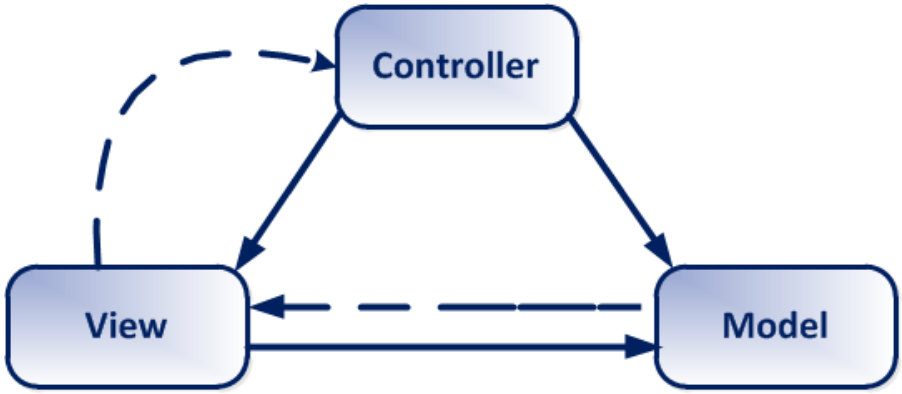
BAB VI

MVC .NET

Model–View–Controller atau dikenal dengan MVC adalah salah satu architectural pattern yang digunakan pada software engineering. Konsep MVC memisahkan domain login dari input dan presentation (UI). Model diagram MVC dan hubungannya dapat dilihat pada gambar berikut.



Model digunakan untuk manajemen informasi dan memberikan notifikasi ke observer jika ada perubahan informasi. Model biasanya dipresentasikan sebagai domain spesifik. Domain logic ditambahkan untuk monitoring perubahan status, sebagai contohnya menghitung apakah hari ini ada yang ulang tahu atau perubahan pengiriman pada shopping. Ketika model berubah statusnya maka model akan menginformasikan ke View yang berasosiasi supaya dilakukan refresh.

Sebagian besar aplikasi menggunakan database sebagai mekanisme persistent storage untuk menyimpan data. Mekanisme ini biasanya dikenal dengan data access layer sedangkan MVC secara eksplisit tidak menyebut sebagai data access layer tetapi semua fungsi dan fitur di enkapsolusi didalam Model sehingga Model tidak bukanlah data access objects. Active Record merupakan design pattern yang mengabungkan domain logic dan data access code yang mana Model dalam memanfaatkan pattern ini.

View berfungsi untuk melakukan render Model kedalam suatu bentuk elemen interakasi contohnya UI (User Interface). Sebuah Model dapat digunakan untuk beberapa View dengan tujuan yang berbeda-beda.

Controller menerima input dan memulai respon yang dibuat oleh objek Model. Sebuah Controller dapat menerima input dari user dan mengintruksikan Model untuk melakukan aksi sesuai dengan input.

MVC bisanya diimplemtasikan pada plikasi web dimana HTML atau XHTML sebagai View yang dihasilkan oleh aplikasi. Controller menerima input GET atau POST dan menentukan apa yang akan dilakukan serta menangani domain object (Model) yang berisi business rules.

Apa Bedanya ASP.NET Web Form dan ASP.NET MVC?

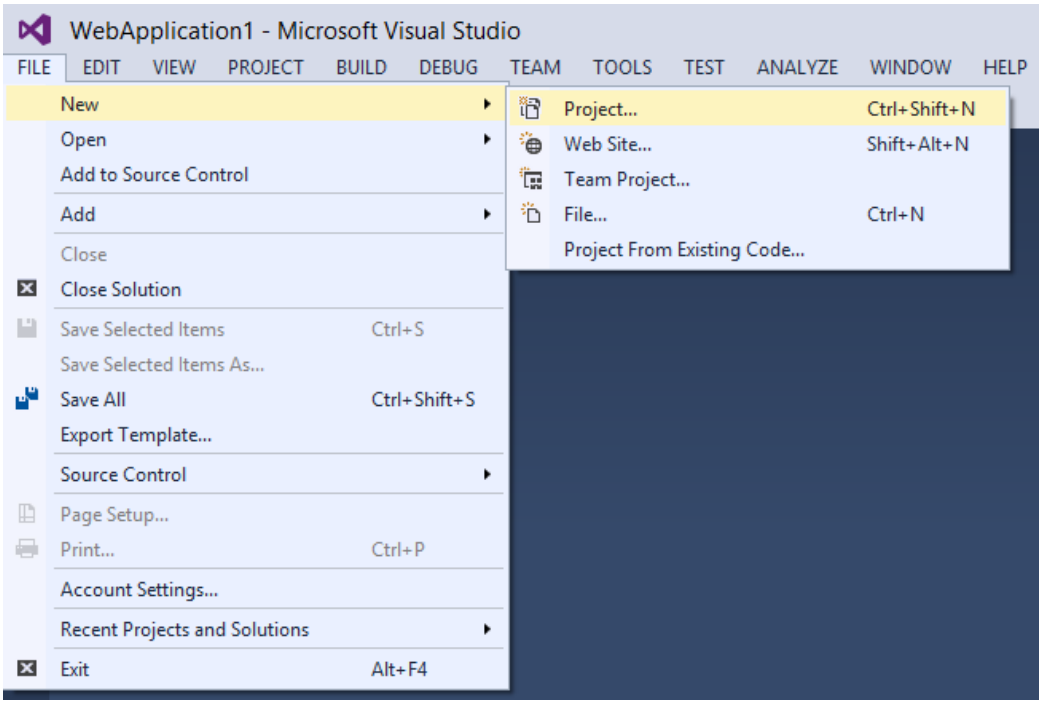
ASP.NET MVC bukanlah pembunuh Web Form ASP.NET ataupun anti Web Form ASP.NET. Dapat dikatakan ASP.NET saat ini sudah berumur 10 tahun lebih. ASP.NET MVC adalah ASP.NET baru yang dibuat untuk menyesuaikan diri atas perkembangan teknologi. Beberapa blog dan artikel banyak sekali membahas apa bedanya ASP.NET Web Form dan ASP.NET MVC. Berikut ini beberapa kondisi fakta antara ASP.NET Web Form dan ASP.NET MVC

1. Web Form sulit untuk di testing
2. ASP.NET MVC memerlukan keahlian scripting HTML
3. ASP.NET MVC bukan salah satu untuk mendapatkan SoC pada ASP.NET
4. Web Form dapat dipelajari dengan cepat
5. ViewState bukannya bermasalah dan ini dapat dikontrol ataupun dimatikan
6. Web Form didesain untuk abstraksi mesin web
7. ASP.NET MVC didesain untuk mudah ditesting
8. ASP.NET MVC dapat menawarkan desain code yang lebih baik
9. Perkembangan ASP.NET MVC masih mudah dan kurang component model
10. ASP.NET MVC bukan anti Web Form

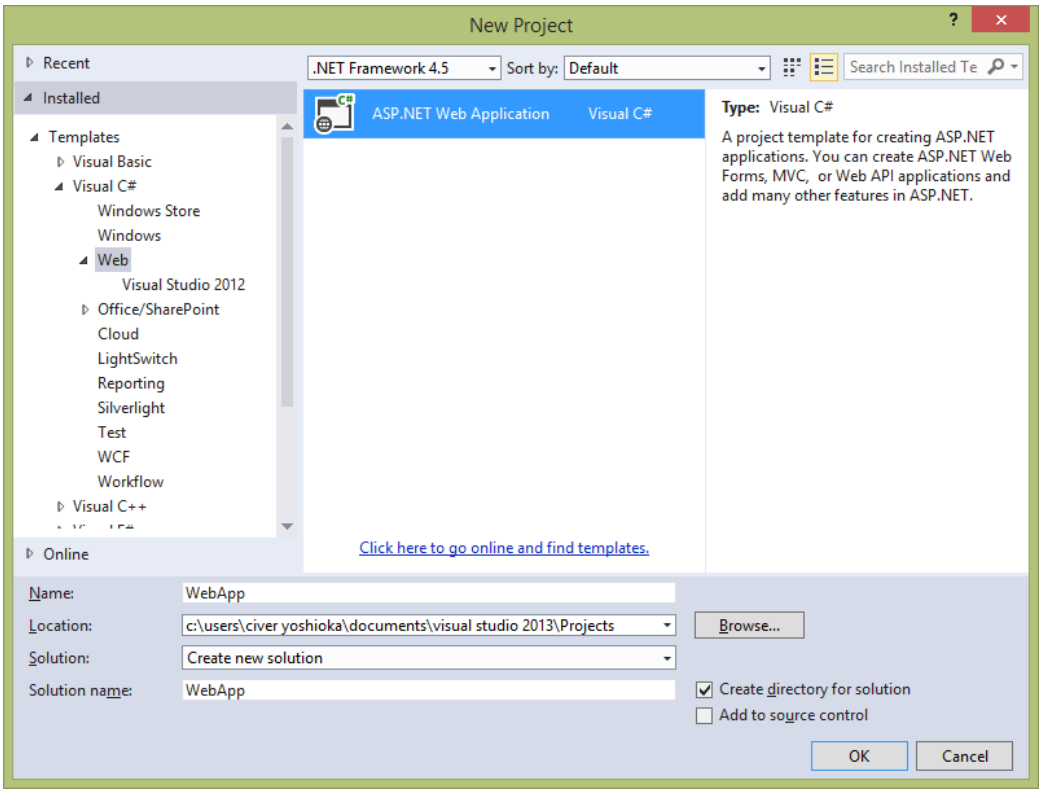
Pada saat modul ini ditulis versi MVC terbaru yang dikeluarkan oleh Microsoft adalah MVC 5. Untuk MVC 5 secara default sudah tersedia pada Visual Studio 2013. Untuk Visual Studo versi sebelumnya membutuhkan beberapa tambahan plugin pada visual studio untuk bisa menjalankan MVC 5. Untuk untuk memperlancar mengikuti modul sebaiknya menggunakan versi Visual Studio 2013 atau yang versi lebih baru.

Untuk membuat projek ASP menggunakan MVC 5 Framework adalah sebagai berikut.

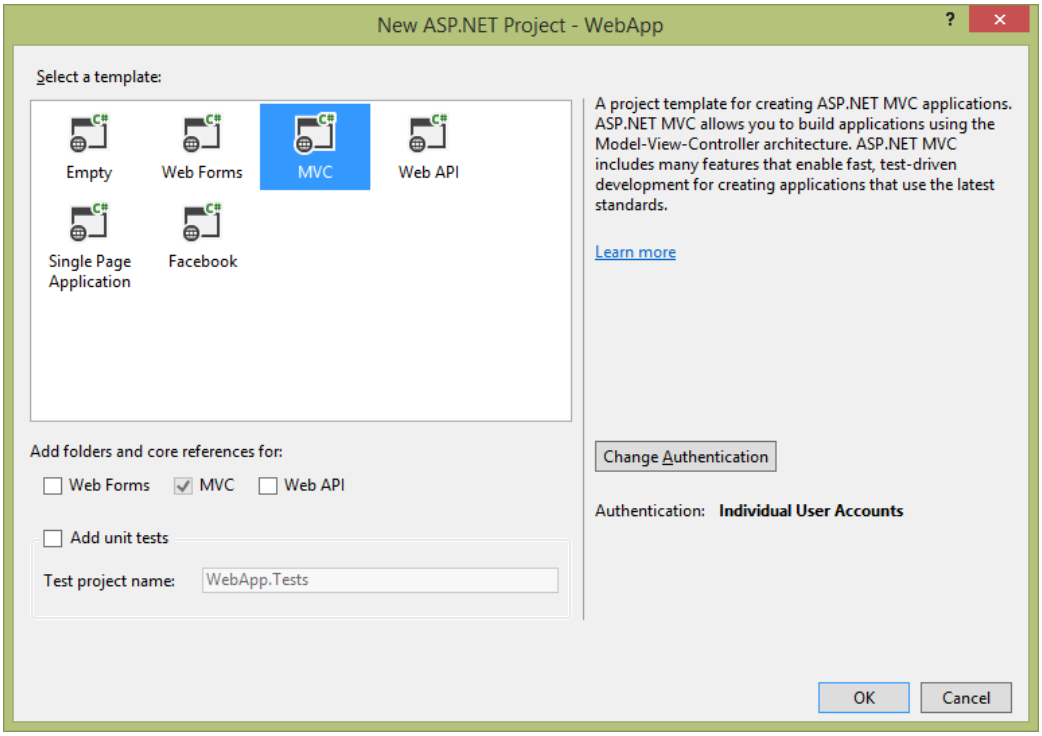
FILE > New > Project



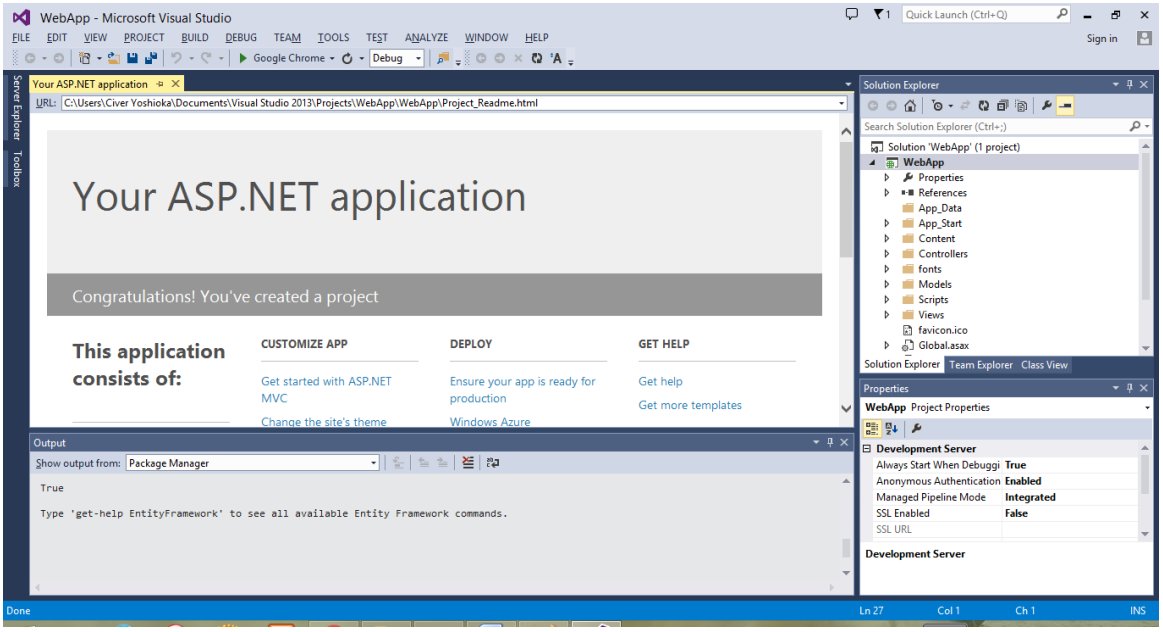
Kemudian pada **Template** silahkan pilih ***Visual C# > Web > ASP .NET Web Application***, kemudian ***OK***.



Setelah Klik OK akan muncul Form pilihan template yang akan digunakan, silahkan pilih MVC. Perhatikan gambar berikut.



Maka akan secara otomatis akan terbentuk template seperti berikut.



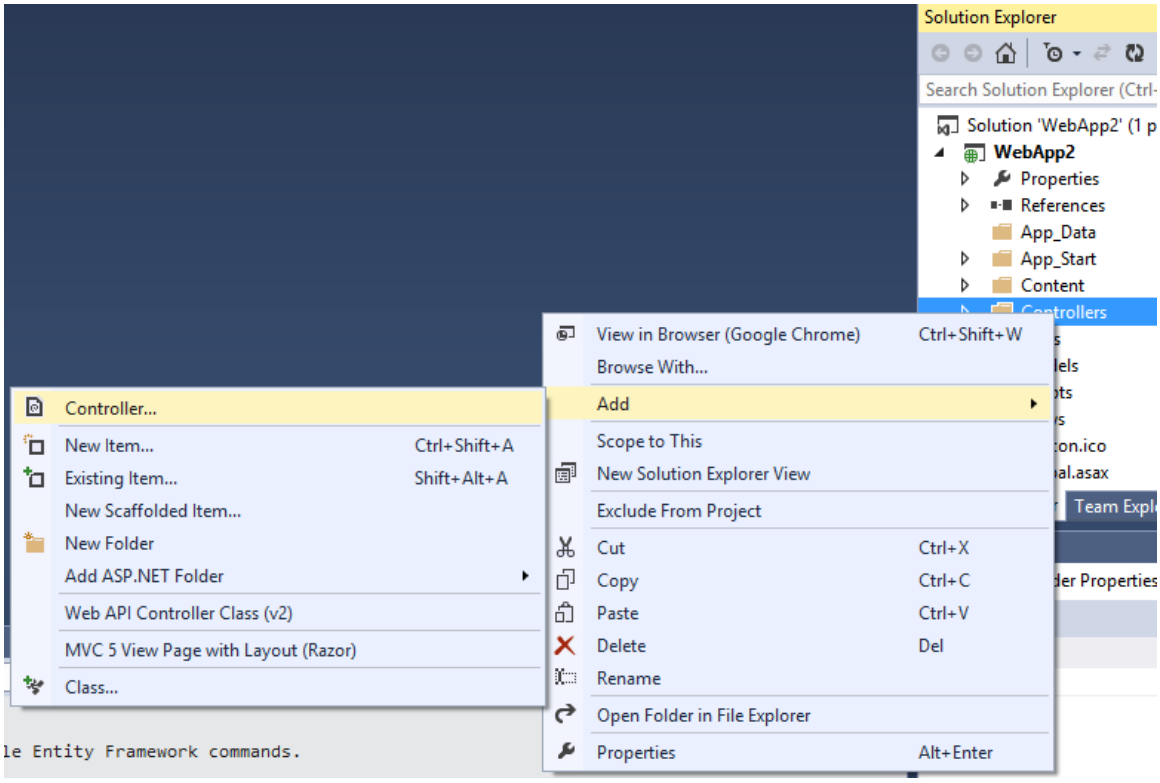
Perhatikan bagian kanan IDE. Perhatikan pada bagian Solution Explorer. Secara default juga telah terbentuk application structure dari MVC 5. Yang terpenting untuk dipahami pada tahap awal adalah folder *model*, *view*, dan *controller*, dimana ketiga folder tersebut adalah kunci untuk konsep MVC pada pembahasan ini.

❖ Controller

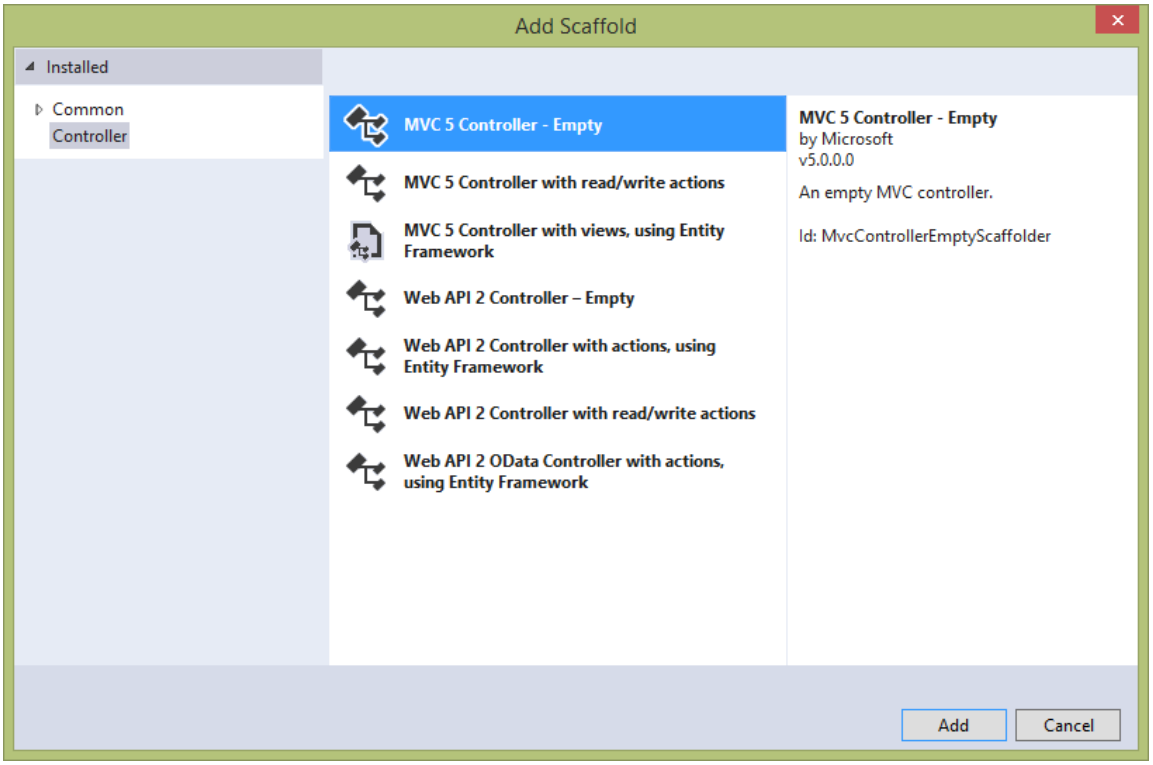
Controller bertanggung jawab untuk menangani request(permintaan) dari Browser, menerima data dari *Model*, mengatur dan memetakan data ke *View* sebagai response ke Browser untuk di tampilkan kepada Pengguna.

Untuk membuat *Controller* bisa dengan menggunakan cara sebagai berikut.

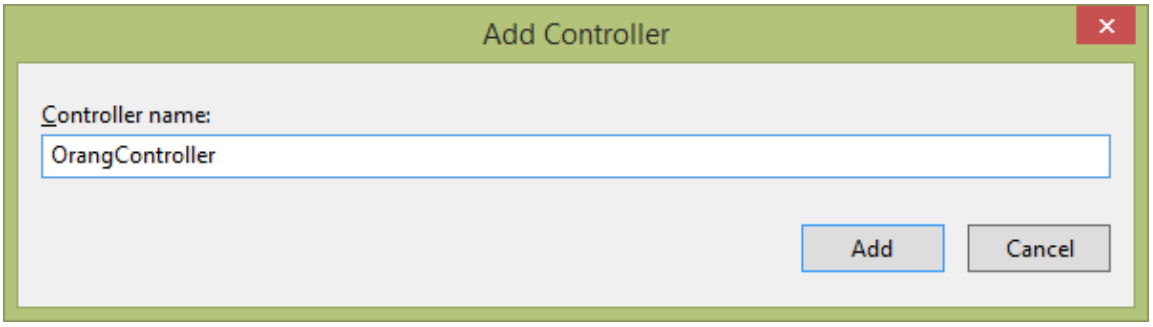
1. Klik kanan pada folder ***Controller > Add > Controller***



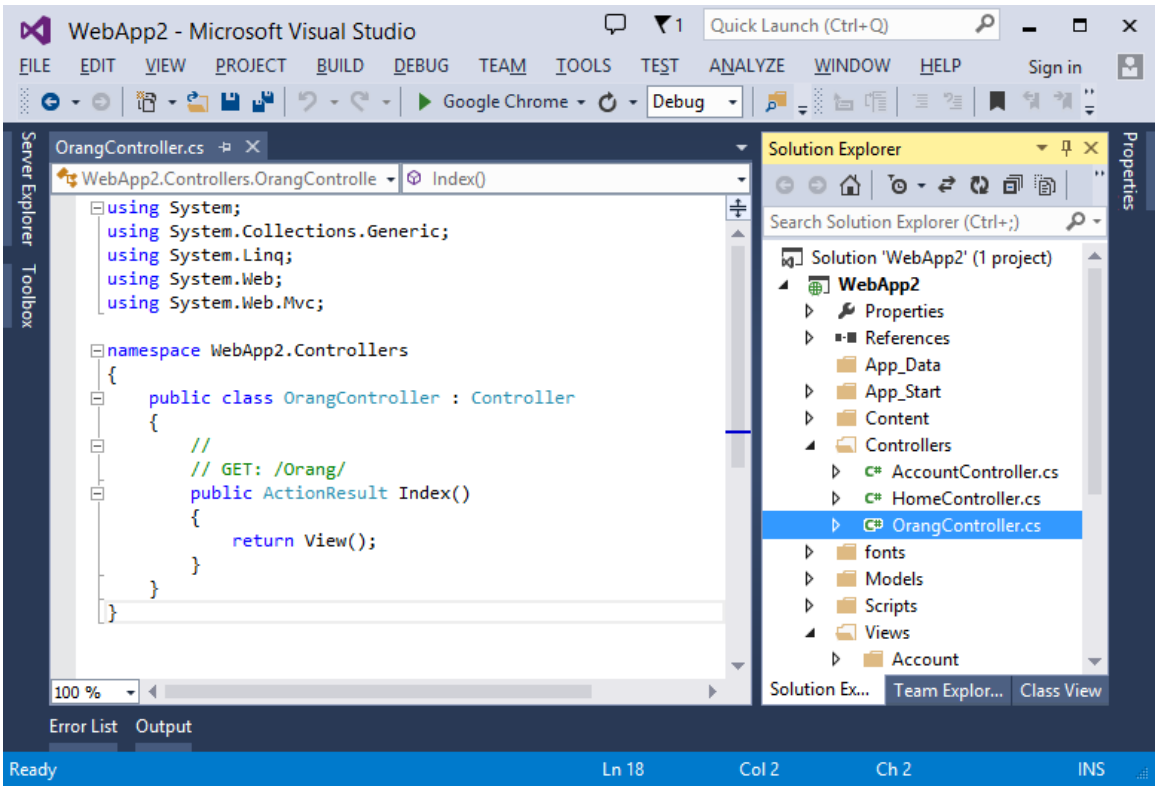
2. Kemudian akan muncul Window **Add Scaffold** sebagai berikut, pilih pada MVC 5 ***Controller – Empty > Add***



3. Berikan nama pada *Controller* yang dibuat. Perlu diingat untuk pemberian nama *Controller* awali dengan menggunakan huruf kapital dan suffix "***Controller***" harus selalu diikutsertakan, misalkan seperti pada contoh berikut



4. Contoh default dari sebuah *Controller*



Controller terdiri dari fungsi-fungsi*(untuk selanjutnya akan disebut action)* yang nantinya setiap fungsi menjadi jembatan antara response dari browser dan model untuk menangani data. Secara default setiap pembuatan *Controller* akan dibuatkan default action yaitu `Index()`. Untuk mengakses dari action-action yang ada didalam *Controller* adalah dengan menggunakan URL pada browser dengan default format sebagai berikut.

```
[nama Host]/[Controller]/[Action]/[Parameter]
```

Bisa dilihat pada `App_Start/RouteConfig.cs`, berikut isi dan configurasi Routing secara default

```
namespace WebApp2
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
```

```
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id =
UrlParameter.Optional }
            );
        }
    }
}
```

Default Controller

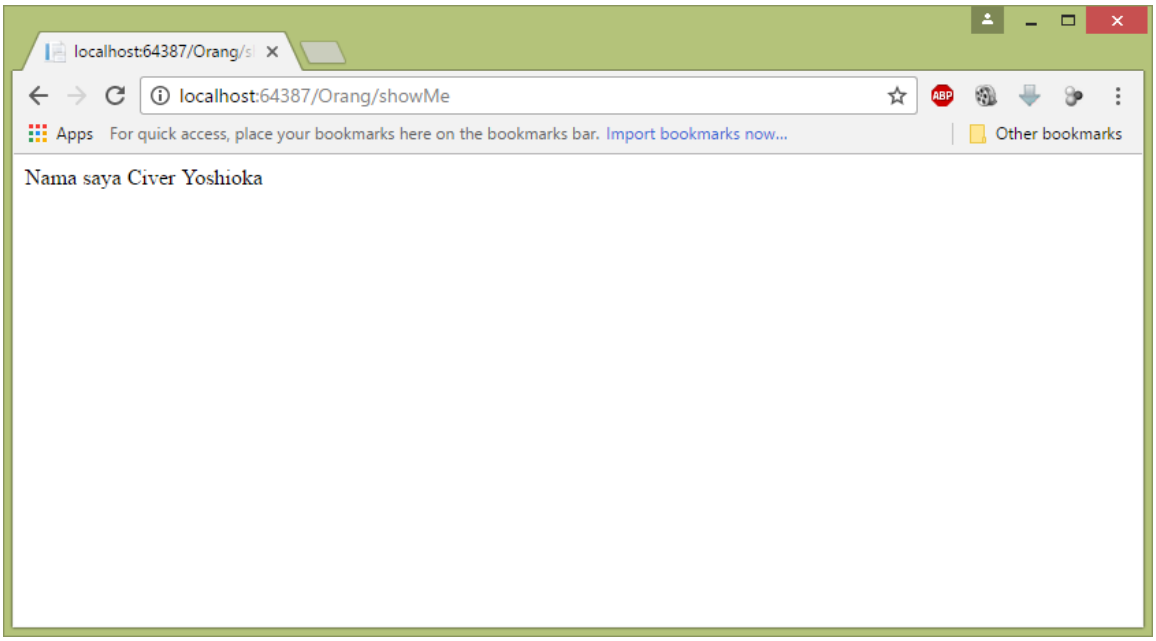
Untuk lebih memahami cobalah anda rubah code dari isi **OrangController** menjadi seperti berikut.

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

namespace WebApp2.Controllers
{
    public class OrangController : Controller
    {
        //
        // GET: /Orang/
        public ActionResult Index()
        {
            return View();
        }

        public string showMe() {
            return "Nama saya Civer Yoshioka";
        }
    }
}
```

Kemudian silahkan buka pada browser dengan URL `http:// namaHost/Orang/showMe`



Passing nilai Melalui Controller

Pada contoh di atas adalah sebuah contoh bagaimana hanya menampilkan sebuah *Controller*. Untuk bagaimana jika ada paramter yang harus di terima uleh action pada sebuah *Controller*. Ubahlah kode di atas menjadi sebagai berikut :

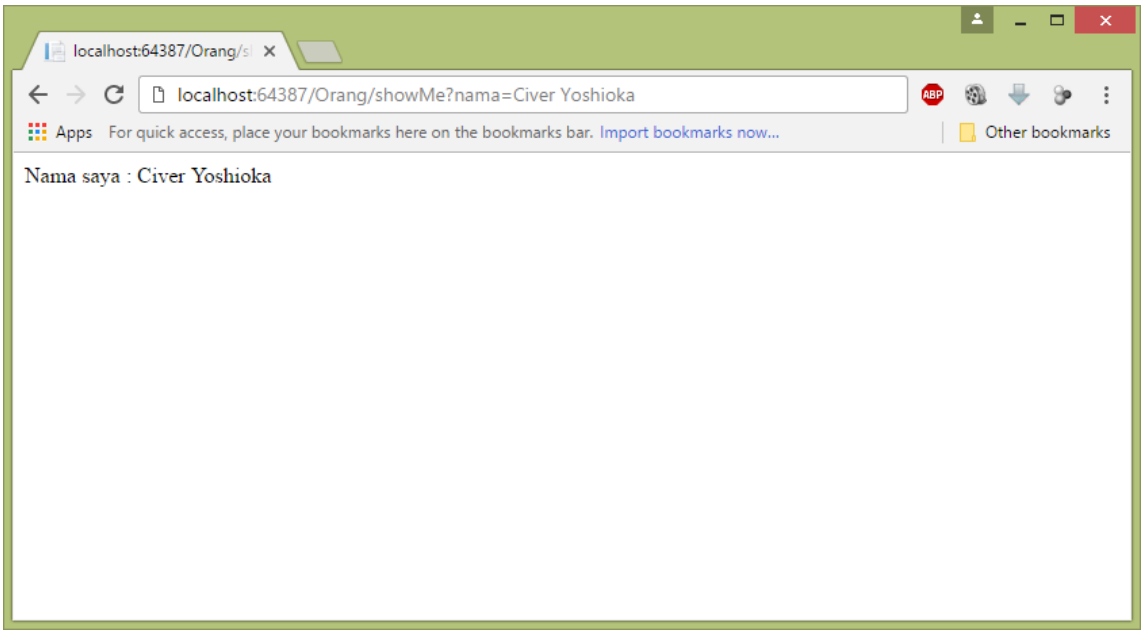
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace WebApp2.Controllers
{
    public class OrangController : Controller
    {
        //
        // GET: /Orang/
        public ActionResult Index()
        {
            return View();
        }

        public string showMe(string nama) {
            return "Nama saya : " + nama;
        }
    }
}
```

Kemudian silahkan buka pada browser dengan URL

`http:// namaHost/Orang/showMe?nama=Civer Yoshioka`



Jika berhasil akan tampak seperti pada gambar di atas.

Ingat bahwa secara default urutan default MVC mapping adalah :

`[nama Host]/[Controller]/[Action]/[Parameter]`

Sedangkan pada contoh sebelumnya belum menggunakan parameter. Untuk menggunakan parameter cobalah dengan mengubah code sebagai berikut :

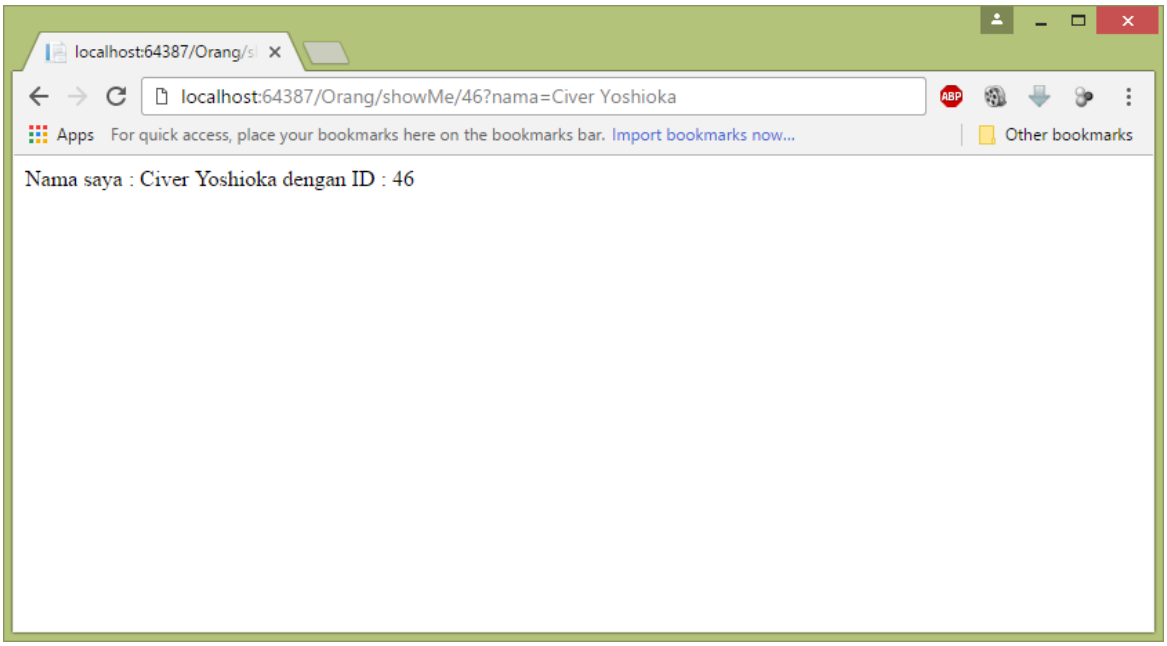
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace WebApp2.Controllers
{
    public class OrangController : Controller
    {
        //
        // GET: /Orang/
        public ActionResult Index()
        {
            return View();
        }
```

```
        public string showMe(string nama, int id) {
            return "Nama saya : " + nama + " dengan ID : " + id;
        }
    }
}
```

Kemudian silahkan buka pada browser dengan URL

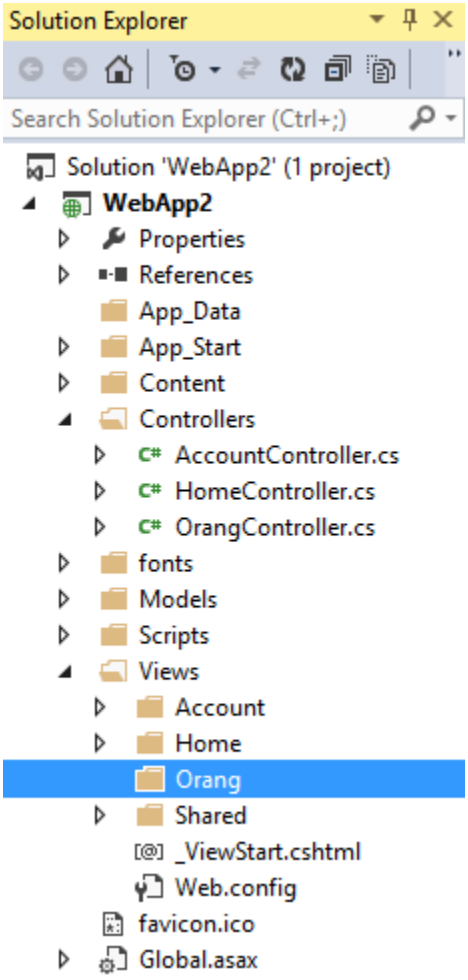
`http:// namaHost/Orang/showMe/46?nama=Civer Yoshioka`



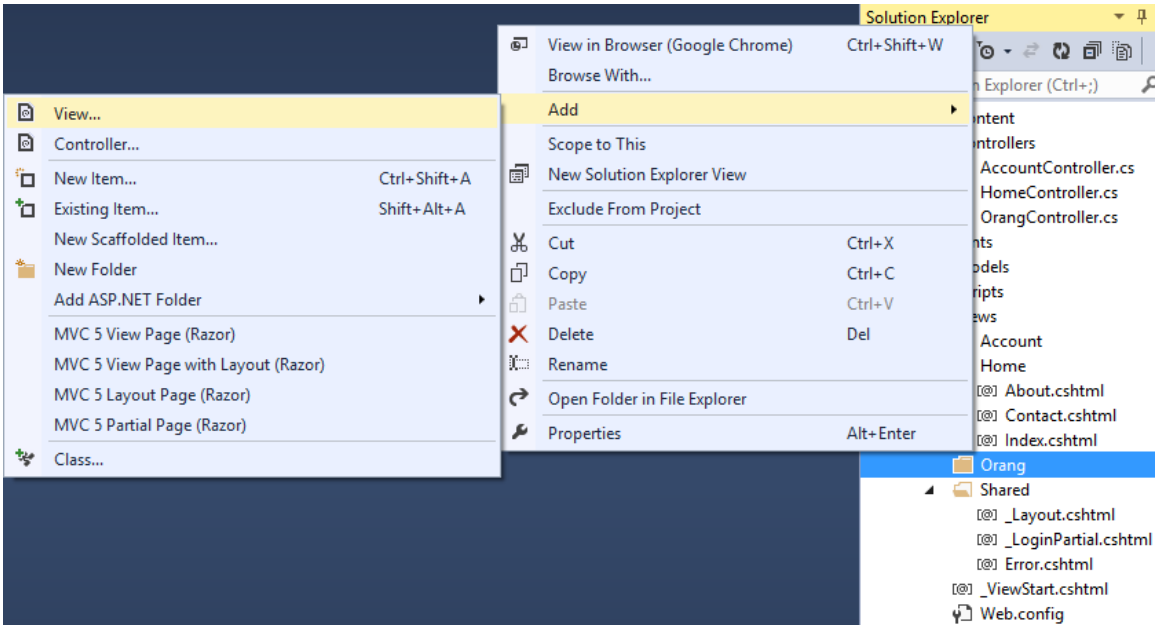
❖ **View**

Adalah template HTML untuk menampilkan hasil response ke pengguna. Pada contoh sebelumnya dengan menggunakan *Controller* bisa menampilkan nilai ke browser tetapi tentunya dengan tampilan yang sangat sederhana. Untuk memperkaya tampilan dan mempermudah untuk membuat tampilan maka pada model MVC harus terpisah. Untuk membuat view pada MVC 5 bisa dengan cara sebagai berikut.

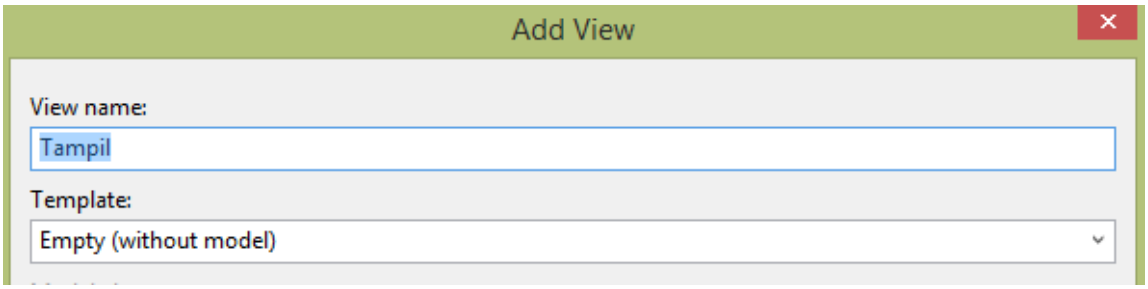
1. Membuat sebuah folder pada direktori Views dengan nama seperti pada nama *Controller*, biasanya ketika ketika membuat *Controller* secara otomatis folder yang namanya sama dengan *Controller* di generate oleh Visual Studio. Jika belum di *Generate* secara otomatis bisa *Generate* secara manual. Misalkan nama *Controller* yang telah dibuat sebelumnya adalah Orang. Maka direktori pada *Views* akan tampak seperti berikut.



2. Untuk membuat *View*, klik kanan pada Direktori ***Orang > Add > View***. Perhatikan Gambar berikut.



3. Berikan nama view yang sesuai, misalkan seperti berikut.



4. Silahkan cek pada Direktori **Orang**, jika berhasil akan ada tambahan sebuah file view yang telah Anda buat. Isi file view secara default adalah sepeti berikut.

```
@{
    ViewBag.Title = "Tampil";
}

<h2>Index</h2>
```

5. Sekarang coba ubah kode pada *Controller* dan disesuaikan dengan view yang telah dibuat supaya bisa saling berkomunikasi bertukar data. Ubahlah kode menjadi seperti berikut.

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

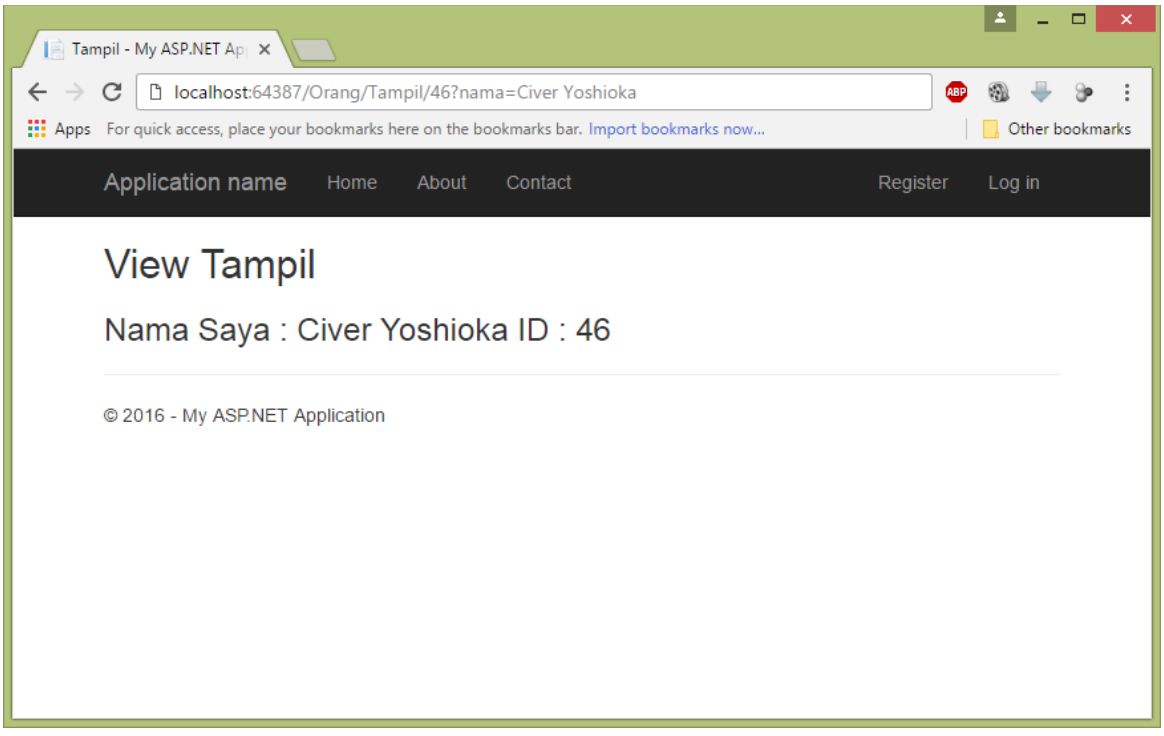
```
using System.Web;
using System.Web.Mvc;

namespace WebApp2.Controllers
{
    public class OrangController : Controller
    {
        //
        // GET: /Orang/
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Tampil(string nama, int id)
        {
            ViewBag.Pesan = "Nama Saya : " + nama + " ID : " + id;
            return View();
        }
    }
}
```

6. Kemudian silahkan buka pada browser dengan URL

   `http:// namaHost/Orang/Tampil/46?nama=Civer Yoshioka`

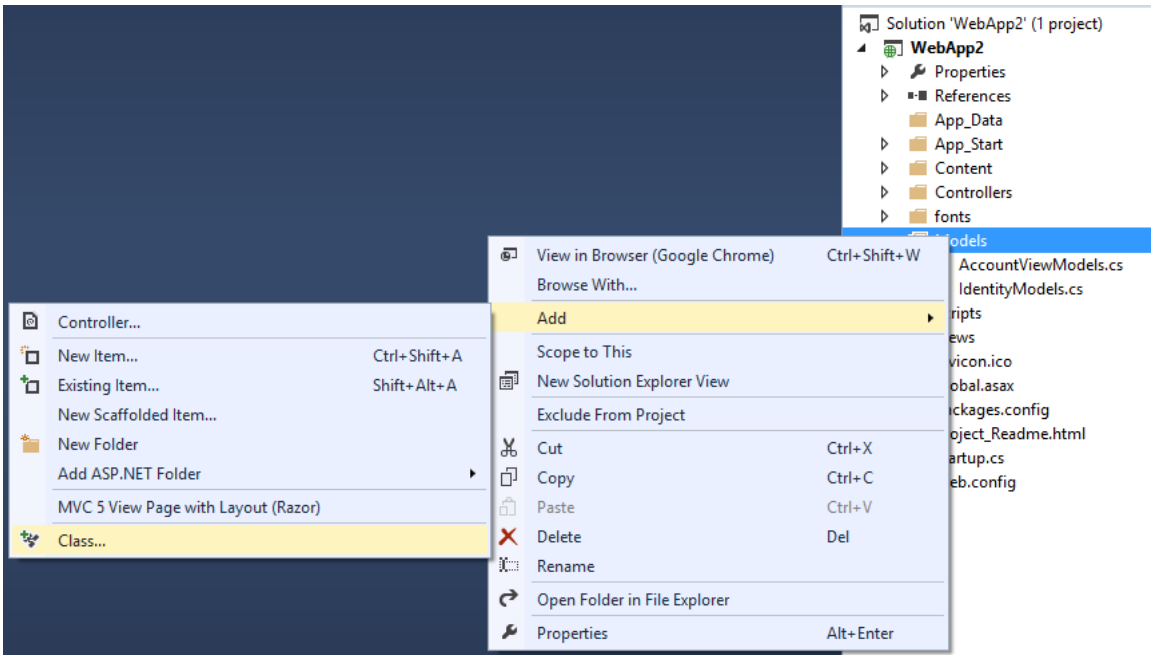


Jika anda perhatikan maka dengan menggunakan *View* untuk menampilkan respon secara otomatis template default dari MVC 5 akan terintegrasi. Sehingga tampilanya adalah seperti pada contoh di atas.

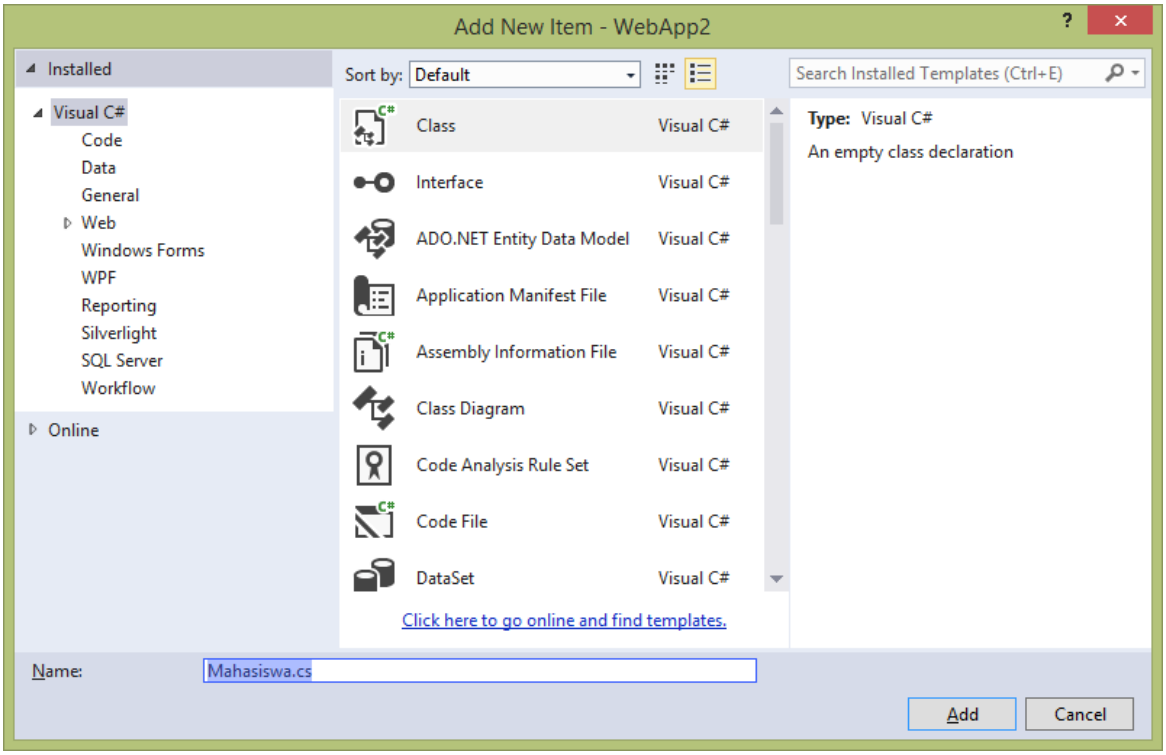
❖ Model

Model dalam MVC model adalah merepresentasikan kelas untuk data. Semua yang berhubungan dengan data akan diolah oleh *Model*. Termasuk data untuk mengambil kedalam database, memasukan kedalam database dan urusan bussines logic dari data semua akan di handle oleh *Model*. Dengan membuat Model, pada Visual Studio sudah difasilitasi untuk membuat Controller dan *View* secara otomatis untuk proses CRUD. Silahkan ikuti langkah-langkah berikut untuk implementasi Model dari *Controller* dan *View*

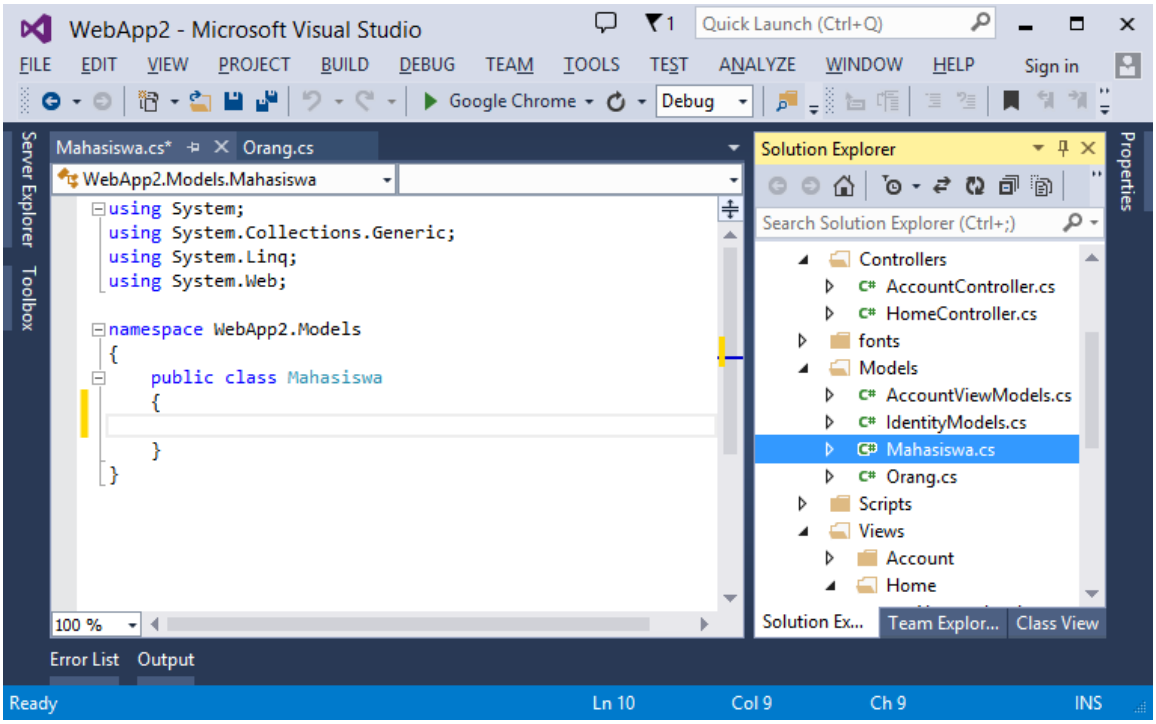
1. Buat kelas *Model* dengan klik kanan pada direktori ***Model > Add > Class***. Perhatikan gambar berikut



2. Pada Window **Add Item**, Pilih ***Visual C# > Class***. Kemudian. Berikan nama kelas yang sesuai seperti berikut.



3. Maka akan terbentuk kelas *Model* default seperti berikut.



4. Buatlah Entitas kelas *Model* untuk **Mahasiswa** dengan mengubah kode *Model* Kelas **Mahasiswa** menjadi seperti berikut.

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

namespace WebApp2.Models
{
    public class Mahasiswa
    {
        public int ID { get; set; }
        public string Nama { get; set; }
        public string Alamat { get; set; }
        public string jenisKelamin { get; set; }
        public int Umur { get; set; }
        public string Jurusan { get; set; }

    }
}
```

5. Pada step di atas adalah membuat Entitas kelas *Model* **Mahasiswa***(Object Mapping dari Mahasiswa)*, yang mana nanti dari setiap entitas akan merefleksikan field data dari Database. Sekarang silahkan ubah kelas di atas menjadi seperti berikut.

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

namespace WebApp2.Models
{
    public class Mahasiswa
    {
        public int ID { get; set; }
        public string Nama { get; set; }
        public string Alamat { get; set; }
        public string jenisKelamin { get; set; }
        public int Umur { get; set; }
        public string Jurusan { get; set; }

    }
    public class MahasiswaDBContext : DbContext
    {
        public DbSet<Mahasiswa> Mhs { get; set; }
    }
}
```

Dengan menambahkan kelas **`MahasiswaDBContext`** yang menjadi turanan dari kelas **`DbContext`** maka secara otomatis kelas tersebut akan menjadi Object Mapping dari kelas Mahasiswa kedalam Entitas data pada Database.

❖ **Koneksi ke SQL Server LocalDB**

Pada sub bab sebelumnya telah dijelaskan bahwa **`MahasiswaDBContext`** adalah Object Mapping dari Objek **Mahasiswa**. Koneksi ke database dan Object Mapping data dari Kelas **Mahasiswa** ke Record Database akan di Handle oleh Kelas tersebut. Membuat Connection String ke dalam Database Local SQL Server. Buka pada file Web.config kemudian carilah Tag `<connectionStrings>` kemudian tambahkan contoh Connection String seperti berikut.

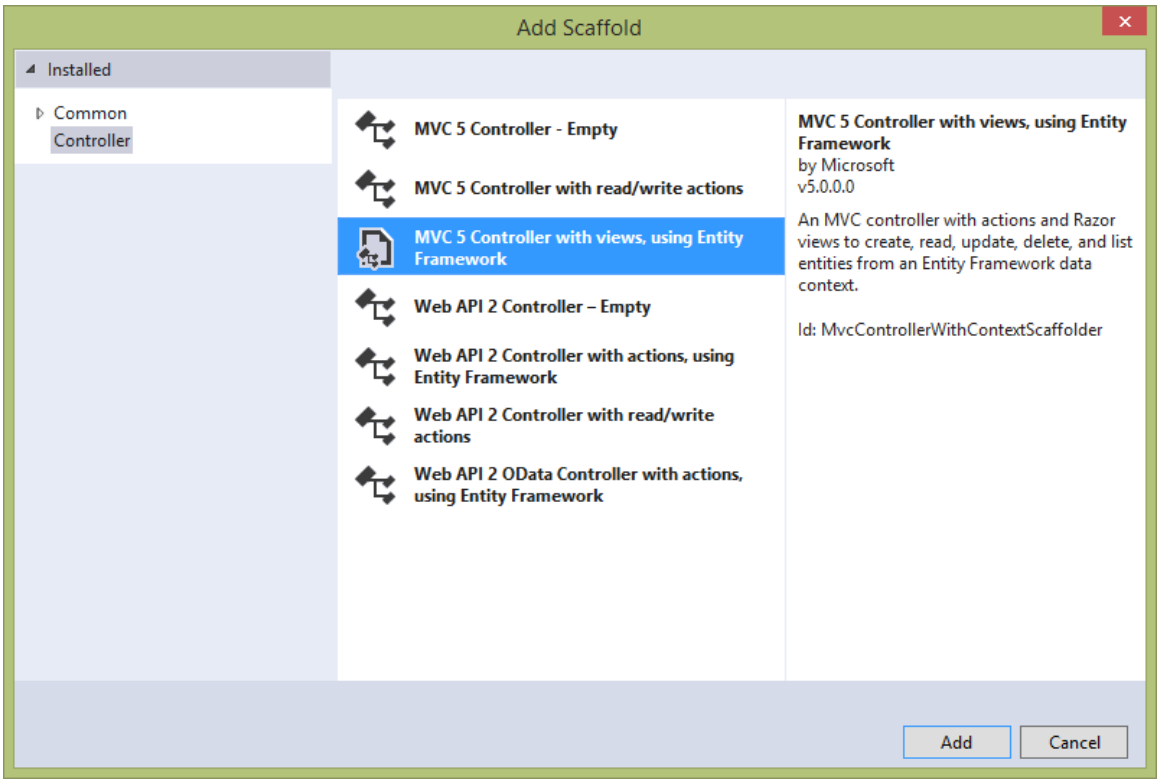
```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\aspnet-WebApp2-2
    providerName="System.Data.SqlClient" />
  <add name="OrangConnection"
       connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\Mahasiswa.mdf;Integrated Security=True"
       providerName="System.Data.SqlClient"
   />
</connectionStrings>
```

Secara default sudah ada Connection String yang telah generate otomatis yaitu **`DefaultConnection`**. Cukup anda tambahkan saja *Connection* String baru seperti contoh di atas.

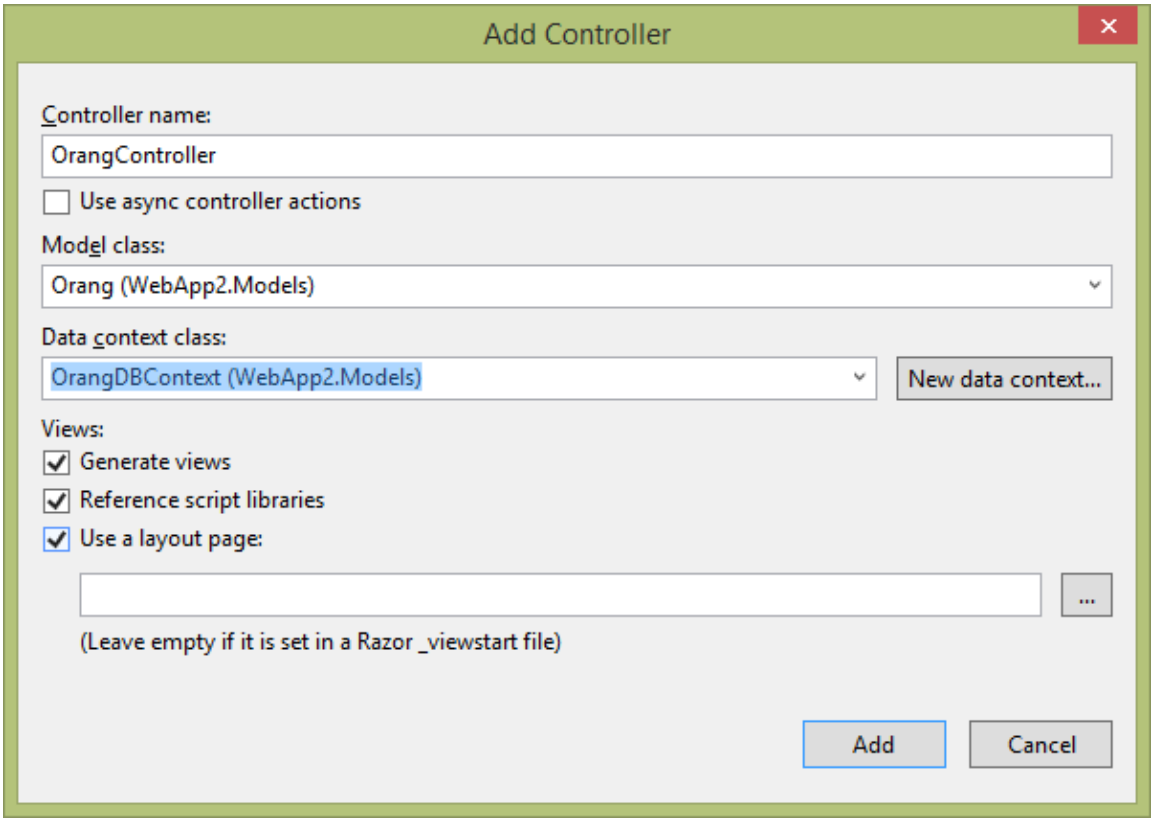
Accessing Data Object Mapping

Pada tahap ini akan dicoba untuk melakukan generate Database dan CRUD aplikasi secara otomatis dari *Model* yang telah dibuat yaitu *Model* **Mahasiswa**. Model Mahasiswa telah dibuat terlebih dahulu sebelumnya. Sekarang adalah bagaimana melakukan generate otomatis *Controller* dan Viewnya

1. Buatlah *Controller* seperti cara biasanya Klik kanan pada Folder ***Controller > Add > Controller***.
2. Pada **Add Scaffold** pilih **MVC 5 Controller with views, using Entity Framework** seperti berikut

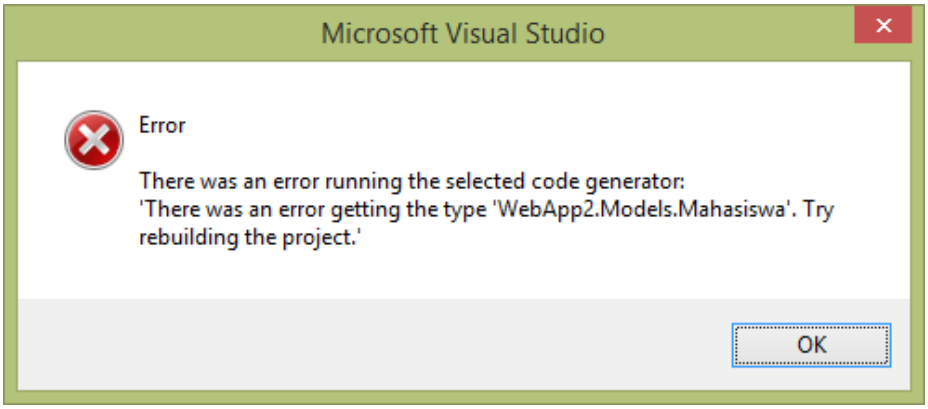


3. Kemudian Klik Add dan akan muncul window baru ,kemudian settinglah *Controller* anda menjadi seperti berikut.



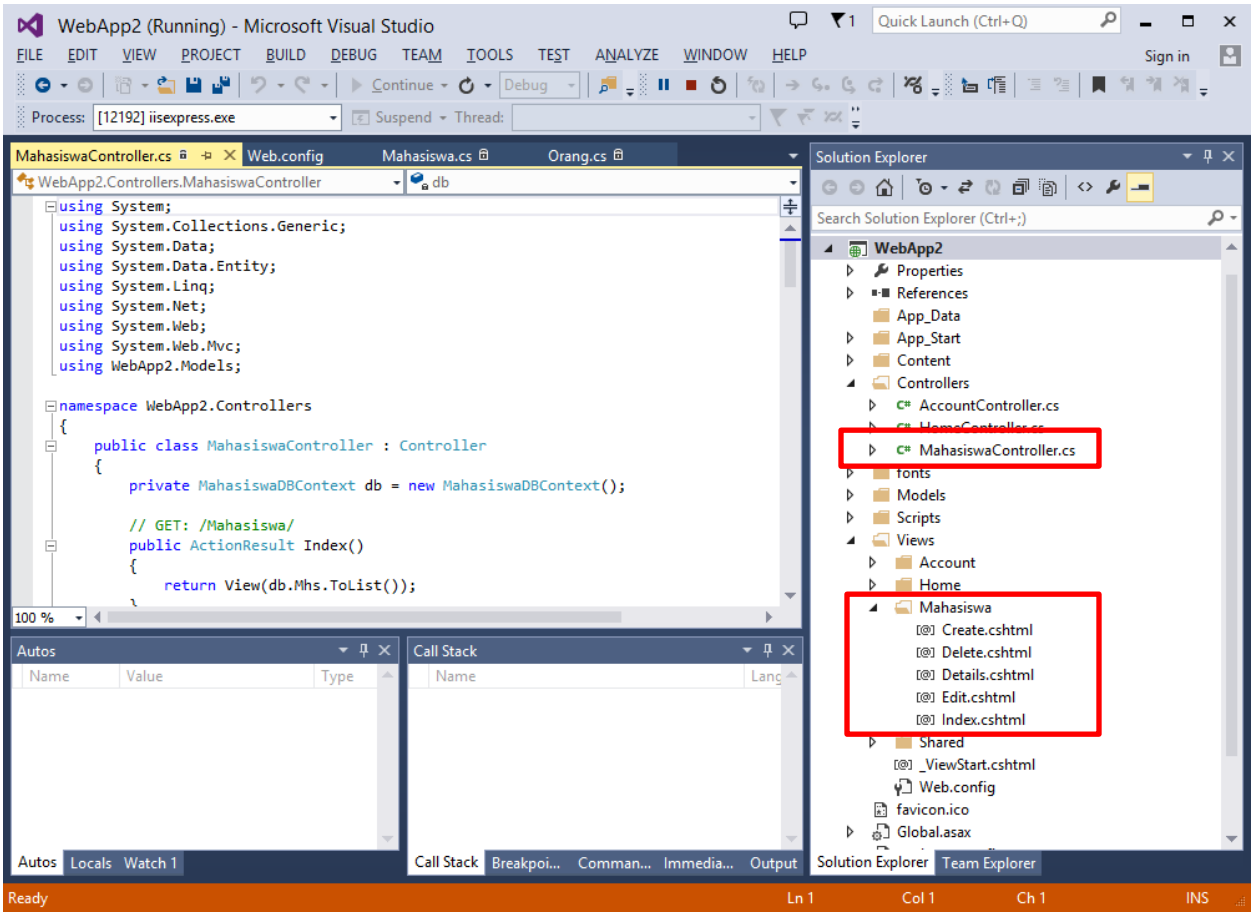
Model Class adalah Model Mahasiswa yang telah dibuat dan Data Context Class adalah Data Context Class adalah kelas Data Context yang dibuat untuk Object Model Mahasiswa. Kemudian Klik Add.

Jika terjadi error seperti Berikut



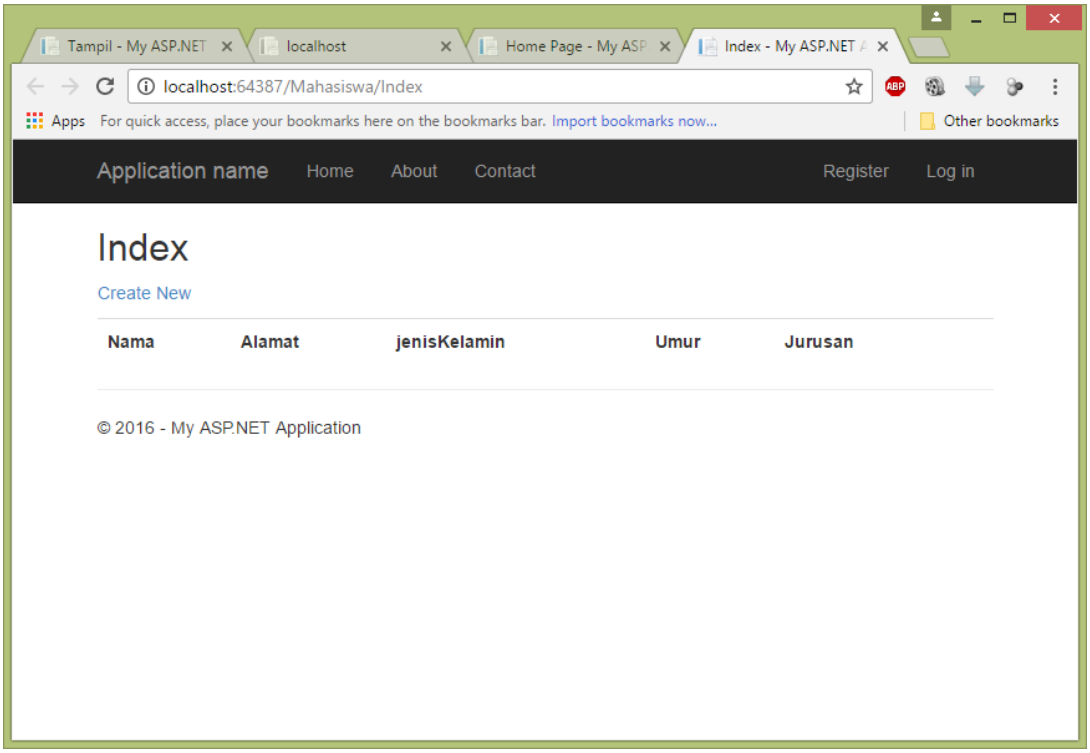
Berarti Binnari dari kelas Mahasiswa belum tersedia karena belum anda lakukan Run ke projek anda. Masuk ke Menu DEBUG > Start Debuging(F5), kemudian ulang langkah sebelumnya maka error tidak akan terjadi lagi.

4. Jika dari langkah sebelumnya anda berhasil maka akan terbentuk beberapa file yang telah di generate secara otomatis seperti berikut.

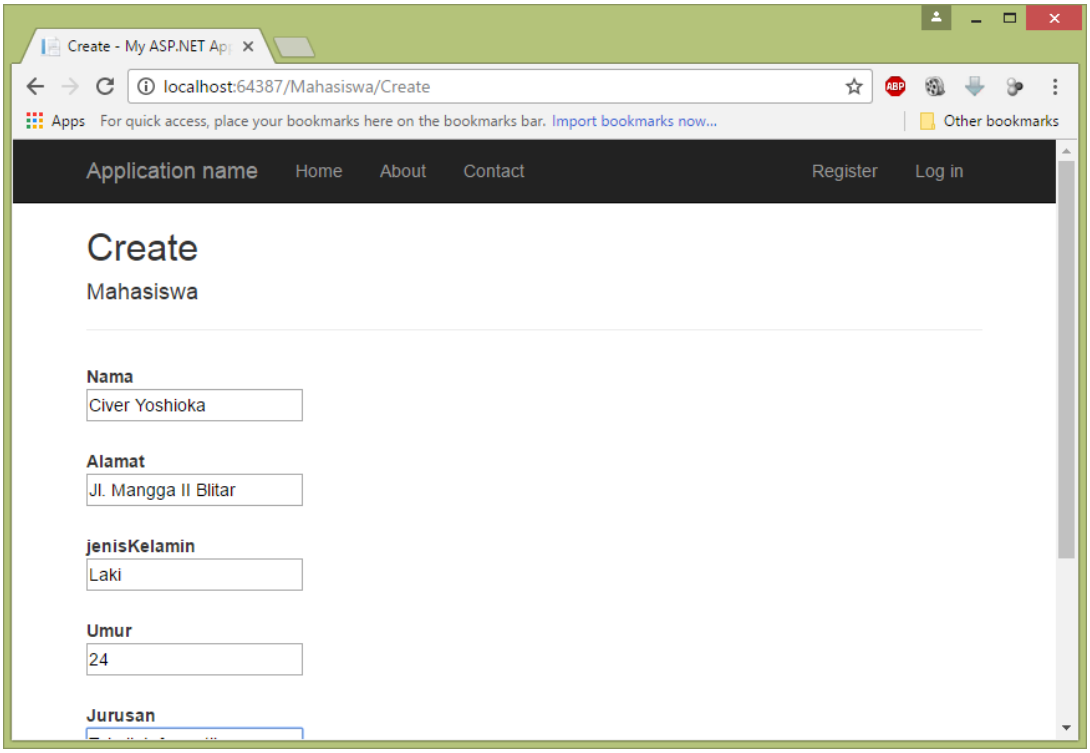


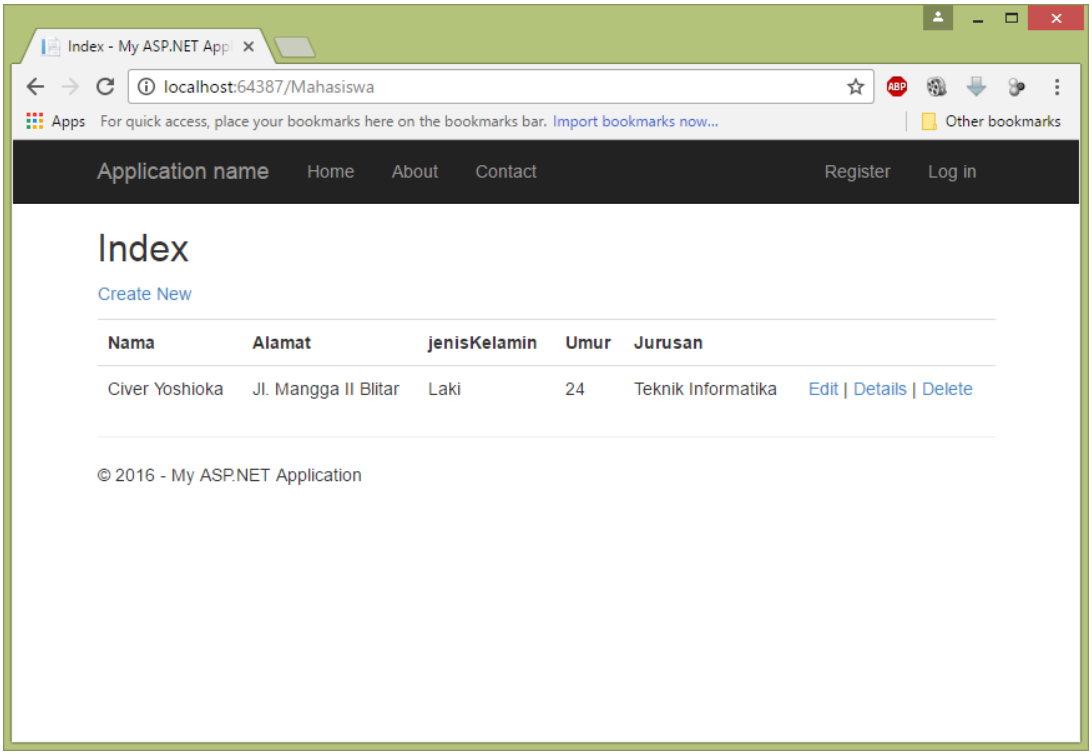
Perhatikan dari kotak yang berwarna merah. Dari Model yang telah dibuat sebelumnya maka dengan membuat Controller dari Model Mahasiswa View untuk CRUD proses juga secara otomatis tebentuk.

5. Cobalah menjalan Program dengan mengetik pada URL Browser [nama Host]/Mahasiswa maka akan muncul tampilan seperti berikut

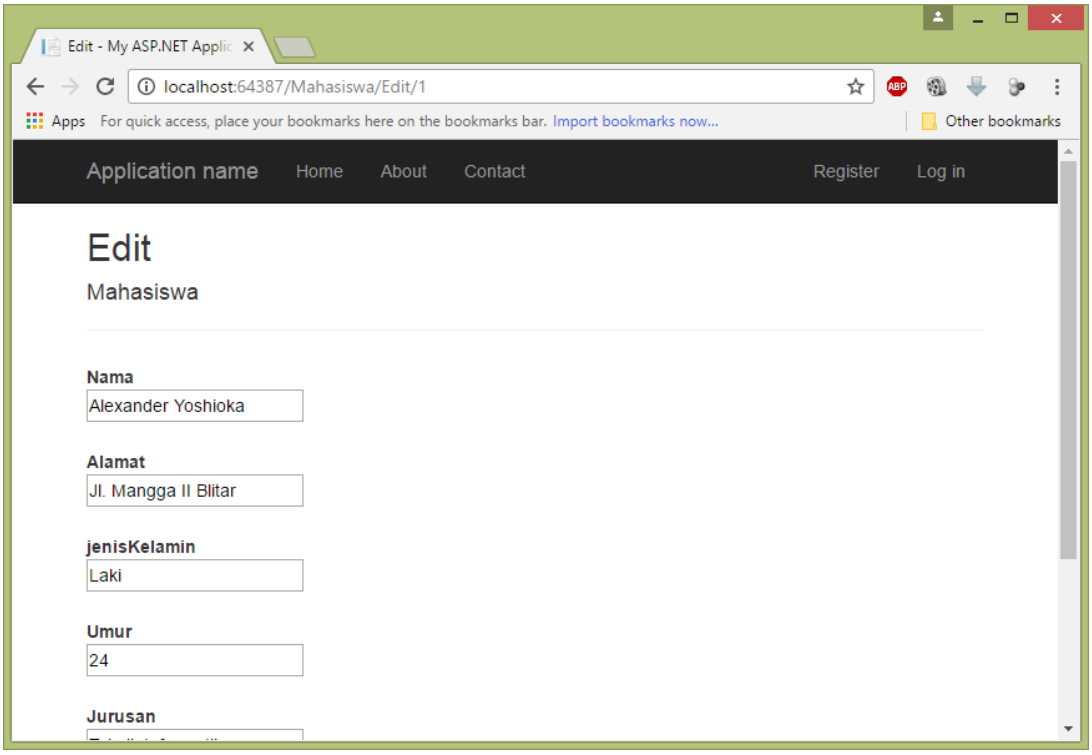


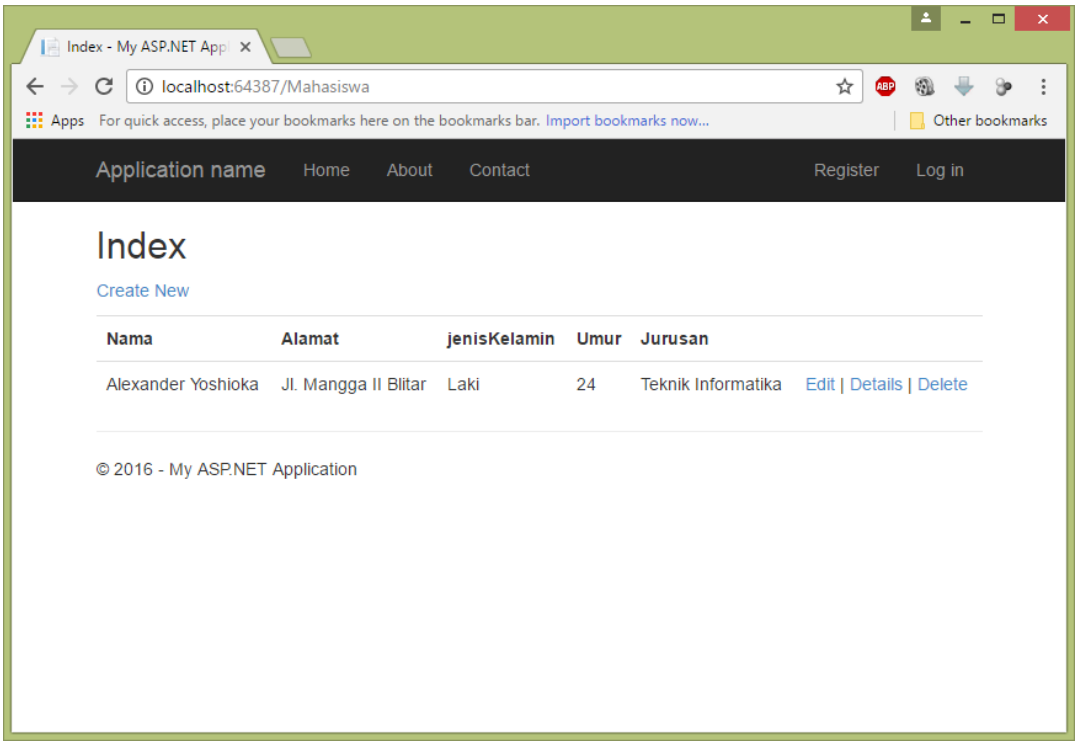
Silahkan Coba Create New



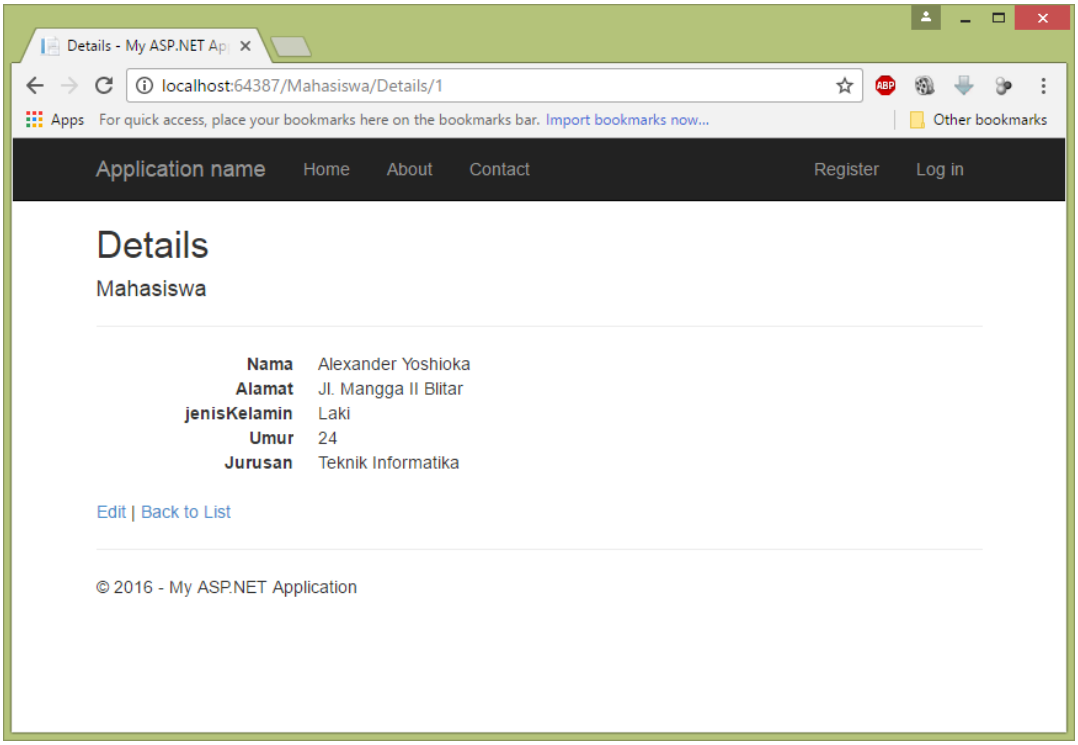

Index - My ASP.NET App
localhost:64387/Mahasiswa
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
Application name Home About Contact Register Log in
Index
Create New
Nama Alamat jenisKelamin Umur Jurusan
Civer Yoshioka Jl. Mangga II Blitar Laki 24 Teknik Informatika Edit | Details | Delete
© 2016 - My ASP.NET Application

Coba Edit


Edit - My ASP.NET Applic
localhost:64387/Mahasiswa/Edit/1
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
Application name Home About Contact Register Log in
Edit
Mahasiswa
Nama
Alexander Yoshioka
Alamat
Jl. Mangga II Blitar
jenisKelamin
Laki
Umur
24
Jurusan



Coba Details



PERCOBAAN

Pada percobaan ini akan memastikan bahwa instalasi dari MVC 5 telah berjalan dengan benar pada komputer anda. Kemudian akan mencoba untuk memahami dasar penggunaan dari Model, View, dan Controller.

TUGAS WORKSHOP

1. Lakukanlah perocobaan di atas