sebelumnya (Pengenalan Dapper) sehingga akan lebih mudah dalam mengikuti dan menyelesaikan latihan pada artikel ini.

Untuk mengetahui lebih lanjut, saya akan menjelaskan bagaimana membuat aplikasi dengan menggunakan Dapper dan stored procedure . Ikuti langkah-langkah dibawah ini.

- Buka SQL Server Management Studio, pada Database Latihan yang telah kita buat sebelumnya expand node “Programmability > stored procedure”.
- Pada node stored procedure klik kanan > New stored procedure, dan ketikkan syntax SQL seperti dibawah.

```
USE [Latihan]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[uspInsertSiswa] (
    @Nim varchar(15),
    @Nama varchar(50),
    @Alamat varchar(255),
    @TempatLahir varchar(50),
    @TglLahir date,
    @Umur int
)
AS
BEGIN
    Insert into tblSiswa (Nim, Nama, Alamat, TempatLahir, TglLahir, Umur)
    Values (@Nim, @Nama, @Alamat, @TempatLahir, @TglLahir, @Umur)
END
```

Syntax diatas merupakan syntax untuk membuat stored procedure Insert pada table Siswa (tblSiswa), dengan nama stored procedure nya adalah “uspInsertSiswa”. Lalu klik button execute untuk membuat stored procedure tersebut, Selanjutnya lakukan hal yang sama untuk membuat 4 (empat) stored procedure lainnya. Dengan syntax dibawah ini.

```
USE [Latihan]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[uspUpdateSiswa] (
    @Nim varchar(15),
    @Nama varchar(50),
    @Alamat varchar(255),
    @TempatLahir varchar(50),
    @TglLahir date,
    @Umur int
)
AS
BEGIN
    Update tblSiswa Set Nama=@Nama, Alamat=@Alamat,
    TempatLahir=@TempatLahir, TglLahir=@TglLahir, Umur=@Umur Where
    Nim=@Nim
END
```

```
USE [Latihan]
GO

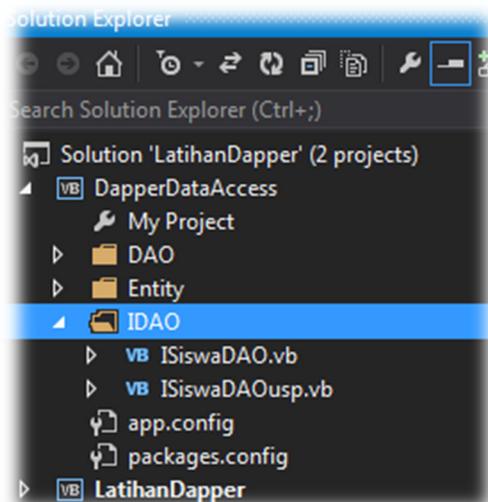
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[uspDeleteSiswa] (
    @Nim varchar(15)
)
AS
BEGIN
    Delete From tblSiswa Where Nim=@Nim
END
```

```
USE [Latihan]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[uspSelectSiswaByNim] (
@Nim varchar(15)
)
AS
BEGIN
SELECT * FROM tblSiswa Where Nim=@Nim
END
```

```
USE [Latihan]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[uspSelectSiswa]
AS
BEGIN
SELECT * FROM tblSiswa Order By Nim
END
```

- Buka Solution sebelumnya (Latihan Dapper). Pada project “DapperDataAccess”, tambahkan satu Interface di folder “IDAO” dengan nama “ISiswaDAOusp.vb”.



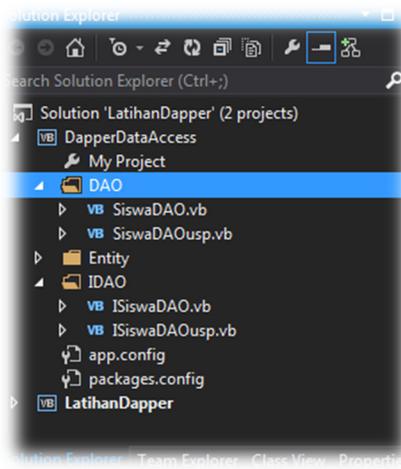
- Lalu ketikkan syntax berikut pada interface yang baru dibuat.

```
Public Interface ISiswaDAOusp

    Sub InsertSiswaUsp(ByVal Entity As Siswa)
    Sub UpdatesiswaUsp(ByVal Entity As Siswa)
    Sub DeleteSiswaUsp(ByVal Nim As String)
    Function SelectSiswaUsp() As List(Of Siswa)
    Function SelectSiswaByNimUsp(ByVal Nim As String) As Siswa

End Interface
```

- Buat sebuah class pada folder DAO dengan nama “SiswaDAOusp.vb”.



- Pada code “Class SiswaDAOusp”, ketikkan “Implements ISiswaDAOusp” dan enter. Maka secara otomatis akan dibuat method sesuai dengan Interface diatas. Seperti gambar dibawah

```
SiswaDAOusp.vb
SiswaDAOusp
- InsertSiswaUsp

References
Public Class SiswaDAOusp
    Implements ISiswaDAOusp

    References
    Public Sub InsertSiswaUsp(ByVal Entity As Siswa) Implements ISiswaDAOusp.InsertSiswaUsp
        Throw New NotImplementedException()
    End Sub

    References
    Public Sub UpdatesiswaUsp(ByVal Entity As Siswa) Implements ISiswaDAOusp.UpdatesiswaUsp
        Throw New NotImplementedException()
    End Sub

    References
    Public Sub DeleteSiswaUsp(ByVal Nim As String) Implements ISiswaDAOusp.DeleteSiswaUsp
        Throw New NotImplementedException()
    End Sub

    References
    Public Function SelectSiswaUsp() As List(Of Siswa) Implements ISiswaDAOusp.SelectSiswaUsp
        Throw New NotImplementedException()
    End Function

    References
    Public Function SelectSiswaByNimUsp(ByVal Nim As String) As Siswa Implements ISiswaDAOusp.SelectSiswaByNimUsp
        Throw New NotImplementedException()
    End Function
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 InsertSiswaUsp(ByVal Entity As Siswa) Implements
ISiswaDAOusp.InsertSiswaUsp
    Using mycon As SqlConnection = New
SqlConnection(My.Settings.myConDapper)
        mycon.Open()
        mycon.Query("uspInsertSiswa", Entity,
commandType:=CommandType.storedProcedure)
    End Using
End Sub

Public Sub UpdatesiswaUsp(ByVal Entity As Siswa) Implements
ISiswaDAOusp.UpdatesiswaUsp
    Using mycon As SqlConnection = New
SqlConnection(My.Settings.myConDapper)
        mycon.Open()
        mycon.Query("uspUpdateSiswa", Entity,
commandType:=CommandType.storedProcedure)
    End Using
End Sub

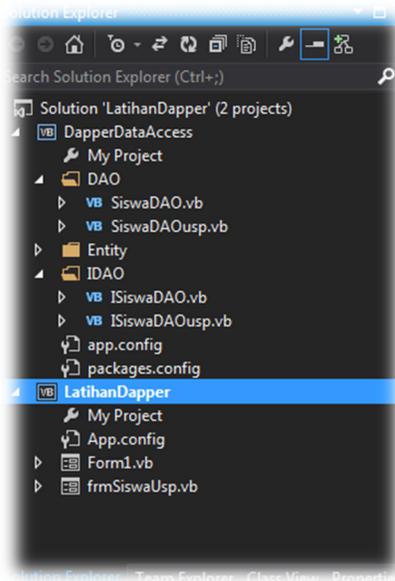
Public Sub DeleteSiswaUsp(ByVal Nim As String) Implements
ISiswaDAOusp.DeleteSiswaUsp
    Using mycon As SqlConnection = New
SqlConnection(My.Settings.myConDapper)
        mycon.Open()
        mycon.Query("uspDeleteSiswa", New With {Nim},
commandType:=CommandType.storedProcedure)
    End Using
End Sub

Public Function SelectSiswaUsp() As List(Of Siswa) Implements
ISiswaDAOusp.SelectSiswaUsp
    Using mycon As SqlConnection = New
SqlConnection(My.Settings.myConDapper)
        mycon.Open()
        Return mycon.Query(Of Siswa)("uspSelectSiswa",
commandType:=CommandType.storedProcedure)
    End Using
End Function
```

```
Public Function SelectSiswaByNimUsp(ByVal Nim As String) As Siswa Implements
ISiswaDAOusp.SelectSiswaByNimUsp
    Using mycon As SqlConnection = New
SqlConnection(My.Settings.myConDapper)
        mycon.Open()
        Return mycon.Query(Of Siswa)("uspSelectSiswaByNim", New With {Nim},
commandType:=CommandType.storedProcedure).FirstOrDefault()
    End Using
End Function
```

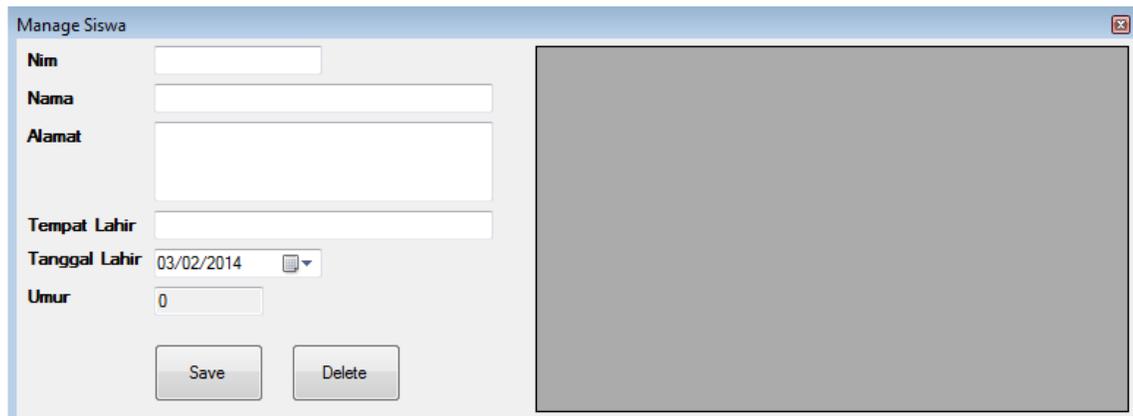
Sekarang kita telah selesai membuat Class SiswaDAOusp dan langkah selanjutnya adalah bekerja pada layer User Interface.

- Membuat User Interface pada aplikasi. (ProjectDynamicDatabase)
- Tambahkan sebuah form dengan nama “frmSiswaUsp”.



- Tambahkan 5 buah textbox pada frmSiswaUsp. 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.



- Import namespace GenericDataAccess.

```
Imports DapperDataAccess
```

- Selanjutnya ketikkan code seperti dibawah.

```
Dim MyAccessSiswa As ISiswaDAOusp = New SiswaDAOusp()  
  
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.SelectSiswaUsp()  
End Sub
```

- Dan pada event frmSiswaUsp_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.SelectSiswaByNimUsp(txtNim.Text)) Then
        MyAccessSiswa.InsertSiswaUsp(siswa)
    Else
        MyAccessSiswa.UpdatesiswaUsp(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.SelectSiswaByNimUsp(txtNim.Text)) Then
        Throw New Exception("Nim tidak ada pada table")
    End If
    MyAccessSiswa.DeleteSiswaUsp(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.

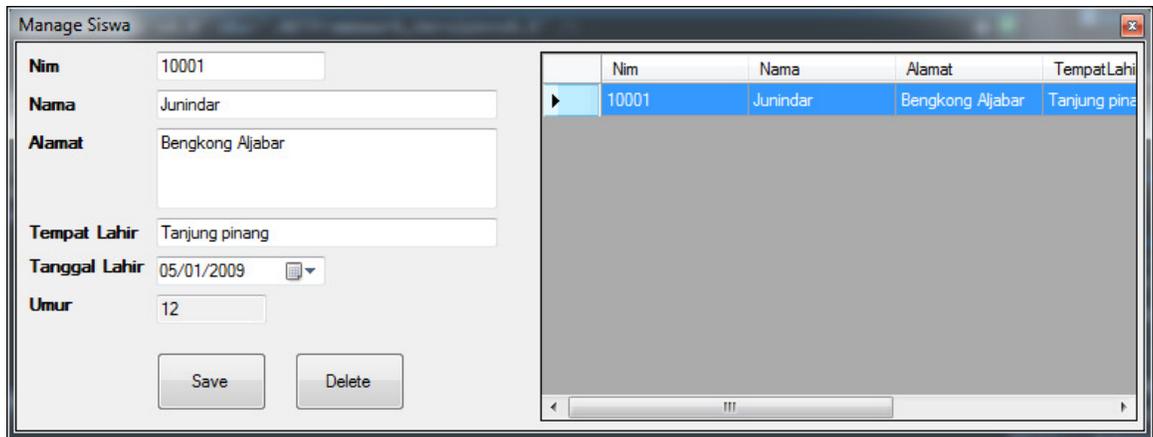
- Selanjutnya buat event CellClick untuk GgvView dan ketikkan sintaks dibawah ini.

```
Try
    If DgvSiswa.Rows(e.RowIndex).Cells("Nim").Value <> Nothing Then
        Dim mysiswa As Siswa =
MyAccessSiswa.SelectSiswaByNimUsp(DgvSiswa.Rows(e.RowIndex).Cells("Nim").Value.ToString())
        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.

Jika kita perhatikan sebenarnya tidak banyak code yang berubah pada user interface ini. Yang perlu diganti adalah nama class dan method-method yang baru kita buat sebelumnya. Ini merupakan salah satu kelebihan jika kita memisahkan code antara user interface dan logic layer. Jika kita pada Interface ISiswaDAOusp kita menggunakan nama-nama method yang sama pada ISiswaDAO, maka pada user interface yang perlu kita lakukan hanya mengganti pada saat mendeklarasikan “MyAccessSiswa”.

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

At the bottom of 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/2015/01/dapper-micro-orm-dan-stored-procedure.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

Referensi

1. www.msdn.microsoft.com
2. www.planetsourcecode.com
3. www.codeproject.com
4. www.aspnet.com

Masih banyak lagi referensi yang ada di Intenet. Anda tinggal cari di www.Google.com.
Dengan kata kunci “**tutorial VB.Net**”