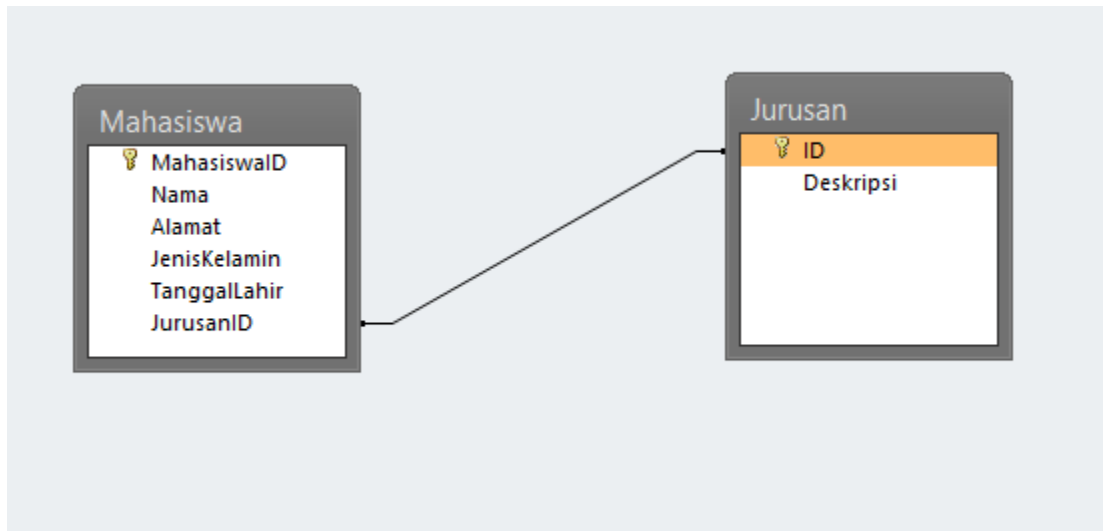


BAB VIII

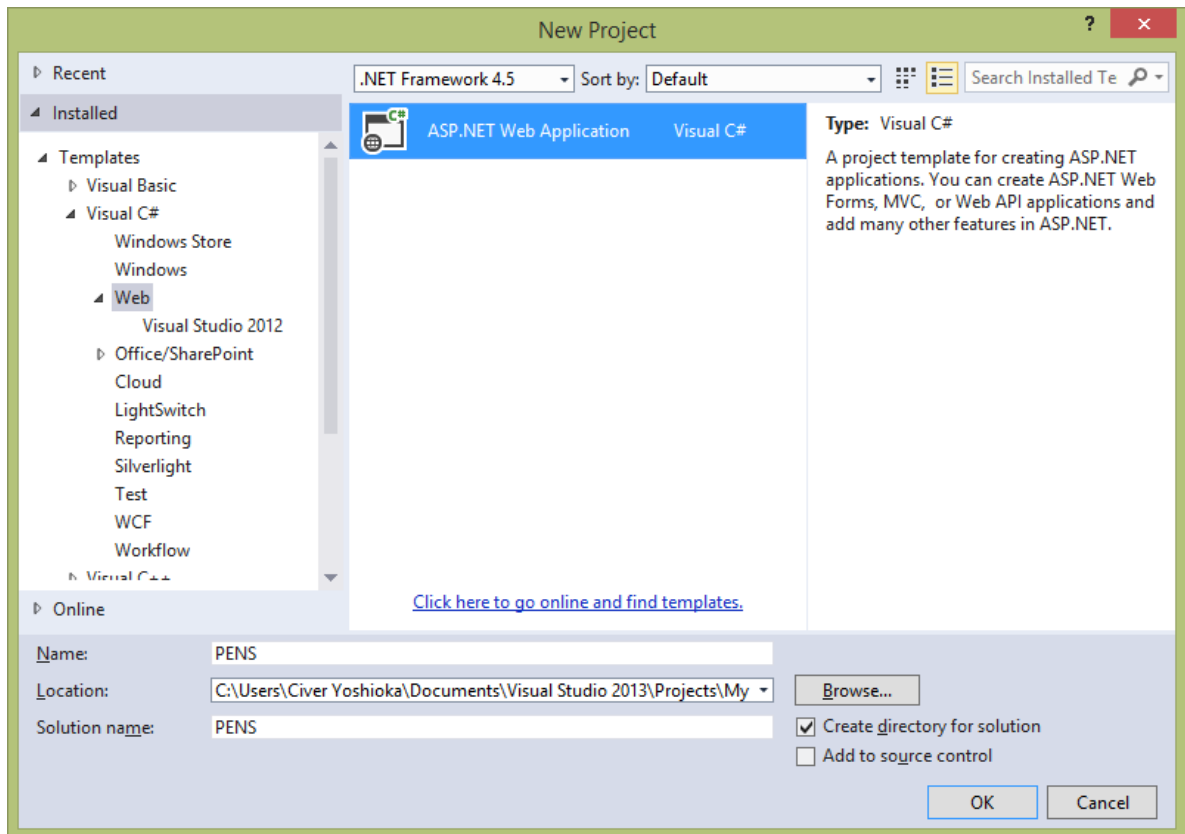
MVC .NET Lanjut

Pada percobaan kali ini adalah mencoba membuat aplikasi generate Model menggunakan MVC 5 dengan dua buah tabel yang berelasi. Misalkan ada dua buah tabel yang berelasi sebagai berikut.

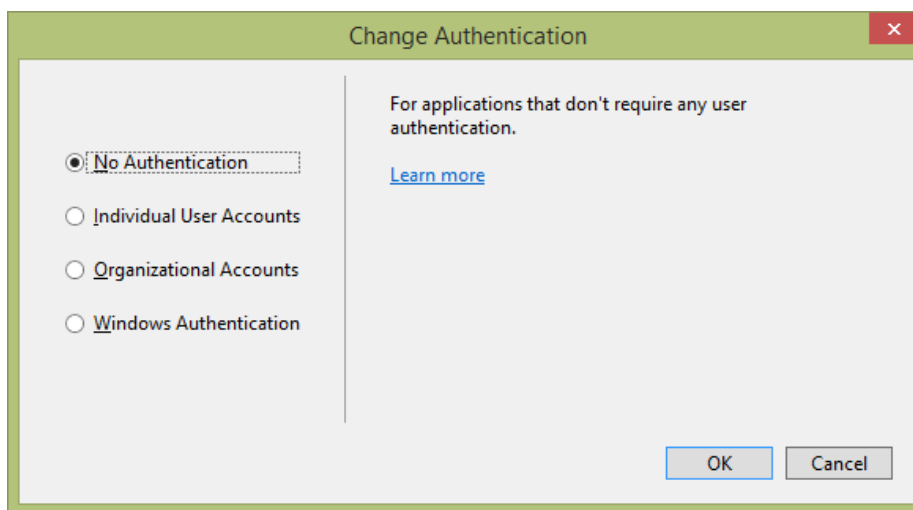


Ada dua buah tabel yang saling berkorelasi. Setiap data pada Tabel **Mahasiswa** yang dimasukan harus memilih Jurusan dari Data Mahasiswa tersebut, sedangkan data Jurusan berada pada tabel Master tersendiri yaitu pada Tabel **Jurusan**. *JurusanID* pada Tabel **Mahasiswa** adalah *Foreign Key* dari *ID* pada tabel **Jurusan**. Untuk membuat form yang menggunakan Generate Data Access Model dari MVC 5 langkahnya adalah sebagai berikut.

1. Membuat Projek dengan menggunakan MVC 5 Template **File > New > Project > Web > ASP.NET Web Application**.



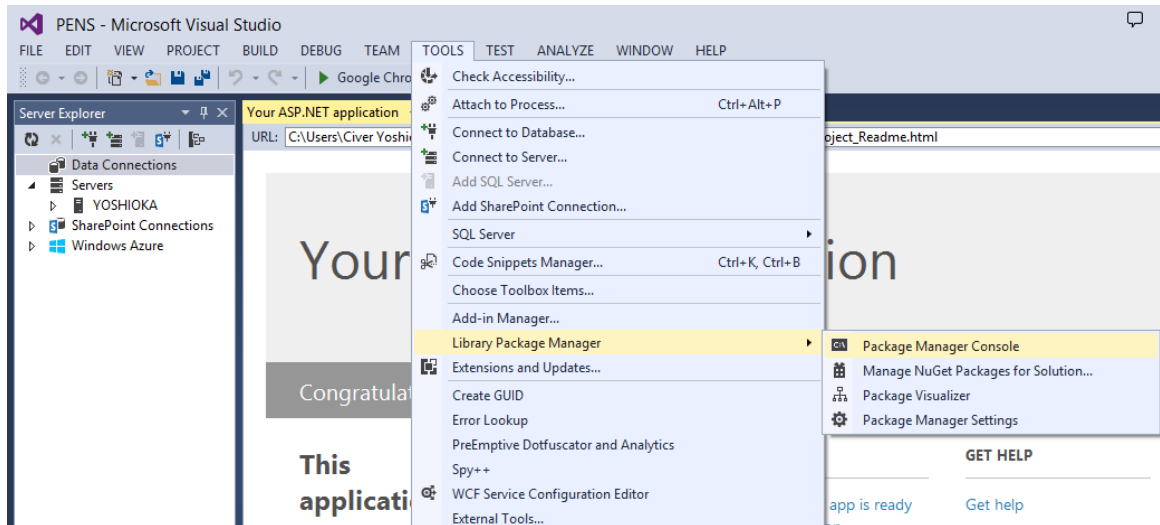
Kemudian Pilih **OK**, selanjutnya pilih **Template MVC** dan klik pada **Change Authentication** pilih **No Authentication**. Karena untuk contoh ini untuk sementara tidak membutuhkan authentication user untuk login.



Kemudian pilih **Ok**, sampai di sini projek telah berhasil dibuat

2. Install Entity Framework 6

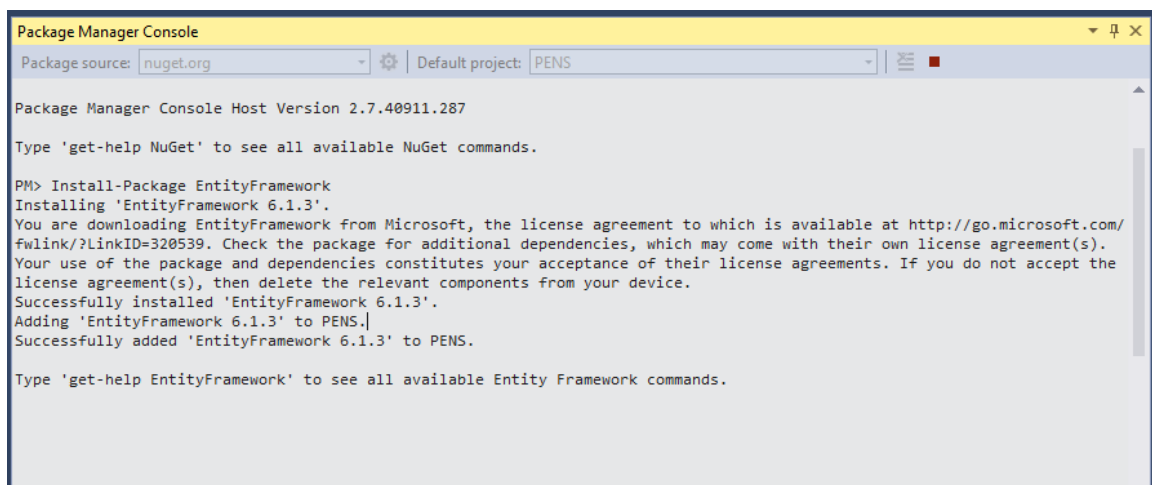
Untuk menginstall Entity Framework 6, buka pada Menu **TOOLS > Library Package Manager > Package Manager Console**



Setelah Console Manager muncul pada Jendela Visual Studio, Ketikkan perintah berikut pada Console Manager.

Install-Package EntityFramework

*Pada instalasi ini koneksi internet adalah direct, jika Anda menggunakan proxy pada instalasi anda silahkan disesuaikan.



Silakan tunggu beberapa saat sampai penambahan **Entity Framework** selesai terinstal. Gambar di atas adalah keadaan setelah **Entity Framework** berhasil di instal.

3. Membuat Data Model dari entity Mahasiswa dan Jurusan

Tambahkan pada direktori dua buah **Model** yaitu *Mahasiswa.cs* dan *Jurusan.cs* yang isinya adalah sebagai berikut.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace PENS.Models
{
    public class Jurusan
    {
        public int ID { get; set; }
        public string Deskripsi { get; set; }
        public virtual ICollection<Mahasiswa> Mahasiswas { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

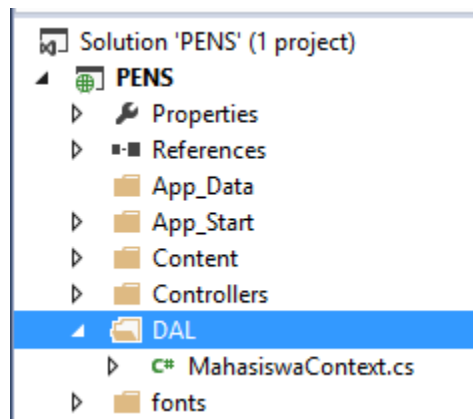
namespace PENS.Models
{
    public class Mahasiswa
    {
        public int MahasiswaID { get; set; }
        public string Nama { get; set; }
        public string Alamat { get; set; }
        public string JenisKelamin { get; set; }
        public DateTime TanggalLahir { get; set; }
        public int JurusanID { get; set; }

        public virtual Jurusan Jurusan { get; set; }
    }
}
```

4. Membuat Kelas Database Context

Kelas ini adalah kelas untuk menghubungkan dan memetakan entiti dari Model ke Field pada Database. Kalau pada modul sebelumnya kelas ini menjadi satu dengan kelas Model. Karena pada Modul ini Model yang ditangani oleh kelas *Database Context* adalah lebih dari satu Model, maka kelas Database Context dibuat terpisah pada Folder dan Kelas tersendiri untuk memudahkan.

Langkah yang harus dilakukan adalah, buatlah Folder sembarang pada projek Anda dan buatlah sebuah kelas dalam folder tersebut **Klick Kanan** pad Projek Anda **Add > New Folder** setelah Folder berhasil dibuat kemudian **Klick Kanan** pada Folder tersebut **Add > Class**. Pada Modul ini langkahnya adalah sebagai berikut. Buat Folder DAL (Data Access Layer) kemudian tambahkan file *MahasiswaContext.cs* sehingga terlihat pada seperti berikut.



Kemudian ubahlah isi dari kelas *MahasiswaContext.cs* menjadi seperti berikut.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

using PENS.Models;
using System.Data.Entity;
using System.Data.Entity.ModelConfiguration.Conventions;

namespace PENS.DAL
{
    public class MahasiswaContext : DbContext
    {
        public MahasiswaContext() : base("MahasiswaContext")
        {
        }

        public DbSet<Mahasiswa> Mahasiswas { get; set; }
    }
}
```

```

        public DbSet<Jurusan> Jurusans { get; set; }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
        }
    }
}

```

Berikut penjelasan Kode

```

public MahasiswaContext() : base("MahasiswaContext")
{
}

```

Kode di atas adalah untuk men-load koneksi ke database lokal yang nanti akan dibahas pada sub-bab selanjutnya.

```

protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
}

```

Kode di atas digunakan untuk mengonversi nama Model ke dalam Nama tabel pada Database yang nantinya di Generate secara Otomatis.

5. Setting Koneksi ke local Database

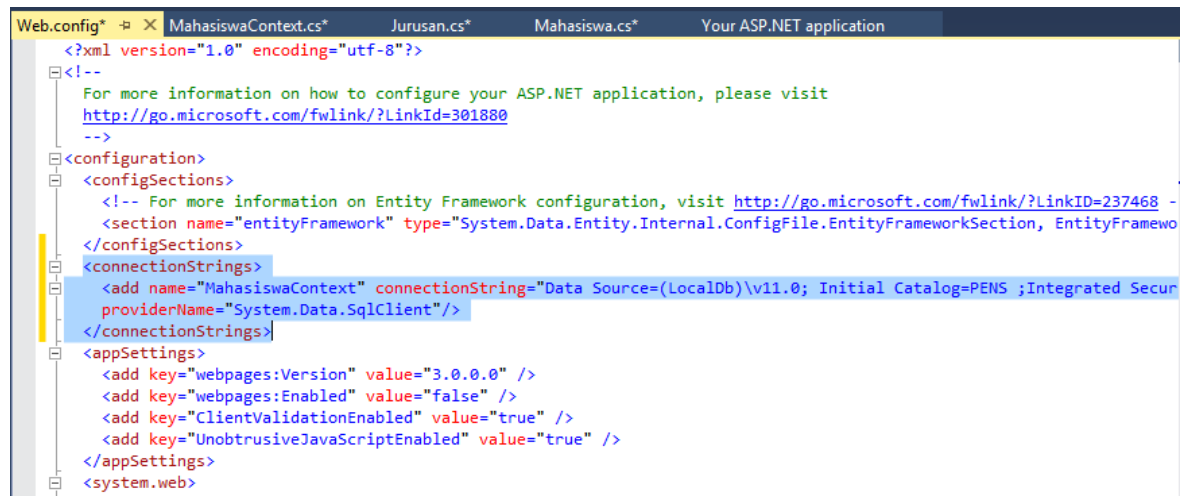
Buka pada file *Web.config* dan tambahkan kode berikut untuk membuat koneksi

```

<connectionStrings>
  <add name="MahasiswaContext" connectionString="Data Source=(LocalDb)\v11.0;
Initial Catalog=PENS ;Integrated Security=SSPI"
  providerName="System.Data.SqlClient"/>
</connectionStrings>

```

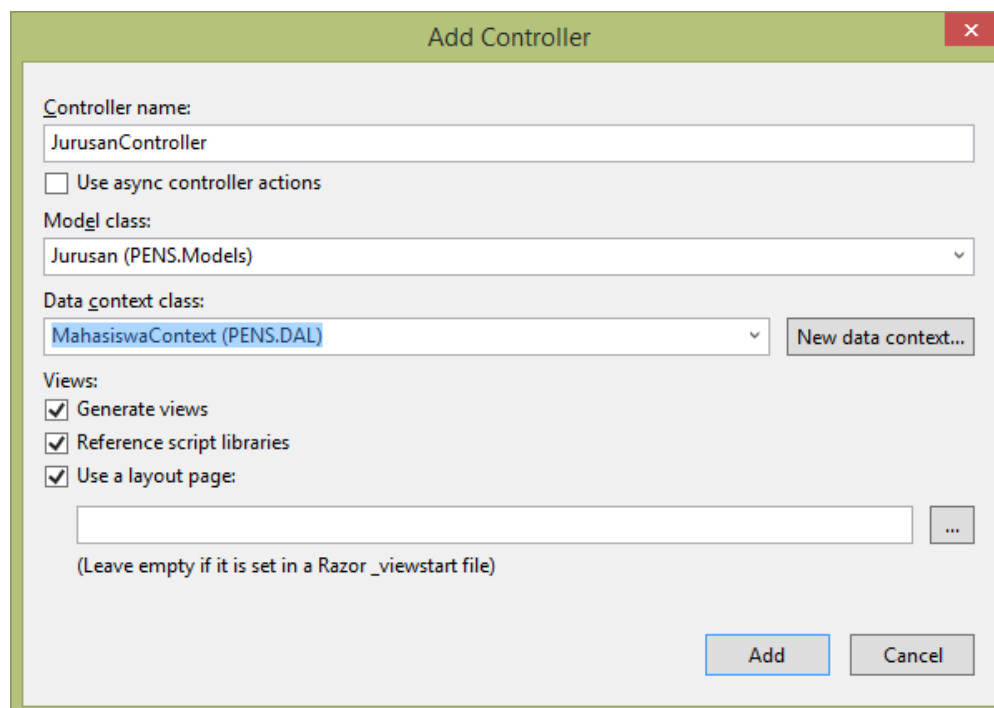
Tambahkan kode koneksi tersebut diantara **Tag** `</configSections>` dan `<appSettings>` sehingga terlihat seperti berikut.



```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=301880
-->
<configuration>
  <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkId=237468 -->
  <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework" />
</configSections>
<connectionStrings>
  <add name="MahasiswaContext" connectionString="Data Source=(LocalDb)\v11.0; Initial Catalog=PENS ;Integrated Security=SSPI" providerName="System.Data.SqlClient"/>
</connectionStrings>
<appSettings>
  <add key="webpages:Version" value="3.0.0.0" />
  <add key="webpages:Enabled" value="false" />
  <add key="ClientValidationEnabled" value="true" />
  <add key="UnobtrusiveJavaScriptEnabled" value="true" />
</appSettings>
</configuration>
```

6. Generate Access Data Object Mapping

Kemudian Generate-lah **Controller** dari masing-masing **Model** yang telah dibuat untuk membuat CRUD prosesnya seperti pada Modul yang telah dipelajari *"pada minggu sebelumnya"*. Berikut adalah Gambar untuk membuat masing-masing **Controller** .



Gambar di atas adalah nama **Controller Jurusan** dari Object Mapping **Model Jurusan**

Add Controller

Controller name:

☐ Use async controller actions

Model class:

Data context class:
 New data context...

Views:
☒ Generate views
☒ Reference script libraries
☒ Use a layout page:
 ...
 (Leave empty if it is set in a Razor _viewstart file)

Add Cancel

Gambar di atas adalah nama **Controller Mahasiswa** dari Object Mapping **Model Mahasiswa**.

Jika Generate Data Access Model yang anda lakukan berhasil maka akan tampak beberapa Form yang bisa anda buka seperti berikut.

Application name Home About Contact

Index

[Create New](#)

Deskripsi	
Teknik Informatik	Edit Details Delete
Teknik Elektronika	Edit Details Delete
Teknik Telekomunikasi	Edit Details Delete
Teknik Komputer	Edit Details Delete
Teknik Mekatronika	Edit Details Delete

© 2016 - My ASP.NET Application

Karena tabel yang dibuat adalah memiliki korelasi yang mana **ID** dari Tabel **Jurusan** akan menjadi *Foreign Key* pada Tabel **Mahasiswa** maka yang pertama harus di isi adalah Data dari tabel Jurusan. Setelah tabel diisi akan terlihat seperti pada gambar di atas

Create
Mahasiswa

Nama: Civer Yoshioka

Alamat: Jl. Mangga II Blitar

JenisKelamin: Laki

TanggalLahir: 03-12-1995

JurusanID:
 Teknik Informatik
 Teknik Elektronika
 Teknik Telekomunikasi
 Teknik Komputer
 Teknik Mekatronika

[Back to List](#)

© 2016 - My ASP.NET Application

Setelah mengisi data dari Tabel jurusan, cobalah untuk mengisi data pada Form Mahasiswa, terlihat pada bagian paling akhir ada Form tambahan yang berisi data dari Tabel Jurusan. Berarti kedua tabel yang telah dibuat telah berhasil berelasi

Index

[Create New](#)

Deskripsi	Nama	Alamat	JenisKelamin	TanggalLahir	
Teknik Informatik	Civer Yoshioka	Jl. Mangga II Blitar	Laki	03/12/1995 0:00:00	Edit Details Delete
Teknik Telekomunikasi	Alibaba	Jl. Menur Surabaya	Laki	03/12/1995 0:00:00	Edit Details Delete

© 2016 - My ASP.NET Application

Tampilan dari Form **Mahasiswa**, terlihat di atas ada Form tambahan yaitu Deskripsi Dari Jurusan Mahasiswa yang telah di masukan pada Form Inputan.

PERCOBAAN

Sebelum mempelajari modul ini pastikan anda sudah memahami percobaan dari modul sebelumnya

TUGAS WORKSHOP

1. Lakukanlah percobaan di atas dan kembangkan korelasi dari beberapa Tabel.