

Bab 7

Halaman ASP.NET Dinamis

7.1 Halaman ASPX secara Dinamis

Halaman-halaman web, terbagi menjadi dua bagian yaitu halaman statis dan halaman dinamis. Gambaran halaman statis berupa sebuah halaman dengan kode HTML, seperti dilukiskan kode HTML dibawah ini

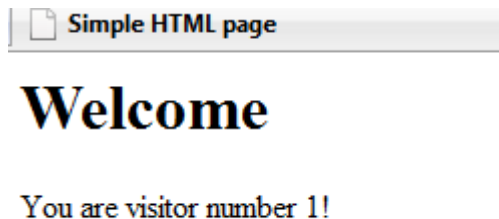
```
<html> <head><title>Simple HTML page</title></head>
<body>
<h1>Welcome</h1>You are visitor number 1!
</body>
</html>
```

File HTML tsb dapat dibuka secara langsung dengan browser, ketika file tsb diletakkan di web server, perilaku file tsb dapat digambar (gb 7.1). Pertama kali browser meminta ke web server file My.html, permintaan tsb diproses oleh web server (IIS) dengan membaca file My.html yang berada di web server, kemudian isi file tsb di kirimkan ke browser. Akhirnya browser

menerjemahkan isi file HTML tsb sehingga hasilnya terlihat pada gambar 7.2



Gambar 7.1 Halaman HTML



Gambar 7.2 Hasil Browser dari file My.html

File My.html menghasilkan angka 1, ketika file tsb diakses oleh banyak browser tanpa menghasilkan sebuah hitungan yang benar dari jumlah pengunjung. Hal ini dapat diselesaikan dengan halaman ASPX secara dinamis, dengan memasukkan hitungan jumlah pengunjung ke dalam file *counter.dat* yang akan di *update* ketika halaman tsb di akses.

```
<%@ Page Language="C#" %><%@ Import Namespace="System.IO" %>
<html><head> <title>Page counter</title> </head>
<body><h1>Welcome</h1>
```

You are visitor number

```
<%  
FileStream s = new FileStream("c:\\Data\\Counter.dat",  
    FileMode.OpenOrCreate);  
int n;  
try { BinaryReader r = new BinaryReader(s);  
    n = r.ReadInt32();} catch { n = 0; }  
n++;s.Seek(0, SeekOrigin.Begin);  
BinaryWriter w = new BinaryWriter(s);  
w.Write(n); s.Close();Response.Write(n);%> !  
</body></html>
```

Script My.aspx

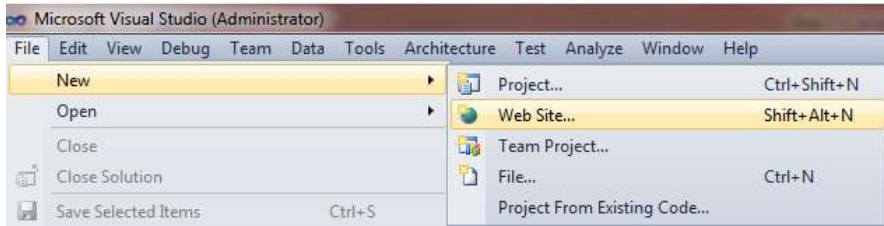
File My.HTML disisipkan kode ASP.NET dalam bahasa C# yang diawali dengan tag <% dan di akhiri dengan tag %>. Hal pertama yang dikerjakan adalah membuka file atau membuatnya jika tidak ada (`OpenOrCreate`), kemudian file tsb dibaca dengan diletakkan ke variable `n`, dan jumlah `n` dinaikkan `n++`, setelah itu nilai `n` dituliskan ke dalam file untuk menyimpan nilai `n`, setelah itu file ditutup dan nilai `n` dituliskan ke browser sebagai *response*: `Response.Write(n)`.

Hasil dari program diatas, adalah browser menampilkan halaman dengan jumlah pengunjung yang dinamis (menaik)

7.2. Bekerja Dengan Visual Studio 2010

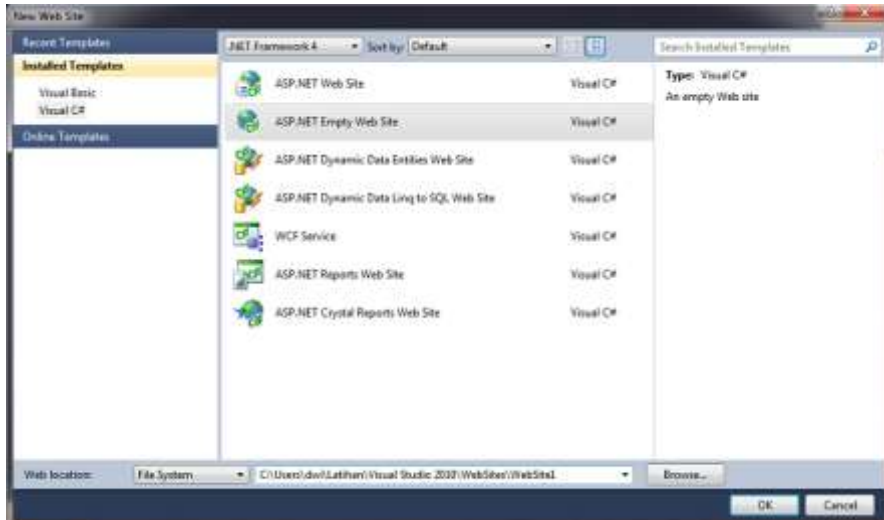
Pada Visual Studio 2010, fitur-fiturnya mirip dengan Visual Studio versi sebelumnya dengan beberapa perbedaan, yang paling menonjol adalah kerangka kerja, pada VS2010 memakai .NET versi 4. Langkah-langkah untuk membuat program yang telah dijelaskan pada subbab sebelumnya yaitu program My.aspx adalah

1. Buka VS2010 dan buat situ baru seperti yang diperlihatkan pada gambar 7.3 dibawah ini:



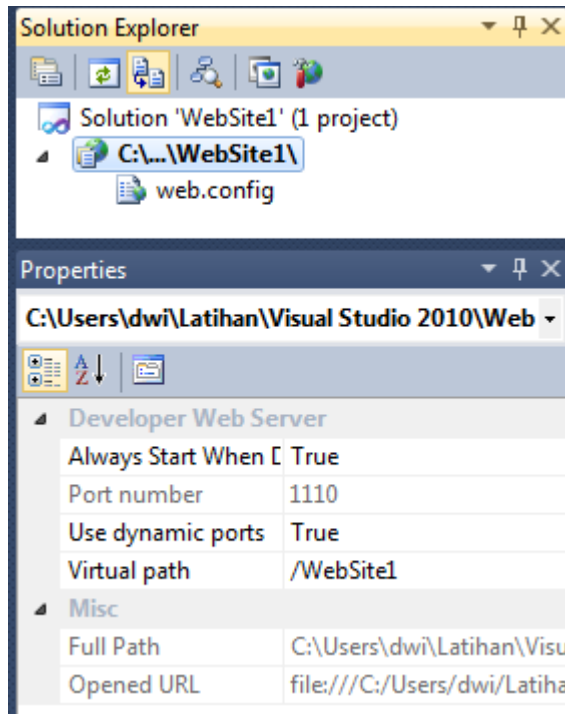
Gambar 7.3 Buat Situs Baru

Pilihan-pilihan yang tersedia pada VS2010 memberikan isian pada pilihan template, direktori situs, tipe bahasa, dan nama situs, seperti gb 7.4 dibawah ini:



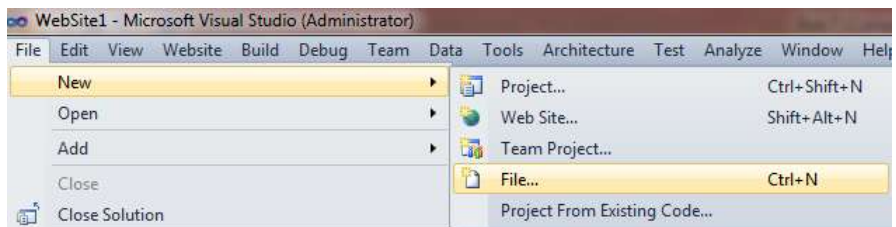
Gambar 7.4 Editor Pilihan Situs Baru

Direktori penyimpanan, penamaan situ dapat diubah sesuai dengan yang di kehendaki, setelah di tekan Ok, jendela VS2010 menampilkan jendela solution dan properti dari situs seperti terlihat pada gambar 7.5:



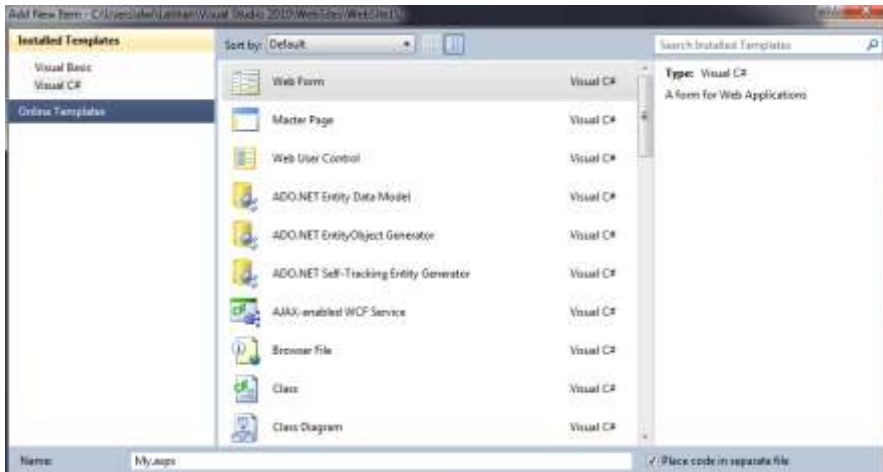
Gambar 7.5 Jendela Solution dan Properti

Pada subbab sebelumnya, terdapat file My.aspx. Hal ini bisa dibuat dengan cara memilih New File atau menekan Ctrl+N seperti window pada gb 7.6



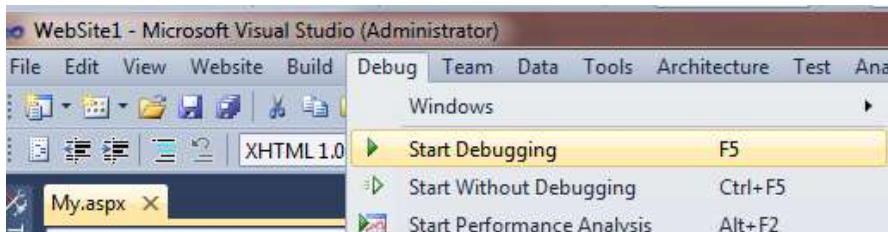
Gambar 7.6 Menu New File

Setelah itu muncul pilihan window template seperti pada gb 7.7 dibawah ini dan isi file name sesuai dengan subbab sebelumnya yaitu My.aspx:



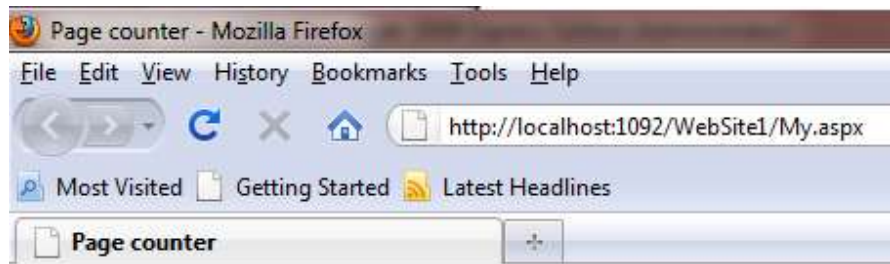
Gambar 7.7 Pembuatan Program My.aspx

Kemudian copikan kode sumber My.aspx dan sisipkan ke jendela yang telah terbuka dan jalankan program diatas, dengan menekan F5 atau memilih menu Start Debugging



Gambar 7.8 Menu Start Debugging

Hasil berupa jumlah pengunjung seperti yang diperlihatkan pada gb 7.9 yang berjalan di localhost dengan port yang disesuaikan.

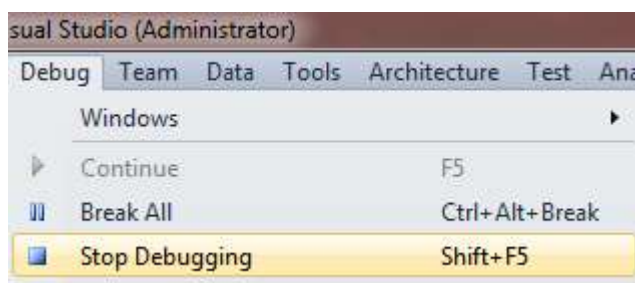


Welcome

You are visitor number 4 !

Gambar 7.9 Hasil Program My.aspx

Kembali pada VS2010 untuk menghentikan program diatas dengan memilih menu->debugging->stop debugging atau menekan tombol shift+F5, seperti terlihat pada gambar 7.8

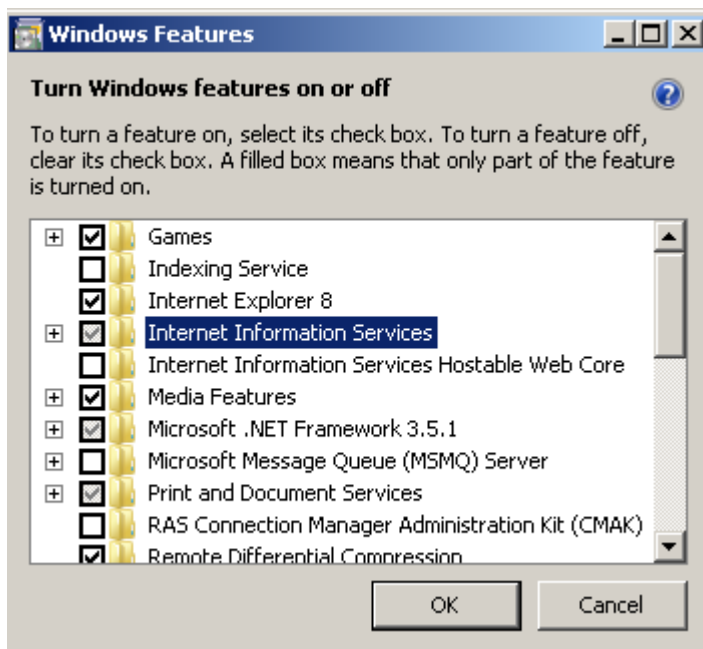


Gambar 7.10 Cara Stop Debugging

Program yang telah teruji pada VS2010 dapat ditempatkan pada server yang sesungguhnya, yaitu web server, sehingga dapat di akses oleh pengguna dari luar komputer.

7.3 Setting IIS pada Windows 7

File yang telah dibuat pada subab sebelumnya, dapat dijalankan pada server IIS, secara default, IIS belum aktif, harus di aktifkan terlebih dahulu, dengan cara memilih menu Start-> Control Panel -> Programs and Features -> Turn Windows features on or off, hingga muncul pilihan seperti gambar dibawah ini:



Gambar 7.11 Setting Windows Features

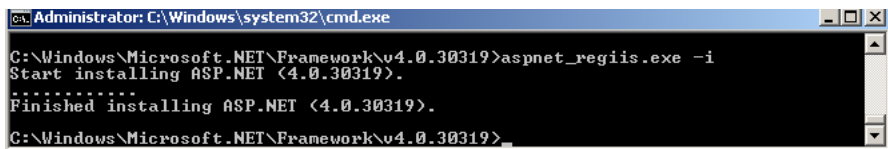
Pada windows (gb. 7.11) centang pilihan Internet Information Services dan klik Ok, setelah itu, windows akan melakukan setup agar web server berjalan pada system operasi windows tsb. Setelah selesai, hasil konfigurasi tsb, dapat dilihat pada menu Start -> Control Panel -> System and Security -> Administrative tools -> Computer Management, seperti yang terlihat pada gambar 7.12



Gambar 7.12 Setting IIS

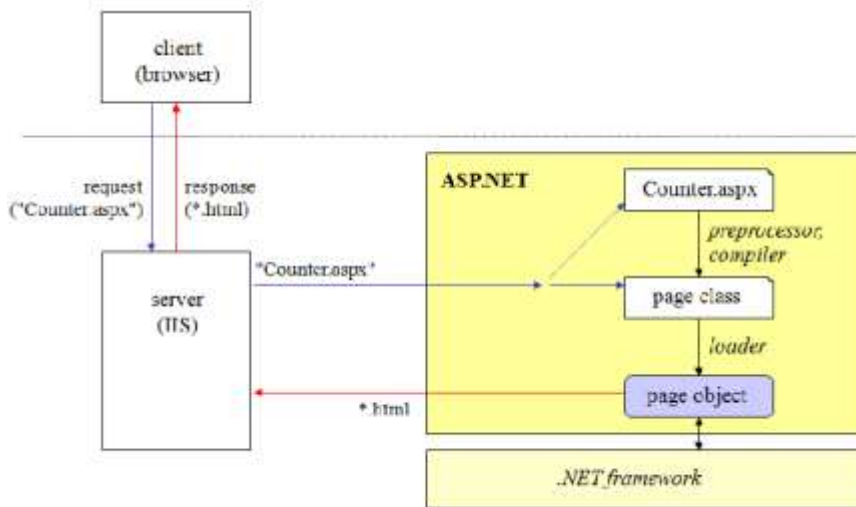
Secara default, lokasi web server ada di c:\inetpub\wwwroot, ketika browser mengakses http://localhost, lokasi yang dimaksud sebagai alamat folder adalah lokasi c:\inetpub\wwwroot. Biasanya, alamat lokasi diubah menjadi virtual directory untuk membuat directory sendiri dengan folder ditempat yang diinginkan.

Setelah Web server IIS disetup, terlebih dahulu harus dilakukan instalasi terhadap framework .NET, dengan cara meregisterkan framework .NET yang di-inginkan, seperti gambar dibawah ini



Gambar 7.13 Regsiter FrameWork .NET

Pada subab sebelumnya terdapat file My.aspx, file tsb dapat diketika di editor umum, seperti notepad. Kemudian disimpan ke dalam directory c:\inetpub\wwwroot. Setelah tersimpan, file tsb dapat dipanggil dari browser dengan mengetik: http://localhost/My.aspx hasil yang didapat seperti diperlihatkan pada gambar 7.9



Gambar 7.14 Cara Kerja Web Server IIS

Dari langkah-langkah pembuatan file My.aspx, sampai file tsb dipanggil oleh browser, gambaran atau cara kerja web server IIS adalah:

1. Pemakai menggunakan browser untuk mengakses file My.aspx
2. Web Server mengenali permintaan akses tsb (*request*) sebagai file ASP.NET, kemudian mengirimkan file My.aspx ke framework ASP.NET untuk diproses lebih lanjut
3. Framework .NET melakukan *preprocessor* dan melaluka *compile* terhadap file My.aspx, sehingga file tsb menjadi bentuk *page class*
4. Setelah *page class* terbentuk, dilakukan proses *loading* untuk menjadi sebuah obyek, dan dari obyek itulah dihasilkan file HTML yang dikirimkan ke web server
5. Web Server menerima hasil proses dari Framework .NET sebagai file HTML, dan hasil tsb dikirimkan ke browser
6. Browser menerjemahkan file HTML dan menampilkan sebagai dokumen.

7.4 File ASPX Sebagai Script

Penulisan hasil sebagai nilai counter seperti pada program My.aspx, dapat pula dituliskan sebagai kode script:

```
<%@ Page Language="C#" %><%@ Import Namespace="System.IO" %>
<html>
<head>
<title>Page counter</title>
<script Language="C#" Runat="Server">
int CounterPage {
    FileStream s = new FileStream("c:\\Data\\Counter.dat",
        FileMode.OpenOrCreate);
    ...
    n = r.ReadInt32();n++;
    ...
    return n;
}
</script></head>
<body><h1>Welcome</h1>
    You are visitor number <%= CounterPage()%> !
</body></html>
```

Bentuk script diatas mirip dengan kode javascript, fungsi CounterPage() dipanggil dengan tag <%= CounterPage()%>

Bentuk penulisan diatas, dapat pula dipecah menjadi dua bagian, satu sebagai program, dan yang lain sebagai sisipan script. File My.aspx dituliskan dengan memakai tag Inherits yang serupa dengan include file

```
<%@Page Language="C#" Inherits="My" Src="My.aspx.cs" %>
<html>
    <head> <title>Page counter</title> </head>
    <body>
        <h1>Welcome</h1>
        You are visitor number <%=CounterPage()%> !
    </body>
</html>
```

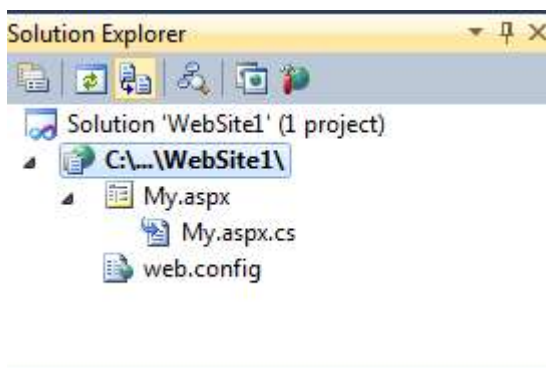
File My.aspx berisi Inherits dari sebuah kelas My yang berada pada file My.aspx.cs. File tsb berisi program yang menghitung

jumlah pengunjung seperti pada program My.aspx sebelum dipecah menjadi dua bagian.

Isi dari file My.aspx.cs adalah:

```
using System.IO;
public class My : System.Web.UI.Page {
public int CounterPage() {
    FileStream s = new FileStream("c:\\Data\\Counter.dat",
        FileMode.OpenOrCreate);
        ...
        n = r.ReadInt32();
        n++;
        ...
        return n;
    }
}
```

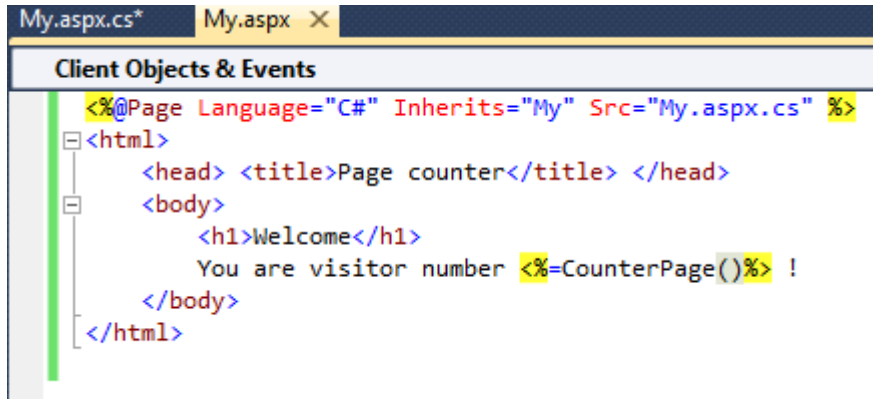
Pada VS2010 ketika membuat file kode program aspx baru, secara otomatis file sumber bahasa juga ikut dibuat, seperti pada pembuatan program My.aspx, maka terdapat pula file My.aspx.cs. Metode pembuatan program yang dipecah menjadi dua bagian dapat dilakukan sesuai pembagian yang terlihat pada gambar 7.14



Gambar 7.14 File My.aspx.cs

Langkah-langkah untuk memecah program My.aspx menjadi dua bagian adalah:

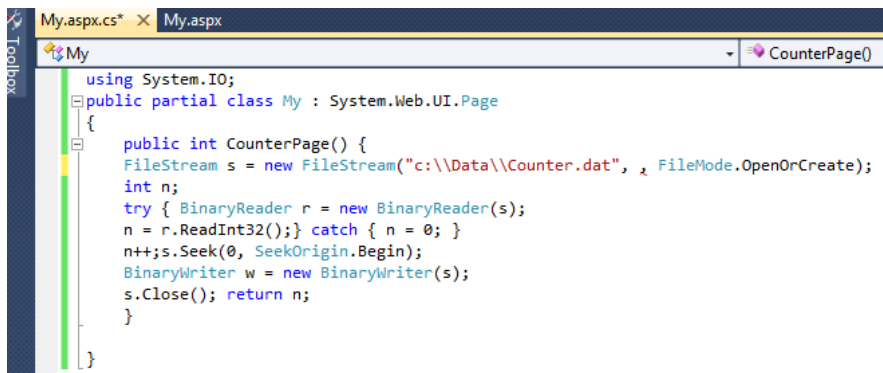
1. Buka File My.aspx dan copikan kode program yang sudah dituliskan sebelumnya, sehingga tampil seperti dibawah ini:



```
<%@Page Language="C#" Inherits="My" Src="My.aspx.cs" %>
<html>
  <head> <title>Page counter</title> </head>
  <body>
    <h1>Welcome</h1>
    You are visitor number <%=CounterPage()%> !
  </body>
</html>
```

Gambar 7.15 File My.aspx pada VS2010

2. Kemudian buka File My.aspx.cs, hapus semua isinya karena belum dibutuhkan, copikan kode program sebelumnya, seperti pada gambar dibawah ini



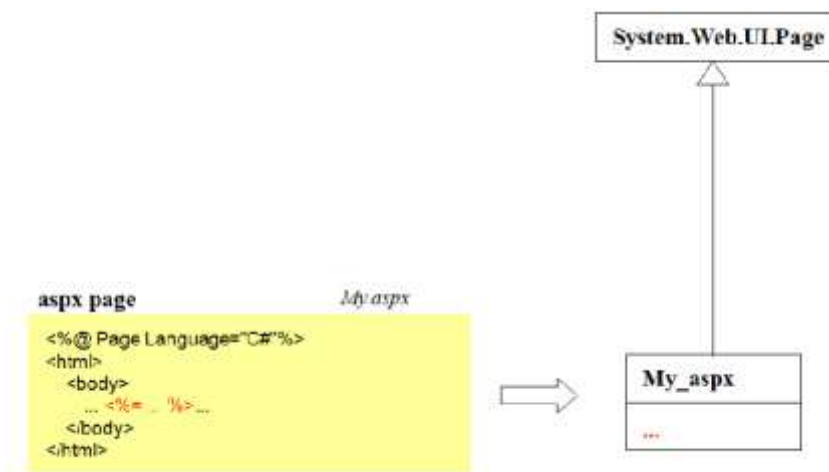
```
using System.IO;
public partial class My : System.Web.UI.Page
{
  public int CounterPage() {
    FileStream s = new FileStream("c:\\Data\\Counter.dat", FileMode.OpenOrCreate);
    int n;
    try { BinaryReader r = new BinaryReader(s);
    n = r.ReadInt32(); } catch { n = 0; }
    n++; s.Seek(0, SeekOrigin.Begin);
    BinaryWriter w = new BinaryWriter(s);
    s.Close(); return n;
  }
}
```

Gambar 7.16 File My.aspx.cs pada VS2010

Jalankan program diatas, hasilnya sama dengan program sebelumnya. Pada gambar 7.9

7.5 Kelas yang dibangkitkan dari File ASPX

Pada gambar 7.9 terlihat peran FrameWork .NET adalah pembangkit kelas dari file ASPX yang dikirimkan oleh web server IIS. File My.aspx akan dibangkitkan sebuah *page class* yang bernama sesuai dengan nama file yang bersangkutan, dalam hal ini My.aspx akan bernama My_aspx, seperti dilukiskan pada gambar 7.17 dibawah ini:



Gambar 7.17 Pembangkitan page Class

Hasil pembangkitan menjadi kelas, secara utuh menjadi 2 kelas dapat terlihat pada kode dibawah ini

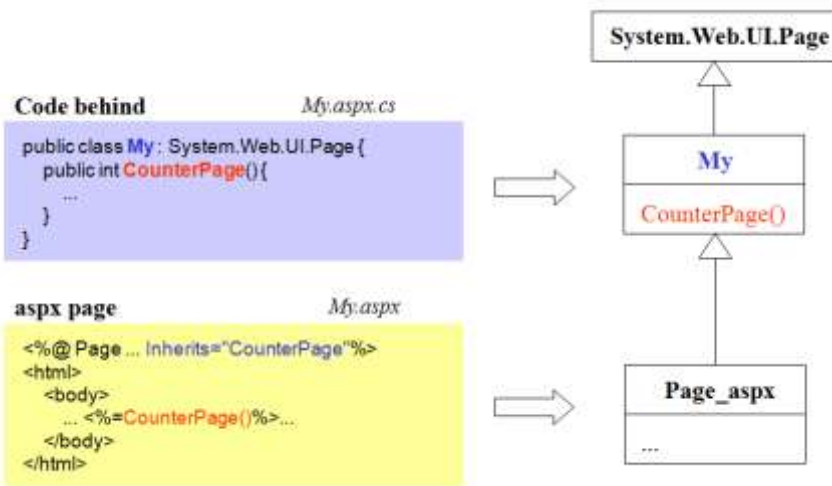
```
namespace ASP {
using System.IO;
...
public class My_aspx : CounterPage {
private static bool __initialized = false;
private static ArrayList __fileDependencies;
public My_aspx() {
ArrayList dependencies;
if ((__initialized == false)) { ... }
public override string TemplateSourceDirectory {
```

```

get { return "/Samples"; } }
private void __BuildControlTree(Control __ctrl) {
__ctrl.SetRenderMethodDelegate(new
RenderMethod(this.__Render__control1));
}
private void __Render__control1(HtmlTextWriter __output,
Control parameterContainer) {
__output.Write("\r\n<html>\r\n\t<head>
<title>Page counter</title>
</head>\r\n\t<body>\r\n\t\t" +
"<h1>Welcome</h1>\r\n\t\tYou are visitor number ");
__output.Write(CounterValue());
__output.Write(" !\r\n\t</body>\r\n</html>\r\n");
protected override void FrameworkInitialize() {
__BuildControlTree(this);
this.FileDependencies = __fileDependencies;
this.EnableViewStateMac = true; this.Request.ValidateInput();
... } }

```

Pada metode file ASPX yang dipisah, pembangkitan kelas juga terjadi dengan dua kelas pula. Hal ini dapat digambarkan pada gb 7.18, dimana file aspx dan file cs menjadi *page class* masing-masing



Gambar 7.18 Pembangkitan dua page Class

