

MODUL PRAKTIKUM  
PEMROGRAMAN BERBASIS WEB 3  
Menggunakan ASP.NET



Oleh:  
Fery Updi

TEKNIK INFORMATIKA  
Ver.1.2016

ASP.NET adalah salah satu teknologi pengembangan web yang paling menarik yang ditawarkan saat ini. ASP.NET adalah platform handal yang dilengkapi dengan banyak sekali fungsi built-in, tingkat kinerja yang mengagumkan, dan merupakan salah satu IDE (Integrated Development Environment) terbaik yang telah terintegrasi dalam paket Visual Studio.

### ***Apa itu ASP.NET***

ASP.NET merupakan teknologi server-side untuk mengembangkan aplikasi web berbasis Microsoft .NET Framework.

Pada saat browser meminta (request) halaman web yang dibuat dengan teknologi Client-side seperti HTML, javascript dan Cascading Style Sheet (CSS), web server hanya mengirimkan balik file-file tersebut pada browser tanpa melakukan proses apapun. Kemudian browserlah yang bertanggung jawab sepenuhnya untuk membaca dan menterjemahkan script atau markup pada file-file tersebut.

Sedangkan teknologi Server-side seperti ASP.NET berbeda, script atau kode tidak diterjemahkan disisi client tapi diinterpretasikan oleh server web. Sehingga kode pada halaman dibaca oleh server dan digunakan untuk menghasilkan HTML, JavaScript dan CSS yang kemudian dikirim ke browser.

Sebelum belajar ASP.NET, pastikan Anda telah menginstal komponen-komponen software yang dibutuhkan dan telah berjalan dengan baik pada komputer Anda.

### ***Webserver IIS (Internet Information Service)***

IIS adalah web server bawaan Microsoft Windows yang dapat digunakan untuk menjalankan aplikasi web ASP.NET. Anda dapat menginstal IIS melalui CD master Windows seperti Windows 2000 Professional, Server dan Advances Server, Windows XP Professional, Windows Vista, Windows 7 dan semua versi Windows Server.

Untuk **menginstal IIS** ikuti langkah dibawah ini:

1. Masukkan CD master Windows
2. Buka Control Panel dan pilih Add or Remove Programs
3. Pilih Add/Remove Windows Components. Daftar komponen window akan ditampilkan
4. Pada daftar komponen berikan tanda ceklis pada Internet Information Service (IIS)
5. Klik Next untuk menginstal IIS

Atau dapat melakukan pengaturan sebagai berikut:

1. Buka Control Panel
2. Pilih Menu Program and Features
3. Pilih link Turn Windows Features on or off
4. Pilih Internet Information Services (IIS)
5. Tunggu beberapa saat
6. Tes jalankan di browser ketik: localhost

Jika terjadi konflik program dengan webserver lain, dapat menonaktifkan sementara IIS dengan cara:

1. Buka Control Panel
2. Pilih System and Security
3. Pilih Administrative Tools
4. Pilih 2x Services
5. Pilih World Wide Web Publishing Service
6. Pada Tab General pilih Startup type :Pilih Manual
7. Pilih tombol Stop utk nonaktif
8. Pilih tombol Start utk aktif
9. Klik Apply dan OK

### ***.NET Framework***

Untuk memulai membuat aplikasi ASP.NET Anda perlu menginstal .NET Framework dan Software Development Kit (SDK). NET Framework berisi file-file yang diperlukan untuk menjalankan dan melihat halaman ASP.NET, sedangkan SDK berisi file-file contoh, dokumentasi dan berbagai peralatan pendukung.

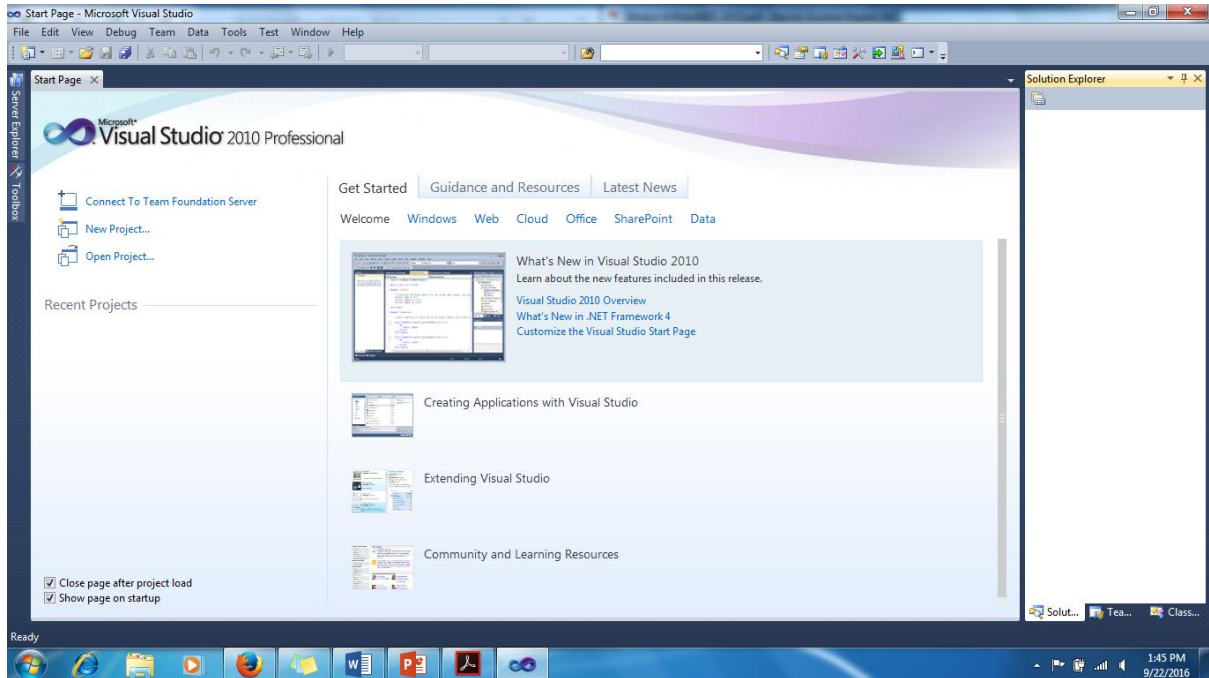
.NET Framework dan SDK dapat didownload melalui situs [www.asp.net](http://www.asp.net).

Selain itu untuk lebih mudahnya Anda dapat menginstal Microsoft Visual Studio 2010 atau Microsoft Visual Web Developers 2010 Express include SQL Server Express.

# 2

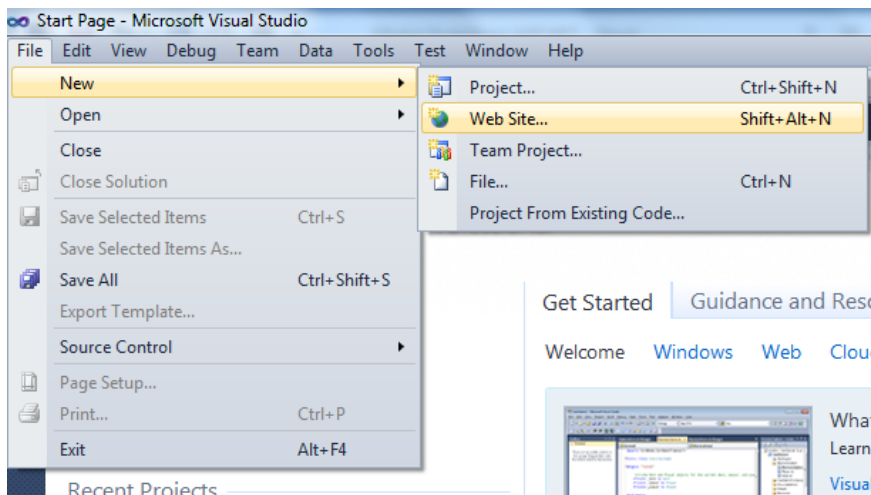
## Menggunakan Microsoft Visual Studio 2010

Membuat file aspx bisa dilakukan melalui notepad dengan mengetikkan semua script baik html maupun bahasa pemrogramannya, akan tetapi kita bisa memanfaatkan Tools yang dikeluarkan oleh Microsoft khusus untuk men-develop aplikasi berbasis .NET Framework yaitu **Microsoft Visual Studio**.

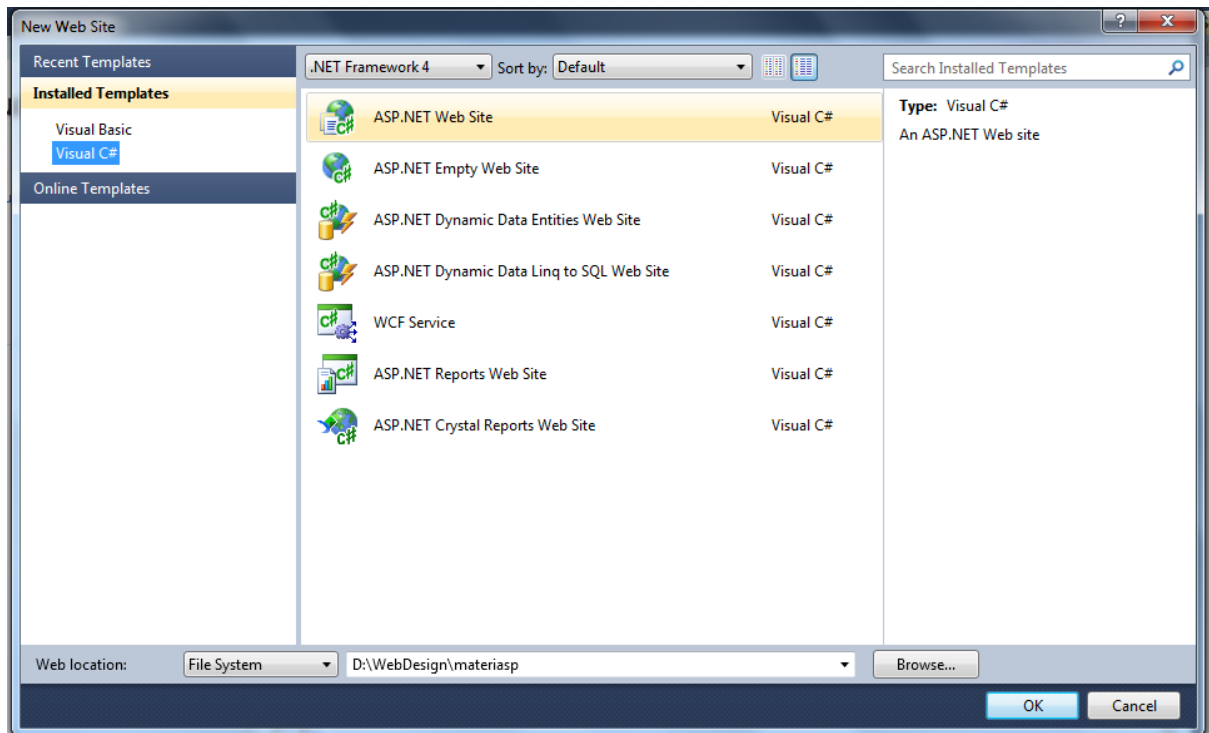


Setelah terinstal dengan baik, jalankan Ms Visual Studio melalui Start > Program Files, kemudian buat website baru untuk menampung file-file latihan yang akan kita buat selama praktikum.

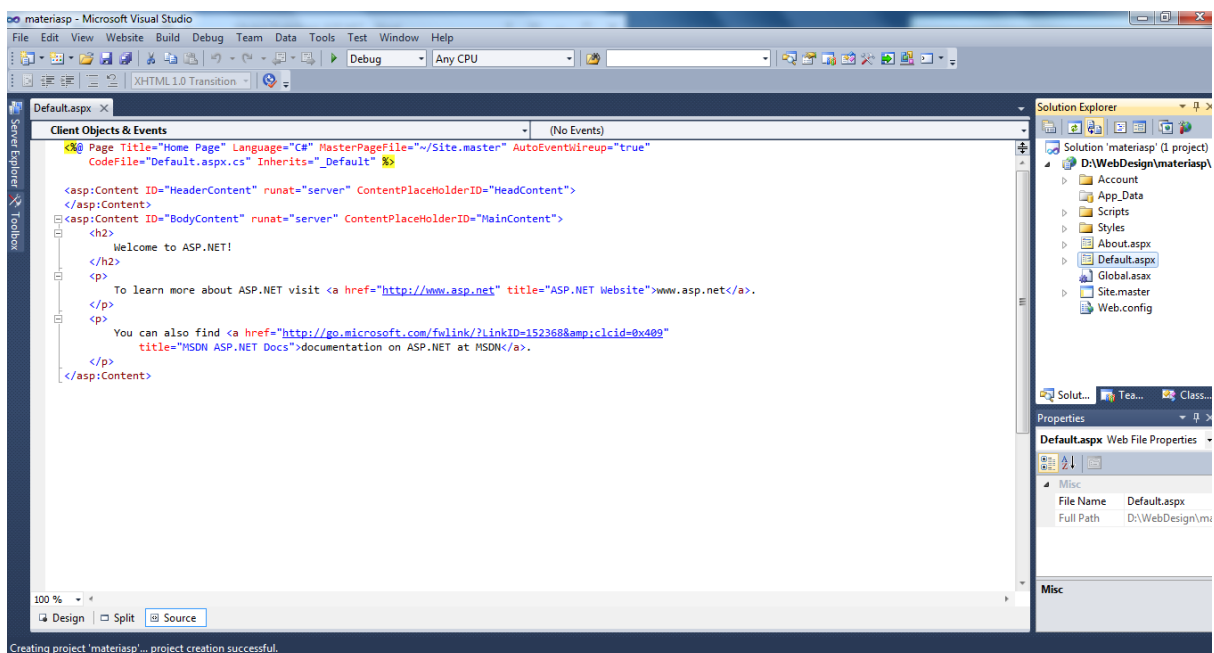
### 1. Pilih menu **File > New > Website**



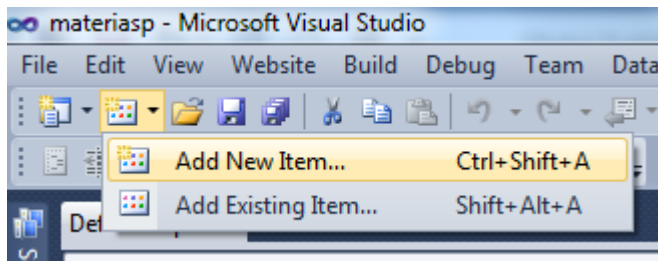
2. Kemudian pilih **Visual C#** pada Installed Templates, pilih ASP.NET Web Site dan tentukan lokasi file-file website akan disimpan



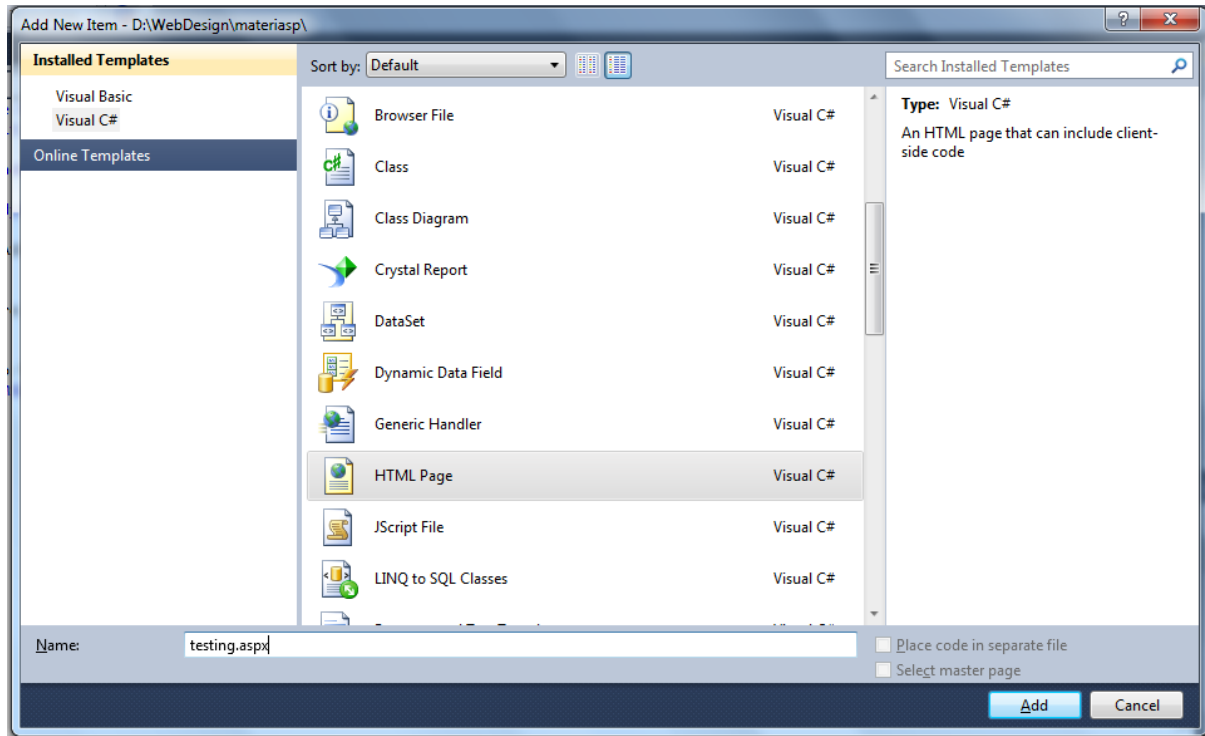
3. Klik **OK** untuk membuat website baru, kemudian akan tampil pada Editor Visual Studio file Default.aspx yang merupakan file default website.



4. Untuk menambahkan file baru pilih icon **Add New Item** pada toolbars



5. Dan pilih **Visual C#** pada **Installed Templates**, pilih **HTML Page** dan ketik nama file yang akan kita buat dengan ekstensi **.aspx**



6. Klik tombol **Add** dan sesuaikan isi file **testing.aspx** dengan yang ada pada modul.
7. Ketik kode program seperti dibawah ini

```
testing.aspx X Default.aspx
Server Objects & Events (No Events)
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Halaman Pertama ASP.NET 2.0</title>
<script runat="server" language="C#">
void Page_Load(Object s, EventArgs e){
timeLabel.Text = DateTime.Now.ToString();
}
</script>
</head>
<body>
<p> Selamat Datang di ASP.NET</p>
<p> Waktu sekarang adalah:
<asp:Label runat="server" ID="timeLabel"/></p>
</body>
</html>
```

8. Klik icon Start Debugging pada toolbars untuk menjalankan file testing.aspx pada browser.

### Control Event dan Subroutines

Event merupakan aksi yang dibangkitkan kemudian code program dijalankan untuk merespon aksi tersebut. Sebagai contoh, pada script dibawah ini terdapat label dan tombol, perhatikan penggunaan atribut onClick pada tombol. Saat tombol diklik maka Subroutines button\_Click akan dijalankan.

```

klikevent.aspx  testing.aspx  Default.aspx
Server Objects & Events  (No Events)
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Klik Button</title>
<script runat="server" language="C#">
public void button_Click(Object s, EventArgs e) {
messageLabel.Text = "BAA KEKOK";
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Button id="button" runat="server" onClick="button_Click" Text="Klik AKU donk..." />
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>

```

**File Name:** klikevent.aspx

Ada banyak event yang dapat digunakan meskipun beberapa event hanya ditemukan pada kontrol tertentu, berikut ini adalah event yang dapat Anda gunakan.

- **OnClick**

Subrutin ini akan dijalankan saat sebuah tombol di klik.

- **OnCommand**

Seperti OnClick, subrutin ini akan dijalankan pada saat sebuah tombol diklik.

- **OnLoad**

Subroutine ini akan dijalankan ketika tombol diloat, biasanya ketika halaman pertama kali diloat.

- **OnInit**

Ketika tombol diinisialisasi, setiap subrutin yang diberikan dalam atribut ini akan jalankan.

- **OnPreRender**

Kita dapat menggunakan atribut ini untuk menjalankan script sebelum tombol ditampilkan.

- **OnUnload**



Subrutin ini akan dijalankan ketika kontrol dihapus dari memori/buffer, dengan kata lain ketika pengguna berpindah ke halaman lain atau saat menutup browser.

- **OnDisposed**

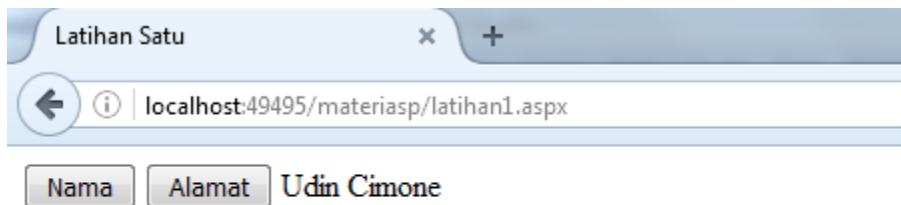
Subroutine yang ditentukan oleh atribut ini dijalankan ketika tombol dilepaskan dari memori.

- **OnDataBinding**

Atribut ini dinyalakan secara otomatis ketika tombol terikat ke sumber data.

## LATIHAN

Buatlah halaman seperti dibawah ini dengan fungsi tombol masing-masing adalah:



- **Page Event**

Selain event diatas yang dapat dibangkitkan melalui kontrol, ada event jenis lain yaitu page event dimana semua event dihasilkan oleh sebuah halaman.

Berikut ini event pada halaman yang sering digunakan.

- **Page\_Init**

Dijalankan ketika halaman diinisialisasi

- **Page\_Load**

Dijalankan pada saat pertama kali browser memproses permintaan (request) halaman, dan semua kontrol didalamnya diperbarui.

- **Page\_PreRender**

Dijalankan pada saat pertama kali sebuah halaman dipanggil, tetapi sebelum browser menerima respon apapun dari server

- **Page\_UnLoad**

Dijalankan ketika halaman tidak lagi dibutuhkan oleh server dan tidak ada aktifitas apapun.

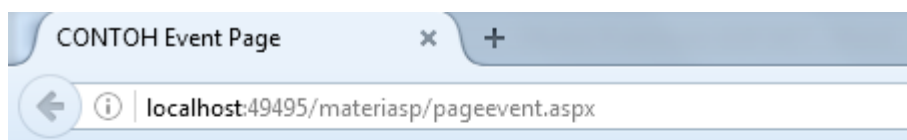
Urutan diatas sesuai dengan peristiwa mereka dieksekusi. Dengan kata lain, event Page\_Init merupakan event pertama yang dipanggil oleh halaman, diikuti oleh Page\_Load, Page\_PreRender, dan akhirnya Page\_UnLoad.

Contoh dibawah ini menggambarkan cara kerja page event, Anda tidak perlu menentukan urutan subroutine yang akan dijalankan karena secara default subroutine akan dijalankan sesuai dengan nama-nama event tersebut.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CONTOH Event Page</title>
<script runat="server" language="C#">
void Page_Init(Object s, EventArgs e)
{
messageLabel.Text = "1. Page_Init <br/>";
}
void Page_Load(Object s, EventArgs e)
{
messageLabel.Text += "2. Page_Load <br/>";
}
void Page_PreRender(Object s, EventArgs e)
{
messageLabel.Text += "3. Page_PreRender <br/>";
}
void Page_UnLoad(Object s, EventArgs e)
{
messageLabel.Text += "4. Page_UnLoad <br/>";
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Hasil output adalah:



1. Page\_Init
2. Page\_Load
3. Page\_PreRender

## Variabel

Variabel merupakan hal yang mendasar dalam pemrograman dan Anda harus mengetahui aturannya dengan baik. Anda harus memberikan nama suatu variabel atau mengidentifikasikannya untuk item data tertentu, selanjutnya variabel dapat digunakan untuk menyimpan suatu nilai, memproses/memodifikasi nilainya atau mengkosongkannya kembali.

Ada banyak macam tipe data, seperti string, integer, dan floating. Sebelum menggunakan variabel Anda harus menentukan tipe data apa yang akan ditampung sebuah variabel, apakah string, integer ataukah desimal.

Perhatikan contoh dibawah ini:

```
String nama;  
Int umur;
```

Baris diatas mendeklarasikan tipe data string untuk variabel nama dan integer untuk variabel umur. Kadang-kadang kita ingin memberikan nilai awal untuk sebuah variabel yang kita deklarasikan, perhatikan contoh dibawah ini:

```
String jenisMobil = "BMW";
```

Atau kita dapat mendeklarasikan lebih dari satu variabel dalam satu grup dan sekaligus memberikan nilai awal seperti contoh dibawah ini.

```
string carType, carColor = "blue";
```

**Tabel 2.1:** Daftar Tipe data yang bisa digunakan

Tipe Data	Keterangan
Int	Bilangan bulat antara -2.147.483.648 s/d 2.147.483.647
Decimal	Angka dengan bilangan desimal hingga 28 digit
String	Semua nilai dengan jenis huruf/teks
Char	Karakter tunggal
Bool	True atau false
Object	Tipe umum untuk sebuah objek

Dalam penggunaannya ketika Anda mendeklarasikan suatu tipe data pada variabel maka hanya tipe data tersebut yang dapat ditampung oleh variabel itu, jadi Anda tidak dapat memberikan nilai integer pada variabel dengan tipe data string. Untuk itu perlu dilakukan konversi suatu data kedalam tipe data lain. Perhatikan kasus dibawah ini:

```
int intX;  
string strY = "35";  
intX = strY + 6;
```

skrip diatas akan menampilkan pesan kesalahan, untuk memperbaikinya perhatikan solusi dibawah ini:

```
<script runat="server" language="C#">  
void Page_Load(){  
int intX;  
string strY = "35";  
intX = Convert.ToInt32(strY) + 6;  
txtLabel.Text = Convert.ToString(intX);  
}  
</script>  
<form runat="server">  
<asp:Label id="txtLabel" runat="server" />  
</form>
```

## Array

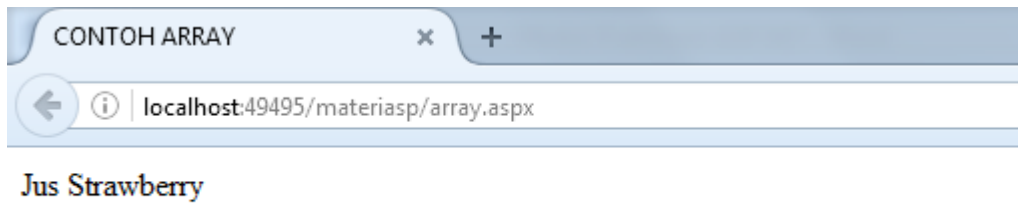
Array merupakan jenis variabel yang spesial karena dapat menyimpan beberapa nilai dengan tipe data yang sama dimana setiap nilai item dalam array dapat dipanggil menggunakan nama variabel array tersebut dengan menentukan posisi itemnya.

Ketiklah perintah dibawah ini dan simpan dengan nama **array.aspx**



```
array.aspx X pageevent.aspx latihan1.aspx klikevent.aspx testing.aspx Default.aspx  
Server Objects & Events (No Events)  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <title>CONTOH ARRAY</title>  
    <script runat="server" language="C#">  
      void Page_Load() {  
        // Deklarasi Array  
        string[] drinkList = new string[4];  
        // Menempatkan nilai  
        drinkList[0] = "Es Jeruk";  
        drinkList[1] = "Jus Strawberry";  
        drinkList[2] = "Teh Manis";  
        drinkList[3] = "Kopi";  
        // Mengakses pemanggilan nilai sesuai urutan yang dipilih  
        drinkLabel.Text = drinkList[1]; }  
    </script>  
  </head>  
  <body>  
    <form runat="server">  
      <asp:Label id="drinkLabel" runat="server" />  
    </form>  
  </body>  
</html>
```

Hasil output adalah:



Perhatikan penggunaan variabel array diatas, deklarasinya menggunakan kurung siku dengan diikuti tipe datanya kemudian menentukan jumlah item yang akan dimiliki oleh variabel array tersebut (pada contoh diatas adalah 4) dengan menggunakan kata kunci 'new'.

Pada variabel array, item pertama selalu pada posisi 0, item kedua pada posisi 1 dan seterusnya. Untuk itu kita dapat menentukan nilai setiap itemnya seperti berikut:

```
drinkList[0] = "Es Jeruk";  
drinkList[1] = "Jus Strawberry";  
drinkList[2] = "Teh Manis";  
drinkList[3] = "Kopi";
```

## ***Function***

Function merupakan satu baris perintah atau lebih yang dibungkus (encapsulate) dengan nama yang telah didefinisikan. Selain itu function dapat mengembalikan nilai.

Skrip dibawah merupakan contoh sederhana sebuah function.

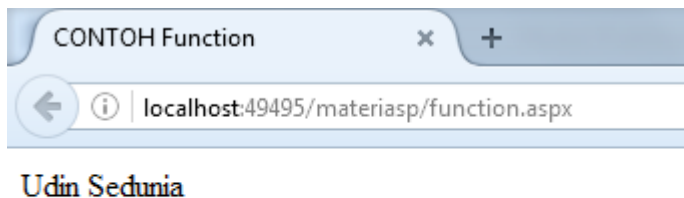
File name: **function.aspx**



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CONTOH Function</title>
<script runat="server" language="C#">
// Disini pembuatan fungsi
string getName() {
return "Udin Sedunia";
}
// Dan kita menggunakan Page_Load handler
void Page_Load() {
messageLabel.Text = getName();
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Jika skrip diatas dijalankan melalui browser, maka saat event **Page\_Load** dijalankan fungsi dengan nama **getName** akan dipanggil. Perhatikan hasil yang ditampilkan pada browser.

Hasil output adalah:



Perintah 'return' pada fungsi getName akan mengembalikan nilai dimana fungsi itu dipanggil. Namun perlu diperhatikan saat mendeklarasikan function, kita harus memastikan tipe data apa yang akan dihasilkan function tersebut. Perhatikan contoh kasus dibawah ini:

File name: **function.aspx**



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CONTOH 2 function</title>
<script runat="server" language="C#">
int addUp(int x, int y){
return x + y;
}
void Page_Load() {
messageLabel.Text = addUp(8,4);
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Jika dijalankan skrip diatas akan menampilkan pesan kesalahan karena nilai yang dikembalikan function addUp ber-tipe data integer sedangkan objek messageLabel.Text ber-tipe data string, maka untuk memperbaikinya tambahkan perintah konversi pada saat memanggil function addUp()

```
messageLabel.Text = addUp(8,4).ToString();
```

Atau

```
messageLabel.Text = Convert.ToString(addUp(8,4));
```

sehingga hasilnya pada browser adalah sebagai berikut:



## Operator

Tabel 2.2: Daftar Operator dalam ASP.NET

Operator	Keterangan
>	Lebih besar
>=	Lebih besar sama dengan
<	Lebih Kecil
<=	Lebih kecil sama dengan
!=	Tidak sama dengan
==	Sama dengan
=	Menentukan nilai suatu variabel
	Atau
&&	Dan
+	Menggabungkan string
New	Membuat objek atau array
*	Kali
/	Bagi
+	Tambah
-	Kurang

### Contoh

#### Penggunaan Function dan Operator

File name: **function3.aspx**



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tri
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CONTOH 3 Functions & Operator</title>
<script runat="server" language="C#">
void Page_Load() {
string user="Udin";
int itemsBought=10;
if (user == "Udin" && itemsBought != 0) {
messageLabel.Text = "Hallo" + user + ", Apakah Anda ingin melanjutkan ke proses " + "pembayaran?";
}
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Hasil output adalah:





HalloUdin, Apakah Anda ingin melanjutkan ke proses pembayaran?

Contoh diatas menggunakan operator **sama dengan**, tidak sama dengan dan operator logical **“and”** dalam perintah If untuk menampilkan pesan konfirmasi pembayaran. Operator sama dengan ‘==’ digunakan untuk membandingkan dua nilai apakah sama atau tidak, sedangkan tanda ‘=’ digunakan untuk menentukan atau memberikan nilai kepada suatu variabel.

### Conditional Logic

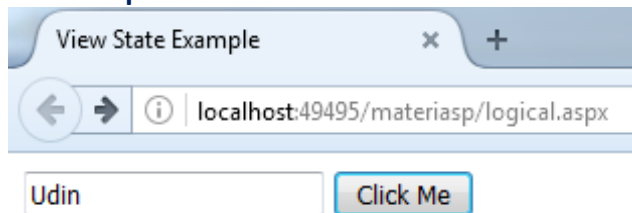
Dalam sebuah aplikasi akan ada suatu aksi/perintah yang akan dijalankan hanya jika suatu kondisi sesuai dengan yang telah ditentukan, misalnya jika pengguna memilih sebuah checkbox atau memilih salah satu item dari DropDownList atau bisa juga mengetikkan suatu kata pada TextBox. Kita cek pilihan tersebut menggunakan kondisi, secara sederhana kita dapat menggunakan perintah if dimana perintah ini sering dipasangkan dengan kondisi else yang akan dijalankan jika kondisinya tidak sesuai.

Perhatikan contoh dibawah ini:

File name: **logical.aspx**

```
logical.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.c
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>View State Example</title>
  <script runat="server" language="C#">
    void Click(Object s, EventArgs e) {
      if(userName.Text == "pageEvent"){
        Response.Redirect("pageevent.aspx");
      }else{
        messageLabel.Text = userName.Text;
      }
    }
  </script>
</head>
<body>
  <form id="Form1" runat="server">
    <asp:TextBox id="userName" runat="server" />
    <asp:Button id="submitButton" runat="server"
      Text="Click Me" OnClick="Click" />
    <asp:Label id="messageLabel" runat="server" />
  </form>
</body>
</html>
```

Hasil output adalah:

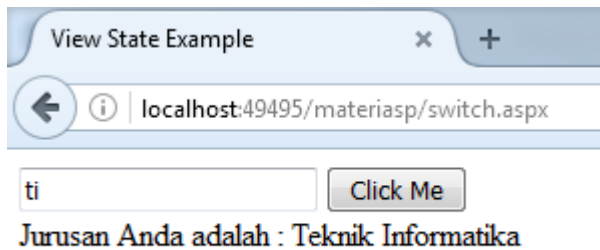


Seringkali kita harus mengecek suatu nilai untuk beberapa kemungkinan dan menentukan perintah yang berbeda untuk beberapa kondisi/kasus. Untuk menangani hal semacam ini kita dapat menggunakan perintah switch:

File name: switch.aspx

```
switch.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.c
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>View State Example</title>
<script runat="server" language="C#">
void Click(Object s, EventArgs e) {
switch (jurusan.Text) {
case "mi":
messageLabel.Text = "Manajemen Informatika";
break;
case "ti":
messageLabel.Text = "Teknik Informatika";
break;
case "si":
messageLabel.Text = "Sistem Informasi";
break;
default:
messageLabel.Text = "Jurusan tidak terdaftar";
break;
}
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:TextBox id="jurusan" runat="server" />
<asp:Button id="submitButton" runat="server"
Text="Click Me" OnClick="Click" /><br />
Jurusan Anda adalah : <asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Hasil output adalah:



## Loops

Loops yang dimaksud adalah baris perintah yang dijalankan secara berulang-ulang selama kondisi pengulangannya bernilai benar. Ada dua macam pengulangan yaitu :

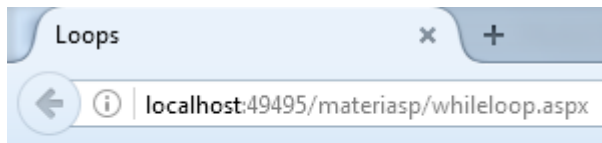
1. While loops, atau Do loops
2. For loops, termasuk perintah For Next dan For Each

while loop merupakan jenis pengulangan yang paling sederhana dimana baris perintah akan dijalankan berulang-ulang selama kondisinya bernilai benar. Berikut ini adalah contohnya:

**File name: whileloop.aspx**

```
whileloop.aspx X
Server Objects & Events
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Loops</title>
  <script runat="server" language="C#">
    void Page_Load()
    {
      // nilai awal counter
      int counter = 0;
      while (counter <= 10) {
        // berikan nilai counter pada label
        messageLabel.Text += counter.ToString();
        // counter ditambah 1
        counter++;
      }
    }
  </script>
</head>
<body>
  <form id="Form1" runat="server">
    <asp:Label id="messageLabel" runat="server"/>
  </form>
</body>
</html>
```

**Hasil output adalah:**



Bentuk lain dari **While loop** adalah **Do While loop**, dimana pengecekan kondisinya berada pada akhir perintah pengulangan bukan pada awal pengulangan itu akan dijalankan seperti pada While loop:

```
void Page_Load()
{
    int counter = 0;
    do {
        messageLabel.Text += counter.ToString();
        counter++;
    }
    while (counter <= 10);
}
```

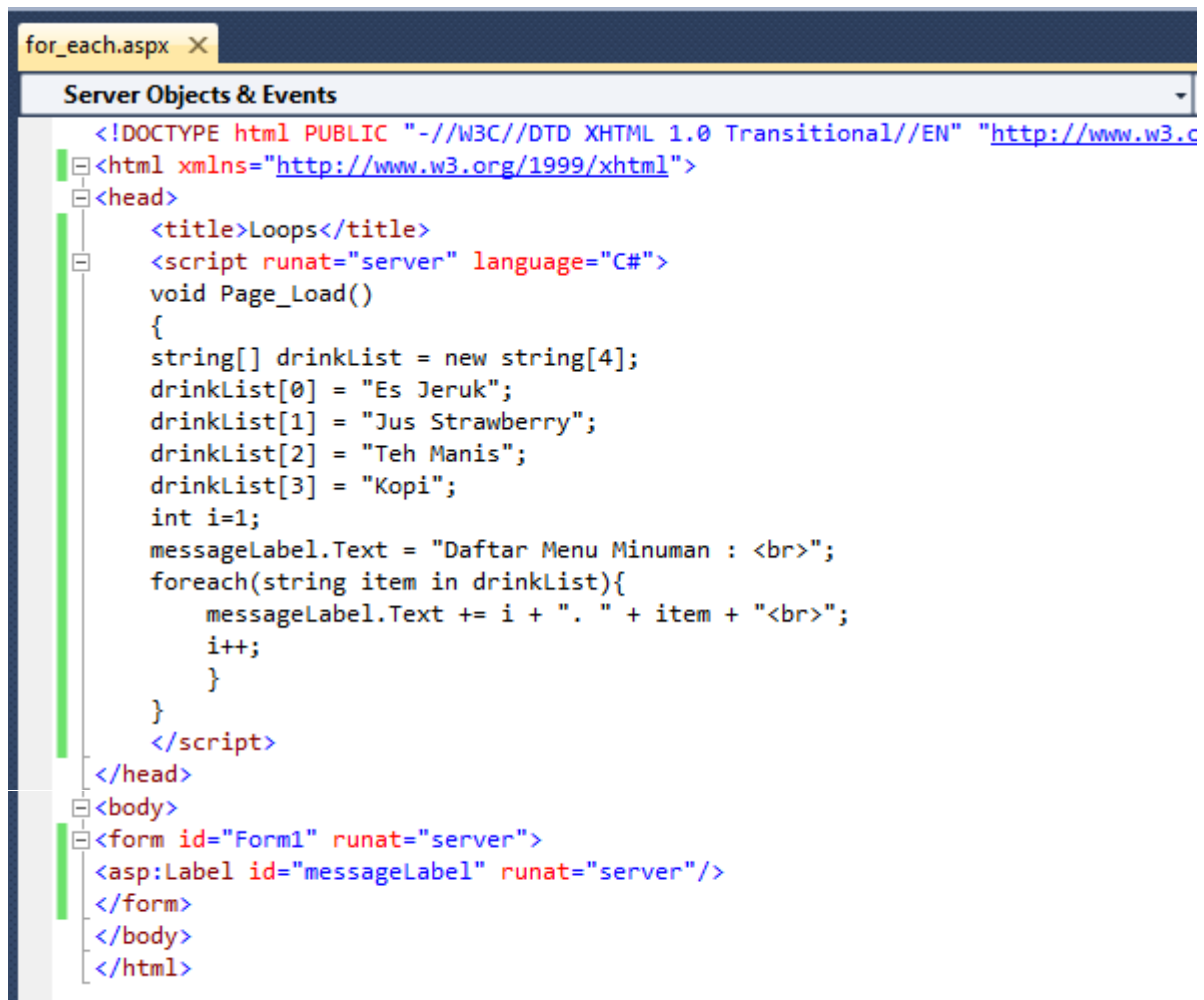
Pengulangan For mirip dengan While loop, tapi biasanya kita menggunakannya ketika sudah mengetahui berapa kali pengulangan yang akan dijalankan. Contoh dibawah ini akan menampilkan hasil yang sama dengan perintah pengulangan sebelumnya:

```
int i;
for (i = 1; i <= 10; i++) {
    messageLabel.Text = i.ToString();
}
```

Pertama kita berikan nilai awal (**i = 1**) kemudian nilai ini akan di cek setiap kali pengulangan (**i <= productList.Items.Count**), dan ditentukan berapa tingkat pertambahannya setiap kali pengulangan (i++).

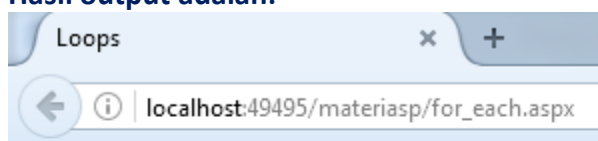
Bentuk lain dari pengulangan jenis For adalah For Each, dimana banyaknya pengulangan ditentukan oleh jumlah item dalam variabel bertipe array. Contoh berikut adalah pengulangan terhadap array dengan nama **drinkList**:

**File name: for\_each.aspx**



```
for_each.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.c
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Loops</title>
<script runat="server" language="C#">
void Page_Load()
{
string[] drinkList = new string[4];
drinkList[0] = "Es Jeruk";
drinkList[1] = "Jus Strawberry";
drinkList[2] = "Teh Manis";
drinkList[3] = "Kopi";
int i=1;
messageLabel.Text = "Daftar Menu Minuman : <br>";
foreach(string item in drinkList){
    messageLabel.Text += i + ". " + item + "<br>";
    i++;
}
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Label id="messageLabel" runat="server"/>
</form>
</body>
</html>
```

Hasil output adalah:

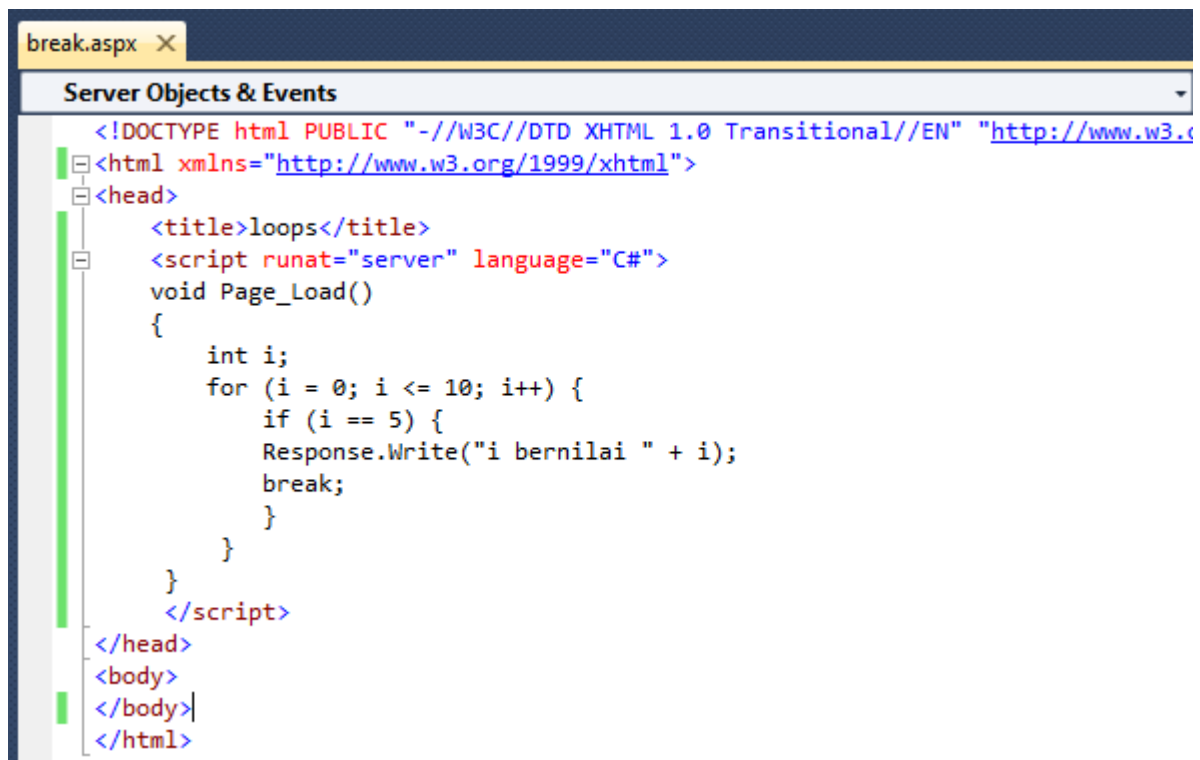


Daftar Menu Minuman :

1. Es Jeruk
2. Jus Strawberry
3. Teh Manis
4. Kopi

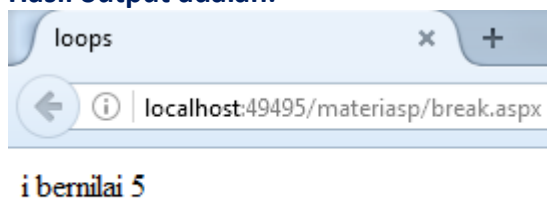
Anda dapat keluar dari pengulangan dengan menggunakan perintah break seperti contoh dibawah ini:

**File name: break.aspx**



```
break.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>loops</title>
<script runat="server" language="C#">
void Page_Load()
{
    int i;
    for (i = 0; i <= 10; i++) {
        if (i == 5) {
            Response.Write("i bernilai " + i);
            break;
        }
    }
}
</script>
</head>
<body>
</body>
</html>
```

Hasil output adalah:



Pada contoh diatas, pengulangan akan dihentikan ketika kondisi  $i = 5$ , dan menampilkan pesan seperti pada contoh gambar diatas.

### Web Forms

Web Form merupakan komponen penting dalam sebuah aplikasi berbasis web dimana dalam pelaksanaannya ini dapat digunakan sebagai alat untuk berkomunikasi antara pengguna dengan aplikasi. Setiap Web Form harus memiliki tag `<form runat="server">` dan terdapat element-element ASP.NET sesuai dengan kebutuhan. Dalam satu halaman web tidak boleh ada lebih dari satu Web Form.

Untuk menggunakan dan memanipulasi web form, kita dapat menggunakan class bawaan Net Framework yaitu `System.Web.UI.page`.

Kita dapat menggunakan beberapa tipe element dalam form termasuk tag HTML standar, tetapi ada pertimbangan lain agar suatu element dapat diidentifikasi dan dimanipulasi baik nilai maupun atributnya pada saat pertama kali halaman web dijalankan (load) atau pada saat sebuah web form disubmit. Element-element ini dalam ASP.NET disebut control, yang terdiri dari:

- HTML Server Controls
- Web server controls
- Web user controls
- Master pages

Ada perbedaan yang signifikan antar jenis control tersebut, namun apa yang membuat mereka mirip adalah semua control dapat mudah diintegrasikan dan digunakan berulang kali pada aplikasi yang kita buat.

### HTML Server Controls

HTML Server Control sangat mirip dengan HTML pada umumnya hanya ada penambahan atribut `runat="server"`. Hal ini dilakukan agar tag HTML yang kita gunakan dapat dikontrol secara dinamis melalui bahasa pemrograman seperti VB atau C#. Lihat pada table daftar Server Controls.

Table 3.1: HTML Server Controls

Class	Associated Tag
HtmlAnchor	<code>&lt;a runat="server" &gt;</code>
HtmlButton	<code>&lt;button runat="server"&gt;</code>
HtmlForm	<code>&lt;form runat="server"&gt;</code>
HtmlImage	<code>&lt;img runat="server"&gt;</code>
HtmlInputButton	<code>&lt;input type="submit" runat="server"&gt;</code> <code>&lt;input type="reset" runat="server"&gt;</code> <code>&lt;input type="button" runat="server"&gt;</code>
HtmlInputCheckBox	<code>&lt;input type="checkbox" runat="server"&gt;</code>
HtmlInputFile	<code>&lt;input type="file" runat="server"&gt;</code>

HtmlInputHidden	<input type="hidden" runat="server">
HtmlInputImage	<input type="image" runat="server">
HtmlInputRadioButton	<input type="radio" runat="server">
HtmlInputText	<input type="text" runat="server"> <input type="password" runat="server">
HtmlSelect	<select runat="server">
HtmlTable	<table runat="server">
HtmlTableRow	<tr runat="server">
HtmlTableCell	<td runat="server"> <th runat="server">
HtmlTextArea	<textarea runat="server">
HtmlGenericControl	<span runat="server"> <div runat="server"> Semua tag HTML lainnya

Semua class HTML Server Control terdapat pada namespace **System.Web.UI.HtmlControls**.

### *Menggunakan HTML Server Control*

Kita akan membuat halaman web yang berisi **HtmlForm**, **HtmlButton**, **HtmlInputText** dan **HtmlSelect**. Ketiklah script dibawah ini dan simpan dengan nama **buku.aspx**

```

buku.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Using ASP.NET Server Control</title>
</head>
<body>
<form id="Form1" runat="server">
<h2>Data Buku</h2>
<p>
ID Buku: <input type="text" id="id_buku" runat="server" />
</p>
<p>
Judul: <input type="text" id="judul" runat="server" />
</p>
<p> Jenis Buku<br />
<select id="jenis" runat="server" multiple="false">
<option>Komputer</option>
<option>Agama </option>
<option>Novel</option>
<option>Bisnis</option>
<option>Anak-anak</option>
</select>
</p>

```

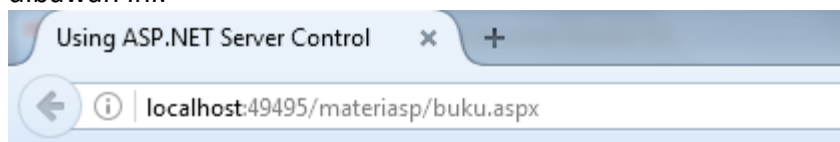


```

<p>
  Penerbit: <input type="text" id="penerbit" runat="server" />
</p>
<p>
  Tahun cetak: <input type="text" id="tahun" runat="server" />
</p>
<p>
  Pengarang: <input type="text" id="pengarang" runat="server" />
</p>
<p>
  Keterangan: <input type="text" id="keterangan" runat="server" />
</p>
<p>
  <button id="confirmButton" runat="server">Confirm</button>
</p>
<p>
  <asp:Label id="messageLabel" runat="server" />
</p>
</form>
</body>
</html>

```

Jika file **buku.aspx** dipanggil melalui browser maka hasilnya akan tampak seperti gambar dibawah ini:



## Data Buku

ID Buku:

Judul:

Jenis Buku

Penerbit:

Tahun cetak:

Pengarang:

Keterangan:

Tombol **Confirm** digunakan untuk menampilkan inputan yang telah diberikan oleh pengguna melalui element `messageLabel.Text`, agar tombol ini berfungsi maka tambahkan atribut `OnServerClick="Click"` pada element button sehingga lengkapnya menjadi seperti dibawah ini:

```
<button id="confirmButton" OnServerClick="Click" runat="server">Confirm</button>
```

dan sisipkan script dibawah ini pada file buku.aspx dibagian **head**:

```
<head>
<title> Using ASP.NET Server Control</title>
<script runat="server" language="C#">
void Click(Object s, EventArgs e)
{
    messageLabel.Text = "Data Buku dengan ID: " + id_buku.Value + "<br />";
    messageLabel.Text += "Judul: " + judul.Value + "<br />";
    messageLabel.Text += "Jenis Buku: ";
    for (int i = 0; i <= jenis.Items.Count - 1; i++) {
        if (jenis.Items[i].Selected) {
            messageLabel.Text += jenis.Items[i].Text + "<br />";
        }
    }
    messageLabel.Text += "Penerbit: " + penerbit.Value + "<br />";
    messageLabel.Text += "Tahun Cetak: " + tahun.Value + "<br />";
    messageLabel.Text += "Pengarang: " + pengarang.Value + "<br />";
    messageLabel.Text += "Keterangan: " + keterangan.Value + "<br />";
}
</script>
</head>
```

Ketika tombol **Confirm** diklik maka akan tampil informasi yang diinputkan pada form.

## Data Buku

ID Buku:

Judul:

Jenis Buku

Penerbit:

Tahun cetak:

Pengarang:

Keterangan:

Data Buku dengan ID: 01  
 Judul: Laskar pelangi  
 Jenis Buku: Novel  
 Penerbit: Erlangga  
 Tahun Cetak: 2015  
 Pengarang: Andi Hirata  
 Keterangan: Novel

## Web Server Control

Web Server Control dapat dikatakan sebagai HTML Server Control versi mahir. Web Server Control akan secara otomatis meng-generate tag HTML sehingga Anda tidak perlu lagi menggunakan tag-tag HTML secara standar, namun jika Anda telah mengenal dengan baik tag-tag HTML maka tidak akan menemukan kesulitan yang berarti ketika menggunakan Web Server Control.

Perhatikan contoh dibawah ini, misal tag HTML untuk input text:

```
<input type="text" name="usernameTextBox" size="30" />
```

Dalam Web Server Control perintahnya menjadi:

```
<asp:TextBox id="usernameTextBox" runat="server" Columns="30">
</asp:TextBox>
```

Perlu diingat, tidak seperti tag HTML standar yang biasa digunakan pada Web Form, web server control diproses pertama kali oleh engine ASP.NET dengan mentransformasikan kedalam bentuk HTML (client side), sehingga Anda perlu memastikan setiap element harus ditutup dengan tag </asp:TextBox> jika tidak maka proses akan mengalami kegagalan sejak pertama kali diproses. Cara lain untuk menutup tag web server control dengan menambahkan garis miring ' / ' diakhir tag element tersebut. Sehingga tag web server control diatas akan terlihat seperti dibawah ini:

```
<asp:TextBox id="usernameTextBox" runat="server" Columns="30" />
```

Untuk memudahkan pada saat menggunakan web server control ada baiknya memperhatikan hal-hal berikut:

- Web server control harus ditempatkan didalam tag <form runat="server"> agar dapat berfungsi dengan baik
- Web server control harus memiliki atribut runat="server"
- Selalu menggunakan prefik asp: setiap menambahkan element web server control

## Standard Web Server Controls

ASP.NET telah memberikan standarisasi terhadap Web Server Control yang dapat dilihat pada .NET Framework 2.0 SDK Documentation. Berikut ini akan dijelaskan beberapa web server control:

### Label

Cara mudah untuk menampilkan teks statis pada halaman web dan dapat juga untuk memodifikasi nilainya melalui script ASP.NET.

```
<asp:Label id="messageLabel" Text="" runat="server" />
```

Untuk menambahkan nilai pada label diatas dapat dilakukan dengan cara:

```
public void Page_Load() {  
    messageLabel.Text = "Hello World";  
}
```

## TextBox

TextBox digunakan untuk membuat inputan yang dapat digunakan pengguna untuk mengetikkan teks atau menampilkan teks standar. Gunakan properti TextMode untuk mengatur apakah isinya akan ditampilkan dalam baris tunggal atau banyak baris, selain itu dapat juga digunakan untuk menyembunyikan teks yang diinput seperti pada inputan password. Perhatikan beberapa contoh penggunaannya dibawah ini.

```
<p>  
Username: <asp:TextBox id="userTextBox" TextMode="SingleLine"  
Columns="30" runat="server" />  
</p>  
<p>  
Password: <asp:TextBox id="passwordTextBox"  
TextMode="Password" Columns="30" runat="server" />  
</p>  
<p>  
Comments: <asp:TextBox id="commentsTextBox"  
TextMode="MultiLine" Columns="30" Rows="10"  
runat="server" />  
</p>
```

## HiddenField

Element ini sering digunakan untuk inputan default yang tidak akan tampil pada halaman web, namun isinya sangat berpengaruh pada jalannya proses aplikasi. Hanya properti value yang dapat kita atur nilainya pada element ini.

```
<asp:HiddenField id="txtHidden" value="hello word" runat="server" />
```

Untuk mengakses nilainya dapat dilakukan dengan cara:

```
messageLabel.Text = txtHidden.Value;
```

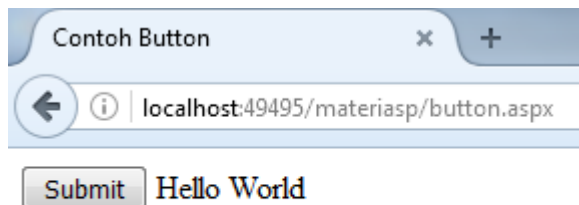
## Button

Secara default, button akan bersifat Submit dimana ketika Button diklik semua isi form dikirimkan ke server untuk diproses lebih lanjut. Contoh:

Nama file: **button.aspx**

```
button.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Contoh Button</title>
<script runat="server" language="C#">
public void WriteText(Object s, EventArgs e) {
    messageLabel.Text = "Hello World";
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Button id="submitButton" Text="Submit" runat="server" OnClick="WriteText" />
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Ketika tombol **Submit** diklik maka akan tampil seperti gambar dibawah ini.



## ImageButton

ImageButton berfungsi sama seperti Button hanya pada ImageButton menggunakan gambar.

Contoh: file name: **imagebutton.aspx**

```
imagebutton.aspx X
Server Objects & Events (No Events)
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DI
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Contoh Image Button</title>
<script runat="server" language="C#">
public void WriteText(Object s, ImageClickEventArgs e) {
    messageLabel.Text = "Coordinate: " + e.X + "," + e.Y;
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:ImageButton id="myImgButton" ImageUrl="feryup.png" runat="server" OnClick="WriteText" />
<asp:Label id="messageLabel" runat="server" />
</form>
</body>
</html>
```

Akan tampil sebagai berikut:



## LinkButton

LinkButton memiliki fungsi yang sama dengan button. Contoh:

```
<asp:LinkButton id="myLinkButon" Text="Click Here" runat="server" />
```

## HyperLink

Element ini akan menampilkan hyperlink yang akan mengarah pada suatu alamat URL sesuai dengan yang telah ditentukan pada properti `navigateUrl`. Tidak seperti LinkButton yang memiliki banyak fitur seperti event Click dan validitas, HyperLink hanya mengarahkan navigasi dari satu halaman kehalaman lain. Contoh:

```
<asp:HyperLink id="myLink" NavigateUrl="http://www.stmikbina.ac.id/"  
ImageUrl="slogo.gif" runat="server">STMIK BINA</asp:HyperLink>
```

## CheckBox

CheckBox dapat Anda gunakan untuk menampilkan pilihan yang hanya memiliki dua kondisi, checked (terpilih) atau unchecked (tidak terpilih).

```
<asp:CheckBox id="surveyCek" Text="Ya, saya suka ASP.NET"
Checked="True" runat="server" />
```

Event utama pada control ini adalah CheckChanged yang dapat dikondisikan melalui atribut onCheckChanged. Properti checked akan bernilai True jika objek di ceklis dan jika tidak di ceklis maka properti checked bernilai False.

## RadioButton

RadioButton merupakan pilihan yang dapat di grup dimana pilihannya hanya boleh satu dari beberapa pilihan yang ada. RadioButton di grup dengan memberikan nilai yang sama pada properti GroupName. Contoh:

```
<asp:RadioButton id="mi" GroupName="jurusan" Text="Manajemen Informatika" runat="server" /><br />
<asp:RadioButton id="ti" GroupName="jurusan" Text="Teknik Informatika" runat="server" /><br />
<asp:RadioButton id="si" GroupName="jurusan" Text="Sistem Informasi" runat="server" /><br />
```

## Image

Control ini akan menampilkan gambar seperti pada HTML pada umumnya, hanya saja pada control ini atribut dan propertisnya dapat dimanipulasi melalui skrip ASP.NET. contoh:

```
<asp:Image id="myImage" ImageUrl="mygif.gif" runat="server"
AlternateText="description" />
```

## Placeholder

Placeholder dapat kita gunakan untuk menambahkan element pada halaman web kapanpun secara dinamis melalui skrip ASP.NET. contoh:

Nama file: **placeholder.aspx**

```
placeholder.aspx X
Server Objects & Events
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Contoh Placeholder</title>
<script runat="server" language="C#">
public void Page_Load()
{
HtmlButton myButton = new HtmlButton();
myButton.InnerText = "My New Button";
placeholder.Controls.Add(myButton);
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Placeholder id="placeholder" runat="server" />
</form>
</body>
</html>
```

## Panel

Panel berfungsi seperti div pada HTML, didalamnya dapat dimanipulasi sebagai grup. Contoh dibawah ini akan menunjukkan bahwa panel dapat menampilkan dan menyembunyikan objek melalui event Click.

Nama file: **panel.aspx**

```
panel.aspx X
Server Objects & Events (No Events)
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tr
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Contoh Panel</title>
<script runat="server" language="C#">
public void HidePanel(Object s, EventArgs e) {
myPanel.Visible = false;
}
</script>
</head>
<body>
<form id="Form1" runat="server">
<asp:Panel id="myPanel" runat="server">
<p>
Username: <asp:TextBox id="usernameTextBox" Columns="30" runat="server" /></p>
<p>Password: <asp:TextBox id="passwordTextBox" TextMode="Password" Columns="30" runat="server" /> </p>
</asp:Panel>
<asp:Button id="hideButton" Text="Hide Panel" OnClick="HidePanel" runat="server" />
</form>
</body>
</html>
```

Hasilnya sebagai berikut:



## DropDownList

DropDownList mirip dengan tag select pada HTML. Element ini membolehkan pengguna untuk memilih melalui list menu. Contoh:

```
<asp:DropDownList id="warnaFav" runat="server">
<asp:ListItem Text="Merah" value="merah" />
<asp:ListItem Text="Biru" value="biru" />
<asp:ListItem Text="Hijau" value="hijau" />
</asp:DropDownList>
```

## CheckBox List

CheckBox List adalah element checkbox namun dalam group, sehingga penggunaannya lebih tepat untuk pilihan yang lebih dari satu. Contoh :

```
<asp:CheckBoxList ID="ekskul" runat="server">
<asp:ListItem Value="pramuka">Pramuka</asp:ListItem>
<asp:ListItem Value="paskibra">Paskibra</asp:ListItem>
<asp:ListItem Value="taekwondo">Tekwondo</asp:ListItem>
</asp:CheckBoxList>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
<asp:Label ID="msgLabel" runat="server" Text="Label"></asp:Label>
```

## CS script

```
msgLabel.Text = "";
for (int n = 0; n < ekskul.Items.Count; n++) {
    if (ekskul.Items[n].Selected) {
        msgLabel.Text += ekskul.Items[n].Text;
    }
}
//atau dengan cara dibawah ini
msgLabel.Text += "<br>";
foreach (ListItem item in ekskul.Items) {
    if (item.Selected)
        msgLabel.Text += item.Text + "<br>";
}
```

## ListBox

ListBox mirip dengan tag select pada HTML dengan multiselect dan size untuk menampilkan kotak pilihan lebih dari satu baris. Jika atribut SelectMode diisi multiselect maka pengguna dapat memilih lebih dari satu dari pilihan yang tersedia. Contoh:

```
<asp:ListBox id="listTechnologies" runat="server"
SelectionMode="Multiple">
<asp:ListItem Text="ASP.NET" Value="aspnet" />
<asp:ListItem Text="JSP" Value="jsp" />
<asp:ListItem Text="PHP" Value="php" />
<asp:ListItem Text="CGI" Value="cgi" />
<asp:ListItem Text="ColdFusion" Value="cf" />
</asp:ListBox>
```

## Latihan:

Buatkan tampilan form sebagai berikut:

### Formulir Pendaftaran

Nama	<input type="text" value="Fery"/>
Alamat	<input type="text" value="Cimone"/>
Agama	<input type="text" value="Islam"/>
Materi Kursus	<input type="text" value="ASP"/>

Nama: Fery  
Alamat: Cimone  
Agama: Islam  
Materi Kursus: ASP

ASP.NET merupakan framework pengembangan untuk membangun halaman web dan situs web dengan HTML, CSS, JavaScript dan server scripting.

ASP.NET mendukung tiga model pengembangan yang berbeda:

1. Halaman web
2. MVC (Model View Controller)
3. Formulir web.

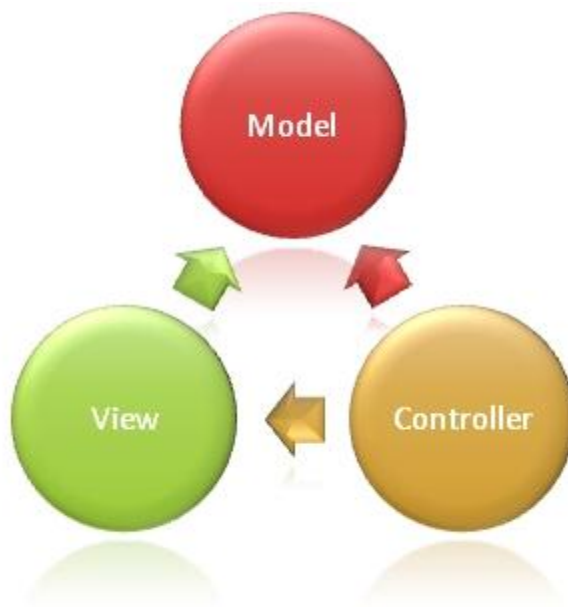
## MVC Programming Model

MVC adalah salah satu dari tiga model pemrograman ASP.NET.

MVC adalah kerangka kerja untuk membangun aplikasi web menggunakan MVC (Model View Controller) desain:

- Model ini merupakan inti aplikasi (misalnya daftar catatan database).
- The View menampilkan data (database catatan).
- Controller menangani input (dengan catatan database).

Model MVC juga menyediakan kontrol penuh atas HTML, CSS, dan JavaScript.



Model MVC mendefinisikan web aplikasi dengan 3 lapisan logika:

- Lapisan bisnis (**Model** logika)
- Layar lapisan (**View** logika)
- Input kontrol (**Controller** logika)

### Model

**Model** adalah bagian dari aplikasi yang menangani logika untuk data aplikasi. Seringkali benda Model mengambil data (dan data store) dari database.

### View

**View** adalah bagian dari aplikasi yang menangani tampilan data. Paling sering dilihat dibuat dari model data.

## Controller

**Controller** adalah bagian dari aplikasi yang menangani interaksi pengguna. Biasanya pengendali membaca data dari input pengguna tampilan, kontrol, dan mengirim data input ke model.

MVC pemisahan membantu Anda mengelola aplikasi yang kompleks, karena Anda bisa fokus pada satu aspek waktu. Misalnya, Anda dapat fokus pada pandangan tanpa tergantung pada logika bisnis. Hal ini juga membuat lebih mudah untuk menguji aplikasi.

MVC pemisahan juga menyederhanakan pengembangan kelompok. pengembang yang berbeda dapat bekerja pada tampilan, logika controller, dan logika bisnis secara paralel.

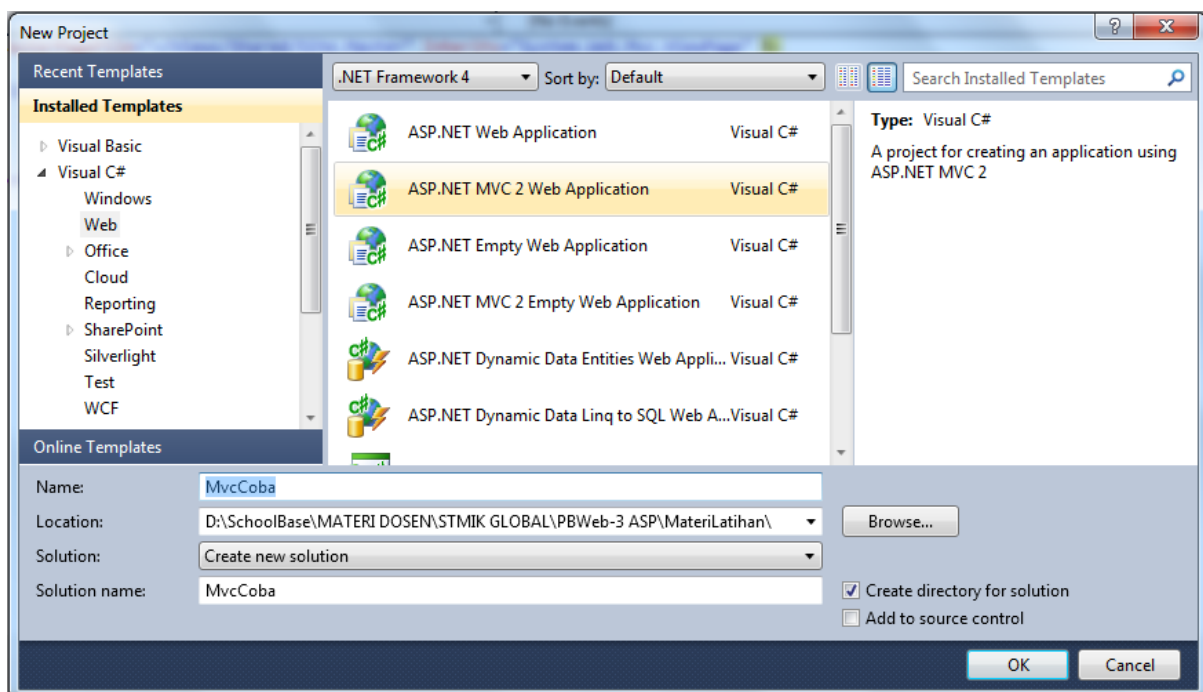
## ASP.NET MVC – Aplikasi Internet

Untuk mempelajari ASP.NET MVC, kita akan Membangun Aplikasi Internet

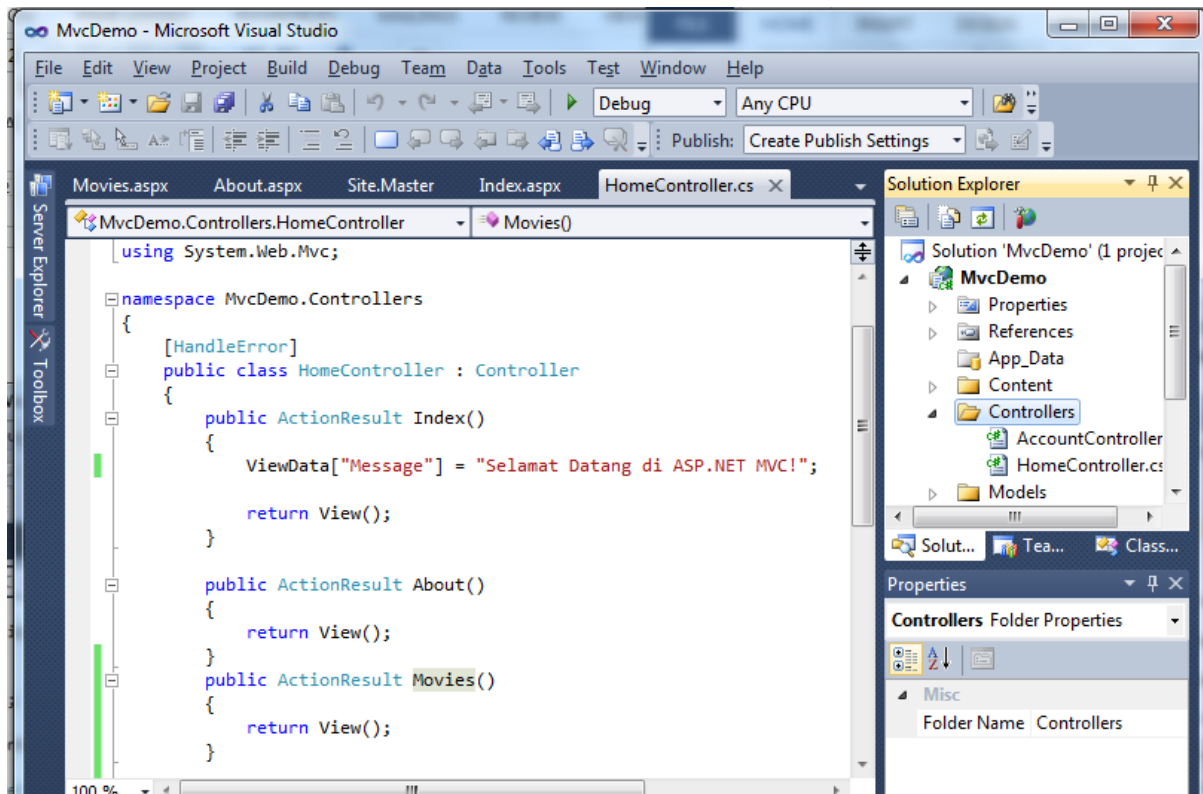
### 1. Membuat Aplikasi Web

membangun sebuah aplikasi internet yang mendukung penambahan, mengedit, menghapus, dan daftar informasi yang disimpan dalam database.

Langkah pertama Mulai Visual Studio 2010 dan pilih **New > Project > Visual C# > Web > pilih ASP.NET MVC 2 Web Application > Isikan Name MvcDemo> tentukan Location > klik OK.**



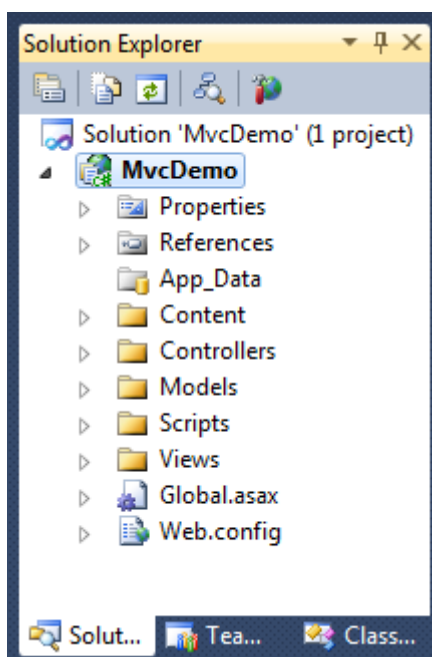
Visual Studio akan membuat sebuah proyek seperti dibawah ini:



## ASP.NET MVC – Aplikasi Folder

Sebuah aplikasi web ASP.NET MVC khas memiliki konten folder berikut:

### 2. Menjelajahi Aplikasi Folder



#### Informasi aplikasi

properti  
Referensi

#### folder aplikasi

App\_Data Folder  
Folder (Content) konten  
controller Folder  
model Folder  
script Folder  
tampilan(Views) Folder

#### file konfigurasi

global.asax  
web.config

Nama folder yang sama di semua aplikasi MVC. Kerangka MVC didasarkan pada standar penamaan. Pengendali dalam folder Controller, Pandangan dalam folder Views, dan Model berada di folder Models. Anda tidak harus menggunakan nama folder dalam kode aplikasi Anda.

Standar penamaan mengurangi jumlah kode, dan membuatnya lebih mudah bagi pengembang untuk memahami proyek MVC. Di bawah ini adalah ringkasan singkat dari isi setiap folder:

### **App\_Data Folder**

Folder **App\_Data** adalah untuk menyimpan data aplikasi. Contohnya penambahan database SQL ke folder App\_Data.

### **Content Folder**

Folder **Konten** digunakan untuk file statis seperti style sheets (file css), ikon dan gambar. Visual Web Developer juga menambahkan file style sheet standar untuk proyek: **Site.css** file dalam folder konten. File style sheet adalah file untuk mengedit ketika Anda ingin mengubah gaya aplikasi.

### **Controller Folder**

Folder Controller berisi kelas kontroler bertanggung jawab untuk menangani input pengguna dan tanggapan. MVC membutuhkan nama semua file controller untuk mengakhiri dengan "Pengendali". Visual Web Developer telah menciptakan Depan controller (untuk Home dan halaman Tentang) dan controller Akun (untuk halaman Login):

### **Model Folder**

Folder Model berisi kelas yang mewakili model aplikasi. Model memegang dan memanipulasi data aplikasi.

### **Views Folder**

Views folder menyimpan file HTML yang berhubungan dengan tampilan dari aplikasi (user interface). Folder Views berisi satu folder untuk setiap kontroler. Visual Web Developer telah menciptakan sebuah folder Akun, folder Home, dan folder bersama (di dalam folder Views). Folder Akun berisi halaman untuk mendaftar dan login ke akun pengguna. Folder Depan digunakan untuk menyimpan halaman aplikasi seperti halaman Home dan halaman About. folder Shared digunakan untuk menyimpan views dibagi antara pengendali (halaman master dan halaman tata letak).

### **Script Folder**

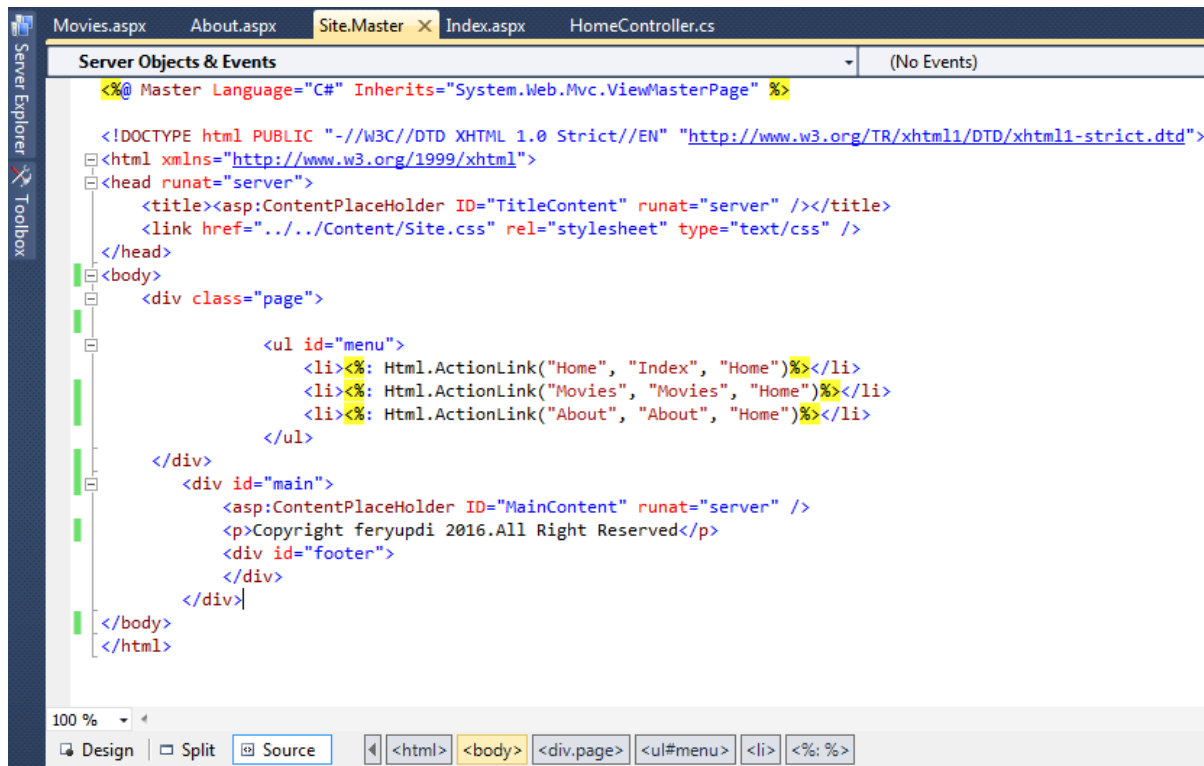
The Script folder menyimpan file JavaScript aplikasi. Secara default Visual Web Developer mengisi folder ini dengan standar MVC, Ajax, dan file jQuery:

## ASP.NET MVC – Styles dan Layout

### 3. Menambahkan Styles dan Layout

#### Menambahkan Layout

file **Site.Master** mewakili tata letak setiap halaman dalam aplikasi. Hal ini terletak di folder bersama di dalam folder Views, Shared



```
<%@ Master Language="C#" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>
  <link href="../../Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <div class="page">
    <ul id="menu">
      <li><%: Html.ActionLink("Home", "Index", "Home")%></li>
      <li><%: Html.ActionLink("Movies", "Movies", "Home")%></li>
      <li><%: Html.ActionLink("About", "About", "Home")%></li>
    </ul>
  </div>
  <div id="main">
    <asp:ContentPlaceHolder ID="MainContent" runat="server" />
    <p>Copyright feryupdi 2016.All Right Reserved</p>
    <div id="footer">
    </div>
  </div>
</body>
</html>
```

## ASP.NET MVC – Controller

### 4. Menambahkan Controller

#### Controller Folder

**Controller Folder** berisi kelas kontroler bertanggung jawab untuk menangani input pengguna dan tanggapan. MVC membutuhkan nama semua kontroler untuk mengakhiri dengan "Pengendali".

Dalam contoh kita, Visual Web Developer telah menciptakan file-file berikut: **HomeController.cs** (untuk Home dan About halaman) dan **AccountController.cs** (Untuk Masuk halaman):

server web biasanya akan memetakan permintaan URL yang masuk langsung ke file disk pada server. Sebagai contoh: permintaan URL seperti "http://www.w3ii.com/default.html" akan memetakan langsung ke file "default.html" di direktori root dari server.

Kerangka MVC peta berbeda. MVC peta URL ke metode. metode ini di kelas yang disebut "Controller".

Controller bertanggung jawab untuk memproses permintaan masuk, penanganan input, menyimpan data, dan mengirim respon untuk mengirim kembali ke klien.

## Home Controller

Controller file dalam **HomeController.cs** aplikasi kita, mendefinisikan **Index** dua kontrol dan **About**.

Menukar isi dari HomeController.cs dengan ini:



```
using System.Web.Mvc;

namespace MvcDemo.Controllers
{
    [HandleError]
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewData["Message"] = "Selamat Datang di ASP.NET MVC!";

            return View();
        }

        public ActionResult About()
        {
            return View();
        }

        public ActionResult Movies()
        {
            return View();
        }
    }
}
```

## Controller Views

File-file **Index.cshtml** dan **About.cshtml** dalam folder Views mendefinisikan pandangan ActionResult Index () dan About () di controller.

## Views Folder

**Views** folder menyimpan file (file HTML) yang terkait dengan tampilan dari aplikasi (user interface). file-file ini memiliki ekstensi html, asp, aspx, cshtml, dan vbhtml, tergantung pada isi bahasa. Folder Views berisi satu folder untuk setiap kontroler.

Visual Web Developer telah menciptakan sebuah folder Akun, folder Home, dan folder bersama (di dalam folder Views).

Folder Akun berisi halaman untuk mendaftar dan login ke akun pengguna.

View Home folder halaman aplikasi seperti halaman index (home page) dan halaman About. folder Shared digunakan untuk menyimpan views dibagi antara pengendali (halaman master dan halaman tata letak).



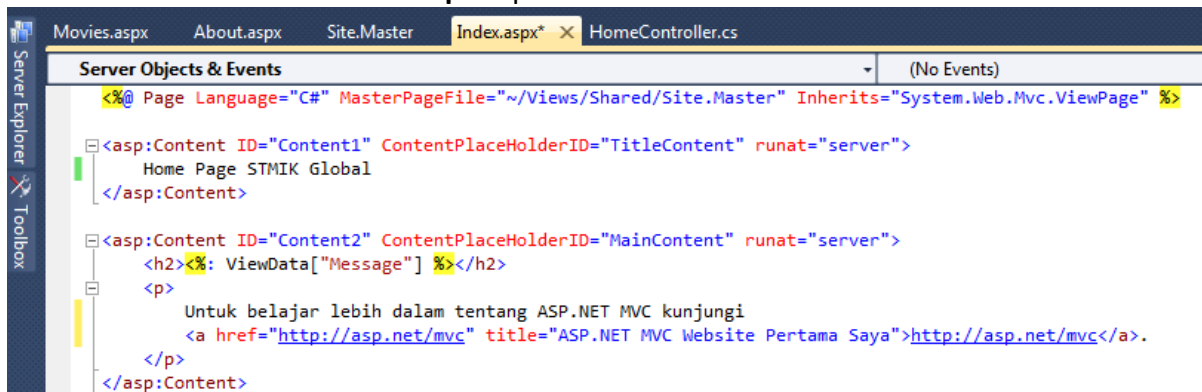
## Jenis File ASP.NET

Berikut jenis file HTML dapat ditemukan di Folder Views:

File Type	Extention
Plain HTML	.htm or .html
Classic ASP	.asp
Classic ASP.NET	.aspx
ASP.NET Razor C#	.cshtml
ASP.NET Razor VB	.vbhtml

## Latihan

- Tambahkan halaman **index.aspx** seperti dibawah ini:



```
<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">
    Home Page STMIK Global
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <h2><%: ViewData["Message"] %></h2>
    <p>
        Untuk belajar lebih dalam tentang ASP.NET MVC kunjungi
        <a href="http://asp.net/mvc" title="ASP.NET MVC Website Pertama Saya">http://asp.net/mvc</a>.
    </p>
</asp:Content>
```

- Tambahkan skrip program di halaman **HomeController.cs** seperti dibawah ini:



```
using System.Web.Mvc;

namespace MvcDemo.Controllers
{
    [HandleError]
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewData["Message"] = "Selamat Datang di ASP.NET MVC!";

            return View();
        }

        public ActionResult About()
        {
            return View();
        }

        public ActionResult Movies()
        {
            return View();
        }
    }
}
```

- Rubahlah kode program **Site.Master** seperti dibawah ini:

```

<%@ Master Language="C#" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>
<link href="../../../Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="page">
<ul id="menu">
<li><%: Html.ActionLink("Home", "Index", "Home")%></li>
<li><%: Html.ActionLink("Movies", "Movies", "Home")%></li>
<li><%: Html.ActionLink("About", "About", "Home")%></li>
</ul>
</div>
<div id="main">
<asp:ContentPlaceHolder ID="MainContent" runat="server" />
<p>Copyright feryupdi 2016.All Right Reserved</p>
<div id="footer">
</div>
</div>
</body>
</html>

```

- Tambahkan halaman **Movies.aspx** untuk menghubungkan link menu Movies

```

<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage" %>

<asp:Content ID="moviesTitle" ContentPlaceHolderID="TitleContent" runat="server">
    Movies
</asp:Content>

<asp:Content ID="moviesContent" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Movies</h2>
    <p>
        Film Terbaru
    </p>
</asp:Content>

```

Hasil layoutnya sebagai berikut:

