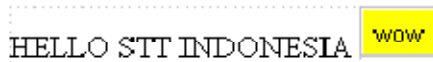


## Mengenal *visual designer* dan IntelliSense

Sejak dahulu, keunggulan Visual Studio dibandingkan *tools* lainnya adalah adanya fitur *visual designer*, yaitu developer melakukan *editing* secara *visual*, dan di balik itu Visual Studio akan *generate* code yang sesuai dengan tampilan *visual* yang dibuat oleh *developer*. Dengan demikian, *developer* tidak perlu menulis *code-code* yang sudah sangat umum. Misalnya, menambahkan suatu tombol pada halaman *website*. *Developer* hanya perlu melakukan *drag-and-drop* serta mengubah tampilan tombol melalui *window* Properties sambil dapat melihat secara langsung perubahan tampilan yang terjadi pada tombol. **Ikuti Langkah Berikut**

### Menambahkan tombol pada halaman *website*

1. Buka Visual Studio 2008, pilih **Open: Web Site...**
2. Bukalah *website* yang terakhir kali Anda buat, yaitu WebASPNET35



HELLO STT INDONESIA Wow

### Penjelasan

Dengan kemampuan visual editing, Visual Studio menghemat waktu dan tenaga dalam membuat dan mengolah *control* pada suatu halaman *website* tanpa perlu mengubah code-nya secara langsung. Dengan mengubah properti *BackColor* menjadi *Yellow* dan *Text* menjadi *Wow*, maka Visual Studio secara otomatis *generate* code berikut:

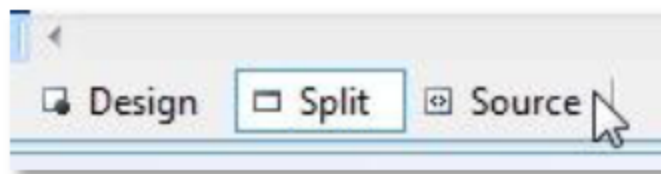
```
<div>  
  
HELLO STT INDONESIA  
    <asp:Button ID="Button1" runat="server" Text="wow" BackColor =Yellow /><p>  
  
</div>
```

Selanjutnya, kita akan memanfaatkan fitur IntelliSense, baik untuk code ASP.NET, C#, bahkan sampai JavaScript.

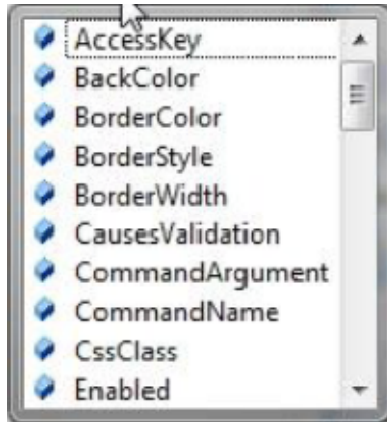
### Ikuti Langkah Berikut

#### Menambahkan *event handler* pada tombol

1. Buka kembali halaman *Default.aspx* pada *website* WebASPNET35 dengan menggunakan Visual Studio 2008.
2. Gunakan mode *Source View* dengan mengklik pilihan *Source*



3. Tambahkan properti Width pada Button1, dengan menekan spasi pada control Button1. Ketika menekan spasi, otomatis Visual Studio akan menampilkan code yang dapat digunakan untuk Button1.



4. Dari pilihan yang ada, pilih Width kemudian ketikkan tanda =

5. Isi properti Width dengan 100. Jangan lupa menambahkan tanda " dan " dalam mengisi suatu properti

```
Width = "100"
```

6. Sekarang *code* untuk Button1 adalah seperti berikut

```
<asp:Button ID="Button1" runat="server" Text="wow" BackColor  
=Yellow Width = "100" />
```

7. Kembali ke Default.aspx dan gunakan mode Source View

8. Tambahkan code JavaScript berikut di dalam tag <head></head> (Anda juga akan dibantu oleh IntelliSense dalam menambahkan *script* ini):

```
<script type="text/javascript"> function sayHi() { alert("Hi..!");  
} </script>
```

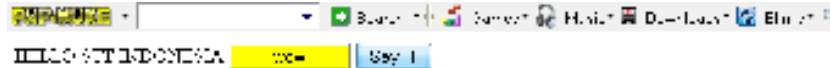
11. Di sebelah Button1, tambahkan sebuah HTML Button yang memanggil method sayHi jika diklik

```
<input type="button" value="Say Hi" onclick="sayHi()" />
```

12. Tes website Anda dengan menekan **Ctrl+F5**

Visual Studio 2008

14. Sekarang klik tombol bertuliskan Say Hi, maka akan muncul window alert yang menyapa Anda



## Penjelasan

Setiap kali Anda mengetikkan sesuatu, Visual Studio akan memberikan Anda pilihan-pilihan yang sesuai dengan apa yang Anda akan ketikkan. Fitur ini disebut IntelliSense. Dengan adanya fitur ini, biasanya kita hanya akan mengetikkan sebuah atau beberapa huruf saja kemudian menekan **Enter** untuk kemudian akan otomatis dilengkapi oleh Visual Studio 2008.

Fitur IntelliSense berlaku untuk code ASP.NET pada *file* .aspx, untuk code C# pada *file* .cs, dan untuk code JavaScript, baik yang *diembed* di suatu halaman web maupun *script* yang berada di dalam *file* .js.

Pada *control* Button1, event OnClick dihandle oleh *method* Button1\_Click yang berada di *code behind* dari halaman Default.aspx, yaitu Default.aspx.cs. Di dalam *method* Button1\_Click, dipanggil *method* Response.Write("I was clicked") sehingga ketika Button1 diklik, akan muncul tulisan **I was clicked**.

Di *control* lainnya, yaitu sebuah HTML input yang bertipe *button*, event onclick dihandle oleh *method* sayHi yang ada di JavaScript. Di dalam *method* sayHi, dipanggil *method* alert("Hi..!"). Dengan demikian jika tombol ini diklik, akan muncul *window* yang bertuliskan **Hi..!**.

# AJAX

## Sekilas tentang AJAX

**Asynchronous JavaScript and XML** atau yang lebih dikenal dengan AJAX mulai bergaung sejak tahun 2005. AJAX bukanlah suatu bahasa pemrograman, bukan pula merupakan suatu *script*, akan tetapi merupakan suatu teknik baru untuk membuat aplikasi web yang lebih interaktif. Dengan menggunakan AJAX, JavaScript yang ada pada suatu halaman web dapat berkomunikasi langsung dengan *server* dengan menggunakan objek XMLHttpRequest. Nah, dengan objek ini, JavaScript dapat mengambil dan mengirimkan data ke *server* tanpa perlu melakukan *reload* atau *refresh* pada halaman web. Mengapa? Karena AJAX melakukan transfer data (*request* Http) secara *asynchronous* antara browser dengan web server sehingga memungkinkan untuk mengambil sebagian data dari server tanpa perlu mengambil semua isi halaman web tersebut dari ulang. Dengan demikian, aplikasi web menjadi lebih cepat dan lebih user-friendly. Dan

untungnya AJAX tidak tergantung dari *web server* tempat AJAX tersebut berada, karena AJAX menggunakan protokol Http dan transfer data melalui request Http. Perlu diketahui bahwa objek XMLHttpRequest hanya didukung oleh browser-browser masa kini, yaitu:

- ✓ Internet Explorer 5 dan yang lebih baru
- ✓ Safari 1.2 dan yang lebih baru
- ✓ Mozilla Firefox 1.0 dan yang lebih baru
- ✓ Opera 8 dan yang lebih baru
- ✓ Netscape 7 dan yang lebih baru

## AJAX pada ASP.NET

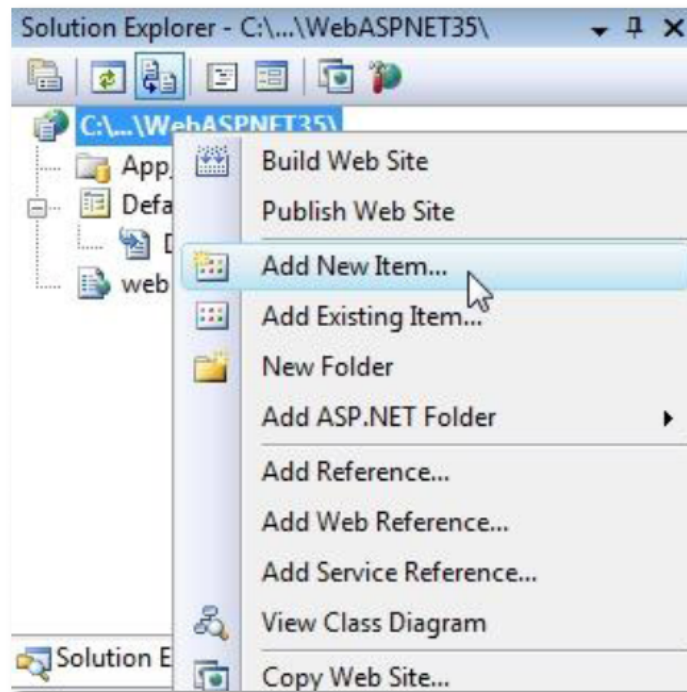
Ketika masih menggunakan framework .NET 2.0, AJAX merupakan “benda asing” sehingga perlu diinstal sebagai suatu “tambahan” pada ASP.NET 2.0. Sekarang, framework .NET 3.5 sangat memudahkan kita dalam menggunakan AJAX di ASP.NET karena AJAX Extensions sudah merupakan bagian dari .NET framework. Control-control yang memungkinkan penggunaan AJAX pun sudah dimasukkan ke dalam Toolbox, yaitu:

1. **ScriptManager**. Berfungsi untuk mengatur *client script* untuk halaman *web* yang menggunakan AJAX.
2. **ScriptManagerProxy**. Memungkinkan komponen yang *nested* seperti *content page* dan *user control* untuk menggunakan AJAX di mana komponen yang menjadi *parent* sudah memiliki ScriptManager.
3. **Timer**. Berfungsi untuk melakukan *postback* dalam setiap kurun waktu tertentu.
4. **UpdatePanel**. Berfungsi sebagai tempat di mana AJAX beraksi. Hanya bagian yang di dalam UpdatePanel yang akan di-reload oleh web server. Bagian di luar UpdatePanel tidak akan mengalami *reload*.
5. **UpdateProgress**. Berfungsi untuk menyediakan informasi status bagian dari halaman web yang sedang di-*update* dalam suatu *control* UpdatePanel.

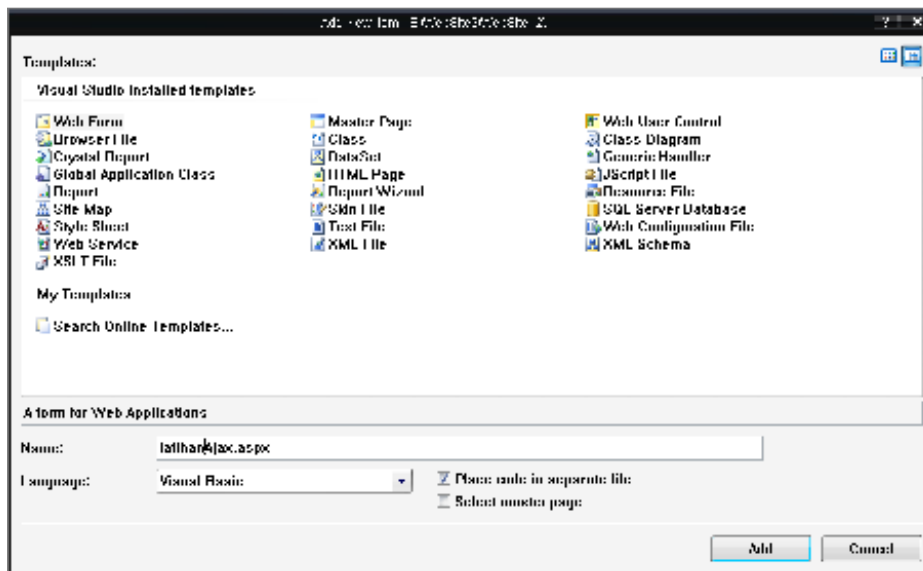
### Ikuti Langkah Berikut

#### Membuat halaman web ASP.NET yang menggunakan AJAX

1. Buka kembali *website* WebASPNET35.
2. Buatlah sebuah halaman *web* baru. Caranya yaitu klik kanan root dari website pada Solution Explorer, kemudian pilih Add New Item...



3. Pilih Web Form pada pilihan Templates. Beri nama LatihanAjax.aspx. Pilih Visual Basic sebagai Language dan jangan lupa untuk memilih check box “Place code in separate file” agar otomatis dibuatkan *code behind* untuk halaman ini oleh Visual Studio.



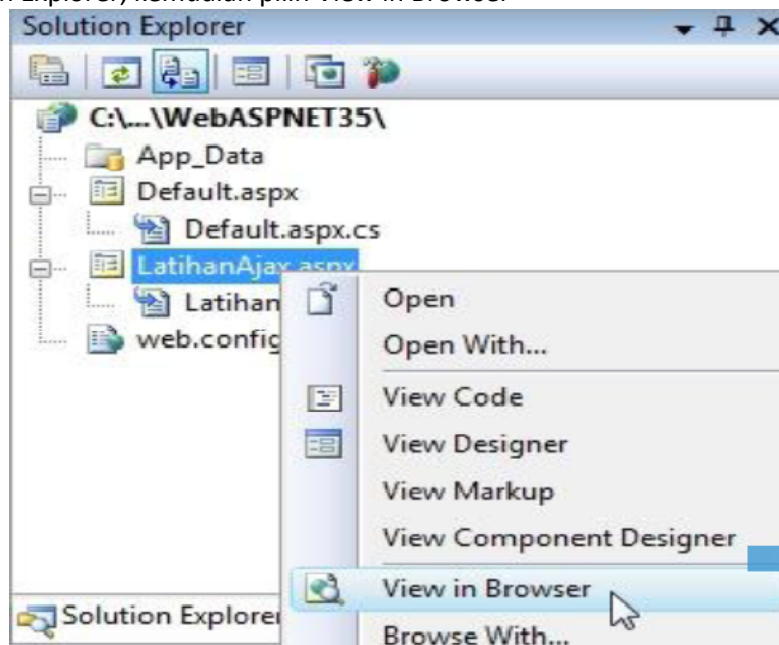
4. Di dalam elemen div, tambahkan sebuah TextBox tanpa tulisan, sebuah Button dengan tulisan “Set Text” di samping TextBox dan sebuah Label tanpa tulisan di bawah TextBox. Tambahkan Button1\_Click sebagai handle OnClick pada Button. Berikut adalah *code* yang terdapat pada elemen div:

```
<div>
  <asp:TextBox ID="TextBox1" runat="server">
  </asp:TextBox> <asp:Button ID="Button1" runat="server" Text="Set
Text" OnClick="Button1_Click" />
  <br /> <asp:Label ID="Label1" runat="server"></asp:Label>
</div>
```

5. Buka code behind LatihanAjax.aspx, yaitu LatihanAjax.aspx.cs. Tambahkan method Button1\_Click di dalamnya seperti berikut:

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = TextBox1.Text;
}
```

6. Jalankan halaman LatihanAjax.aspx pada browser dengan cara klik kanan LatihanAjax.aspx pada Solution Explorer, kemudian pilih View in Browser



7. Coba tuliskan suatu kata di text box, kemudian klik tombol "Set Text". Halaman akan di -postback ke server dan terjadi reload atau refresh yang ditandai dengan sempatnya muncul "layar putih" di browser.



8. Pada contoh di atas, kita belum menggunakan kemampuan AJAX. Sekarang kita akan gunakan AJAX agar dalam memanggil *method* Button1\_Click tidak perlu postback seluruh halaman web ke server sehingga tidak terjadi *refresh*. Buka kembali halaman LatihanAjax.aspx.
9. Tambahkan UpdatePanel dan isi UpdatePanel tersebut dengan ContentTemplate. Pindahkan TextBox, Button serta Label yang sebelumnya ada ke dalam ContentTemplate. Berikut adalah code pada elemen div setelah langkah ini selesai

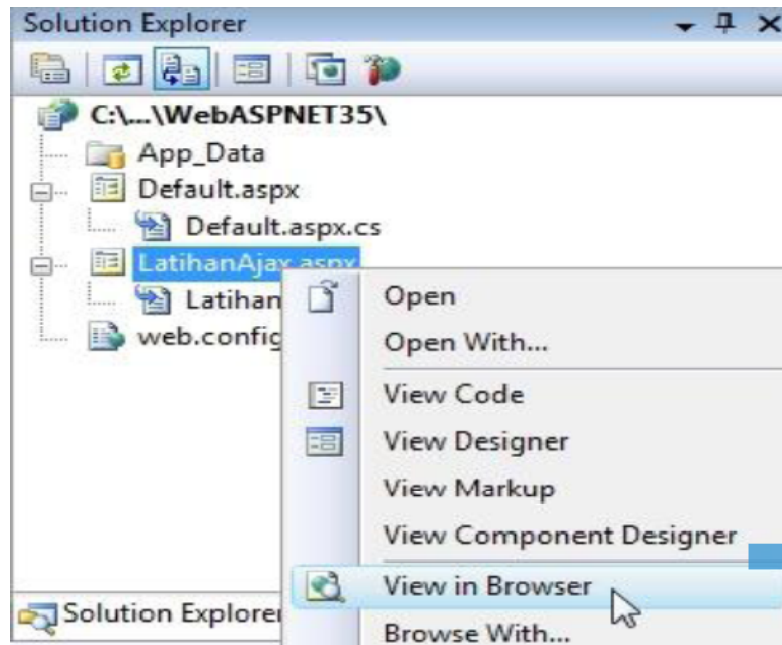
```
</div>

<asp:UpdatePanel ID="UpdatePanel1" runat="server">
  <ContentTemplate>
    <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    <asp:Button ID="Button2" runat="server"
      Text="Set Text" OnClick="Button1_Click" />
    <br />
    <asp:Label ID="Label2" runat="server"></asp:Label>
  </ContentTemplate>
</asp:UpdatePanel>
</div>
```

10. Tambahkan sebuah ScriptManager sebelum UpdatePanel sehingga code LatihanAjax.aspx menjadi seperti berikut

```
</div>
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<asp:UpdatePanel ID="UpdatePanel2" runat="server">
<ContentTemplate> <asp:TextBox ID="TextBox3" runat="server">
  </asp:TextBox> <asp:Button ID="Button3" runat="server"
    Text="Set Text" OnClick="Button1_Click" />
<br />
  <asp:Label ID="Label3" runat="server"></asp:Label>
</ContentTemplate> </asp:UpdatePanel>
</div>
```

12. Isi TextBox1 dengan suatu tulisan, kemudian klik Button1. Akan muncul tulisan pada Label1 sesuai dengan tulisan yang ada di TextBox1. Dan hebatnya, hal ini terjadi tanpa adanya refresh pada halaman web.



11. Untuk menjalankan LatihanAjax.aspx pada browser, klik kanan LatihanAjax.aspx pada Solution Explorer kemudian pilih View in Browser



### Penjelasan

Hingga langkah ke-7, halaman LatihanAjax.aspx merupakan halaman website ASP.NET biasa, karena halaman web tersebut “bersikap” seperti biasa, yaitu terjadi refresh ketika browser melakukan request ke web server.

Pada langkah ke-9, kita memasukkan semua *control* yang akan diberikan kemampuan untuk melakukan *request* ke *web server* secara *asynchronous* ke dalam *control* UpdatePanel. Dengan demikian, TextBox1, Button1, serta Label1 memiliki kemampuan AJAX, yaitu melakukan update dan terupdate tanpa perlu me-refresh seluruh halaman web.

Namun jangan lupa bahwa agar LatihanAjax.aspx diperlakukan secara AJAX, maka perlu ditambahkan sebuah ScriptManager pada halaman tersebut. ScriptManager ini harus muncul sebelum adanya UpdatePanel sehingga biasanya memang ditempatkan di awal halaman website.

Dengan adanya ScriptManager serta masuknya control-control ke dalam UpdatePanel, maka halaman LatihanAjax.aspx sudah diberikan kemampuan AJAX, yaitu melakukan *update* tanpa perlu melakukan *refresh*.



Ada kalanya update yang terjadi di belakang layar butuh waktu lama dan karena halaman web yang diberikan kemampuan AJAX tidak Nampak untuk refresh, maka sebaiknya pengunjung web perlu diberitahu bahwa proses update sedang dilakukan. Untuk kebutuhan ini, kita bisa menggunakan control UpdateProgress.