



**Soetam Rizky**

**Learning by Sample**

# **ASP.NET 3.5**



PRESTASI PUSTAKA  
PUBLISHER

Jakarta 2009  
PRESTASI PUSTAKARAYA  
Penerbit Buku Berkualitas Prima  
Copyright © 2009

**SOETAM RIZKY**  
**Learning by Sample**  
**ASP.NET 3.5**

Editor: **Tim Prestasi**  
Desain Cover: **Soetam Rizky**  
Setting: **Tim Prestasi**

Hak Penerbitan ada pada PT. Prestasi Pustakaraya – Jakarta

Hak cipta dilindungi undang-undang  
Dilarang mengutip, memperbanyak, dan menerjemahkan sebagian  
atau seluruh isi buku ini tanpa izin tertulis dari Penerbit Prestasi  
Pustakaraya – Jakarta - Indonesia – 2009

**PRESTASI PUSTAKA**

**SOETAM RIZKY**  
**Learning by Sample**  
**ASP.NET 3.5**

ISBN: 978-602-8470-35-3

Cetakan Pertama: Juli 2009

**Learning by sample**

**ASP .NET 3.5**

Soetam Rizky

〔 Untuk istri dan putri kecilku, pelipur jiwa di kala lara dan duka 〕

〔 Kekayaan adalah merasa bahagia dan bersyukur dengan apa yang telah kita miliki  
Kemiskinan adalah merasa sedih dan mengeluh dengan apa yang telah kita miliki 〕

## **Kata Pengantar**

Alhamdulillah, buku yang secara langsung membahas contoh aplikatif tentang pemrograman web dengan menggunakan ASP .NET ini akhirnya terselesaikan. Dan tentu saja ini adalah buku pertama yang ditulis sejak kepindahan domisili penulis ke kota Malang, yang dengan hawa sejuknya telah sangat banyak membantu dalam proses penulisan buku.

Tentu saja, banyak pihak yang secara tidak langsung ikut mendorong dan membantu saat buku ini disusun. Karenanya, penulis sangat berterima kasih kepada para rekan penulis yang tidak pernah bosan membakar semangat dalam menulis buku, Pak Yuswanto, Pak Sholiq yang selalu membantu secara moril dan juga materil dan Pak David yang tidak pernah bosan bertanya tentang naskah buku baru. Tak lupa para mantan mahasiswa di STIKOM Surabaya dan juga rekan lain yang tidak pernah bosan berkomentar di Facebook sehingga menjadikan penulis tetap tak pernah lekang dari semangat.

Semoga buku ini tidak hanya menjadi contoh instant dalam membangun sebuah situs dengan menggunakan ASP .NET, tetapi juga mampu menjadi

bahan referensi untuk pengembangan lebih lanjut bagi para akademisi.

Selamat berkarya !!!

Malang

Jumadil Akhir 1430 H / Juni 2009

# Daftar isi

<b>PENDAHULUAN .....</b>	<b>1</b>
<b>SIMPLE DOWNLOAD MANAGER .....</b>	<b>7</b>
PENGANTAR .....	9
STRUKTUR SITUS .....	10
STYLE SHEET : EXPLORER.CSS .....	12
MODIFIKASI WEB.CONFIG .....	16
HALAMAN UTAMA : DEFAULT.ASPX .....	18
TESTING DAN PENGEMBANGAN .....	38
<b>SIMPLE COMPANY PROFILE .....</b>	<b>41</b>
PENGANTAR .....	43
STRUKTUR SITUS .....	45
PEMBUATAN FILE TEKS .....	47
MODIFIKASI WEB.CONFIG .....	50
STYLE SHEET : COMPANYPROFILE.CSS.....	52
HALAMAN ADMINISTRATOR : ADMIN.ASPX.....	56
HALAMAN UTAMA : DEFAULT.ASPX .....	71
TESTING DAN PENGEMBANGAN .....	80
<b>ONLINE LOTTERY .....</b>	<b>83</b>
PENGANTAR .....	85
STYLE SHEET : LOTTERY.CSS .....	87
HALAMAN UTAMA : DEFAULT.ASPX .....	89
TESTING DAN PENGEMBANGAN .....	98
<b>PERSONAL REMINDER .....</b>	<b>101</b>
PENGANTAR .....	103
PEMBUATAN DATABASE : PERSONALREMINDER.MDF .....	105
HALAMAN UTAMA : DEFAULT.ASPX .....	109
TESTING DAN PENGEMBANGAN .....	139
<b>SLIDE SHOW GALLERY.....</b>	<b>145</b>
PENGANTAR .....	147

STRUKTUR SITUS .....	149
STYLE SHEET : <i>SLIDESHOW.CSS</i> .....	151
MASTER PAGE : <i>SLIDESHOW.MASTER</i> .....	156
STRUKTUR DATABASE DAN STORED PROCEDURE: <i>SLIDESHOW.MDF</i> .....	161
FILE JAVASCRIPT : <i>SHOWIMAGE.JS</i> .....	168
MENAMPILKAN THUMBNAIL : <i>THUMBNAIL.ASPX</i> .....	173
KUMPULAN FUNGSI AKSES DATABASE : <i>DATAACCESS.VB</i> .....	178
HALAMAN ADMINISTRATOR : <i>ADMIN.ASPX</i> .....	186
HALAMAN UTAMA : <i>DEFAULT.ASPX</i> .....	204
TESTING DAN PENGEMBANGAN .....	224
<b>IKLAN MINI.....</b>	<b>229</b>
PENGANTAR .....	231
STRUKTUR SITUS .....	234
PEMBUATAN DATABASE : <i>MINIADS.MDF</i> .....	235
STYLE SHEET : <i>MINIADS.CSS</i> .....	249
PEMBUATAN SITE MAP .....	253
MODIFIKASI WEB.CONFIG.....	257
PEMBUATAN CLASS (I) : <i>MISC.VB</i> .....	260
PEMBUATAN CLASS (II) : <i>DATAACCESS.VB</i> .....	263
MASTER PAGE : <i>MINIADS.MASTER</i> .....	270
HALAMAN REGISTRASI : <i>REGISTER.ASPX</i> .....	273
HALAMAN LOGIN : <i>LOGIN.ASPX</i> .....	286
FOLDER CMS – MASTER PAGE : <i>ADMIN.MASTER</i> .....	290
FOLDER CMS – HALAMAN UTAMA : <i>DEFAULT.ASPX</i> .....	294
FOLDER CMS – USER PROFILE : <i>USERPROFILE.ASPX</i> .....	299
FOLDER CMS – ADMINISTRASI USER : <i>MANAGEUSER.ASPX</i> .....	305
FOLDER CMS – SETTING SITUS : <i>SETTING.ASPX</i> .....	309
FOLDER CMS – PENGUMUMAN : <i>ANNOUNCEMENT.ASPX</i> .....	312
FOLDER CMS – KATEGORI IKLAN : <i>CATEGORY.ASPX</i> .....	320
FOLDER CMS – PASANG IKLAN : <i>NEWADS.ASPX</i> .....	332
FOLDER CMS – EDIT IKLAN : <i>EDITADS.ASPX</i> .....	343
HALAMAN UTAMA : <i>DEFAULT.ASPX</i> .....	347
TESTING DAN PENGEMBANGAN .....	366
<b>APPENDIX : PEMROGRAMAN WEB.....</b>	<b>370</b>
<b>APPENDIX : ASP .NET ? .....</b>	<b>374</b>



<b>APPENDIX : VWD EXPRESS VS VS PRO .....</b>	<b>376</b>
<b>APPENDIX : KONSEP DASAR .....</b>	<b>378</b>
<b>APPENDIX : KOMPONEN PENTING.....</b>	<b>382</b>
<b>APPENDIX : AKSES DATA .....</b>	<b>394</b>
<b>APPENDIX : ASP .NET AJAX.....</b>	<b>396</b>

# Pendahuluan

---

Dari sekian banyak buku yang telah terbit dari penulis, buku ini memang sedikit unik dari buku-buku yang sudah ada sebelumnya. Jika buku-buku dari penulis sebelumnya (khususnya yang bersifat programming) lebih banyak bercerita tentang teori pemrograman yang disertai dengan studi kasus, di buku ini yang terjadi adalah kebalikannya.

Dalam buku ini memang sengaja langsung menampilkan contoh-contoh situs yang telah dibagi berdasarkan level para pembaca, mulai dari level pemula hingga mahir. Sehingga nantinya diharapkan buku ini dapat menjangkau berbagai level programmer baik bagi pemula yang baru saja memulai belajar ASP .NET maupun yang sudah merasa di level mahir dan ingin mengembangkan teknik pemrogramannya secara lanjut.

Bagi Anda penggemar ebook ataupun buku-buku berbahasa asing untuk pemrograman, tentu melihat gaya penulisan yang sama dengan beberapa buku yang memiliki judul *recipe* atau *instant result*. Dan memang benar bahwa saat pertama kali menulis buku ini, penulis sangat terinspirasi dengan buku-buku jenis tersebut.

Karena dari buku-buku pemrograman yang langsung “menusuk” ke contoh pemrograman, penulis sangat merasakan manfaat langsung dan merasa lebih cepat paham tentang beberapa teknik pengembangan program. Sehingga gaya itulah yang dianut dalam penulisan buku ini.

Selain itu, penulis yakin bahwa masih banyak programmer (baik level pemula hingga mahir) yang umumnya malas membaca mengenai teori-teori dasar seperti pengenalan komponen dan lainnya. Tetapi mereka (mungkin juga termasuk Anda) jauh lebih suka belajar dengan langsung mempraktekkan ke dalam sebuah studi kasus atau seringkali disebut sebagai *learning by doing*.

Dan meski buku ini terinspirasi oleh buku-buku berbahasa asing, bukan berarti bahwa contoh yang ada dalam buku ini juga mengambil dari contoh-contoh yang banyak beredar di internet ataupun buku lain. Sehingga salah satu kelebihan utama dari buku ini adalah orisinalitas contoh yang benar-benar langsung dibuat oleh penulis. Karenanya, meski dibuat dengan teknik yang mungkin terasa naif (dengan menggunakan alibi bahwa semuanya dilakukan demi kesederhanaan pemahaman bagi pembaca), tetapi semua contoh di buku ini dijamin orisinal.

Meski demikian, dengan memperkenalkan berbagai variasi teknik pemecahan kasus dalam situs ini diharapkan pembaca dapat pula terinspirasi untuk memecahkan masalah yang mungkin sejenis untuk kasus yang lain. Tidak semua contoh situs yang ada dibuat dengan menggunakan teknik yang efisien. Sekali lagi, semua dilakukan hanya demi keragaman teknik yang dapat ditampilkan dan juga demi kemudahan penulisan dan juga saat pembaca berusaha mempraktekkan situs ini.

Semua contoh situs yang ada dalam buku ini memang dibuat dengan menggunakan Visual Web Developer Express Edition 2008, meski demikian seluruh contoh juga telah dites dengan baik saat dibuka dan dieksekusi menggunakan Visual Studio 2008. Tetapi beberapa situs juga sudah dites berjalan dengan baik dengan menggunakan Visual Web Developer Express Edition 2005, meski hal tersebut sangat tidak disarankan.

Dengan alasan terbatasnya ruang dan waktu, tidak semua fitur *advanced* dari ASP .NET 3.5 ditampilkan di buku ini. Sebagai contoh, adalah tiadanya kasus situs yang menggunakan akses database dengan LINQ ataupun tidak adanya contoh situs dengan menggunakan WPF dan Silverlight.

Hal tersebut sengaja dilakukan karena pada saat buku ini selesai ditulis, Microsoft telah mengeluarkan rilis preview untuk Visual Studio 2010 dan ASP .NET 4.0. Sehingga diharapkan fitur-fitur baru tersebut dapat dibahas lebih lanjut di buku selanjutnya yang memang khusus membahas tentang ASP .NET 4.0. Karena seluruh fitur tersebut nantinya jauh lebih terintegrasi dan lebih mudah untuk diimplementasikan di dalam Visual Studio 2010 dibandingkan menggunakan Visual Studio 2008.

Satu hal penting yang sangat perlu diingat sebelum mencoba contoh-contoh yang ada di situs ini adalah bahwa dalam kegiatan pemrograman tidak boleh ada kata *putus asa* dan kata *tidak bisa*. Karena sesungguhnya seorang programmer adalah orang-orang paling tangguh dalam menghadapi tantangan sekaligus orang yang paling sabar dalam menghadapi kesalahan. Karena semua tantangan dalam pemrograman pasti terdapat jawaban didalamnya, dan semua kesalahan pasti ada jalan keluarnya.

Selain itu, Anda bisa mencoba contoh-contoh yang ada di tiap bab tanpa harus memperdulikan urutan bab yang ada (bukan urutan sub bab). Sebagai misal, jika Anda telah merasa mahir dan langsung ingin mencoba contoh yang terakhir dari buku ini, maka

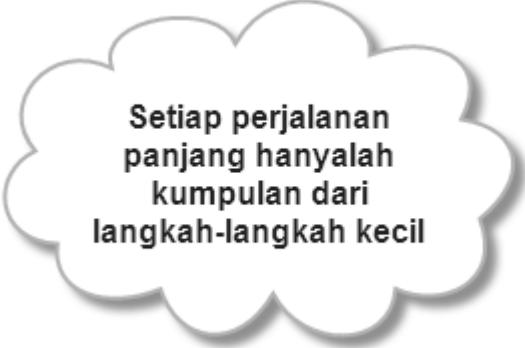
Anda tidak perlu harus melalui bab-bab awal yang ditujukan untuk pemula.

Akhir kata, selamat berkarya !!!

**Halaman ini sengaja dikosongkan**

# **Simple Download Manager**

---



**Setiap perjalanan  
panjang hanyalah  
kumpulan dari  
langkah-langkah kecil**



**Level :** Pemula

**Tujuan :**

Membuat situs download manager secara sederhana

**Fitur situs :**

- Melakukan pengaturan upload dan download file
- Memanfaatkan file web.config untuk menyimpan setting situs
- Injeksi script javascript ke dalam Gridview
- Memanfaatkan komponen Panel untuk fitur tambahan situs
- Proses login dengan model *single user*
- Modifikasi Gridview dengan field buatan sendiri tanpa menggunakan database
- Menggunakan prosedur buatan sendiri untuk "menangkap" event di dalam Gridview

**Estimasi waktu pembuatan :** 30 – 100 menit

## **Pengantar**

---

---

Contoh di bagian ini merupakan contoh sederhana yang dapat dikerjakan oleh pemula secara cepat. Salah satu faktor yang menyebabkan contoh situs ini dapat dikerjakan dengan cepat adalah kesederhanaannya. Hanya terdiri dari satu halaman web dan tidak menggunakan database sama sekali didalamnya adalah bukti dari kesederhanaan situs ini.

Situs yang akan dibuat hanya akan melakukan browsing terhadap sebuah folder yang didalamnya terdiri dari file-file yang sebelumnya telah diupload oleh administrator. Secara singkat, situs ini merupakan situs yang berfungsi sebagai download manager bagi user tanpa harus melakukan registrasi.

Bagi anda yang terbiasa melakukan download melalui situs-situs download seperti mediafire atau box.net, situs ini adalah kloning sederhana dari situs sejenis. Tetapi di dalam contoh ini, tidak terdapat registrasi yang tersedia untuk pengguna majemuk, hanya disediakan untuk satu administrator.

Diharapkan contoh situs ini dapat menjadi contoh yang cepat dan tepat khususnya bagi para pemula, tetapi juga dapat menjadi contoh singkat bagi para programmer dengan level menengah.

## **Struktur Situs**

---

---

Situs yang dibuat hanya terdiri dari bagian sederhana yaitu :

1. Default.aspx

Merupakan halaman utama yang terdiri dari sebuah Gridview dan dua buah panel. Gridview yang ada nantinya akan memiliki dua mode yaitu mode user dan mode administrator yang dapat melakukan modifikasi terhadap status file. Sedangkan panel yang ada nantinya digunakan untuk mode administrator dan hanya akan tampak jika administrator telah berhasil melakukan proses login.

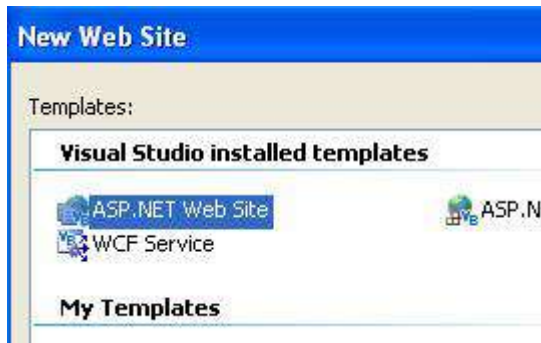
2. Explorer.css

Merupakan file style sheet yang hanya terdiri dari tiga class utama dan setting bagian body untuk halaman utama.

3. Web.config

Di dalam file konfigurasi ini nantinya dimodifikasi sebuah bagian yaitu appSetting yang akan diisi dengan password untuk administrator. Hal ini dilakukan karena hanya terdapat satu orang administrator yang diasumsikan dalam situs ini sehingga hanya dibutuhkan password, tidak lagi nama user.

Kini langkah pertama yang harus dilakukan dalam membuat situs ini adalah melakukan pembuatan web site baru di dalam IDE Visual Studio 2008 atau Visual Web Developer Express Edition 2008. Beri nama web site yang baru tersebut dengan nama *SimpleDownloadManager* atau nama lain sesuai selera.



Selanjutnya, dari web site yang dibuat telah terbentuk satu halaman utama, yang akan dimodifikasi di bagian berikutnya. Sedangkan untuk style sheet dijelaskan di sub bab setelah sub bab ini.

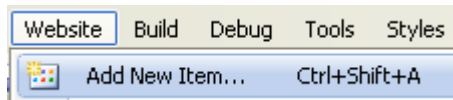
## **Style Sheet : Explorer.css**

---

Di dalam style sheet yang dirancang dalam situs ini menggunakan style sederhana yaitu satu kolom besar di tengah browser dengan menggunakan sebuah header sebagai judul situs dan sebuah footer untuk keterangan copyright. Bagian utama memiliki lebar *non stretch* atau tidak selebar ukuran layar.

Sedangkan font yang digunakan untuk keseluruhan situs adalah font Verdana dengan ukuran 10pt. Anda juga bisa mengubah style font ini sesuai selera masing-masing.

Kini, untuk langkah pembuatannya, klik pada menu *Website* -> *Add New Item* atau dengan menekan kombinasi tombol *Ctrl+Shift+A*.



Setelah itu pilih tipe file *Style Sheet* dan beri nama *Explorer.css* untuk file tersebut.



Kemudian, di dalam file style sheet yang tersedia, ketikkan listing berikut :

```
body
{
    font-family:Verdana;
    font-size:10pt;
}
.header
{
    font-size:14pt;
    font-weight:bold;
```

```
text-align:center;
padding-bottom:5px;
border-bottom:dashed 2px red;
margin-bottom:5px;
}
.utama
{
position:relative;
margin-left:auto;
margin-right:auto;
width:70%;
}
.footer
{
font-size:8pt;
font-weight:bold;
text-align:center;
margin-top:5px;
border-top:solid 1 px black;
}
```



---

Untuk mendapatkan posisi di tengah layar dengan lebar tertentu, perhatikan di class style sheet *utama* yang diatur dengan lebar 70%, tetapi margin kiri dan kanannya diset menjadi auto sehingga akan mendapatkan lebar

margin yang sama (dengan kata lain tepat di tengah layar berapapun resolusinya). Sedangkan untuk header dan footer hanya cukup dengan menambahkan garis batas (border) serta margin (atas/top untuk header dan bawah/bottom untuk footer) untuk memberi jarak.

---

---



## Modifikasi web.config

---

File web.config merupakan file konfigurasi yang tidak akan dapat terlihat ataupun terbaca oleh browser yang digunakan oleh pengguna. Meski terlihat aman, web.config seringkali masih ditambah dengan pengamanan tambahan seperti proses enkripsi. Di dalam contoh ini, file web.config tidak ditambah dengan pengamanan ekstra karena memang ditujukan untuk pemula.

Di dalam Solution Explorer cari terlebih dulu file web.config, lalu double klik untuk menampilkan isi dari file tersebut.



Modifikasi dari file ini terletak pada tag `<appSettings>`. Carilah terlebih dulu tag tersebut, kemudian modifikasi menjadi seperti listing berikut ini :

```
<appSettings>
  <add key="Password" value="222" />
</appSettings>
```



---

---

Atribut *key* merupakan pengenal yang akan diakses dalam listing program nanti, sedangkan atribut *value* adalah isi dari password itu sendiri yang nanti akan digunakan dalam proses login. Anda bisa mengganti isi dari atribut *value* sesuai selera.

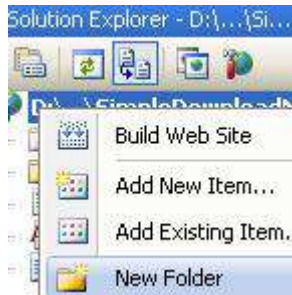
---

---

## **Halaman Utama : Default.aspx**

---

Kini untuk halaman utama, langkah pertama yang harus dilakukan adalah melakukan pembuatan folder yang akan menampung file hasil upload. Klik kanan di hirarki tree teratas pada Solution Explorer dan pilih sub menu *New Folder*.



Selanjutnya, beri nama folder baru tersebut dengan nama *DownloadFile* , pemberian nama ini jangan diganti terlebih dulu sesuai selera karena akan berpengaruh pada listing program nantinya.

Kemudian double klik pada file *Default.aspx* untuk melakukan modifikasi di halaman utama situs ini. Langkah pertama yang dilakukan adalah melakukan drag file style sheet ke bagian bawah tag `<title>...</title>`, sekaligus memodifikasi bagian tag tersebut. Hasil dari proses drag tersebut tampak pada listing berikut (juga bisa langsung diketikkan jika proses drag gagal dilakukan) :

```
<title>Simple Download Manager</title>
<link href="Explorer.css"
      rel="stylesheet"
      type="text/css" />
```

Kemudian ketikkan listing java script berikut tepat dibawahnya:

```
<script type="text/javascript" >
function Confirm() {
  if (confirm('Are you sure to delete it?')) {
    return true;
  }
  else {
    return false;
  }
}
</script>
```



---

Script tersebut nanti akan digunakan sebagai media konfirmasi saat proses penghapusan file akan dilakukan oleh administrator. Karena proses penghapusan sangat rentan dilakukan, maka diperlukan satu penghalang berupa kotak dialog konfirmasi. Selain itu, juga mendemonstrasikan cara melakukan

injeksi javascript ke dalam sebuah  
proses di dalam GridView.

---

---

Berikutnya, berpindahlah ke mode *Source*

 dan di bawah tag `<form...>`

ketikkan listing berikut untuk membentuk layout dari web site menggunakan style sheet yang sebelumnya telah didefinisikan.

```
<div class="header">
...: Simple Download Manager :...
</div>
<div class="utama">

</div>
<div class="footer">
<asp:LinkButton
    ID="LinkButton1" runat="server">
    Administrator</asp:LinkButton>
<br />
copyright @ Soetam Rizky 2009<br />
</div>
```

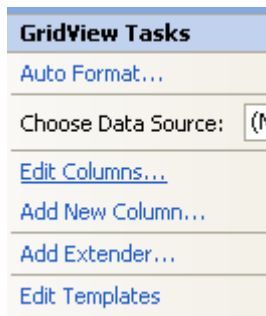


Perhatikan bahwa dalam layout utama terdapat tiga bagian utama yaitu *header* yang berisi judul, *utama* yang nanti akan berisi Gridview serta *footer* yang berisi keterangan copyright serta sebuah Linkbutton untuk menuju ke panel administrator.

Untuk melihat hasil layout yang telah diketikkan, berpindah ke mode *Design* maka akan terlihat layout seperti pada gambar berikut :

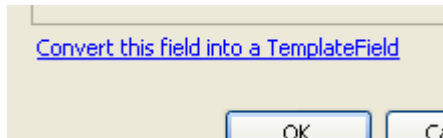


Selanjutnya, drag sebuah Gridview tepat di bagian class utama atau bagian kotak kosong yang terletak di tengah-tengah. Lalu klik pada smart tag di Gridview tersebut (smart tag adalah icon panah kecil yang umumnya terletak di bagian pojok kanan atas dari komponen), dan pilih sub menu *Edit Columns*.

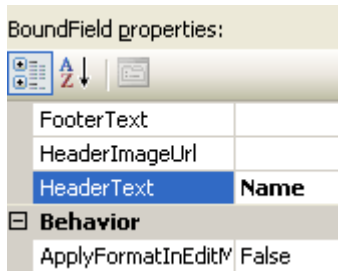
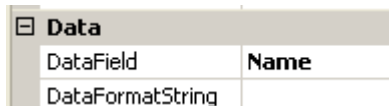


Di dalam kotak dialog yang tersedia, tambahkan field dengan melakukan klik di tombol *Add*. Tambahkan field-field berikut :

1. Field *No.*, klik pada link *Convert this field into template field* agar nanti dapat dimodifikasi di mode Source



2. Field *Name*, isikan *Name* di dalam property *DataField* dan property *HeaderText*



3. Field *File Size*, isikan *Length* di dalam property *DataField* dan *File Size* di property *Header Text*.
4. Field *Date Modified*, isikan *LastWriteTime* di dalam property *DataField* dan *Date Modified* di property *Header Text*.

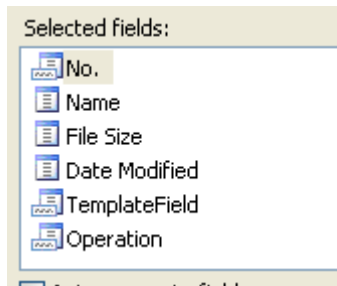
5. Sebuah Template Field yang nanti akan digunakan untuk menempatkan Linkbutton *Download*.



Set property *ShowHeader* menjadi nilai *False*

Behavior	
ConvertEmptyString	True
InsertVisible	True
ShowHeader	False
SortExpression	

6. Terakhir buat sebuah Template Field lagi, tetapi set property *HeaderText* agar berisi *Operation*.



Selanjutnya, kembalilah ke mode *Source* dan modifikasi bagian Gridview sesuai dengan petunjuk berikut:

1. Di dalam field atau columns *No.* modifikasi tag didalamnya seperti listing berikut ini:



```
<asp:TemplateField HeaderText="No.">
  <ItemTemplate>
    <%# Container.DisplayIndex + 1 %>
  </ItemTemplate>
</asp:TemplateField>
```



---

---

Modifikasi tersebut untuk menempatkan sebuah field yang berisi nomor urut hanya berdasarkan jumlah record. Jumlah record dalam kasus ini sama dengan jumlah file yang terdapat di dalam folder *DownloadFile*.

---

---

2. Di dalam field *Template Field* yang tidak memiliki header, modifikasi listingnya menjadi seperti listing berikut :

```
<asp:TemplateField ShowHeader="False">
  <ItemTemplate>
    <asp:LinkButton
      ID="LinkButton1"
      runat="server"
      CausesValidation="false"
      OnClick="downloadFile"
      Text="Download"
      CommandArgument=
'<%# Eval("Name") %>'>
    </asp:LinkButton>
```

```
</ItemTemplate>  
</asp:TemplateField>
```

Dalam field tersebut, ditempatkan sebuah Linkbutton yang nanti berfungsi untuk melakukan proses download file. Dalam Linkbutton tersebut, terdapat beberapa property yang penting diperhatikan yakni :

- *CausesValidation* yang diisi dengan nilai *false*, berarti bahwa Linkbutton tersebut tidak terpengaruh dengan komponen validasi yang terdapat di dalam halaman web yang sama.
- *OnClick* yang diisi dengan nilai *downloadFile* menandakan bahwa pada saat diklik, Linkbutton tersebut akan melakukan eksekusi prosedur dengan nama *downloadFile* (akan dibuat di proses berikutnya).
- *CommandArgument* yang diisi dengan nilai '*<%# Eval("Name") %>*' akan berfungsi sebagai parameter yang akan diambil di dalam prosedur sekaligus membedakan file mana yang akan didownload oleh pengguna. Ini juga berarti bahwa field *Name* menjadi parameter utama dalam proses download, sehingga jika terjadi dua buah file yang sama dalam satu folder, maka akan dianggap sebagai satu file saja.

3. Kemudian, Template Field yang terakhir yaitu field *Operation*, modifikasi listingnya seperti listing berikut :

```
<asp:TemplateField
HeaderText="Operation">
<ItemTemplate>
    <asp:LinkButton ID="LinkButton2"
runat="server"
    OnClick="deleteFile"
    CommandArgument=
'<%# Eval("Name") %>'
    OnClientClick='return Confirm();'
    Enabled='<%#Session("Admin") %>'\>
Delete</asp:LinkButton>
</ItemTemplate>
</asp:TemplateField>
```

Hampir sama dengan Linkbutton yang terdapat dalam field sebelumnya, tetapi terdapat beberapa perbedaan property di dalam field ini yaitu :

- *OnClientClick* yang diisi dengan nilai *return Confirm()* yang berarti pada saat Linkbutton diklik, maka akan melakukan eksekusi terlebih dulu ke script javascript yang telah dibuat di penjelasan sebelumnya, sehingga pada saat proses hapus dilakukan, administrator akan diberikan konfirmasi terlebih dulu.

- *Enabled* yang diisi dengan nilai `<%#Session("Admin") %>` yang berarti bahwa Linkbutton tersebut hanya akan dapat berfungsi jika session bernama *admin* memiliki nilai *true*, dengan kata lain jika proses login sebagai administrator berhasil dilakukan, maka Linkbutton ini baru dapat berfungsi. Pemberian nilai pada session bernama *admin* ini akan dilakukan pada saat proses login dilakukan.

4. Untuk melihat hasilnya, berpindahlah lagi ke mode *Design*, dan hasilnya akan tampak seperti pada gambar

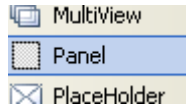
asp:GridView#GridView1

No.	Name	File Size	Date Modified	Operation
Databound	Databound	Databound	Databound	<a href="#">Download</a> <a href="#">Delete</a>
Databound	Databound	Databound	Databound	<a href="#">Download</a> <a href="#">Delete</a>
Databound	Databound	Databound	Databound	<a href="#">Download</a> <a href="#">Delete</a>
Databound	Databound	Databound	Databound	<a href="#">Download</a> <a href="#">Delete</a>
Databound	Databound	Databound	Databound	<a href="#">Download</a> <a href="#">Delete</a>

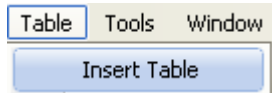
Langkah berikutnya adalah kembali ke mode *Source* dan menambahkan satu layer baru dengan style dari class *utama* di bagian paling bawah, atau tepat di bawah bagian footer.

```
<div class="utama">  
  
</div>
```

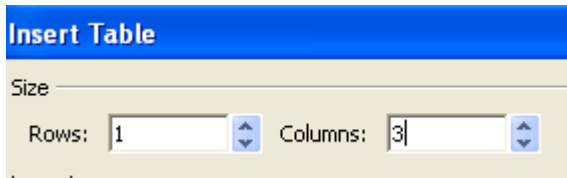
Kemudian, kembali ke mode *Design* dan tempatkan dua buah komponen Panel di dalam layer tersebut.



Di dalam Panel yang pertama set property *Visible* menjadi *false*, lalu tempatkan sebuah *table* dengan memilih menu *Table à Insert Table*.



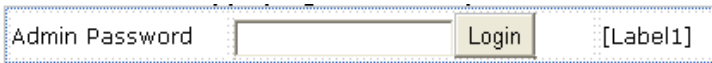
Pada kotak dialog yang tersedia, set jumlah baris sebanyak satu dan jumlah kolom sebanyak 3. Model layout seperti ini (dengan menggunakan Table, akan membantu penempatan komponen agar dapat lurus dan teratur).



Kemudian di kolom yang pertama, ketikkan keterangan *Admin Password*, sedangkan di kolom yang kedua drag sebuah Textbox dan sebuah Button. Untuk Textbox, set property *TextMode* menjadi *Password*, dan untuk Button set property *Text* menjadi *Login*.



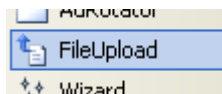
Untuk kolom yang terakhir, tempatkan sebuah Label yang property *Text*-nya dikosongkan. Layout untuk Panel pertama akan terlihat seperti pada gambar berikut :



Sedangkan untuk Panel yang kedua, nantinya akan berfungsi bagi administrator melakukan proses upload file dan juga penggantian password. Sama seperti Panel yang pertama, set property *Visible* dari Panel ini menjadi *false*. Lalu sisipkan Table juga didalamnya, tetapi dengan jumlah baris sebanyak 3 dan kolom juga sebanyak 3.

Di dalam baris pertama, ikuti petunjuk berikut :

- Kolom pertama, ketikkan keterangan *File*.
- Kolom kedua, tempatkan komponen *FileUpload*



- Kolom ketiga tempatkan komponen Button dan set property *Text* menjadi *Upload File*.

Sedangkan untuk baris kedua, ikuti petunjuk berikut ini :

- Kolom pertama, ketikkan teks *New Password*
- Kolom kedua, tempatkan sebuah Textbox

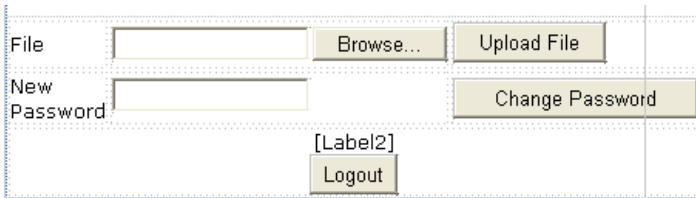
- Kolom ketiga, tempatkan sebuah Button dan set property *Text* menjadi *Change Password*.

Di baris yang terakhir, blok ketiga kolom yang ada didalamnya dan klik kanan lalu pilih sub menu *Modify* à *Merge Cells* untuk menggabungkan ketiga kolom tersebut menjadi satu.



Kemudian klik icon untuk melakukan proses *alignment* dan set menjadi *Justify Center* agar baris yang ketiga menjadi rata tengah.

Lalu, tempatkan sebuah Label dengan property *Text* yang dikosongkan serta sebuah Button dengan property *Text* berisi *Logout*. Hasil layout dari Panel kedua terlihat seperti pada gambar berikut ini.



Langkah selanjutnya adalah melakukan proses pengetikan listing untuk situs ini. Langkah-langkah dalam pengetikan listing adalah :

1. Dobel klik di bagian kosong halaman web sehingga akan menuju ke bagian *Code* dan terbentuk

prosedur *Page\_Load*. Didalam prosedur tersebut ketikkan listing berikut:

```
RefreshGrid()
```

2. Prosedur *RefreshGrid* yang dipanggil di dalam prosedur *Page\_Load* diketikkan tepat dibawahnya dan memiliki listing sebagai berikut :

```
Sub RefreshGrid()  
Dim directory As _  
    System.IO.DirectoryInfo = _  
    New System.IO.DirectoryInfo _  
    (Server.MapPath("~/DownloadFile/"))  
Me.GridView1.DataSource = _  
    directory.GetFiles()  
Me.GridView1.DataBind()  
End Sub
```



---

---

Prosedur *RefreshGrid* berfungsi untuk menempatkan daftar file dalam folder *DownloadFile* menjadi sebuah sumber data yang dapat ditampilkan di Gridview. Sekaligus pula melakukan refresh data di Gridview jika terdapat file baru yang telah diupload oleh administrator.

---

---



3. Berikutnya, ketikkan prosedur untuk menangani perilaku GridView saat ada data baru dengan isi sebagai berikut:

```
Protected Sub GridView1_DataBound _  
    (ByVal sender As Object, _  
    ByVal e As System.EventArgs) _  
Handles GridView1.DataBound  
    Panel1.Visible = False  
    Panel2.Visible = _  
        Session("Admin")  
End Sub
```



---

---

Saat GridView menerima sumber data (datasource) dari daftar file di folder *DownloadFile*, maka akan dicek pula status dari session yang sedang berjalan. Jika terjadi operasi file (hapus atau delete) maka berarti mode situs sedang dalam posisi login administrator, dan jika hanya terjadi operasi download, berarti dalam posisi user biasa. Hal ini dideteksi agar Panel kedua yang hanya berfungsi untuk administrator tidak terlihat pada saat posisi sedang dalam keadaan user biasa.

---

---

4. Kemudian ketikkan juga prosedur *downloadFile* yang sebelumnya telah didefinisikan terlebih dulu di

dalam GridView1 (lihat lagi penjelasan pada pengaturan layout GridView).

```
Sub downloadFile _  
    (ByVal sender As Object, _  
    ByVal e As EventArgs)  
    Response.AppendHeader _  
        ("content-disposition", _  
        "attachment; filename=" + _  
        sender.CommandArgument)  
    Response.WriteFile _  
        (Server.MapPath _  
        ("~/downloadfile/") & _  
        sender.CommandArgument)  
    Response.End()  
End Sub
```



---

Proses download untuk user biasa memiliki proses yang sedikit berbeda. Karena dalam situs ini tidak dibatasi jenis file yang bisa diupload, maka dapat terjadi resiko pada saat user melakukan download akan "lari" ke halaman web yang menampilkan file tersebut (misal: file image). Untuk mencegah hal tersebut, maka dilakukan proses download file dengan menggunakan perintah *Response* yang

memposisikan file sebagai sebuah *attachment*. Akibatnya, apapun jenis file yang diklik pada Linkbutton *Download* akan menampilkan kotak dialog popup untuk menyimpan file tersebut.

---

---

5. Prosedur buatan sendiri yang terakhir adalah prosedur *deleteFile* yang nanti akan dieksekusi saat administrator akan melakukan penghapusan file dari folder. Isi dari prosedur tersebut adalah :

```
Sub deleteFile _  
    (ByVal sender As Object, _  
     ByVal e As EventArgs)  
        IO.File.Delete(Server.MapPath _  
            ("~/downloadfile/") & _  
            sender.CommandArgument)  
        RefreshGrid()  
End Sub
```

6. Kini, kembalikan lagi ke mode desainer, dan double klik pada Linkbutton1 atau Linkbutton *Administrator* yang terletak pada bagian footer. Lalu ketikkan listing berikut ini:

```
Panel1.Visible = True
```

7. Selanjutnya, double klik pada Button *Login*, dan ketikkan listingnya

```
If TextBox1.Text = _
    System.Web.Configuration. _
    WebConfigurationManager. _
    AppSettings("Password") Then
    Panel1.Visible = False
    Panel2.Visible = True
    Session("Admin") = True
    RefreshGrid()
    Label1.Text = ""
Else
    Label1.Text = "Login error !"
End If
```



---

---

Proses login untuk administrator hanya mengambil parameter password dari file web.config di dalam bagian tag *appSetting* (lihat lagi pembahasan modifikasi web.config). Jika login error, maka di dalam Label1 akan ditampilkan pesan error. Sedangkan jika benar, maka Panel kedua akan ditampilkan dan sebagai tanda bahwa situs sedang dalam mode admin, maka session dengan nama *Admin* diset dengan nilai *True*. Setelah itu, mode Gridview juga diperbarui dengan memanggil ulang prosedur *RefreshGrid*, sehingga Linkbutton *Delete* dapat terlihat di Gridview.

---

---

8. Lalu, untuk listing di Button *Upload File*, dobel klik Button tersebut dan di dalam prosedur yang tersedia, tempatkan listing berikut :

```
Try
    With FileUpload1
        If .HasFile Then
            .SaveAs(Server.MapPath _
                ("~/DownloadFile/") & _
                .FileName)
            RefreshGrid()
            Label2.Text = ""
        Else
            Label2.Text = _
                "No file to upload !"
        End If
    End With
Catch ex As Exception
    Label2.Text = "Upload error !"
End Try
```



---

Seperti telah dijelaskan sebelumnya, proses upload file tidak membatasi jenis file yang akan diupload oleh administrator. Validasi hanya dilakukan untuk komponen FileUpload, apakah telah memiliki nama dan lokasi file yang siap

diupload. Jika nama dan lokasi file telah ada, maka proses penyimpanan file hasil upload dilakukan dengan nama asli. Ini berarti jika ada nama file yang sama, maka akan dilakukan proses *overwrite*.

9. Terakhir, double klik pada Button *Logout* dan isikan listingnya

```
Session.Clear()
```

```
RefreshGrid()
```

## Testing dan Pengembangan

---

Testing dari situs ini dapat dilakukan dengan melakukan langkah-langkah berikut :

1. Login terlebih dulu sebagai administrator dengan password yang telah ditentukan (dalam contoh listing, password default administrator adalah 222, tetapi juga bisa diganti sesuai selera).
2. Setelah itu, cobalah untuk melakukan upload beberapa jenis file yang berbeda ke dalam situs tersebut. Tampilan mode administrator akan tampak seperti pada gambar berikut :



3. Selanjutnya, beralihlah ke mode user dengan melakukan proses logout. Pada saat beralih ke mode user dari mode administrator, perhatikan perubahan yang terjadi di dalam Gridview. Linkbutton untuk proses hapus akan terlihat *disabled* sehingga tidak bisa lagi diakses oleh user.

...: Simple Download Manager ...

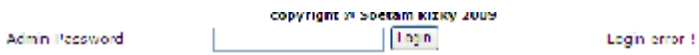
No.	Name	File Size	Date Modified	Operation
1	Petualangan Tom Sawyer.pdf	282425	4/24/2008 18:03:06 PM	<a href="#">Download</a> <a href="#">Delete</a>
2	PPA.xls	57344	4/12/2009 8:55:49 AM	<a href="#">Download</a> <a href="#">Delete</a>

Administrator  
copyright © Soetam Rizky 2009

4. Kemudian cobalah untuk melakukan proses download terhadap salah satu file, maka akan muncul kotak dialog berikut :



5. Coba juga untuk melakukan kesalahan pada saat proses login dengan melakukan kesalahan input pada password, sehingga akan muncul pesan kesalahan.



6. Kini, berpindahlah lagi ke mode administrator dengan melakukan proses login. Lalu cobalah untuk melakukan proses penghapusan salah satu file. Saat Linkbutton *Delete* diklik, maka akan muncul kotak dialog konfirmasi berikut yang merupakan hasil dari injeksi javascript.





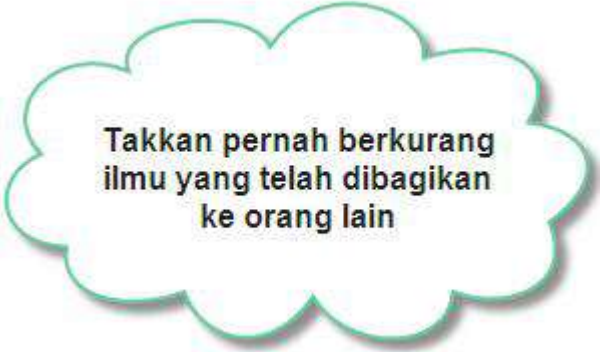
7. Cobalah juga untuk melakukan proses penggantian password di dalam mode administrator. Dan setelah penggantian password berhasil, lakukan login ulang untuk verifikasi, apakah password sudah berhasil digantikan.

Sedangkan pengembangan yang dapat dilakukan bagi Anda yang ingin lebih bereksplorasi antara lain :

1. Membuat download manager agar dapat menangani folder majemuk, sehingga tidak hanya terdiri dari satu folder.
2. Membuat halaman web registrasi, sehingga download manager dapat menangani proses download sesuai dengan hak akses dari tiap user yang diberikan oleh administrator, hampir mirip dengan apa yang terjadi di banyak situs file hosting seperti rapidshare.com atau filefactory.com.
3. Membuat download manager dapat melakukan pengelompokan berdasarkan jenis file yang akan didownload oleh pengguna.

# **Simple Company Profile**

---



**Takkan pernah berkurang  
ilmu yang telah dibagikan  
ke orang lain**

**Level :** Pemula

**Tujuan :**

- Membuat situs company profile sederhana
- Membuat content management system untuk sebuah situs dengan basis file teks tanpa menggunakan database.

**Fitur situs :**

- Membuat situs yang *customizable* dengan menggunakan teknik pembuatan content management system dalam satu halaman web
- Modifikasi layout situs dengan memanfaatkan file teks, tanpa menggunakan database
- Menggunakan komponen jenis Rich Text Editor dalam melakukan pengeditan file teks dengan format teks yang diterjemahkan ke format HTML.

**Estimasi waktu pembuatan :** 60 – 120 menit

## **Pengantar**

---

---

Salah satu fitur situs yang selalu menjadi perhatian utama dalam proses pembelajaran pemrograman web adalah membuat CMS atau *content management system*. CMS sendiri sebenarnya lebih banyak diartikan sebagai fitur situs yang mampu melakukan update isi situ secara dinamis, sehingga pengguna "awam" dapat melakukan update isi situs tanpa bantuan webmaster.

Contoh dalam bab ini menampilkan cara pembuatan CMS secara sederhana, tetapi dengan cara yang berbeda. Karena contoh ini ditujukan untuk pemula, maka data yang diupdate dalam situs ini bukan berupa database, tetapi berupa file teks. Sedangkan untuk contoh CMS dengan menggunakan database akan dibahas di bab yang berbeda.

Selain untuk para pemula, salah satu keuntungan membuat situs dengan CMS yang terdiri dari file teks adalah kemudahan dalam mencari web hosting yang "murah meriah". Karena tidak memerlukan database, proses instalasi situs juga tergolong sangat mudah dan cepat.

Tetapi, harga yang harus dibayar karena tidak memiliki database adalah kesederhanaan dari situs itu sendiri. Itulah alasan mengapa situs ini lebih sesuai

untuk sebuah situs *company profile* yang didalamnya hanya terdiri dari data-data sederhana mengenai profil sebuah perusahaan. Dan situs dengan fitur tersebut, di kenyataan masih sangat dibutuhkan. Dengan kata lain, fitur yang ditawarkan oleh contoh berikut ini adalah cepat, tepat, singkat, sederhana dan murah meriah.

Untuk kepentingan pembelajaran, maka Anda akan melakukan modifikasi terhadap file `web.config` dan file teks, serta menggunakan *rich text editor* yaitu dengan memanfaatkan salah satu komponen editor teks gratis yang sudah terkenal yakni FCKEditor. Selain itu, juga diperlihatkan cara untuk membuat sebuah situs menjadi *customizable* sehingga lebih mudah "dilepas" ke pengguna awam.

## **Struktur Situs**

---

---

Struktur situs dalam contoh ini terdiri dari :

1. Default.aspx

Merupakan halaman utama yang terdiri dari bagian utama (company profile) dengan struktur layout yang terdiri dari header, footer dan bagian utama yang terbagi lagi menjadi dua bagian yaitu bagian menu dan data perusahaan.

2. Admin.aspx

Merupakan halaman untuk administrator yang akan digunakan untuk mengatur tampilan sekaligus melakukan proses manipulasi file teks

3. CompanyProfile.css

Merupakan file style sheet untuk mengatur tampilan situs secara keseluruhan.

4. Web.config

Dalam file konfigurasi ini, akan ditempatkan data yang menjadi acuan dari situs, seperti lokasi penyimpanan file teks dan setting situs umum seperti header dan footer.

5. Folder *App\_Data*

Dalam folder yang memang dikhususkan untuk menyimpan data ini akan ditempati oleh file-file teks yang dibutuhkan dalam eksekusi situs.

6. Folder *FCKEditor*

Merupakan folder yang nanti berisi file-file bawaan dari komponen FCKEditor.

## Pembuatan File Teks

---

Langkah pertama dalam pembuatan situs ini, sama dengan contoh sebelumnya yaitu membuat web site baru di dalam IDE Visual Studio atau Visual Web Developer Express Edition dengan tipe *ASP .NET Website*. Di dalam solution yang tersedia, juga akan disediakan sebuah file *Default.aspx*, file *web.config* dan sebuah folder *App\_Data*.

Berikutnya, buatlah empat buah file teks di dalam folder *App\_Data* dengan cara melakukan klik pada folder *App\_Data* di *Solution Explorer* lalu memilih menu *Website → Add New Item*. Di dalam kotak dialog yang tersedia, pilih jenis *Text File* dan ketikkan nama file teks yang akan dibuat. Lalu ketikkan isi dari file teks yang dimaksud di dalam mode listing di IDE Visual Studio.



---

Penempatan file-file teks di dalam folder *App\_Data* bertujuan agar file-file tersebut tidak dapat diakses secara langsung oleh



pengguna di dalam browser. Hal ini dikarenakan secara default seluruh file yang ada di dalam folder App\_Data tidak akan dapat diakses oleh browser secara langsung oleh pengguna. Bahkan dari browser folder ini dianggap "tidak ada".

---

---

Terdapat empat buah file teks yang dibuat dalam situs ini yakni :

1. File *ContactUs.txt*

Merupakan file teks untuk mendefinisikan isi dari sub menu *Contact Us* atau alamat dari perusahaan. Untuk langkah awal, Anda bisa mengisikan apa saja di dalam file teks ini.

2. File *Home.txt*

Merupakan file teks yang akan menjadi bagian utama dari situs dan muncul pertama kali saat situs dijalankan. Sama dengan file teks sebelumnya, file ini bisa diisi dengan content apa saja, karena nanti akan diedit ulang melalui editor di mode administrator.

3. File *Password.txt*

Merupakan file penting untuk kepentingan login dan hanya boleh berisi 1 baris saja untuk pengisian password. Dalam contoh ini Anda juga bisa mengisi password sesuai selera.

4. File *Profile.txt*

Sama seperti file teks *ContactUs* dan *Home*, file ini nanti akan menjadi isi dari halaman utama, dan pada langkah awal bisa diisi sesuai selera.

## Modifikasi web.config

---

Langkah berikutnya adalah melakukan modifikasi web.config. Hampir sama dengan modifikasi yang dilakukan di contoh situs *Simple Download Manager*, di dalam web.config ditambahkan bagian baru di dalam tag *appSetting*. Dan di dalam contoh ini juga ditambahkan konfigurasi untuk melindungi halaman web admin.aspx.

Selengkapnya modifikasi web.config dapat mengikuti langkah berikut ini :

1. Cari bagian tag `<appSetting/>` dan modifikasi seperti listing berikut:

```
<appSettings>
  <add key="header"
    value=
      "...: Simple Company Profile :..." />
  <add key="logo"
    value="~/logo.gif" />
  <add key="copyright"
    value="copyright @ Antologi 2009" />
</appSettings>
```



---

Dalam setting tersebut, selain mengatur teks yang akan tampil di bagian header dan footer, juga mengatur logo yang akan muncul di

bagian header. Diasumsikan bahwa nama file image yang akan dijadikan logo adalah *logo.gif* dan sudah tercopy ke root folder dari situs ini. Anda dapat mencari file image lain (dengan format selain format gif) lalu melakukan operasi *copy-paste* ke dalam Solution Explorer.

2. Lalu cari bagian tag `<authentication>` dan modifikasi isinya.

```
<authentication mode="Forms" >
    <forms defaultUrl="~/admin.aspx"
        loginUrl="~/default.aspx" />
</authentication>
```



---

Modifikasi tag *authentication* akan menyebabkan halaman web yang telah dideklarasikan sebagai *defaultUrl* akan terproteksi dan tidak akan bisa diakses secara langsung oleh pengguna tanpa melalui proses login. Sedangkan alamat halaman web di atribut *loginUrl* merupakan alamat halaman web yang nantinya akan ditempati oleh proses login.

---

## Style Sheet : CompanyProfile.css

---

Setelah menyelesaikan proses pembuatan file teks dan modifikasi web.config, kini saatnya menyelesaikan pembuatan file style sheet. Yang harus dilakukan adalah memilih menu *Website à Add New Item* dan di dalam kotak dialog yang tersedia pilih tipe *Style Sheet*. Beri nama file style sheet tersebut dengan nama *CompanyProfile.css*.



Di dalam style sheet tersebut, ketikkan listing berikut :

```
body
{
    font-family:Verdana;
    font-size:10pt;
}
```

```
.utama
{
    margin-left:auto;
    margin-right:auto;
    width:80%;
    position:relative;
    border-left:double 2px blue;
    border-right:double 2px blue;
    padding-left:2px;
    padding-right:2px;
}

.kiri
{
    float:left;
    width:30%;
    padding-right:2px;
    border-right:dashed 1px red;
    height:50%;
}

.kanan
{
    float:right;
    width:65%;
    padding-right:2px;
}

.header
```

```
{
    width:80%;
    font-weight:bold;
    font-size:14pt;
    text-align:center ;
    height:50%;
    margin-left:auto;
    margin-right:auto;
}
.footer
{
    margin-top:5px;
    margin-left:auto;
    margin-right:auto;
    border-top:solid 3px black;
    font-size:8pt;
    font-weight:bold;
    color:Black;
    text-align:center ;
    width:80%;
}
```



---

Dalam listing style sheet layout yang dibuat hampir sama dengan layout pada contoh *Simple Download Manager*, yaitu terletak di tengah browser dengan sisa ruang yang sama

besar di kiri dan kanan (ditandai dengan penggunaan margin kiri dan kanan yang diberi nilai *auto*). Tetapi di dalam contoh ini, diberikan bagian kiri dan kanan yang akan digunakan sebagai tanda pemisah untuk menu dan isi company profile. Meski demikian, Anda juga bisa membuat sendiri layout yang sesuai selera, dengan melihat template-template CSS yang bertebaran di dunia maya, misal : [www.csszengarden.com](http://www.csszengarden.com)

---



## Halaman Administrator : Admin.aspx

---

Sebelum memulai pembuatan halaman untuk administrator ini, Anda wajib untuk melakukan download komponen FCKEditor di situs resminya yaitu [www.fckeditor.net](http://www.fckeditor.net) . Komponen ini merupakan jenis komponen *rich text editor* (RTE) yang berarti bahwa komponen ini berfungsi untuk melakukan entri teks dengan format seperti halnya pada aplikasi pengolah kata sederhana seperti Wordpad. Tetapi format teks yang diberikan (seperti bold, italic atau underline) nantinya akan dikonversi ke dalam tag HTML yang bersesuaian.



---

Jika Anda mencari komponen RTE lain yang gratis dan “bagus” sebenarnya masih sangat banyak. FCKEditor hanyalah salah satu contoh yang digunakan dalam buku ini. Contoh komponen RTE lain adalah FreeTextBox yang akan digunakan di contoh yang lain. Sedangkan kriteria untuk bagus dan cocok dalam pemilihan komponen RTE lebih mengarah pada penilaian subyektif, khususnya untuk memenuhi kebutuhan dari situs itu sendiri. Jadi, dalam pengembangannya, Anda juga bisa mengganti komponen

FCKEditor dalam contoh ini dengan komponen RTE yang lain, atau bahkan dengan komponen Textbox biasa.

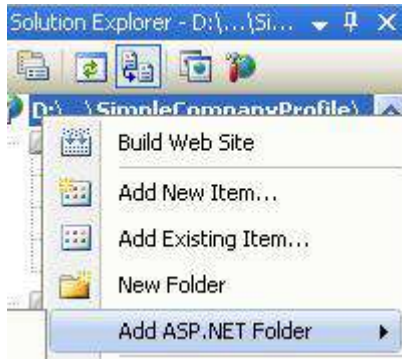
---

---

Komponen FCKEditor merupakan komponen RTE yang gratis dan telah terbukti banyak digunakan di berbagai portal dan juga dengan berbagai pilihan bahasa pemrograman. Saat proses download, pilihlah komponen FCKEditor yang memang dikhususkan untuk kepentingan ASP .NET (hingga buku ini ditulis hanya sampai versi untuk ASP .NET 2.0).

Setelah selesai melakukan proses download (dan juga berhasil melakukan ekstraksi file hasil download), lakukan operasi copy-paste folder FCKEditor ke dalam Solution Explorer.

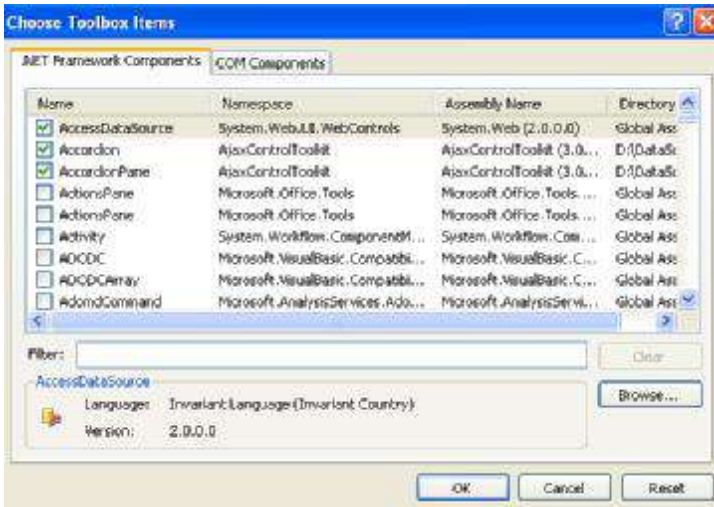
Kemudian, di dalam Solution Explorer , klik kanan di bagian paling atas hirarki dan pilih sub menu *Add ASP .NET Folder à Bin* untuk membuat folder built-in baru sebagai tempat penampung komponen FCKEditor.




Lalu lakukan operasi copy-paste file dengan nama *FredCK.FCKeditorV2.dll* (dari hasil download) ke dalam folder *Bin* tersebut. Dan kini Anda telah siap untuk melakukan penambahan komponen FCKEditor ke dalam toolbox.

Untuk menambahkan komponen baru ke dalam Toolbox, lakukan langkah berikut :

1. Klik kanan pada Toolbox dan pilih sub menu *Choose Item*
2. Di dalam kotak dialog yang tersedia, klik tombol Browse. Setelah muncul kotak dialog, arahkan ke folder Bin tempat aplikasi dibuat, lalu pilih file *FredCK.FCKeditorV2.dll* yang sebelumnya telah dicopykan.



3. Jika proses berhasil maka di dalam Toolbox akan muncul icon baru untuk FCKEditor  FCKeditor, dan itu berarti kita siap untuk melanjutkan ke proses berikutnya.

Untuk melakukan proses desain dari layout di halaman admin, berpindahlah terlebih dulu ke mode Source dan ikuti langkah-langkah selanjutnya :

1. Drag file style sheet *CompanyProfile .css* tepat di bawah tag `<title>...</title>` sehingga listing akan tampak sebagai berikut :

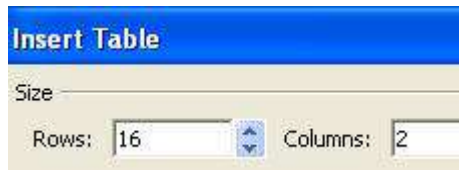
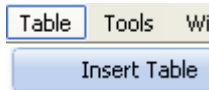
```
<link href="CompanyProfile.css"
rel="stylesheet" type="text/css" />
```

2. Kemudian modifikasi tag yang terletak di bawah tag `<form>` dengan listing berikut :

```
<div class="header">
...: Admin Page :...
</div>
<div class="utama">

</div>
```

3. Berikutnya, berpindahlah ke mode Design, dan tepat di bawah header atau layer *utama*, sisipkan sebuah Table dengan cara memilih menu *Table à Insert Table*. Kemudian di dalam kotak dialog, pilih jumlah baris (row) sebanyak 16 dan jumlah kolom (column) sebanyak 2.



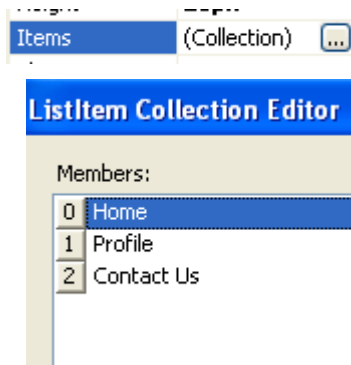
4. Kemudian, pada baris ke 1, 5, 7, 10 dan 13, masing-masing blok dua kolom dalam tiap baris dan klik kanan, lalu pilih sub menu *Modify à Merge Cells* untuk menjadikan dua kolom dalam baris-baris tersebut menjadi satu. Dan juga, di tiap baris yang telah digabungkan tersebut, pilih alignment *Justify*

*Center* dari icon yang tersedia di toolbar dan format tulisan **Bold**



5. Kini ketikkan teks-teks berikut pada posisi masing-masing :
  - a. Baris pertama (pada kolom yang telah digabung) : *Save Content*
  - b. Baris kedua kolom pertama : *Header*
  - c. Baris ketiga kolom pertama : *Footer*
  - d. Baris keempat kolom pertama : *Logo*
  - e. Baris ketujuh : *Site Content*
  - f. Baris kedelapan kolom pertama : *Part*
  - g. Baris ketigabelas : *Change Password*
  - h. Baris keempatbelas kolom pertama : *Password*
6. Kemudian tempatkan komponen-komponen pada kolom-kolom berikut ini :
  - a. Baris kedua kolom kedua : Textbox
  - b. Baris ketiga kolom kedua : Textbox
  - c. Baris keempat kolom kedua : FileUpload
  - d. Baris kelima : Linkbutton dengan property *Text* diset dengan nilai *Save Setting*
  - e. Baris kedelapan kolom kedua : DropDownList dengan setting property :

- AutoPostBack : True
- ItemCollection → dengan melakukan klik pada tombol kecil di jendela Property



---

Property Items di dalam DropDownList, pada contoh-contoh yang lain akan diisi menggunakan data dari tabel di dalam database. Tetapi dalam contoh ini diisi melalui property karena memiliki nilai yang statis.

Sedangkan property *AutoPostBack* dengan nilai *True* akan menyebabkan DropDownList bersifat interaktif saat dipilih langsung oleh pengguna.

---

- f. Baris kesepuluh kolom kedua : FCKEditor dengan setting property :
- BasePath : ~/FCKeditor/
  - Width : 100%

- Height : 200px
- g. Baris kesebelas : Linkbutton dengan setting property *Text* diberi nilai *Save Content*
- h. Baris kedubelas kolom kedua : satu Textbox dan satu Linkbutton dengan property *Text* berisi *Save Password*
- i. Baris keempatbelas kolom kedua : Linkbutton dengan property *Text* berisi *Logout*.

Hasil dari layout teks dan penempatan komponen akan tampak seperti pada gambar berikut ini :



**...: Admin Page ...**

**Setting**

Header

Footer

Logo

[Save Setting](#)

**Site Content**

Part  ▼

FCKeditor V2 - FCKeditor1

[Save Content](#)

**Change Password**

Password

[Logout](#)

Lalu, untuk listing yang ada dalam halaman web ini, ikuti langkah penyelesaiannya :

1. Dobel klik pada salah satu bagian kosong di halaman web, sehingga akan menuju ke area listing

program. Lalu di dalam prosedur *Page\_Load* ketikkan listing berikut ini:

```
If Not Session("login") Then
    Response.Redirect("default.aspx")
Else
    If Not IsPostBack Then
        TextBox1.Text = _
            System.Web.Configuration. _
                WebConfigurationManager. _
                    AppSettings("header")
        TextBox2.Text = _
            System.Web.Configuration. _
                WebConfigurationManager. _
                    AppSettings("copyright")
        TextBox3.Text = _
            IO.File.ReadAllText _
                (xFolder & "Password.txt")
        DropDownList1.SelectedIndex = 0
        DropDownList1_SelectedIndexChanged _
            (sender, e)
    End If
End If
```



---

Pada prosedur ini memiliki dua fungsi sekaligus yakni untuk melakukan proses proteksi jika halaman web admin.aspx diakses

tanpa melalui proses login (pada kondisi percabangan pertama). Sedangkan fungsi yang kedua adalah untuk membaca file web.config yang dilakukan untuk membaca konfigurasi dari situs dan ditampilkan ke bagian yang diinginkan (dalam Textbox) agar bisa diedit. Selain itu juga melakukan proses refresh dari isi komponen FCKEditor yang nanti akan dilakukan di dalam prosedur *DropDownList1\_SelectedIndexChanged* dengan menggunakan parameter default.

2. Di bagian atas dari deklarasi *Public Class ....* Ketikkan deklarasi variabel berikut ini yang menyatakan lokasi folder tempat penyimpanan file teks untuk konfigurasi serta isi dari situs itu sendiri.

```
Dim xFolder As String = _  
    Server.MapPath("~/App_Data/")
```

3. Setelah itu ketikkan sebuah fungsi *ChoosePart* yang berguna untuk menentukan file konfigurasi mana yang sedang diedit untuk kemudian diakses oleh Linkbutton *Save Content*.

```
Function choosePart _  
    (ByVal xindex As Integer) As String  
    Dim xpart As String  
    Select Case xindex  
        Case 0
```

```
xpart = "Home.txt"
Case 1
    xpart = "Profile.txt"
Case 2
    xpart = "ContactUs.txt"
End Select
Return xFolder & xpart
End Function
```

4. Berikutnya double klik di Linkbutton *Save Setting* dan ketikkan listing berikut ini yang berguna untuk hasil isian di tiga Textbox pertama ke dalam file konfigurasi web.config. Setting yang disimpan dalam prosedur ini adalah setting untuk *header*, *footer* dan *logo*. Khusus untuk kepentingan logo, sekaligus juga dilakukan proses penyimpanan file hasil upload ke dalam direktori root dari situs.

```
Dim xconfig As Configuration = _
Web.Configuration. _
    WebConfigurationManager. _
        OpenWebConfiguration("~/")
xconfig.AppSettings. _
    Settings("header"). _
        Value = TextBox1.Text
xconfig.AppSettings. _
    Settings("copyright"). _
        Value = TextBox2.Text
```

```
With FileUpload1
  If .HasFile Then
    xconfig.AppSettings. _
      Settings("logo"). _
        Value = "~/\" & .FileName
    .SaveAs(.FileName)
  End If
End With
xconfig.Save()
```

5. Kemudian double klik di komponen `DropDownList1` yang terletak di baris kesebelas dan di dalam `DropDownList1_SelectedIndexChanged` ketikkan listing berikut ini untuk membaca isi dari file teks ke dalam komponen `FCKEditor` yang dianggap sedang aktif berdasarkan pilihan user di dalam `Dropdownlist`.

```
FCKeditor1.Value = _
  IO.File.ReadAllText _
  (choosePart(DropDownList1. _
    SelectedIndex))
```

6. Berikutnya double klik di `Linkbutton Save Content` untuk menyetikkan listing yang nantinya akan mengeksekusi penyimpanan file teks berdasarkan isi yang diketikkan oleh administrator di dalam komponen `FCKEditor`.

```
IO.File.WriteAllText _  
    (choosePart(DropDownList1. _  
        SelectedIndex), FCKeditor1.Value)
```

7. Lalu double klik di Linkbutton *Save Password* untuk melakukan penyimpanan Password baru ke dalam file teks *Password.txt*. Perlu diingat bahwa password dalam contoh ini tidak melalui proses pengamanan apapun (seperti enkripsi) sehingga dapat dikembangkan lebih lanjut bagi Anda yang ingin lebih bereksplorasi dengan menambahkan fitur enkripsi.

```
If Trim(TextBox3.Text) <> "" Then  
    IO.File.WriteAllText _  
        (xFolder & "Password.txt", _  
        TextBox3.Text)  
End If
```

8. Tambahkan atribut *ValidateRequest="false"* di dalam source *admin.aspx*, tepat di bagian paling atas source yaitu di dalam tag *<%@ Page Language="VB".....*. Hal ini dilakukan agar saat proses penyimpanan di komponen FCKEditor yang menggunakan format HTML tidak dianggap sebagai suatu injeksi script ke dalam halaman web.

9. Listing terakhir adalah dengan melakukan dobel klik di Linkbutton *Logout* yang nanti akan dieksekusi agar pengguna beralih lagi ke halaman utama.

```
Session.Clear()
```

```
Response.Redirect _  
("default.aspx")
```

## Halaman Utama : Default.aspx

---

Langkah terakhir dalam pembuatan situs ini adalah membuat halaman utama dari situs company profile. Halaman web utama dari situs ini yaitu default.aspx sesungguhnya telah tersedia saat pembuatan situs dilakukan. Tetapi perlu dimodifikasi agar dapat sesuai dengan tujuan situs ini.

Yang pertama dilakukan adalah melakukan modifikasi layout dari halaman default.aspx, dengan mengikuti langkah-langkah berikut ini:

1. Berpindahlah terlebih dulu ke mode Source, kemudian tepat di bawah tag `<title>...</title>` ketikkan listing untuk menghubungkan file CSS ke dalam halaman web.

```
<link href="CompanyProfile.css"
rel="stylesheet"
type="text/css" />
```

2. Kemudian di bawah tag `<form>` ketikkan listing HTML untuk mendapatkan layer berdasarkan class style sheet yang telah didefinisikan dalam file *CompanyProfile.css*.

```
<div class="header">

</div>
```



```
<div class ="utama">
    <div class="kiri">

    </div>
    <div class="kanan">

    </div>
</div>
<div class="footer">

</div>
```



---

---

Halaman utama terdiri dari bagian header yang akan diisi logo, kemudian bagian utama yang terpisah menjadi dua yaitu seksi kiri untuk menu dan seksi kanan untuk isi dari company profile. Bagian terakhir adalah footer yang nanti akan diisi dengan data copyright (berdasarkan isi file *footer.txt*) dan sebuah Linkbutton untuk menampilkan komponen Panel yang berfungsi untuk kepentingan login ke halaman *admin.aspx*.

---

---

3. Lalu setelah berpindah di dalam mode Design, tepat di bagian atas (header) drag sebuah komponen Image didalamnya. Sedangkan di bagian utama

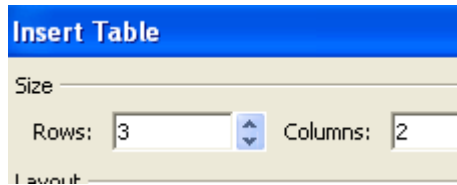
seksi kiri, drag tiga buah Linkbutton yang masing-masing property *Text* diisi dengan nilai *Home*, *Profile* dan *Contact Us*. Dan di seksi kanan, drag sebuah komponen Literal didalamnya.



4. Sedangkan di bagian footer, drag sebuah Linkbutton dan set property *Text* dengan nilai *Admin Login*, kemudian sisipkan di bagian bawah Linkbutton tersebut (dengan menekan tombol Enter setelah Linkbutton) sebuah Label dengan property *ID* berisi *lblfooter*.

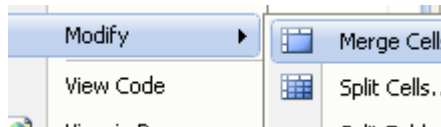


5. Selanjutnya, di bagian bawah footer, drag sebuah komponen Panel dengan property *Visible* yang bernilai *False*. Kemudian di dalam Panel tersebut, sisipkan sebuah Table (dengan mengakses menu *Table à Insert Table*) dengan jumlah baris/row sebanyak 3 dan jumlah kolom/column sebanyak 2.



6. Di dalam Table yang baru terbentuk, set nilai berikut di tiap baris dan kolomnya :

- a. Di baris pertama, pilih kedua kolom (select dengan menggunakan mouse) dan klik kanan serta pilih sub menu *Modify à Merge Cells*. Dan didalamnya ketikkan teks *Admin Login*



- b. Di baris kedua kolom pertama, ketikkan teks *Password* dan di kolom kedua drag sebuah Textbox dengan property *TextMode* diarahkan ke pilihan *Password*.
- c. Di baris ketiga kolom pertama, drag sebuah Button dan set property *Text* menjadi *Login*. Dan di kolom kedua drag lagi sebuah Button dan set property *Text* menjadi *Cancel*. Di sebelah Button *Cancel* drag sebuah Label yang nanti akan digunakan untuk menampilkan pesan kesalahan saat login gagal. Property *Text* untuk Label tersebut dapat dikosongkan.

The screenshot shows a web application interface. At the top, there is a navigation menu with links for 'Home', 'Profile', and 'Contact Us'. Below the menu is a horizontal line, followed by a button labeled 'Admin Login' and a 'Label'. Below this is a form titled '...: Admin Login :...' containing a 'Password' input field with a red asterisk, and two buttons: 'Login' and 'Cancel'. A '[Label1]' is positioned to the right of the 'Cancel' button.

Setelah modifikasi layout selesai dilakukan, ikuti langkah-langkah berikut untuk menyelesaikan listing program, sekaligus mengakhiri pembuatan situs:

1. Dobel klik di bagian kosong dari halaman web sehingga akan menuju ke prosedur *Page\_Load*, dan ketikkan listing berikut :

```
LinkButton1_Click(LinkButton1, e)
If Not Session("login") Then _
    refreshPage()
```



---

Pada saat halaman utama web ditampilkan oleh pengguna, maka hal pertama yang dilakukan adalah melakukan pengisian seksikan dengan memanggil prosedur dari Linkbutton1 (dianggap sebagai isi default yaitu untuk bagian *Home*). Isi dari prosedur ini akan dijelaskan di urutan selanjutnya. Kemudian dilakukan pengecekan apakah proses login

dilakukan oleh pengguna, agar dapat melakukan proses refresh halaman utama jika telah terjadi modifikasi di bagian administrator.

---

---

2. Sebelumnya ketikkan terlebih dulu deklarasi variabel untuk deklarasi letak file teks, tepat di bawah deklarasi *Partial Class*.....

```
Dim xFolder As String = _  
    Server.MapPath("~/App_Data/")
```

3. Selanjutnya, ketikkan dua prosedur yang nanti akan dimanfaatkan untuk menampilkan hasil kustomisasi dari situs ini, yaitu prosedur *refreshPage* untuk melakukan pengaturan isi logo header, tulisan copyright pada footer dan judul halaman web yang muncul di caption browser. Sedangkan prosedur kedua adalah *refreshContent* yang akan digunakan untuk menangani penekanan Linkbutton di seksi kiri dari halaman utama web dan akan dieksekusi saat salah satu Linkbutton di seksi kiri (yang juga berfungsi sebagai menu) diklik oleh pengguna.

```
Sub refreshPage()  
    Page.Title = System.Web. _  
        Configuration. _  
        WebConfigurationManager. _  
        AppSettings("header")
```

```
lblFooter.Text = System.Web. _  
    Configuration. _  
    WebConfigurationManager. _  
    AppSettings("copyright")  
Image1.ImageUrl = _  
    System.Web. _  
    Configuration. _  
    WebConfigurationManager. _  
    AppSettings("logo")  
End Sub  
  
Sub refreshContent _  
    (ByVal xfile As String)  
    Literall1.Text = IO.File. _  
        ReadAllText(xFolder & xfile)  
End Sub
```

4. Berikutnya, dobel klik pada Linkbutton *Home* dan ketikkan listing berikut ini:

```
Select Case sender.text  
    Case "Home"  
        refreshContent("Home.txt")  
    Case "Profile"  
        refreshContent("Profile.txt")  
    Case "Contact Us"  
        refreshContent("ContactUs.txt")  
End Select
```

Di belakang deklarasi prosedur *Linkbutton1\_Click*, setelah *Handles Linkbutton1.Click*, tambahkan tanda koma dan listing berikut ini:

```
LinkButton2.Click, LinkButton3.Click
```



---

---

Penambahan tersebut merupakan implementasi dari teknik *multiple handles* yang menyebabkan prosedur *Linkbutton1\_Click* juga akan berlaku untuk *Linkbutton2* dan *Linkbutton3*. Dan untuk membedakan tiap Linkbutton tersebut, maka di dalam prosedur *refreshContent* mengambil parameter dari property *Text* tiap Linkbutton yang diambil dari *sender.text*. *Sender* adalah parameter default dari prosedur *built-in* yang berarti "diri sendiri", sehingga dalam konteks ini *sender* adalah Linkbutton yang sedang diklik oleh pengguna (baik kesatu, kedua ataupun ketiga).

---

---

5. Lalu double klik di Linkbutton *Admin Login* untuk mengetikkan listing yang akan menampilkan komponen Panel yang berisi layout untuk proses login.

```
Panel1.Visible = True
```

6. Di dalam Panel, tepatnya di Button *Login*, dobel klik untuk mengetikkan listing yang akan membawa pengguna level administrator menuju ke halaman *admin.aspx*.

```
If IO.File.ReadAllText(xFolder & _
    "Password.txt") = _
    TextBox1.Text Then
    Label1.Text = ""
    Session("login") = True
    Response.Redirect("admin.aspx")
Else
    Label1.Text = "Login error !"
End If
```

7. Sedangkan listing untuk Button *Cancel* adalah:

```
Panel1.Visible = False
Session.Clear()
```



## **Testing dan Pengembangan**

---

Setelah menyelesaikan pembuatan situs company profile ini, maka dapat dilakukan proses testing dengan mengikuti langkah-langkah berikut ini:

1. Pastikan terlebih dulu bahwa file *Password.txt* telah memiliki isi teks yang akan dijadikan acuan untuk proses login.
2. Kemudian eksekusi situs tersebut hingga muncul ke browser (dengan asumsi pembuatan situs telah berhasil dan proses kompilasi tidak mengalami kesalahan), kemudian masuk terlebih dulu ke halaman *admin.aspx* melalui proses login (klik di Linkbutton *Admin Login*).
3. Di dalam halaman administrator tersebut, isikan masing-masing teks untuk setting header, footer dan logo yang akan ditampilkan. Juga isikan teks untuk masing-masing file teks yang akan ditampilkan yaitu untuk *home*, *profile* serta bagian *contact us*. Jangan lupa untuk melakukan penyimpanan di tiap pengisian tersebut.



4. Lalu klik Linkbutton *Logout* untuk melihat hasil dari pengisian tersebut.



5. Untuk kepentingan testing, cek pula kondisi saat proses login dengan melakukan entri password yang tidak sesuai untuk melihat hasil dari pesan error.

Dari hasil situs ini, dapat dipikirkan beberapa pengembangan antara lain :


1. Dapat ditambahkan fitur AJAX di dalam halaman utama web
2. Dapat ditambahkan fitur untuk dapat melakukan entri file teks dengan jumlah sesuai dengan

keinginan pengguna, sehingga tidak terbatas pada tiga file teks tersebut.

3. Dapat pula dibuat tambahan kustomisasi *sidebar* di bagian seksi kiri, sehingga dapat disisipi fitur lain seperti counter atau banner.

## **Online Lottery**

---



**Bahkan mentari yang tenggelam di sore hari masih akan terbit di esok pagi, begitu pula yang harus terjadi pada semangat manusia**

**Level :** Pemula

**Tujuan :**

- Membuat sebuah situs model lotere dan mengacak tiga digit angka agar bisa cocok dengan *lucky number* yang tersedia
- Mengaplikasi teknik AJAX secara sederhana

**Fitur situs :**

- Penggunaan ASP .NET AJAX, khususnya untuk penggunaan Timer secara majemuk dan simultan
- Menggunakan komponen UpdatePanel dari ASP .NET AJAX
- Menggunakan fungsi *Rnd* untuk mendapatkan angka acak dengan jangkauan tertentu

**Estimasi waktu pembuatan :** 30 – 60 menit

## **Pengantar**

---

---

Pada saat penulis mengeluarkan buku mengenai *AJAX dengan ASP .NET 2.0*, terdapat banyak respon mengenai contoh-contoh yang ada dalam buku tersebut. Salah satu respon dari pembaca menginginkan sebuah contoh situs yang konkrit khususnya untuk implementasi komponen Timer yang ada di dalam ASP .NET AJAX.

Penggunaan Timer dalam ASP .NET AJAX memang seringkali terlupakan untuk situs-situs yang berbasis database. Itulah salah satu alasan dari pembuatan contoh situs *Online Lottery* ini. Dengan mengambil konsep seperti layaknya penarikan sebuah undian yang dilakukan secara online, maka situs ini juga mampu melakukan simulasi penggunaan Timer secara majemuk.

Dan apabila teknik AJAX dilakukan dengan baik, maka pengguna umumnya tidak merasakan bahwa mereka sedang berada dalam sebuah aplikasi berbasis web, bukan di sebuah aplikasi berbasis desktop.

Contoh situs ini sangatlah sederhana dan diharapkan dapat diselesaikan dalam waktu yang sangat cepat. Hal ini disebabkan situs ini hanya memiliki sebuah halaman web dan sebuah file style sheet tanpa ada unsur penyimpanan data, baik berupa

file teks ataupun database. Sehingga dapat menjadi contoh cepat dan mampu memancing para pemula untuk lebih dalam bereksplorasi dengan fitur-fitur yang ada di dalam ASP .NET AJAX.

## Style Sheet : Lottery.css

---

Langkah pertama dalam pembuatan situs ini adalah dengan membuat website baru di dalam lingkup Visual Studio atau Visual Web Developer Express Edition. Kemudian pilih tipe ASP .NET Website sehingga di dalam website yang baru langsung tersedia satu halaman web *default.aspx* yang akan dimodifikasi di bagian berikutnya.



---

Khusus untuk Anda yang masih menggunakan Visual Web Developer Express Edition 2005 atau Visual Studio 2005, maka Anda wajib memilih tipe website *ASP .NET AJAX Website* dengan asumsi bahwa Anda telah melakukan instalasi ASP .NET AJAX yang dapat didownload dari situs [www.asp.net](http://www.asp.net)

---

Selanjutnya, buatlah sebuah file style sheet baru dengan nama *Lottery.css* dengan isi listing sebagai berikut :

```
body
{
    font-family:Arial;
    font-size:10pt;
}
```



```
.kotakAngka
{
    font-size:52pt;
    font-weight:bold ;
    text-align:center;
    border:dashed 2px red;
    color:Blue;
    padding:1px;
}
```

Di dalam style sheet tersebut hanya terdapat dua bagian yaitu satu bagian *body* yang menyatakan jenis dan ukuran font default untuk layout situs, dan satu class style sheet *kotakAngka* yang akan digunakan untuk mendefinisikan style kotak angka yang akan diacak sebagai simulasi lotere.

Dalam kotak angka tersebut, nantinya ukuran font akan dibuat sebesar mungkin dan terletak di dalam kotak berwarna merah dengan warna angka biru. Sedangkan sintaks *padding* berfungsi untuk menentukan jarak pixel antara *border* dengan angka, sehingga angka tidak terlihat terhimpit dengan garis batasnya.

## Halaman Utama : Default.aspx

---

Langkah berikutnya adalah dengan melakukan modifikasi layout serta pengetikan listing untuk halaman utama dan satu-satunya dalam situs ini. Untuk memodifikasi layout dari halaman web ini, berpindahlah ke mode Source dan ikuti langkah berikut :

1. Tepat di bawah tag `<title>...</title>` sisipkan listing untuk menempelkan style sheet ke dalam halaman web.

```
<link href="Lottery.css"
      rel="stylesheet" type="text/css" />
```

2. Kemudian berpindahlah lagi ke dalam mode Design dan ketikkan teks *Online Lottery*. Untuk melakukan format pada teks tersebut, terlebih dulu blok teks tersebut, kemudian di bagian toolbar set teks tersebut dengan mengaturnya seperti pada gambar berikut :



3. Selanjutnya, drag sebuah komponen *ScriptManager* di bagian paling atas dari halaman web.

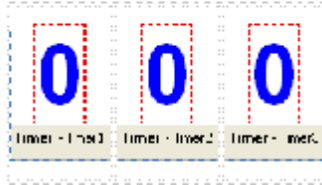


4. Lalu di bawah teks *Online Lottery* sisipkan sebuah Table dengan jumlah baris/row sebanyak 1 dan jumlah kolom/column sebanyak 3. Di dalam kolom pertama, drag sebuah komponen UpdatePanel, dan drag sebuah Label di dalam UpdatePanel tersebut. Untuk Label yang telah didrag, set property *CssClass* menjadi *kotakAngka* agar dapat berubah style seperti CSS yang telah didefinisikan sebelumnya. Set juga property *Text* label tersebut menjadi angka 0 untuk nilai awal. Kemudian drag sebuah komponen Timer ke dalam UpdatePanel yang sama, dan set property *Enabled* menjadi *False* dan property *Interval* menjadi 50 (yang berarti Timer akan melakukan update setiap 50 milidetik). Set juga property *UpdateMode* pada UpdatePanel menjadi *Conditional* (yang berarti bahwa isi dari UpdatePanel tersebut hanya akan direfresh dengan kondisi tertentu. Lakukan langkah ini, untuk kolom kedua dan ketiga dari Table tersebut, sehingga

terdapat tiga buah UpdatePanel dengan isi masing-masing sebuah Label dan sebuah Timer.

Script Manager - ScriptManager1

## ONLINE LOTTERY



5. Lalu di bawah Table tersebut, drag dua buah Button, yang masing-masing property *Text* dari Button tersebut diisi dengan nilai *Start* dan *Stop*.
6. Selanjutnya, di bawah Button, drag sebuah UpdatePanel baru (jangan lupa untuk melakukan setting property *UpdateMode* menjadi *Conditional*) dan didalamnya ketikkan teks *Today Lucky Number is :* dan disebelahnya drag sebuah Label baru, begitu juga drag sebuah Label tepat dibawahnya.



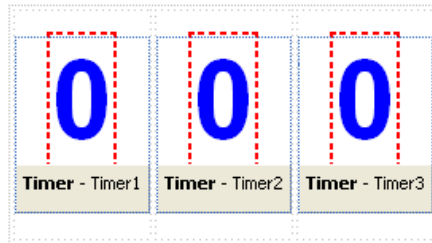
Perlu diingat bahwa komponen UpdatePanel tidak akan tampak oleh pengguna di dalam browser. Penempatan UpdatePanel di contoh situs berikutnya umumnya malah dilakukan pada

saat terakhir proses layout. Hal ini dikarenakan UpdatePanel secara otomatis akan mengisolir komponen yang ada didalamnya, sehingga seringkali terjadi kesulitan jika terdapat setting property secara interaktif,

---

---

## ONLINE LOTTERY



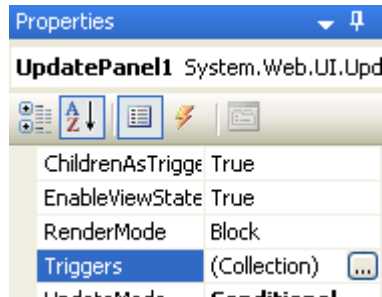
Start

Stop

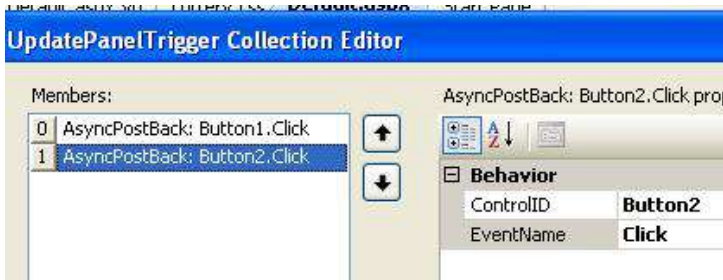
**Today Lucky Number is : [Label4]  
[Label5]**

---

7. Kini, klik di UpdatePanel yang pertama, lalu di jendela property tepat di property *Triggers*, klik tombol kecil untuk menuju kotak dialog setting Triggers.



Di dalam kotak dialog tersebut, tambahkan dua Trigger *AsyncPostBack* ke *Button1* dan *Button2* dan dua-duanya ke event *Click*.



---

Ini berarti bahwa UpdatePanel tersebut hanya akan direfresh pada saat penekanan *Button1* dan *Button2* dan pada saat penekanan tersebut, proses refresh tidak akan mempengaruhi bagian halaman web yang lain. Lakukan setting ini pada dua UpdatePanel yang lain di kolom kedua dan kolom ketiga dan juga pada UpdatePanel yang terletak di bagian paling bawah halaman web.

---

Berikutnya, lakukan langkah-langkah berikut untuk menyelesaikan listing situs ini:

1. Dobel klik di bagian kosong halaman web dan di dalam prosedur *Page\_Load* ketikkan listing berikut:

```
If Not IsPostBack Then _  
    randomNumber(Label14, 999, 100)
```

2. Kemudian ketikkan prosedur *randomNumber* yang nanti akan dieksekusi untuk mendapatkan bilangan acak bulat dengan batas atas dan batas bawah tertentu. Perlu diingat terlebih dulu bahwa fungsi *rnd* hanya akan menghasilkan bilangan desimal antara 0 dan 1, sehingga diperlukan bilangan pengali untuk mendapatkan bilangan bulat dengan batas atas dan bawah yang lebih besar.

```
Sub randomNumber _  
    (ByVal xLabel As Label, _  
    ByVal xtop As Integer, _  
    ByVal xbottom As Integer)  
    xLabel.Text = _  
    CInt(Int((xtop) * _  
        Rnd() + xbottom))  
End Sub
```

3. Lalu, dobel klik pada Timer, dan modifikasi prosedur untuk *Timer1\_Tick* menjadi sebagai berikut :

```
Protected Sub Timer1_Tick _
```

---

```
(ByVal sender As Object, _  
ByVal e As System.EventArgs) _  
Handles Timer1.Tick, _  
Timer2.Tick, _  
Timer3.Tick  
Select Case sender.id  
Case "Timer1"  
randomNumber(Label1, 9, 1)  
Case "Timer2"  
randomNumber(Label2, 9, 1)  
Case "Timer3"  
randomNumber(Label3, 9, 1)  
End Select  
End Sub
```



---

Modifikasi prosedur *Timer1\_Tick* menyatakan bahwa prosedur ini juga akan dieksekusi oleh *Timer2* dan *Timer3*. Dan untuk membedakan eksekusi tiap Timer, dibedakan dengan mengenali property *ID* dari tiap Timer tersebut. Saat interval Timer memenuhi syarat (yaitu tiap 50 milidetik) maka akan dilakukan proses pengacakan angka dengan batas atas 9 dan batas bawah 0 (bukan 1, karena batas bawah yang dipanggil dari fungsi *randomNumber* dikurangi dengan 1). Ini akan menyebabkan ketiga angka yang



ada mengalami proses random secara  
simultan secara cepat.

---

---

4. Kini dobel klik pada Button *Start* dan ketikkan listing berikut ini:

```
Timer1.Enabled = True  
Timer2.Enabled = True  
Timer3.Enabled = True  
Label5.Text = ""
```

5. Listing terakhir adalah pada Button *Stop* yang nanti akan berfungsi untuk menghentikan proses pengacakan angka dan melakukan proses pencocokan dengan angka keberuntungan yang didefinisikan pada saat halaman web pertama kali ditampilkan (pada prosedur *Page\_Load*).

```
Timer1.Enabled = False  
Timer2.Enabled = False  
Timer3.Enabled = False  
If CInt(Label1.Text & _  
Label2.Text & _  
Label3.Text) = Label4.Text Then  
Label5.Text = _  
"Congratulation ! You win !"  
Else  
Label5.Text = _
```

"You lose...Try again..."

End If

## Testing dan Pengembangan

---

Dari pembuatan situs sederhana ini, testing dapat dilakukan dengan cara berikut :

1. Saat situs dieksekusi dan tampil di browser, klik tombol atau Button *Start* untuk memulai permainan. Dan untuk mendapatkan angka keberuntungan, klik Button *Stop*.

### ONLINE LOTTERY



Today Lucky Number is : 425  
You lose...Try again...

2. Perhatikan proses pengacakan angka, tidak akan mempengaruhi bagian halaman web yang lain dan menunjukkan bahwa penggunaan AJAX telah berhasil.

Pengembangan yang dapat dilakukan dalam situs ini antara lain :


1. Menambahkan level permainan sehingga dapat diatur kecepatan dari interval Timer maupun range angka yang akan ditebak.

2. Menyimpan hasil statistik permainan, berapa kali pemain mengalami kekalahan maupun kemenangan saat tepat menebak angka keberuntungan.

**Halaman ini sengaja dikosongkan**

## **Personal Reminder**

---



Saat berusaha berenang menyeberang sungai mungkin saja tenggelam atau sampai ke seberang, tetapi jika tidak pernah berusaha berenang pasti Anda takkan pernah sampai ke seberang dan tetap tenggelam

**Level :** Pemula - Menengah

**Tujuan :**

- Membuat situs yang mampu menampilkan data agenda harian
- Mempelajari cara akses database dari SQL Server Express Edition 2005

**Fitur situs :**

- Penggunaan dan pembuatan database SQL Server Express Edition 2005 (dan dapat juga diaplikasikan untuk SQL Server edisi yang lain)
- Mengkoneksikan komponen akses data dalam ASP .NET (GridView dan Formview) dengan tabel yang ada dalam database
- Modifikasi komponen akses data dengan komponen lain
- Melakukan proses data binding, baik dari *design time* maupun dari *run time*
- Memanfaatkan komponen Multiview dan View untuk membuat situs dengan teknik *single page solution*

**Estimasi waktu pembuatan :** 60 – 100 menit

## **Pengantar**

---

---

Jika Anda merasa sebagai pemula yang belum pernah sama sekali membuat situs dengan menggunakan ASP .NET (versi manapun), memang sangat disarankan untuk tidak langsung meloncat ke contoh yang ada di dalam bab ini. Meski demikian, bukan berarti bab ini mustahil dikerjakan oleh para pemula. Karena dengan struktur situs yang sangat sederhana (hanya menggunakan satu halaman web) sehingga dapat juga diselesaikan oleh pemula secara cepat.

Dalam contoh situs ini, akan didemonstrasikan mengenai cara akses database dalam sebuah situs menggunakan ASP .NET khususnya database dari SQL Server Express Edition. Tetapi dengan cara yang hampir sama, juga dapat diaplikasikan ke SQL Server dengan edisi yang lain. Cara akses yang dibahas, mulai dari pembuatan, cara akses ke komponen data secara otomatis (dengan cara drag-drop atau lazim disebut dengan cara *design time*) maupun dengan cara memodifikasi query dari listing program (juga disebut sebagai cara akses *run time*).

Contoh situs ini hanya menggunakan satu halaman web dengan memanfaatkan komponen Multiview yang memiliki banyak komponen View



didalamnya. Untuk sebuah situs yang profesional, masih banyak pihak yang meragukan tingkat keamanan penggunaan Multiview, terutama untuk menyembunyikan fitur penting dalam sebuah situs (seperti yang dilakukan dalam contoh ini). Tetapi penggunaan Multiview sangat membantu untuk melakukan penyederhanaan jumlah halaman web dalam sebuah situs.

## **Pembuatan Database : PersonalReminder.mdf**

---

---

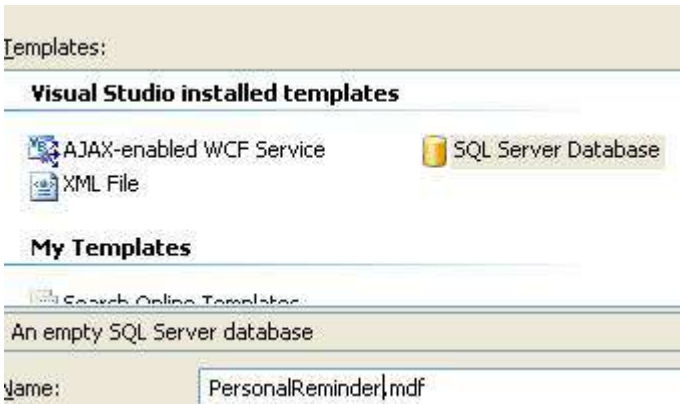
Langkah pertama dalam pembuatan situs ini sama dengan langkah yang juga dilakukan di contoh lainnya, yaitu dengan membuat sebuah solution website baru di lingkup Visual Studio atau Visual Web Developer Express Edition dengan tipe *ASP .NET Website*.

Kemudian perhatikan bahwa didalamnya akan langsung terdapat sebuah folder bernama *App\_Data*. Secara default, tiap database akan langsung diarahkan ke folder tersebut dengan asumsi bahwa folder tersebut tidak akan terlihat dan tidak dapat diakses oleh pengguna di dalam browser.

Kini untuk pembuatan database dalam situs ini, klik kanan di folder *App\_Data* dalam Solution Explorer dan pilih sub menu *Add New Item*.



Dalam kotak dialog yang tersedia, pilih tipe *SQL Server Database* dan beri nama database tersebut dengan nama *PersonalReminder.mdf*



---

---

Jika Anda mengalami kegagalan pada pembuatan database, terdapat beberapa kemungkinan penyebab kegagalan tersebut. Salah satunya adalah service dari SQL Server Express tidak sedang dalam keadaan running. Untuk mengatasinya cobalah mengakses aplikasi *SQL Server Configuration Manager* dan pastikan bahwa service SQL Server dalam keadaan running.


---


---

Setelah selesai, dobel klik pada database yang telah terbentuk untuk menuju ke jendela *Database Explorer* (jika Anda menggunakan Visual Studio akan menuju ke jendela *Server Explorer*). Kini klik di dalam treewiew untuk database tersebut, tepatnya di item *Table* lalu klik kanan dan pilih sub menu *Add New*

Table. Pada kotak dialog tabel baru tersebut, buat tabel dengan struktur seperti pada gambar:

Column Name	Data Type	Allow Nulls
IDSchedule	int	<input type="checkbox"/>
DateSchedule	datetime	<input type="checkbox"/>
DateToSchedule	datetime	<input type="checkbox"/>
TitleSchedule	varchar(50)	<input type="checkbox"/>
TextSchedule	varchar(MAX)	<input checked="" type="checkbox"/>

Lalu klik di field *IDSchedule* dan jadikan field tersebut sebagai *primary key* dengan cara melakukan klik di icon  yang ada dalam Toolbar.

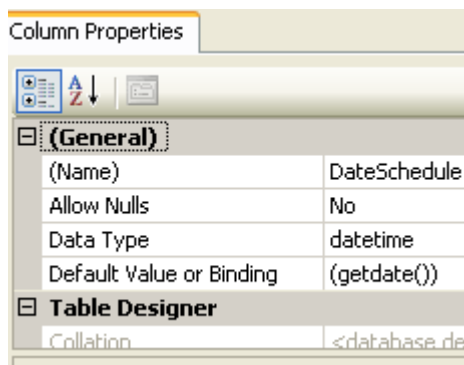
Column Name
 IDSchedule

Kemudian, berpindahlah ke bagian bawah dari property field, dan di bagian property *Identity Specification* ganti nilai *IsIdentity* menjadi *yes*, sehingga nantinya field ini akan memiliki nilai otomatis angka yang bersifat incremental (berurutan) pada saat terjadi proses penyimpanan data baru.

Mas Non-SQL Server Subscriber		NO
<input checked="" type="checkbox"/> Identity Specification	(Is Identity)	Yes
	Identity Increment	1

Kemudian klik di field *DateSchedule* dan di dalam jendela *Column Property* set property *Default Value or Binding* menjadi *GETDATE()* yang berarti

bahwa isi dari field tersebut secara otomatis akan diisi dengan tanggal sistem dari server.



Selanjutnya, tutuplah kotak dialog tersebut, dan simpan tabel yang baru terbentuk dengan nama *Schedule*.

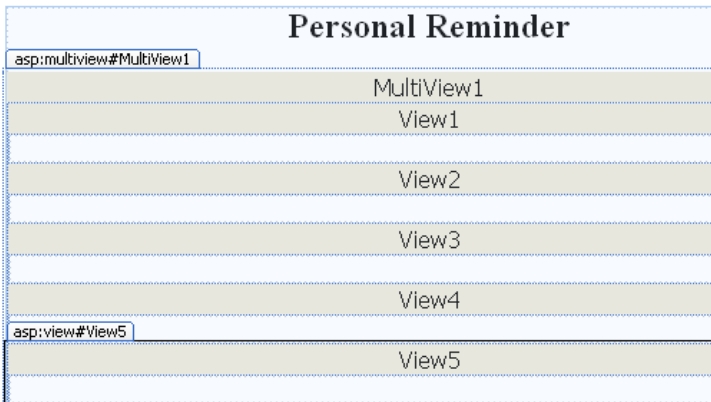


## Halaman Utama : Default.aspx

---

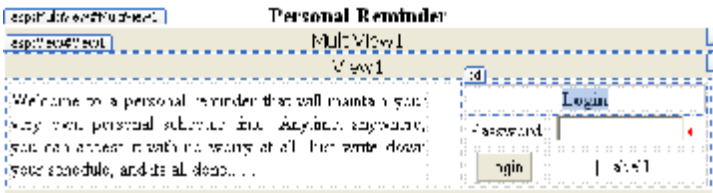
Kini saatnya untuk memodifikasi layout dari halaman utama situs ini. Berikut adalah yang harus dilakukan untuk memodifikasi layout:

1. Ketikkan teks untuk judul situs di bagian atas halaman web yaitu *Personal Reminder*.
2. Di bawah teks tersebut, drag sebuah komponen Multiview, dan langsung double klik komponen View di dalam Multiview tersebut sebanyak 5 kali, sehingga terbentuk sebuah Multiview dengan 5 komponen View didalamnya.



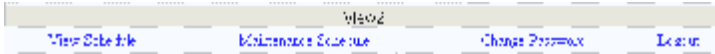
Set Property *ActiveViewIndex* dalam Multiview tersebut menjadi 0, sehingga pada saat halaman web muncul akan langsung menuju ke View yang pertama.

3. Kemudian di dalam View yang pertama tempatkan sebuah tabel dengan jumlah baris sebanyak 1 dan jumlah kolom sebanyak 3. Di dalam kolom pertama ketikkan teks pengantar, kemudian kolom kedua dikosongkan sebagai jarak dengan kolom ketiga. Dan di kolom ketiga tempatkan lagi sebuah tabel dengan jumlah baris sebanyak 3 dan jumlah kolom sebanyak 3. Di baris pertama, lakukan operasi *merge cells* dan ketikkan teks *Login*. Sedangkan di baris kedua kolom pertama ketikkan teks *Password*, lalu drag sebuah Textbox di kolom yang kedua dengan property *TextMode* diarahkan ke nilai *Password*. Untuk baris ketiga, di kolom pertama drag sebuah Button dan set property *Text* menjadi *Login* dan di kolom kedua drag sebuah Label dan kosongkan isi dari property *Text* untuk Label tersebut.

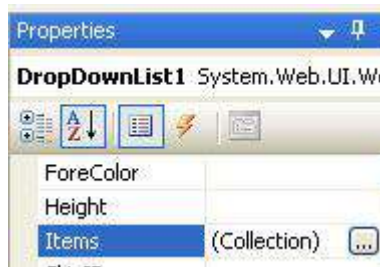


4. Kini, di dalam View yang kedua, tempatkan sebuah tabel dengan jumlah baris sebanyak satu dan jumlah kolom sebanyak empat. Di tiap kolom tersebut, tempatkan masing-masing sebuah

Linkbutton yang property *Text*nya diisng dengan *View Schedule*, *Maintenance Schedule*, *Change Password* dan *Logout*.



5. Selanjutnya View yang ketiga, ketikkan teks *Show Schedule* yang diikuti dengan sebuah Dropdownlist disebelahnya. Pada Dropdownlist tersebut, set property *AutoPostBack* menjadi *True*. Kemudian ketikkan item dari Dropdownlist melalui tombol kecil yang ada di property *Item*.



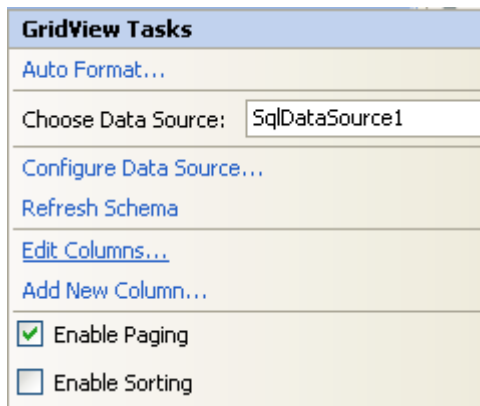
Kemudian isikan item Dropdownlist seperti pada gambar



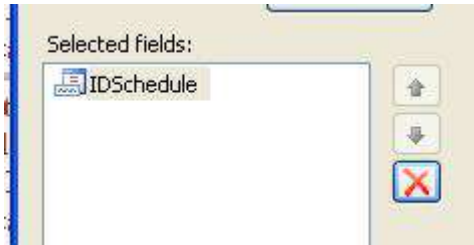


6. Berikutnya, dari jendela Database Explorer (atau Server Explorer jika Anda menggunakan Visual Studio), drag tabel *Schedule* ke bawah Dropdownlist yang baru dibuat. Hasil dari operasi drag tabel ini akan membentuk dua komponen baru yaitu Gridview dan SqlDataSource. Selain itu, hasil operasi drag tersebut juga akan memodifikasi file web.config secara otomatis dengan menambahkan tag *connectionstring* yang berisi deklarasi koneksi ke dalam database.

Klik kanan pada Gridview, dan pilih opsi *Enable Paging* agar Gridview dapat terpisah menjadi halaman yang berbeda jika record yang akan ditampilkan banyak. Lalu pilih sub menu *Edit Columns* untuk mengedit tampilan Gridview.



Di dalam kotak dialog yang tersedia, hapus semua field kecuali satu field yaitu *IDSchedule*.



Setelah itu field yang tersisa dikonversi menjadi *Template Field* dengan mengeksekusi Linkbutton *Convert this field into a TemplateField* sehingga nantinya dapat dimodifikasi dengan menggunakan tag HTML.



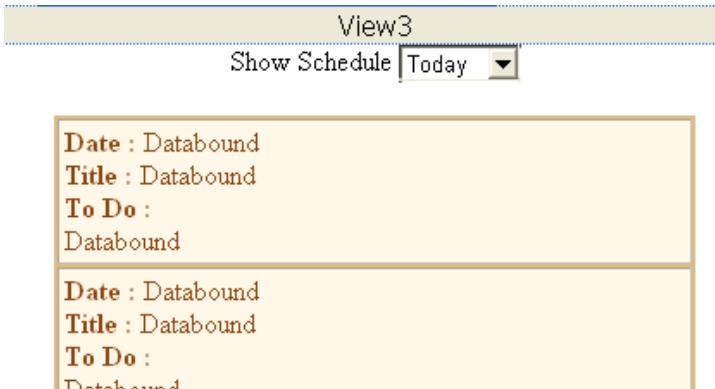
Selanjutnya berpindahlah ke mode Source, dan carilah rangkaian tag `<Columns>...</Columns>` di dalam GridView, dan modifikasi menjadi listing berikut ini :

```
<asp:TemplateField
  HeaderText="IDSchedule"
  SortExpression="IDSchedule">
  <ItemTemplate>
    <p style="text-align:justify">
```

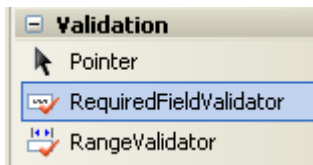
```
<b>Date      : </b>
<%=Format(Eval("DateToSchedule"), _
          "dd-MM-yyyy")%> <br />
<b>Title     : </b>
<%=eval("TitleSchedule") %><br />
<b> To Do    : </b> <br />
<%=Eval("TextSchedule")%></p>
</ItemTemplate>
</asp:TemplateField>
```

Pada field jenis template tersebut, dimodifikasi sehingga di field tersebut nantinya dapat menampilkan data dalam bentuk tabular, tidak lagi kolumnar (berdasar kolom atau field). Dalam listing layout tersebut terdapat *injeksi script server side* yaitu listing yang diapit dengan tanda tag `<%=#....%>` yang berarti bahwa semua yang ada dalam tag tersebut merupakan script server side (termasuk didalamnya nanti pemanggilan fungsi atau prosedur dari listing program). Perintah *Eval* berarti memanggil field yang bersesuaian dan ditampilkan ke dalam Gridview.

Setelah selesai, kembalilah lagi ke mode Design, dan set property *ShowHeader* Gridview menjadi *False*. Untuk kepentingan layout, Anda juga dapat mengganti *Auto format* dari Gridview sesuai selera.



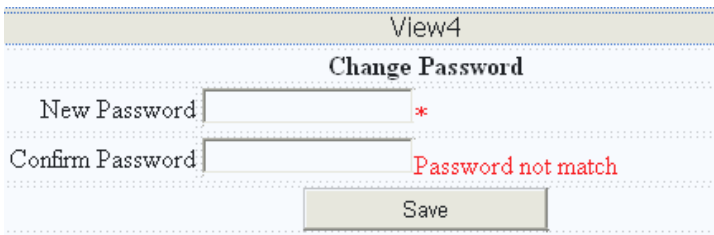
7. Di dalam View yang keempat, tempatkan sebuah tabel dengan jumlah baris sebanyak empat dan jumlah kolom sebanyak dua. Di baris yang pertama lakukan operasi *Merge Cells* dan ketikkan judul tabel yaitu *Change Password*
- Setelah itu di baris kedua, untuk kolom pertama ketikkan teks *New Password* dan di kolom kedua drag sebuah *Textbox* dan sebuah komponen *RequiredFieldValidator* disebelahnya.



Untuk komponen *RequiredFieldValidator*, set property *Error Message* menjadi tanda asterisk (\*), dan property *ControlToValidate* mengarah ke *Textbox3*.

Di baris yang yang ketiga kolom pertama ketikkan teks *Change Password*, dan di kolom yang kedua, drag lagi sebuah Textbox serta disebelahnya drag sebuah komponen *CompareValidator*. Pada komponen *CompareValidator* tersebut set property *ErrorMessage* menjadi *Password not match*. Lalu di property *ControlToValidate* diarahkan ke *Textbox4* sedangkan property *ControlToCompare* diset menjadi *Textbox3*. Untuk tiap Textbox yang telah didrag set masing-masing property *TextMode* menjadi mode *Password*.

Selanjutnya di baris terakhir, lakukan operasi *Merge cells* sehingga menjadi satu kolom dan drag sebuah Button dengan property *Text* berisi teks *Save*.



The screenshot shows a web form titled "View4" with the heading "Change Password". It contains two password input fields: "New Password" with a red asterisk and "Confirm Password" with a red error message "Password not match". A "Save" button is at the bottom.



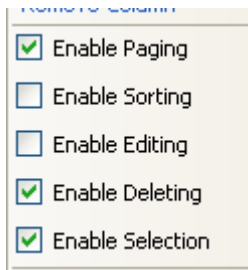
Komponen jenis validator merupakan alat bantu yang dapat melakukan validasi terhadap isian pengguna. Untuk komponen *RequiredFieldValidator* berfungsi sebagai validator agar isian pengguna tidak kosong.

Sedangkan komponen CompareValidator dalam contoh ini digunakan untuk membandingkan isi antara dua Textbox agar pengguna tidak salah memasukkan password yang baru, sebab saat melakukan pengisian Textbox dalam keadaan *masked* atau tertutup oleh karakter bullet.

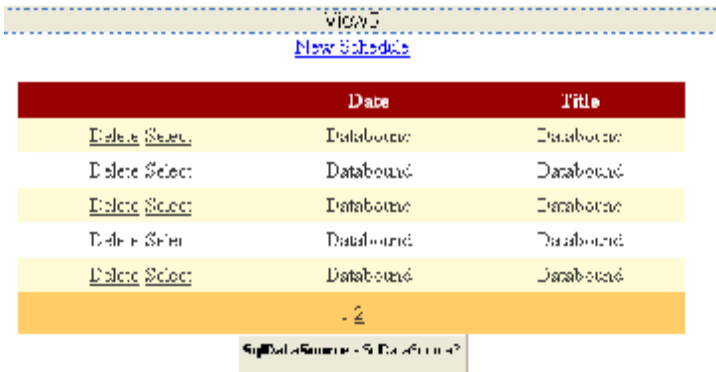
---

---

8. Kini di View yang terakhir, drag terlebih dulu sebuah Linkbutton dengan property *Text* yang diisi dengan teks *New Schedule*. Lalu di bawah Linkbutton tersebut, drag tabel *Schedule* dari Database Explorer sehingga terbentuk sebuah Gridview dan sebuah SqlDataSource yang baru. Klik pada *Smart Tag* (tanda panah kecil yang umumnya terletak di pojok kanan atas untuk menampilkan menu bantuan komponen) yang ada di Gridview, dan pilih opsi *Enable Paging*, *Enable Deleting* dan *Enable Selection*. Lalu pilih sub menu *Edit Columns*.



Di dalam kotak dialog untuk pemilihan kolom tersebut, sisakan dua field yakni field *DateToSchedule* dan field *TitleSchedule*. Di field *TitleSchedule* set property *HeaderText* menjadi *Title*. Untuk field *DateToSchedule* atur property *HeaderText* menjadi *Date* dan property *DataFormatString* menjadi *{0:dd-MM-yyyy}*.



Lalu drag lagi tabel *Schedule* di bawah GridView tersebut, tetapi hapus GridView baru yang terbentuk dari hasil drag tabel dan sebagai gantinya drag sebuah komponen Formview. Pada Formview tersebut, klik di Smart Tag dan set property *DataSource* ke *SqlDataSource3*. Kemudian set property *DefaultMode* dari Formview tersebut menjadi *Insert*.

**FormView Tasks**

Auto Format...

Choose Data Source:

Langkah berikutnya adalah melakukan edit dari template Formview. Berpindahlah ke mode Source dan di dalam tag Formview modifikasi menjadi listing layout berikut:

```
<EditItemTemplate>
<center>
<b>...:Edit Schedule:...</b></center>
<table>
  <tr>
    <td>Date</td>
    <td>
      <asp:Calendar ID="Calendar2"
        runat="server"
        SelectedDate=
        '<%# Bind("DateToSchedule") %>'
        Width="200px">
      </asp:Calendar>
    </td>
  </tr>
  <tr>
    <td>Title</td>
    <td>
      <asp:TextBox
        ID="TitleScheduleTextBox"
```



```
        runat="server"
        Text=
'<%# Bind("TitleSchedule") %>' />
    </td>
</tr>
<tr>
    <td colspan="2">
        <asp:TextBox
            ID="TextScheduleTextBox"
            runat="server"
            Height="86px"
            Text=
'<%# Bind("TextSchedule") %>'
            TextMode="MultiLine"
            Width="273px" />
    </td>
</tr>
<tr>
    <td>
        <asp:LinkButton
            ID="UpdateButton"
            runat="server"
            CausesValidation="True"
            CommandName="Update"
            Text="Update" />
    </td>
    <td>
```

```
<asp:LinkButton
  ID="UpdateCancelButton"
  runat="server"
  CausesValidation="False"
  CommandName="Cancel"
  Text="Cancel" />
</td>
</tr>
</table>
</EditItemTemplate>
<InsertItemTemplate>
<center>
<b>...:Input Schedule:...</b></center>
<table>
  <tr>
    <td>Date</td>
    <td>
      <asp:Calendar
        ID="Calendar1"
        runat="server"
        SelectedDate=
          '<%#now.date %>'>
      </asp:Calendar>
    </td>
  </tr>
  <tr>
    <td>Title</td>
```

```
<td>
    <asp:TextBox
        ID="TitleScheduleTextBox"
        runat="server"
        Text=
'<%# Bind("TitleSchedule") %>' />
    </td>
</tr>
<tr>
    <td colspan="2">
        <asp:TextBox
            ID="TextScheduleTextBox"
            runat="server"
            Height="101px"
            Text='<%#
                Bind("TextSchedule") %>'
            TextMode="MultiLine"
            Width="295px" />
    </td>
</tr>
<tr>
    <td>
        <asp:LinkButton
            ID="InsertButton"
            runat="server"
            CausesValidation="True"
            CommandName="Insert"
```

```
        Text="Save" />
    </td>
    <td>
        <asp:LinkButton
            ID="InsertCancelButton"
            runat="server"
            CausesValidation="False"
            CommandName="Cancel "
            Text="Cancel" />
    </td>
</tr>
</table>
</InsertItemTemplate>
```

...: Input Schedule ...:

< May 2009 >						
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Date

Title

Databound

[Save](#) [Cancel](#)

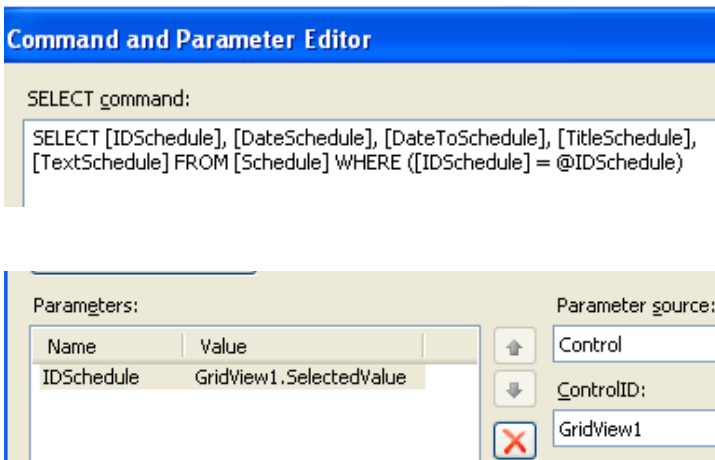


Perlu diperhatikan bahwa listing tersebut tidak perlu diketikkan semuanya, karena sebenarnya listing tersebut hanyalah modifikasi dari listing template Formview yang telah ada. Sedangkan hasil dari capture gambar, merupakan hasil dari pilihan *AutoFormat* di komponen Formview. Secara singkat, modifikasi Formview tersebut lebih menekankan pada penempatan Calendar

sebagai pengganti Textbox biasa di dalam Formview agar saat pengguna memasukkan tanggal tidak mengalami kesulitan menentukan format tanggal yang tepat. Sedangkan penempatan layout sesungguhnya adalah tergantung dari selera Anda masing-masing.

---

9. Selanjutnya di komponen *SqlDataSource3* (komponen *SqlDataSource* terakhir yang didrag), klik pada property *SelectQuery*, lalu klik di tombol kecil pada nilai property tersebut untuk menuju kotak dialog yang telah disediakan. Pada kotak dialog, set perintah query dan parameternya seperti pada gambar berikut :



Set juga di property *InsertQuery* seperti pada kotak dialog berikut ini :

### Command and Parameter Editor

INSERT command:

```
INSERT INTO Schedule(DateToSchedule, TitleSchedule, TextSchedule) VALUES  
(@DateToSchedule, @TitleSchedule, @TextSchedule)
```

Dan terakhir, set property *UpdateQuery* seperti pada gambar :

### Command and Parameter Editor

UPDATE command:

```
UPDATE Schedule SET DateToSchedule = @DateToSchedule, TitleSchedule =  
@TitleSchedule, TextSchedule = @TextSchedule WHERE (IDSchedule = @IDSchedule)
```



Perhatikan bahwa edit tiap perintah query yang telah dilakukan merupakan penyesuaian dari field-field yang akan dimanipulasi dalam halaman web khususnya di View yang terakhir.

Dalam perintah *Select* diambil sebuah parameter yang berjenis *Control* dari Gridview, sehingga saat Gridview diklik oleh pengguna, maka Formview akan secara otomatis melakukan tampilan record yang bersesuaian.

Sedangkan di perintah *Insert* terdapat beberapa field yang ditinggalkan dalam parameter karena telah memiliki *default value* (lihat lagi di sub bab pembuatan

database). Dan di perintah *Update* hanya terdapat tiga field yang boleh diedit oleh pengguna karena di dalam Formview telah didefinisikan terlebih dulu.

---

---

10. Terakhir, drag sebuah Linkbutton di bawah komponen Multiview dan set property *ID* menjadi *linkBack* dan property *Text* berisi nilai *Back*. Linkbutton ini nanti akan menjadi alat eksekusi untuk kembali ke menu administrator atau di View yang kedua.

Sebelum berlanjut ke proses pembuatan listing program, maka langkah penting berikutnya adalah melakukan modifikasi pada file web.config. Pada file web.config, carilah tag *<appSetting>* dan modifikasi menjadi listing berikut :

```
<appSettings>
  <add key="Password" value="123" />
</appSettings>
```



---

---

Anda boleh mengganti isi dari atribut *value* sesuai keinginan, tetapi isi dari atribut yang sudah diganti perlu diingat dengan seksama, karena nanti akan digunakan sebagai password pada saat proses login.

---

---



Setelah selesai melakukan proses pengaturan layout, maka saatnya untuk melakukan proses pembuatan listing program berdasarkan langkah-langkah ini:

1. Dobel klik di Button *Login* untuk mengetikkan listing yang akan dieksekusi pada saat user (dengan asumsi level administrator) telah selesai melakukan pengetikan password yang kemudian akan dicocokkan dengan nilai dari atribut password di dalam file web.config. Jika ternyata pengetikan password telah sesuai, maka akan dieksekusi prosedur *changeView* yang isinya dijelaskan di langkah berikutnya.

```
If TextBox2.Text = _  
    ConfigurationManager. _  
    AppSettings.Item(0) Then  
    Session("LoggedIn") = "True"  
    changeView(1)  
End If
```

2. Berikutnya ketikkan dua buah prosedur yaitu prosedur *changeView* yang nanti akan berfungsi untuk memindahkan View yang sedang aktif di dalam Multiview dan prosedur *viewLinkBack* yang akan menentukan apakah Linkbutton *Back* perlu

ditampilkan atau tidak dalam halaman web. Linkbutton tersebut nantinya hanya muncul pada saat administrator telah berhasil melakukan login.

```
Sub changeView _  
    (ByVal indexView As Integer)  
        MultiView1.ActiveViewIndex = _  
            indexView  
        viewLinkBack()  
End Sub
```

```
Sub viewLinkBack()  
    linkBack.Visible = _  
        IIf(Session("LoggedIn") _  
            <> "", Session("LoggedIn"), _  
            False)  
    If Session("LoggedIn") = _  
        "False" Then  
        MultiView1.ActiveViewIndex = 0  
    End If  
End Sub
```

3. Kemudian double klik di bagian kosong halaman web sehingga akan menuju ke prosedur *Page\_Load* untuk menyetikkan listing berikut :

```
viewLinkBack()
```

4. Selanjutnya double klik di Linkbutton *View Schedule* untuk menyetikkan listing berikutnya yang akan berguna untuk mengubah View yang aktif ke View ketiga (karena index diawali dari angka 0). Dan agar GridView langsung menampilkan jadwal dengan parameter hari ini (dari pilihan pertama di Dropdownlist), maka indeks yang aktif di Dropdownlist diarahkan ke indeks pertama yaitu ke nol.

Pada saat Dropdownlist dipindahkan indeks aktifnya, maka akan mengeksekusi prosedur *DropDownList1\_SelectedIndexChanged* yang isinya akan diterangkan di langkah berikutnya.

```
changeView(2)
DropDownList1. _
    SelectedIndex = 0
DropDownList1_SelectedIndexChanged _
    (sender, e)
```

5. Langkah berikutnya double klik di Linkbutton *Maintenance Schedule*. Dalam Linkbutton ini akan menuju ke View yang kelima yaitu View yang berfungsi untuk melakukan proses edit dan entri data baru di jadwal. Karenanya Formview di dalam View tersebut diarahkan menjadi mode *Insert*

sehingga di keadaan awal siap untuk melakukan entri data baru.

```
changeView(4)
FormView1.ChangeMode _
    (FormViewMode.Insert)
```

6. Kini double klik di Linkbutton *Change Password* lalu ketikkan listing berikut ini :

```
changeView(3)
```

7. Selanjutnya double klik di Linkbutton *Logout* dan ketikkan listing untuk melepas proteksi sesudah user administrator melakukan proses login, sehingga keadaan halaman web kembali lagi ke mode *guest*.

```
Session("LoggedIn") = _
    "False"
changeView(0)
```

8. Langkah berikutnya adalah melakukan double klik di Dropdownlist sehingga menuju ke prosedur *DropDownList1\_SelectedIndexChanged*. Isi dari prosedur ini akan dieksekusi pada saat pengguna mengganti pilihan di dalam Dropdownlist. Perubahan di dalam Dropdownlist akan langsung dieksekusi karena property *Autopostback* sebelumnya telah diset menjadi *true*.

Di dalam prosedur ini diimplementasikan parameter query secara interaktif secara *run time* (hanya akan terlihat hasilnya pada saat situs dijalankan). Dengan menggunakan teknik seperti ini, maka hasil dari query akan lebih luas penggunaannya karena parameter query dapat diambil dari variabel apapun selama masih dalam halaman web yang sama (bahkan bisa juga mengambil dari halaman web yang lain jika menggunakan session didalamnya).

Pada percabangan yang diterapkan, di kondisi pertama (*Case 0* atau pada saat Dropdownlist diarahkan ke pilihan pertama yaitu *Today*), maka parameter query akan menghitung perbedaan tanggal di dalam tabel dengan tanggal sistem menggunakan fungsi *DateDiff* (dengan hasil 0 berarti tanggal sistem sama dengan tanggal dalam tabel). Tanggal sistem yang diambil adalah tanggal dari server dengan menggunakan fungsi *GETDATE()* (fungsi ini bukan fungsi ASP .NET tetapi fungsi dari perintah SQL khusus di SQL Server). Satu catatan penting, karena mengambil tanggal server, maka jika situs ini dihosting di server luar Indonesia (seperti server US), maka akan terjadi perbedaan waktu yang cukup signifikan dengan waktu di

Indonesia (ingat bahwa Indonesia termasuk dalam area GMT +7).

Di kondisi berikutnya mirip dengan kondisi pertama, tetapi hanya mengambil selisih tanggal untuk mendapatkan jadwal yang sudah dianggap kadaluwarsa (expired) dengan asumsi kurang dari tanggal sistem, serta jadwal yang dianggap belum terjadi (dengan asumsi tanggal di tabel masih lebih besar dari tanggal sistem).

```
With SqlDataSource1
    Dim xParam As String
    Select Case DropDownList1. _
        SelectedIndex
    Case 0
        xParam = _
            "where DATEDIFF(day, " & _
            "DateToSchedule, " & _
            "GETDATE()) =0 "
    Case 1
        xParam = _
            "where DATEDIFF(day, " & _
            " GETDATE(),DateToSchedule) " & _
            ">0 "
    Case 2
        xParam = _
            "where DATEDIFF(day, " & _
            " GETDATE(),DateToSchedule) <0 "
```

Case 3

```
xParam = ""
End Select
.SelectCommand = _
    "select * from schedule " & _
    xParam & _
    "order by idschedule desc"
.Select _
    (DataSourceSelectArguments. _
    Empty)
.DataBind()
End With
```

9. Lalu double klik di Button *Save* di dalam View yang berfungsi untuk mengganti password. Kemudian ketikkan listing berikut untuk melakukan penyimpanan password baru yang sesungguhnya melakukan edit atribut *Password* di dalam file *web.config*.

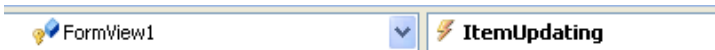
```
Dim xconfig As Configuration = _
    Web.Configuration. _
    WebConfigurationManager. _
    OpenWebConfiguration("~/")
xconfig.AppSettings. _
    Settings("Password").Value = _
    TextBox3.Text
xconfig.Save()
```

```
changeView(1)
```

10. Di dalam View untuk *Maintenance Schedule*, double klik pada Linkbutton *New Schedule*.

```
FormView1.ChangeMode _  
(FormViewMode.Insert)
```

11. Kemudian double klik pada Formview, dan gantilah pada combobox *event* (terletak di bagian kanan atas di *code view*) ke event *ItemUpdating* sehingga menuju ke dalam prosedur *FormView1\_ItemUpdating*.



Prosedur ini nanti akan dieksekusi pada saat Formview *sedang* (perhatikan akhiran *ing* yang dalam bahasa Inggris berarti *present tense* atau sedang dilakukan, sedangkan untuk prosedur dengan akhiran *ed* berarti *past tense* atau telah dilakukan) melakukan proses penyimpanan data baru.



 **ItemUpdating**


 **(Declarations)**


 DataBinding


 DataBound

 Disposed

 Init

 ItemCommand

 ItemCreated

 ItemDeleted

 ItemDeleting

 **ItemInserted**

 **ItemInserting**

 **ItemUpdated**

 **ItemUpdating**

 Load

Fungsi utama dari prosedur ini nanti akan mengarahkan parameter untuk isian field *DateToSchedule* ke komponen Calendar yang terdapat dalam Formview saat Formview dalam mode Insert. Untuk mendapatkan nilai dari komponen yang berada dalam Formview (juga di komponen lain yang bersifat komposit seperti Gridview, Datalist atau Repeater), digunakan perintah *FindControl* yang diikuti oleh nama komponen tersebut. Selanjutnya dikonversi dengan fungsi *Ctype* sesuai dengan tipe obyeknya agar dapat diambil property dari komponen tersebut.

```
With SqlDataSource3
```

```
    .UpdateParameters _
```

```
    ("DateToSchedule"). _
```

```
DefaultValue = _  
    CType(FormView1. _  
        FindControl("Calendar2"), _  
        Calendar).SelectedDate  
End With
```

12. Selanjutnya arahkan event dari Formview ke *ItemInserted* sehingga terbentuk prosedur *FormView1\_ItemInserted*. Dalam prosedur ini hanya akan mengembalikan Formview ke mode awal yakni mode *Insert* dan melakukan proses refresh di Gridview dengan metode *Databind*.

```
FormView1.ChangeMode _  
    (FormViewMode.Insert)  
GridView1.DataBind()
```

13. Masih tetap di komponen Formview, arahkan ke event *ItemInserting* sehingga terbentuk prosedur baru bernama *FormView1\_ItemInserting*

```
With SqlDataSource3  
    .InsertParameters _  
        ("DateToSchedule"). _  
        DefaultValue = _  
        CType(FormView1.FindControl _  
            ("Calendar1"), Calendar). _  
            SelectedDate  
End With
```

14. Yang terakhir adalah melakukan double klik pada GridView sehingga akan menuju ke prosedur *GridView1\_SelectedIndexChanged*. Prosedur ini nantinya secara otomatis akan mengarahkan komponen Formview ke mode edit tepat di record yang dipilih di dalam GridView. Arahkan terhadap record yang akan diedit tersebut merupakan efek dari setting query sebelumnya (lihat lagi pada pengaturan layout khususnya di property *SelectQuery* pada komponen *SqlDataSource3* atau pada komponen *SqlDataSource* yang menyertai Formview).

```
FormView1.ChangeMode _  
    (FormViewMode.Edit)
```

## **Testing dan Pengembangan**

---

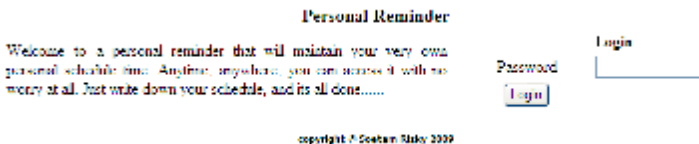
Setelah selesai melakukan proses pengaturan layout dan juga pengetikan listing program, maka kini saatnya untuk mengeksekusi situs yang telah dibuat. Jika terjadi kesalahan atau error pada saat eksekusi (proses debug), maka berikut ini adalah kemungkinan-kemungkinan yang harus dicek :

1. Perhatikan pengetikan huruf kecil dan huruf besar, terutama pada saat pengetikan untuk proses layout di Formview (misal : `calendar1`, seharusnya `Calendar1`), dan juga pada saat proses pengetikan listing.
2. Perhatikan modifikasi pada file `web.config`, pastikan bahwa tidak terjadi salah ketik untuk modifikasi di tag `appSetting`
3. Perhatikan penempatan komponen `SqlDataSource`, terutama dari hasil drag-drop tabel melalui Database Explorer atau Server Explorer. Pastikan bahwa proses drag-drop tidak terulang lebih dari satu kali sehingga nama/property ID dari `SqlDataSource` tetap seperti pada petunjuk langkah-langkah pengaturan layout.
4. Perhatikan pengetikan query pada property `SqlDataSource`, pastikan bahwa semua tanda koma

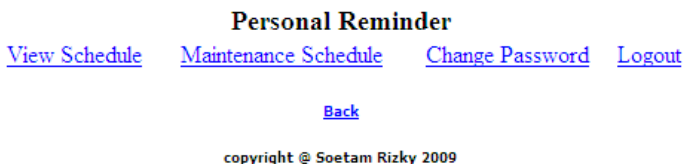
ataupun tanda kurung (brackets) telah diketikkan dengan benar.

Apabila semua telah berhasil, maka langkah proses testing adalah :

1. Saat situs pertama kali dibuka, maka akan muncul tampilan dari View yang pertama. Cobalah untuk melakukan login dengan password yang telah dientrikan di dalam file web.config (jika Anda mengikuti semua petunjuk, maka password default adalah 123). Cobalah pula dengan mengetikkan password yang salah agar muncul Label yang menyatakan kesalahan.



2. Jika login sukses, maka tampilan akan menuju ke View yang kedua yakni menampilkan menu kecil dan sebuah Linkbutton navigasi untuk kembali ke menu utama. Ini menandakan bahwa situs dalam mode administrator.



3. Selanjutnya, cobalah terlebih dulu untuk melakukan penambahan jadwal atau schedule dengan mengeksekusi Linkbutton *Maintenance Schedule*. Di dalam tampilan berikutnya, cobalah untuk menambahkan beberapa jadwal dengan minimal satu jadwal yang memiliki rentang waktu satu hari sebelum tanggal sistem, satu jadwal lagi dengan rentang satu hari setelah tanggal sistem dan minimal sebuah jadwal yang sama dengan tanggal sistem.

Personal Reminder  
[New Schedule](#)

	Date	Title
<a href="#">Delete</a> <a href="#">Select</a>	18-04-2009	test
<a href="#">Delete</a> <a href="#">Select</a>	17-05-2009	Test lagi
<a href="#">Delete</a> <a href="#">Select</a>	18-05-2009	test

...: Input Schedule ...:

Date

May 2009						
≤						≥
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Title

[Save](#) [Cancel](#)

4. Jika sudah selesai melakukan proses entri, dapat juga dicoba proses edit ataupun penghapusan. Tetap diingat bahwa variasi entri jadwal tetap harus dipertahankan agar saat proses penampilan jadwal tetap bisa muncul ke Gridview. Kini klik Linkbutton *Back* untuk kembali ke menu, lalu coba klik Linkbutton *View Schedule* untuk menampilkan data jadwal yang sudah selesai diinputkan.

Personal Reminder

Show Schedule Expired

Date : 18-05-2009 Title : test To Do : test
Date : 17-05-2009 Title : Test lagi To Do : Jadwal untuk besok
Date : 18-04-2009 Title : test To Do : test

[Back](#)

copyright © Sistem Italky 2009

5. Jangan lupa juga untuk melakukan testing penggantian password dengan melakukan klik pada Linkbutton *Change Password*. Dan setelah selesai melakukan proses tersebut, lakukan proses logout dengan mengeksekusi Linkbutton *Logout* dan lakukan proses login ulang.

Apabila proses testing telah berhasil dilakukan, berikut ini adalah pemikiran untuk pengembangan lebih lanjut :

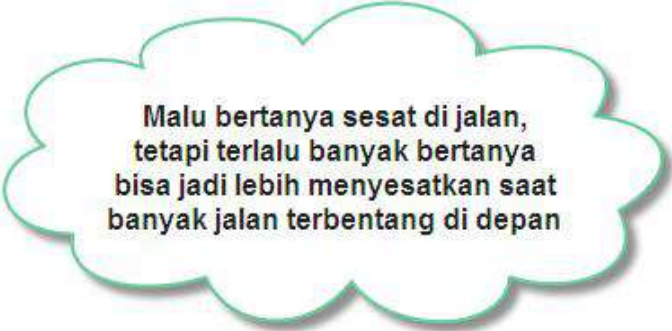
1. Tambahkan teknik AJAX sehingga solusi *single page website* dapat lebih "terasa".
2. Tambahkan fitur untuk pencarian jadwal berdasarkan kriteria tertentu, baik dari judul ataupun isi agenda.



3. Bisa juga ditambahkan fitur *alert* sehingga jadwal yang telah masuk tanggalnya (atau mendekati tanggal sistem) dapat benar-benar menjadi pengingat bagi pengguna.

## **Slide Show Gallery**

---



**Malu bertanya sesat di jalan,  
tetapi terlalu banyak bertanya  
bisa jadi lebih menyesatkan saat  
banyak jalan terbentang di depan**

**Level :** Pemula - Menengah

**Tujuan :**

- Membuat image gallery sederhana
- Membuat auto slide show dari image gallery dengan teknik AJAX
- Mengaplikasikan in-stream bitmap untuk pembuatan thumbnail image secara otomatis

**Fitur situs :**

- Implementasi komponen Timer dari ASP .NET AJAX untuk kepentingan tampilan data dari database
- Modifikasi Gridview dengan menggunakan ASP .NET AJAX
- Implementasi ASP .NET AJAX untuk situs yang menggunakan database
- Implementasi *content management system* dengan menggunakan ASP .NET AJAX

**Estimasi waktu pembuatan :** 30 – 100 menit

## **Pengantar**

---

---

Contoh situs atau website yang dibahas di bab ini adalah implementasi sederhana dari ASP .NET AJAX serta penggunaan GDI+. Situs yang dibuat dalam bab ini nantinya akan menampilkan sebuah galeri gambar atau image gallery dengan sebuah fitur unik yaitu automatic slide show.

Pembuatan automatic slide show sesungguhnya merupakan contoh lain dari penerapan komponen *Timer* yang ada di dalam ASP .NET AJAX. Selain itu, dengan menggunakan sedikit pengembangan melalui GDI+, maka slide show yang dihasilkan juga dapat melakukan pembuatan thumbnail (gambar yang ukurannya lebih kecil untuk tujuan preview) secara otomatis.

Pembuatan automatic slide show sesungguhnya juga dapat dilakukan dengan menggunakan fasilitas yang terdapat di dalam AJAX Control Toolkit, yakni dengan menggunakan extender *Slide Show*. Tetapi di dalam contoh ini sengaja menggunakan penggunaan komponen *Timer* biasa dari ASP .NET AJAX untuk mendemonstrasikan penggunaannya secara sederhana.

Dengan hanya memiliki dua buah halaman web atau web page yang akan ditampilkan dan dibantu

dengan sebuah halaman web page yang hanya akan berfungsi untuk menampilkan thumbnail sebuah class untuk melakukan akses data, maka situs ini diharapkan dapat dibuat dengan waktu yang tidak terlalu lama.

Teknik lain yang didemonstrasikan dalam contoh ini adalah penggunaan *stored procedure* di dalam database SQL Server, sekaligus cara aksesnya. Dan juga di dalam situs ini digunakan teknik akses database dengan cara yang sedikit berbeda, yaitu dengan menggunakan kumpulan fungsi yang terdapat dalam satu class. Sehingga akses data dapat dilakukan dengan jalan yang lebih fleksibel dan lebih singkat.

Selamat berkarya !!

## **Struktur Situs**

---

---

Situs yang akan dibuat terdiri dari tiga buah webpage dan sebuah class serta sebuah master page untuk halaman web yang bersangkutan. Selain itu juga terdapat sebuah file CSS (Cascading Style Sheet) untuk mengatur halaman master (sekaligus juga halaman web lainnya). Keterangan mengenai tiap file tersebut adalah :

1. SlideShow.css

Merupakan file style sheet dengan model layout dua kolom dan terletak di tengah browser dengan sebuah bagian header dan footer untuk menampilkan keterangan serta login.

2. SlideShow.Master

Merupakan master page untuk template halaman web default.aspx dan admin.aspx

3. Default.aspx

Didalamnya merupakan halaman awal dari website sekaligus berfungsi untuk menampilkan galeri gambar dan slide show. Di dalamnya juga sekaligus terdapat fasilitas untuk melakukan login ke halaman admin.aspx.

4. Admin.aspx

Halaman ini memiliki fungsi majemuk, selain untuk mengatur album dari galeri gambar, juga untuk

melakukan proses upload gambar itu sendiri. Selain itu, juga berfungsi untuk melakukan perubahan password administrator.

5. Thumbnail.aspx

Halaman web ini bukanlah halaman web yang akan ditampilkan secara langsung. Tetapi merupakan halaman web yang didalamnya hanya berisi fungsi untuk menampilkan gambar dengan mode thumbnail, sehingga hanya dapat ditampilkan dengan menggunakan parameter query string.

6. DataAccess.vb

Merupakan class sederhana yang didalamnya berisi kumpulan fungsi untuk mengakses database sekaligus mengambil konfigurasi *connection string*. Fungsi yang ada diusahakan agar dapat digunakan sefleksibel mungkin sehingga juga dapat digunakan untuk aplikasi website lainnya.

## Style Sheet : *SlideShow.css*

---

---

Setelah Anda membuat sebuah solution baru untuk situs jenis *ASP .NET Website*, maka langkah pertama yang dilakukan dalam pembuatan situs ini adalah membuat file style sheet terlebih dulu untuk mengatur tampilan situs.



---

---

Bagi Anda yang masih menggunakan ASP .NET 2.0 maka Anda harus melakukan instalasi ASP .NET AJAX terlebih dulu. Kemudian pada pembuatan solution situs baru, pilih jenis ASP .NET AJAX Website.

---

---

Style sheet yang dibuat untuk situs ini, terdiri dari dua bagian yakni :

1. Bagian utama, merupakan bagian style sheet yang akan mengatur tampilan utama dari situs dengan mode tampilan seperti pada contoh sebelumnya yaitu berada di tengah dengan sisa ruang kosong yang sama besar di bagian kanan-kiri.

```
body
{
font-family:Verdana;
font-size:10pt;
```



```
}  
.kotak  
{  
border:solid 1px black;  
padding:1px;  
margin:1px;  
}  
.header  
{  
text-align:center;  
font-size:14t;  
font-weight:bold;  
width:100%;  
}  
.footer  
{  
text-align:center;  
font-size:8pt;  
font-weight:bold;  
width:100%;  
}  
.utama  
{  
margin-left:auto;  
margin-right:auto;  
width:80%;  
}
```

```
.kiri
{
float:left;
width:40%;
border-right:dashed 1px blue;
padding:1px;
}
.kanan
{
float:right;
padding:1px;
width:58%;
}
```



---

---

Sekali lagi perlu diingat bahwa model tampilan ini dapat dimodifikasi sesuai selera. Hanya karena pada saat pembuatan contoh ini, penulis sedang “gemar” dengan model tampilan seperti ini bukan berarti bahwa model tampilan ini yang terbaik.

---

---

2. Bagian berikutnya adalah bagian kotak yang akan digunakan sebagai tampilan untuk menampilkan gambar atau image dalam ukuran aslinya. Kotak ini nantinya akan bertindak seperti sebuah kotak *modal pop up* yang muncul di dalam browser. Di

dalam kotak tersebut nantinya akan juga terdapat sebuah Linkbutton yang berfungsi untuk menutup kotak dialog dan kembali ke tampilan situs yang asli.

```
.kotak{
position:absolute ;
z-index:99999;
padding:10px;
border:1px solid #CCC;
background-color:#999999;
padding-bottom:30px;
margin-left:auto;
margin-right:auto;
overflow:auto ;
text-align:center;
top:0;
width:auto;
}

.linkclose
{
color:white ;
background-color:red;
padding:3px;
margin:3px;
font-family:Times New Roman ;
font-size:12pt;
```

```
font-weight:bold ;  
border:double 2px white;  
}
```

## Master Page : *Slideshow.master*

---

Setelah file style sheet selesai dibuat, maka langkah selanjutnya adalah menambahkan file master page yang akan digunakan sebagai template untuk dua halaman web berikutnya. Pilihlah menu *Website* à *Add New Item* dan di dalam kotak dialog yang tersedia pilih tipe *Master Page* dan beri nama *Slideshow.master*.

Di dalam tampilan Master Page tersebut, berpindahlah ke mode Source lalu drag file *Slideshow.css* ke bawah tag `<title>...</title>` sehingga menjadi seperti listing berikut :

```
<title>Untitled Page</title>
<link href="Slideshow.css"
rel="stylesheet"
type="text/css" />
```

Selanjutnya, drag sebuah komponen *ScriptManager* tepat di bawah tag `<form....>`. Lalu ketikkan listing berikut ini dibawahnya :

```
<div class="header">
AJAX Image Gallery
with Auto Slide Show</div>
<div class="utama">
<asp:ContentPlaceHolder
id="ContentPlaceHolder1"
runat="server">
```

```
</asp:ContentPlaceHolder>
</div>
<div class="footer">
copyright @Soetam Rizky
2009<br />
<asp:UpdatePanel ID="UpdatePanel1"
runat="server" UpdateMode="Conditional">
<ContentTemplate>
<asp:MultiView ID="MultiView1"
runat="server" ActiveViewIndex="0">
<asp:View ID="View1" runat="server">
  <asp:LinkButton ID="LinkButton3"
  runat="server">
    Administrator Login
  </asp:LinkButton>
</asp:View>
<asp:View ID="View2" runat="server">
  <table width="50%">
    <tr>
      <td style="text-align: right">
        Password
      </td>
      <td style="text-align: center">
        <asp:TextBox ID="TextBox1"
        runat="server"
        style="text-align: center"
        TextMode="Password">
```



Master Page ini sungguh sulit untuk diilustrasikan. Karenanya untuk mencegah kesalahan eksekusi situs, khususnya bagi Anda yang berada dalam level menengah atau pemula, maka hasil dari drag-drop tersebut langsung dituliskan dalam bentuk listing, Tetapi bagi Anda yang berada dalam level mahir, dapat melihat langsung ke hasil capture dari listing tersebut dan mencoba untuk melakukan operasi drag-drop sendiri sehingga dapat menyerupai contoh.

---

---



Perlu diperhatikan bahwa dalam listing (yang terlihat) panjang tersebut, sesungguhnya terdiri dari beberapa bagian yakni :

1. Bagian header, yang ditempati oleh teks untuk penanda situs



2. Bagian utama yang hanya terdiri dari sebuah komponen ContentPlaceholder bawaan dari Master Page
3. Bagian footer yang terdapat dalam sebuah komponen UpdatePanel dan terdiri dari dua bagian yaitu :
  - a. Bagian footer yang terdiri dari sebuah teks untuk footer biasa.
  - b. Bagian footer yang memiliki sebuah komponen Multiview dengan dua buah View didalamnya.

Di dalam View yang pertama terdapat Linkbutton *Administrator Login* untuk menampilkan View yang kedua. Sedangkan di dalam View yang kedua terdapat sebuah Table dengan sebuah Textbox dan sebuah Linkbutton untuk melakukan proses login.

## Struktur Database dan Stored Procedure: *Slideshow.mdf*

---


Langkah berikutnya adalah melakukan pembuatan database yang didalamnya hanya memiliki dua buah tabel. Untuk membuat database tersebut, klik pada menu *Website à Add New Item* kemudian pilih tipe *SQL Server Database* dan beri nama database tersebut dengan nama *Slideshow.mdf*. Jika terjadi kotak konfirmasi dialog yang menyatakan agar database disimpan di dalam folder *App\_Data*, klik tombol yang menyatakan setuju.

Dalam Database Explorer (atau Server Explorer bagi Anda yang menggunakan Visual Studio), klik kanan pada hirarki tree *Table* dan pilih sub menu *New Table*.

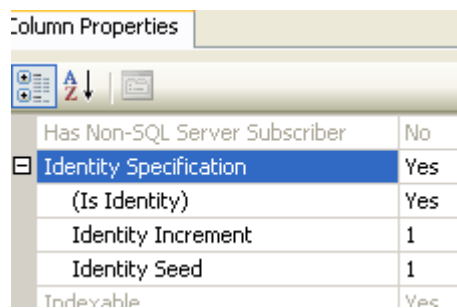


Kini buatlah dua buah tabel baru yaitu :



### 1. Album

	Column Name	Data Type	Allow Nulls
	AlbumID	int	<input type="checkbox"/>
	AlbumName	varchar(50)	<input type="checkbox"/>
	AlbumStatus	bit	<input type="checkbox"/>

Perhatikan bahwa field *AlbumID* merupakan primary key. Set keterangan tersebut dengan melakukan klik di icon dengan gambar kunci berwarna kuning di toolbar. Selain itu, field *AlbumID* juga didefinisikan untuk memiliki sifat *identity* yang berarti pada saat terjadi proses input data baru, akan langsung memiliki nilai angka secara urut. Agar field tersebut memiliki sifat *identity*, klik pada field atau column tersebut, dan di dalam jendela *Column Property* set nilai property *Identity Specification* dan property *IsIdentity* menjadi *yes*.

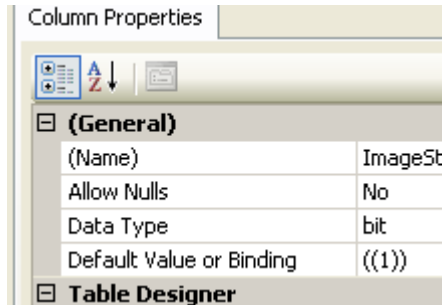


## 2. AlbumDetail

	Column Name	Data Type	Allow Nulls
	AlbumID	int	<input type="checkbox"/>
	ImageName	varchar(50)	<input type="checkbox"/>
	ImageTitle	varchar(50)	<input type="checkbox"/>
	ImageDate	datetime	<input type="checkbox"/>
	ImageStatus	bit	<input type="checkbox"/>

Di dalam tabel yang kedua terdapat dua buah field yang menjadi primary key. Untuk menjadikan





Berikutnya adalah melakukan proses pembuatan Stored Procedures yang nanti akan diakses melalui halaman web dan juga class yang akan dijelaskan di sub bab berikutnya.



---

---

Stored Procedures lebih mirip sebuah fungsi yang *menempel* di dalam sebuah database. Bagi Anda yang ingin lebih dalam mempelajari mengenai Stored Procedures dan penggunaannya, dapat membaca buku berjudul *Panduan Belajar SQL Server 2005 Express Edition* dari penulis yang sama.

---

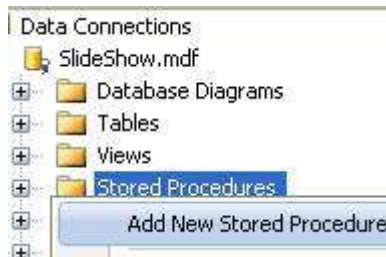
---

Terdapat tiga buah stored procedures yang dibuat yaitu :

1. InsertAlbum

Stored procedures ini digunakan untuk melakukan proses input data ke dalam tabel *Album*. Cara

pembuatan stored procedures baru ini adalah dengan melakukan klik kanan pada Database Explorer (atau Server Explorer jika menggunakan Visual Studio) tepatnya di hirarki tree *Stored Procedures* kemudian memilih sub menu *Add New Stored Procedures*



Selanjutnya di dalam mode listing untuk stored procedures, ketikkan listing berikut :

```
CREATE PROCEDURE dbo.InsertAlbum
    (@AlbumName varchar(50))
AS
    INSERT INTO Album
        (AlbumName, AlbumStatus)
    VALUES    (@AlbumName, 1)
RETURN
```

## 2. InsertImage

Lakukan langkah yang sama dengan pembuatan stored procedures pertama, lalu ketikkan listing untuk stored procedures yang nanti akan digunakan

untuk memasukkan data ke dalam tabel *AlbumImage*

```
CREATE PROCEDURE dbo.InsertImage
(
    @AlbumID int,
    @ImageName varchar(50),
    @ImageTitle varchar(50)
)
AS
INSERT INTO AlbumDetail
(AlbumID, ImageName,
ImageTitle, ImageDate,
ImageStatus)
VALUES (@AlbumID,
@ImageName,@ImageTitle,
GETDATE(), 1)
RETURN
```

### 3. SelectAlbum

Stored procedures yang terakhir adalah stored procedures yang akan diakses di dalam halaman web utama (dan juga administrator) untuk menampilkan data dari tabel *Album*. Isi dari stored procedures tersebut adalah :

```
CREATE PROCEDURE dbo.Selectalbum
AS
SELECT Album.*
```

FROM

Album

RETURN



## File Javascript : ShowImage.js

---

Langkah berikutnya adalah melakukan pembuatan file javascript yang nanti akan digunakan untuk menampilkan gambar atau image di dalam browser sebagai mode pop up khusus. File javascript yang diberi nama *ShowImage.js* tersebut, dibuat dengan menggunakan cara klik di menu *Website* → *Add New Item* lalu pilih tipe file *Javascript* dan beri nama file tersebut.



---

Javascript merupakan salah satu jenis bahasa pemrograman web yang berbasis *client side*. Ini berarti bahwa apapun fungsi dalam javascript, nantinya akan didownload terlebih dulu oleh browser client dan baru akan dieksekusi di browser client tersebut. Tentu saja dengan menggunakan teknik ini banyak efek visual yang dapat ditambahkan, tetapi seluruh fungsi dapat dengan mudah dilihat oleh pengguna. Beberapa script dapat juga langsung diinjeksikan ke dalam halaman web secara langsung, tetapi ada juga yang memilih cara seperti pada sub bab ini, yaitu dengan membuat sebuah file Javascript yang

terpisah dan nanti akan diinjeksikan ke  
dalam halaman web.

---

---

Terdapat dua fungsi yang terdapat dalam file script tersebut yakni :

1. ShowImage

Fungsi ini nantinya akan menampilkan gambar yang asli dengan ukuran orisinal di dalam browser dengan teknik pop up. Tampilan tersebut dilakukan dengan menggunakan bantuan dari layer HTML (ditandai dengan sintaks *div*). Dalam layer tersebut, kemudian disisipkan sebuah tag *img* untuk menampilkan gambar atau image berdasarkan parameter tertentu (nanti akan diambil melalui halaman web yang asli atau *server side*).

Setelah layer tersebut diisi gambar, juga diisi dengan sebuah Linkbutton yang nanti akan berfungsi untuk menutup dari kotak pop up buatan tersebut. Dan untuk menambahkan efek visual transparan pada browser, maka style sheet diberikan perintah *alpha* yang akan membuat tampilan latar belakang menjadi buram (sekilas tampak transparan).

Dengan menggunakan teknik ini, maka tampilan pop up tidak akan terganggu meski di dalam browser telah terdapat fasilitas *pop up blocker*,

karena pop up dalam script ini sebenarnya adalah bagian dari halaman web yang ditampilkan secara terpisah.



---

---

Perlu diperhatikan dalam pengetikan Javascript, memiliki sifat *case sensitive* atau membedakan huruf kecil dan besar. Sehingga sangat penting untuk memperhatikan hal tersebut saat proses pengetikan listing.

---

---

## 2. CloseImage

Fungsi yang kedua adalah fungsi untuk menutup kotak pop up yang telah tampil. Pada saat kotak pop up ditutup, proses yang dilakukan sesungguhnya adalah mencari layer yang telah ditampilkan (dalam script ini diberi nama *test*), kemudian menghapus layer tersebut dari halaman web secara langsung serta mengembalikan style sheet ke tampilan semula.

Listing lengkap dari file ini adalah sebagai berikut :

```
function showimage(imageurl) {  
var xbody =  
document.getElementsByTagName("body")[0];  
var loading = document.createElement("div");
```

```
loading.id = "test";
loading.className = "kotak";
loading.innerHTML = "<a href='#' " +
    "onclick='closeimage()'" +
    "class='linkclose' >Close " +
    "</a><br> " + imageurl;
document.body.appendChild(loading);
xbody.style.filter =
    'alpha(opacity=20)';
}

function closeimage() {
var xbody =
document.getElementsByTagName("body")[0];
var loading =
document.getElementById("test");
    document.body.removeChild(loading);
xbody.style.filter = 'alpha(opacity=100)';
}
```



---

Sebenarnya telah terdapat banyak script *jadi* yang memiliki fungsi sama seperti file ini, seperti *lightbox*, *lightbox*, *DHTML Pop Up* dan masih banyak lainnya (Anda bisa mencari di search engine dengan kata kunci *pop up script* atau juga bisa mencari dari framework Javascript yang

telah ada seperti scriptaculous dan moo). Script-script tersebut memiliki efek visual yang jauh lebih *menawan* dan lebih terstruktur dibanding hasil yang akan muncul dari script ini. Tetapi untuk kepentingan pembelajaran, pembuatan script ini akan sangat membantu Anda untuk memahami bagaimana cara melakukan injeksi Javascript ke dalam suatu halaman web.

---

---

## Menampilkan Thumbnail : *Thumbnail.aspx*

Kini saatnya membuat sebuah halaman web yang unik yaitu halaman web yang tidak memiliki tampilan layout dan memang tidak dibutuhkan tampilan layout didalamnya. Proses pembuatan halaman web ini sama dengan halaman web yang lain yaitu melalui menu *Website* → *Add New Item*, kemudia pilih tipe *Web Form* dan beri nama sebagai *Thumbnail.aspx*.

Halaman web ini nantinya akan berfungsi sebagai *pemanggil* untuk menampilkan gambar atau image dalam mode *thumbnail* (atau ukuran yang lebih kecil). Mode thumbnail berbeda dengan menampilkan gambar yang kemudian ukurannya (height dan width) diperkecil. Karena dalam mode thumbnail, bukan hanya ukuran gambar (dimensi) yang diperkecil, tetapi juga ukuran file dan resolusinya. Sehingga proses loading gambar akan menjadi lebih cepat dibandingkan menampilkan gambar dalam ukuran yang asli.

Teknik yang digunakan dalam situs ini adalah menampilkan gambar dalam mode thumbnail dengan metode *in stream bitmap* yang berarti bahwa sesungguhnya tidak pernah ada gambar yang tersimpan dalam mode thumbnail secara fisik, tetapi hanya *dialirkan* pada saat pengguna melakukan proses browsing. Kelebihan dari teknik ini adalah keamanan

gambar yang asli untuk tidak dapat *dileech* oleh user secara brutal. Tetapi di satu sisi, teknik ini pada saat loading pertama kali akan menyebabkan halaman web sedikit *berhenti*, terutama jika koneksi internet yang digunakan lamban.

Di dalam file halaman web ini, masalah ke dalam mode pengetikan listing dan ikuti langkah berikut ini untuk menyelesaikannya.

1. Di bagian paling atas dari listing tersebut, ketikkan perintah *Imports* untuk menyertakan *library* GDI+ di dalam halaman web tersebut dengan mengetikkan listing berikut :

**Imports System.Drawing**

2. Selanjutnya double klik di halaman web yang kosong tersebut sehingga akan menuju ke prosedur *Page\_Load*. Dalam prosedur ini, merupakan inti dari apa yang akan dilakukan oleh halaman web saat nanti dipanggil oleh halaman web yang lain yaitu halaman default dan halaman administrator.

Alur untuk menampilkan gambar dalam mode thumbnail adalah dengan melakukan pembuatan sebuah obyek bitmap baru yang mengambil duplikat (copy) dari file gambar yang sesungguhnya. File gambar yang sesungguhnya tersebut nantinya akan

diambil dari parameter *query string* pada saat pemanggilan halaman web .

Query string merupakan parameter yang dipanggil dari browser dengan menempatkan kata kunci tertentu saat pengguna memanggil URL tertentu, misal : `www.xyz.com/id=saya` .

Dari hasil pemanggilan query string tersebut, kemudian ditentukan dua macam penampilan thumbnail, yaitu mode slide yang menampilkan thumbnail dengan tinggi gambar sebesar 200 pixel. Dan yang kedua yaitu mode thumbnail pada halaman administrator yang menampilkan gambar thumbnail dengan tinggi lebih kecil yakni 100 pixel.

Isi dari listing adalah sebagai berikut :

```
Dim xfilename As String = _
    Server.MapPath("~/images/") & _
    Request.QueryString("name")
Dim xbitmap As New Bitmap(xfilename)
Dim xthumb As New Bitmap(100, 100)
Dim xabort As New _
    Image.GetThumbnailImageAbort _
    (AddressOf xok)
With xbitmap
    If Request.QueryString("mode") = _
        "slide" Then
        Dim xSize As Integer = _
```



```
(200 * .Height) / .Width
If xSize < 200 Then _
    xSize = 200
xthumb = .GetThumbnailImage _
    (xSize, xSize, xabort, _
    IntPtr.Zero)
Else
    xthumb = .GetThumbnailImage _
    (100, 100, xabort, _
    IntPtr.Zero)
End If
xthumb.Save(Context. _
    Response.OutputStream, _
    Imaging.ImageFormat.Jpeg)
.Dispose()
xthumb.Dispose()
End With
```

3. Berikutnya dibuat sebuah fungsi yang sebenarnya hanya digunakan untuk pelengkap pada pembuatan m0ode thumbnail.

```
Public Function xok() As Boolean
    Return True
End Function
```



Beberapa situs memilih cara yang lebih “mudah” untuk menampilkan gambar thumbnail yaitu dengan melakukan penyimpanan gambar dalam dua buah file yaitu file asli dan sebuah file yang memang diperkecil ukurannya. Cara ini juga dapat dilakukan, tetapi jika situs telah memiliki banyak file gambar maka akan membuat webmaster pengelolaan situs akan menjadi kesulitan dalam mengatur kuota file.

## **Kumpulan Fungsi Akses Database : *DataAccess.vb***

Berikutnya adalah melakukan pembuatan sebuah file *class* yang berisi kumpulan fungsi dan prosedur untuk mengakses database. Class yang hampir serupa juga akan Anda temui di beberapa contoh lainnya dalam buku ini. Tetapi mengingat bahwa tujuan dari buku ini adalah agar pembaca dapat mengerjakan contoh tanpa mempedulikan urutan bab, maka di tiap contoh kasus akan tetap dibahas satu persatu dari awal.

Langkah pembuatan class ini mirip dengan pembuatan halaman web biasa, yaitu dengan memilih menu *Website* à *Add New Item*, tetapi kali ini yang dipilih adalah tipe file *Class*. Perlu diperhatikan bahwa secara default, pembuatan class akan langsung membuat sebuah folder baru (jika belum ada) yaitu folder *App\_Code* di dalam struktur folder situs. Karenanya, jika terdapat kotak konfirmasi dialog untuk pembuatan folder baru, maka pilih jawaban ya. Dan jangan lupa untuk memberi nama class ini sebagai *DataAccess.vb*.



Meski dalam situs ini penggunaan class masih belum memanfaatkan konsep OOP (Object Oriented Programming) secara penuh, tapi diharapkan dapat memberikan gambaran bagaimana membangun sebuah situs yang lebih efisien.

Di dalam class ini terdapat dua buah prosedur dan sebuah fungsi dan beberapa deklarasi variabel serta referensi. Langkah pembuatan selengkapnya adalah :

1. Pertama kali yang harus diketikkan adalah impor referensi yang akan digunakan dalam class tersebut nantinya. Referensi tersebut adalah untuk akses data ke SQL Server serta akses data dari file web.config. Pengetikan listing untuk impor referensi ini ditempatkan di bagian paling atas dari class (sebelum deklarasi *Public class...*).

```
Imports Microsoft.VisualBasic  
Imports System.Data.SqlClient  
Imports System.Web.Configuration
```

2. Berikutnya di bawah deklarasi *Public Class..* ketikkan variabel-variabel awal yang akan digunakan di prosedur dan fungsi selanjutnya.

Variabel tersebut antara lain untuk mendefinisikan metode koneksi serta dataset dan pendukungnya.

```
Dim xConn As New SqlConnection _  
    (ConfigurationManager. _  
        ConnectionStrings _  
        ("SlideShowConnectionString"). _  
        ConnectionString)  
Dim xDataAdapter As _  
    New SqlDataAdapter  
Dim xSet As New Data.DataSet
```



---

Wajib diingat bahwa class ini nanti akan tergantung dengan langkah pembuatan pada sub bab berikutnya yakni pembuatan halaman web administrator.

Sebab pada saat pembuatan halaman web administrator akan sekaligus juga memodifikasi file `web.config` secara otomatis, khususnya pada atribut *connection string*. Sehingga pada tahap ini, akan bersambung di sub bab berikutnya.

---

3. Berikutnya adalah membuat fungsi yang akan digunakan untuk memanggil sebuah tabel secara utuh dengan memanfaatkan konsep *disconnected*. Untuk lebih jelas tentang konsep *disconnected*

dapat membaca buku *Sistem Informasi Terintegrasi dengan menggunakan XML Web Services* dari pengarang yang sama.

```
Public Function GetTable _
    (ByVal xTable As String) _
    As Data.DataSet
    With xDataAdapter
        Dim xCommand As _
            New SqlCommand _
                (xTable, xConn)
        xCommand.CommandType = _
            Data.CommandType.Text
        xCommand.CommandText = _
            "select * from " & xTable
        Dim xTempSet As _
            New Data.DataSet
        .Fill(xTempSet)
        Return xTempSet.Copy
    End With
End Function
```

4. Fungsi yang kedua adalah fungsi untuk mengambil hasil eksekusi stored procedures ke dalam sebuah dataset. Dalam fungsi ini, parameter dari stored procedures terlebih dulu dianggap sebagai sebuah string utuh yang dipisah dengan tanda semicolon (;) atau titik koma. Dan berdasarkan tanda pemisah

tersebut, maka akan dipecah sesuai dengan parameter yang ada di dalam stored procedures.

```
Public Function GetDataSet _
    (ByVal xSP As String, _
        ByVal xParam As String) _
    As Data.DataSet
    With xDataAdapter
        Dim xCommand As _
            New SqlCommand(xSP, xConn)
        xCommand.CommandType = _
            Data.CommandType._
            StoredProcedure
        .SelectCommand = xCommand
        If Trim(xParam) <> "" Then
            Dim xParamAsli() _
                As String = _
                    Split(xParam, ";")
            For i As Integer = _
                0 To UBound(xParamAsli)
                .SelectCommand._
                    Parameters(i).Value = _
                        xParamAsli(i)
            Next
        End If
        Dim xTempSet As _
            New Data.DataSet
        .Fill(xTempSet)
    End With
End Function
```

```
Return xTempSet.Copy  
End With  
End Function
```



---

---

Dalam fungsi ini , setelah hasil stored procedures ditempatkan ke dalam sebuah dataset, maka dataset tersebut tidak langsung diambil ke halaman web, tetapi terlebih dulu dicopy ke dataset yang lain sebagai nilai balik dari fungsi ini. Jika perlu bandingkan terlebih dulu dengan fungsi sebelumnya yang melakukan pengisian dataset melalui perintah query biasa.

---

---

5. Langkah terakhir dalam pembuatan class ini adalah membuat prosedur untuk melakukan penyimpanan data. Hampir mirip dengan fungsi yang kedua, yakni sama-sama menggunakan stored procedures untuk melakukan tugasnya. Tetapi karena penyimpanan data tidak membutuhkan nilai balik berupa data seperti tampilan data, maka tidak ada dataset yang diumpangkan dari hasil eksekusi prosedur ini.

```
Public Sub SaveData _  
(ByVal xSP As String, _  
    ByVal xParam As String, _
```



```
        ByVal xvalue As String)
Dim xCommand As _
    New SqlCommand(xSP, xConn)
With xCommand
    .CommandType = _
        Data.CommandType. _
        StoredProcedure
    If Trim(xParam) <> "" Then
        Dim xParamAsli() _
            As String = _
                Split(xParam, ";")
        Dim xParamValue() _
            As String = _
                Split(xvalue, ";")
        For i As Integer = _
            0 To UBound(xParamAsli)
            Dim xtempParam As _
                New SqlParameter
            xtempParam.ParameterName = _
                xParamAsli(i)
            xtempParam.Value = _
                xParamValue(i)
            xtempParam.Direction = _
                Data.ParameterDirection. _
                Input
            .Parameters.Add(xtempParam)
        Next
    End With
End Sub
```

```
End If
If xConn.State = _
    Data.ConnectionState.Closed _
    Then xConn.Open()
    .ExecuteNonQuery()
xConn.Close()
End With
End Sub
```



---

---

Pada saat setiap kali Anda melakukan langkah pembuatan file baru, maka jangan lupa untuk melakukan proses penyimpanan file. Dan satu hal penting yang perlu diperhatikan, jika Anda termasuk kategori pemula-menengah, dan merasa capek atau jenuh pada tahap tertentu, maka beristirahatlah. Karena dalam kondisi capek dan jenuh umumnya akan malah menyebabkan ketelitian berkurang dan jika terjadi error pada saat eksekusi hanya akan menambah keputusasaan dalam proses belajar.

---

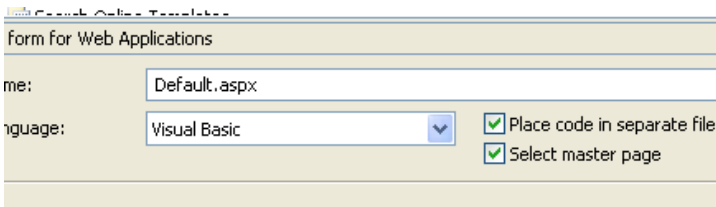
---

## Halaman Administrator : *Admin.aspx*

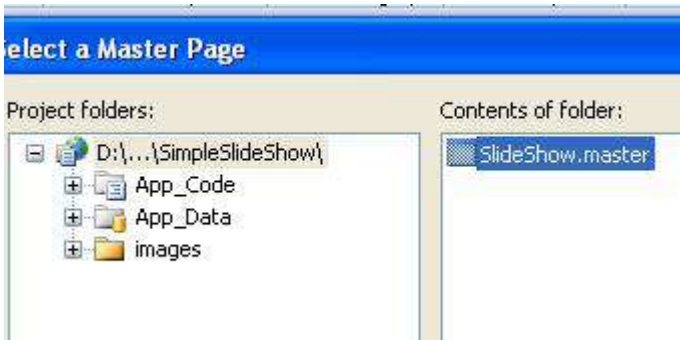
---

Setelah melalui tahapan-tahapan yang tidak memiliki tampilan dan layout didalamnya, kini saatnya melakukan pembuatan halaman web (yang sesungguhnya) dengan menggunakan tampilan layout. Halaman web untuk administrator ini, dibuat dengan menambahkan satu buah halaman web baru dengan nama *admin.aspx*.

Satu hal yang patut dicermati adalah pada saat pembuatan halaman web administrator ini adalah melakukan pemilihan Master Page sehingga di dalam kotak dialog yang tersedia, pilihlah opsi *Select Master Page* sehingga halaman web terkoneksi dengan Master Page yang telah dibuat sebelumnya.



Dan karena hanya terdapat satu buah Master Page dalam situs ini, maka di kotak dialog berikutnya dapat dipilih satu-satunya Master Page yang tersedia.



---

---

Dalam sebuah situs dapat terdiri dari banyak Master Page. Untuk kemudahan, dalam situs ini hanya terdapat sebuah Master Page.

Selain itu, Master Page dapat juga mengikutsertakan Master Page lain atau sering kali disebut sebagai *nested Master Page*.

---

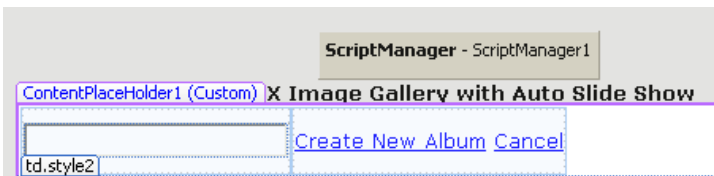
---

Dalam halaman web ini terdapat tiga fitur utama yaitu maintenance album galeri, maintenance data gambar atau image gallery dan penggantian password administrator. Ketiganya tidak terhubung secara langsung, tetapi memiliki keterkaitan satu sama lain.

Langkah lengkap pembuatan halaman web ini dimulai dari pembentukan layout terlebih dulu. Dan sebelum melakukan langkah-langkah ini, pastikan terlebih dulu bahwa semua langkah pembuatan yang

telah dijelaskan di sub bab sebelumnya telah dikerjakan dengan benar.

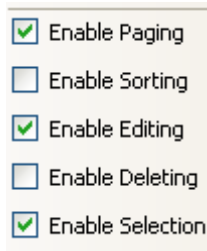
1. Buat sebuah tabel di dalam halaman web dengan jumlah baris sebanyak 6 dan jumlah kolom sebanyak 2.
2. Di dalam baris pertama letakkan komponen berikut ini :
  - a. Di kolom pertama, drag sebuah Panel lalu di dalam Panel tersebut tempatkan sebuah Textbox. Di komponen Textbox tersebut ganti property *ID* menjadi *txtAlbum*.
  - b. Di kolom kedua, drag dua buah Linkbutton. Pada Linkbutton pertama ganti property *ID* menjadi *InkSave* dan property *Text* menjadi *Create New Album*. Sedangkan di Linkbutton yang kedua, set property *ID* menjadi *InkCancel* dan property *Text* menjadi *Cancel*.



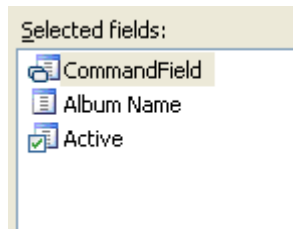
3. Di baris kedua, lakukan operasi *Merge Cells* sehingga hanya tersedia satu kolom (dengan cara blok kedua kolom dan klik kanan sehingga muncul menu, dan pilih sub menu *Merge Cells*). Kemudian drag tabel *Album* dari jendela *Database Explorer*

(atau jendela *Server Explorer* jika menggunakan Visual Studio), sehingga terbentuk sebuah Gridview dan sebuah komponen *SqlDataSource*.

Kemudian, klik di *Smart Tag* pada Gridview sehingga muncul menu kecil di Gridview tersebut dan set opsi yang tersedia agar Gridview dapat menjadi mode *paging* dan didalamnya dapat melakukan operasi edit dan *select* atau pemilihan record.



Kemudian klik di sub menu *Edit Columns* dan hapus field *AlbumID* dari tampilan Gridview. Edit juga property *HeaderText* dari field *AlbumStatus* menjadi *Active*.



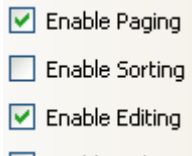


Perlu dicamkan bahwa operasi drag tabel seharusnya dilakukan satu kali (jangan dilakukan berulang). Sebab pada saat operasi drag tersebut sekaligus pula memodifikasi file `web.config` sehingga terbentuk nilai dari tag `connection string`. Nilai atribut dari tag tersebut yang dimanfaatkan untuk koneksi ke dalam database. Dan di dalam contoh situs ini sekaligus juga dipanggil di class `DataAccess` (lihat lagi penjelasan di sub bab mengenai pembuatan class)

---

4. Di baris ketiga, lakukan operasi *Merge Cells* seperti halnya baris kedua. Lalu drag tabel *AlbumImage* didalamnya sehingga terbentuk lagi sebuah GridView dan komponen `SqlDataSource` yang kedua.

Seperti halnya yang terjadi di baris kedua, klik di *smart tag* dari GridView dan set opsi editing serta paging. Lalu klik sub menu *Edit columns* untuk mengedit field yang akan ditampilkan.



Selanjutnya, berpindahlah ke mode Source dan mulai untuk mencari tag *GridView2*. Kemudian di antara tag `<Column>....</Column>`, edit menjadi seperti listing berikut :

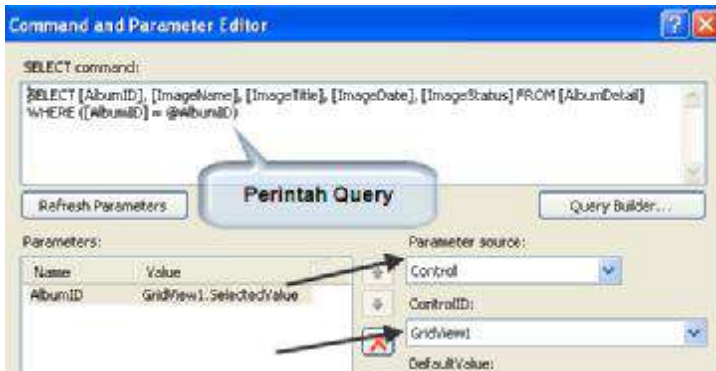
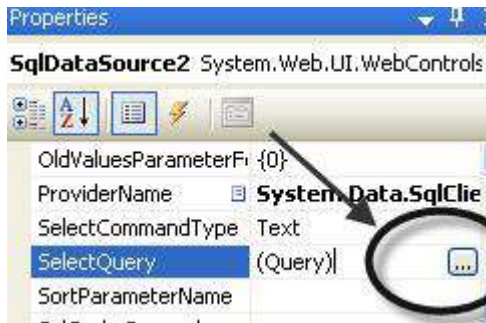
```
<asp:CommandField
ShowEditButton="True" />
<asp:TemplateField>
    <ItemTemplate>
        <%#setGrid()%>
    </ItemTemplate>
</asp:TemplateField>
<asp:BoundField
DataField="ImageTitle"
HeaderText="Title"
SortExpression="ImageTitle" />
<asp:CheckBoxField
DataField="ImageStatus"
HeaderText="Active"
SortExpression="ImageStatus" />
<asp:TemplateField
HeaderText="Image" >
<EditItemTemplate>
<img src=
'thumbnail.aspx?name=<%# eval &
_ ("ImageName") %>' alt='' />
</EditItemTemplate>
<ItemTemplate>
```



```
<img src=
'thumbnail.aspx?name=<%# eval &
_ ("ImageName") %>' alt='' />
</ItemTemplate>
</asp:TemplateField>
```

Di dalam hasil pengeditan kolom tersebut, terdapat beberapa hal penting antara lain :

- a. Penempatan sebuah fungsi *setGrid* yang nanti akan menyatakan sebuah Linkbutton untuk menampilkan gambar dalam ukuran aslinya. Fungsi *setGrid* akan dibahas di bagian berikutnya dari sub bab ini.
  - b. Penempatan kolom bertipe *ItemTemplate* untuk menampilkan gambar dalam ukuran thumbnail dengan memanggil halaman web *thumbnail.aspx* dengan menggunakan parameter query string dari kolom *ImageName*.
5. Kemudian klik di komponen *SqlDataSource2* atau komponen *SqlDataSource* hasil drag tabel *AlbumImage*. Di dalam property *SelectQuery*, klik tombol kecil disebelahnya hingga muncul kotak dialog, dan modifikasilah seperti pada gambar berikut ini :



Masih di komponen yang sama, klik pada property *InsertQuery* dan modifikasi perintah query yang ada seperti pada gambar kotak dialog berikut ini :



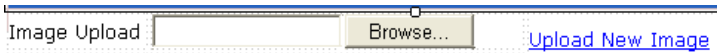
Dan yang terakhir, modifikasi di property *UpdateQuery* seperti pada kotak dialog berikut :

### Command and Parameter Editor

UPDATE command:

```
UPDATE AlbumDetail SET ImageTitle = @ImageTitle, ImageStatus = @ImageStatus  
WHERE (AlbumID = @AlbumID) AND (ImageName = @ImageName)
```

6. Berikutnya di baris yang keempat, lakukan langkah berikut :
  - a. Di kolom pertama ketikkan teks *Image Upload*
  - b. Di kolom kedua, secara berturut-turut drag sebuah komponen FileUpload dan sebuah Linkbutton yang diset property *Text* menjadi *Upload New Image*



7. Sedangkan di baris kelima, dapat dilakukan langkah-langkah berikut ini :
  - a. Di kolom pertama ketikkan teks *Image Title*
  - b. Di kolom kedua, drag sebuah Textbox yang property *ID* didalamnya diset menjadi *txtImageTitle* dan drag sebuah Label yang property *ID* diganti menjadi *lblErrorUpload*.



8. Dan yang terakhir di baris keenam, langkah yang dilakukan mirip dengan dua baris sebelumnya :

- a. Di kolom pertama, ketikkan teks *Admin Password*
- b. Di kolom kedua drag tiga buah komponen yaitu Textbox dengan property *ID* berisi *txtPassword*, sebuah Linkbutton dengan property *Text* berisi *Save Password*, serta sebuah Label dengan property *ID* berisi *lblErrorPass*.



Setelah selesai melakukan proses pembentukan tampilan atau layout, kini saatnya untuk melanjutkan ke tahapan selanjutnya yaitu pengetikan listing program. Tahapan untuk pengetikan listing program adalah :

1. Dobel klik pada bagian kosong dari halaman web (ingat, jangan dobel klik pada bagian abu-abu atau pada bagian Master Page), sehingga menuju ke bagian prosedur *Page\_Load* lalu ketikkan listing berikut :

```
If Session("Login") _  
    <> True Then  
    Response.Redirect _  
    ("default.aspx")  
Else  
    If Not IsPostBack Then  
        txtPassword.Text = _
```

```
System.Web.Configuration. _  
WebConfigurationManager. _  
AppSettings("password")  
End If  
End If
```

Di dalam prosedur ini, terdapat dua hal penting yang dilakukan yaitu :

- a. Melakukan proteksi terhadap halaman web itu sendiri dengan mengecek apakah session dengan nama *Login* telah terisi dengan nilai *True*. Jika tidak, maka diasumsikan bahwa pengguna belum melakukan login dan kendali halaman web dikembalikan ke halaman utama.
  - b. Jika memang pengguna telah melakukan login, maka Textbox yang digunakan untuk mengganti password diisi dengan nilai dari atribut yang didapat dari file web.config. Modifikasi file web.config akan dijelaskan di bagian terakhir dari sub bab ini.
2. Kemudian, deklarasikan obyek dari class *DataAccess* yang telah dibuat di sub bab sebelumnya. Deklarasi tersebut diletakkan tepat di bawah deklarasi *Partial Class...* dan *Inherits...*

```
Dim xData As New DataAccess
```

3. Selanjutnya, ketikkan prosedur dan fungsi yang akan digunakan dalam halaman web ini yaitu :
- a. Fungsi *setgrid* yang nanti akan digunakan untuk menampilkan sebuah Linkbutton yang saat diklik oleh pengguna langsung mengeksekusi file Javascript untuk menunjukkan file gambar dalam ukuran yang sesungguhnya.

```
Function setGrid()  
Dim xaddress() As String = _  
    Split(Request.Url. _  
        AbsoluteUri, "/")  
Dim address As String  
For i As Integer = 0 To _  
    UBound(xaddress) - 1  
    address &= xaddress(i) & "/"  
Next  
Dim xtemp As String = _  
    "<table><tr> <td> "  
xtemp &= _  
    "<a href='#' onclick='" & _  
    "showimage('""') > View </a>"  
xtemp &= "</td><td >"  
  
xtemp &= "</td></tr></table>"
```

```
Return xtemp  
End Function
```

- b. Prosedur *refreshGrid* yang digunakan untuk melakukan refresh tampilan data agar tetap sinkron antara Gridview yang pertama (untuk tabel *Album*) dan yang kedua (untuk tabel *AlbumImage*).

```
Sub refreshGrid()  
    GridView1.DataBind()  
    GridView2.DataBind()  
End Sub
```

- c. Prosedur *LinkUpdate* yang akan dieksekusi saat Linkbutton *Create New Album* dan Linkbutton *Cancel* diklik oleh pengguna.

```
Sub LinkUpdate _  
    (ByVal sender As Object, _  
    ByVal e As EventArgs)  
    Dim xVisible As Boolean  
    Select Case sender.text  
        Case "Create New Album"  
            sender.text = "Save"  
            xVisible = True  
        Case "Save"  
            sender.text = _  
                "Create New Album"
```

```
xVisible = False
xDatA.SaveData _
("InsertAlbum", _
"@AlbumName", _
txtAlbum.Text)
refreshGrid()
Case "Cancel"
lnkSave.Text = _
"Create New Album"
xVisible = False
End Select
Panell.Visible = xVisible
lnkCancel.Visible = xVisible
End Sub
```

4. Lalu double klik di Linkbutton *Create New Album* dan ketikkan listing berikut didalamnya.

```
LinkUpdate(sender, e)
```

5. Sama halnya dengan langkah sebelumnya, double klik di Linkbutton *Cancel* dan ketikkan listing yang sama yaitu :

```
LinkUpdate(sender, e)
```

6. Selanjutnya, double klik pada Linkbutton *Upload New Image* dan ketikkan listing berikut ini didalamnya :

```
If Not IsDBNull _
```



```
(GridView1. _
    SelectedValue) Then
If FileUpload1.FileName _
    <> "" Then
    Dim xrumus _
        As String
    xrumus = _
        CStr(Year(Now.Date)) & _
        CStr(Format(Month _
            (Now.Date), "MM")) & _
        CStr(Day(Now.Date)) & _
        CStr(Hour(Now)) & _
        CStr(Minute(Now)) & _
        CStr(Second(Now)) & _
        FileUpload1.FileName
    FileUpload1.SaveAs _
        (Server.MapPath _
            ("~/images/") & _
            xrumus)
    xData.SaveData _
        ("InsertImage", _
"@AlbumID;@ImageName;@ImageTitle", _
        GridView1.SelectedValue & _
        ";" & xrumus & ";" & _
        txtImageTitle.Text)
    refreshGrid()
    lblErrorUpload.Text = ""
```

```
txtImageTitle.Text = ""  
Else  
    lblErrorUpload.Text = _  
        "You must select file to upload"  
End If  
Else  
    lblErrorUpload.Text = _  
        "Please, select an album"  
End If
```



---

Dalam prosedur ini, proses yang dilakukan adalah melakukan proses entri data ke dalam tabel *AlbumImage* dengan menggunakan stored procedures yang dieksekusi melalui class *DataAccess*. Didalamnya pula sekaligus dilakukan proses upload data gambar ke dalam folder *images* dengan memberi nama ulang (rename) file gambar yang diupload dengan sebuah rumus tertentu yakni menggabungkan tahun, bulan, tanggal, jam, menit hingga detik menjadi sebuah kode nama file baru. Hal ini dilakukan agar file gambar tidak memiliki nama yang sama. Sebab jika nama file yang diupload sama, maka akan terjadi proses tindih (overwrite).

---

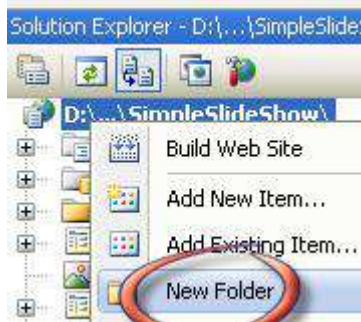
7. Kemudian double klik di Linkbutton *Save Password* dan ketikkan listing yang akan dieksekusi untuk menyimpan password baru ke dalam file `web.config`.

```
If Not Trim _
    (txtPassword.Text) = _
    "" Then
    Dim xconfig As _
        Configuration = _
            Web.Configuration. _
                WebConfigurationManager. _
                    OpenWebConfiguration("~/")
    xconfig.AppSettings. _
        Settings("password"). _
            Value = txtPassword.Text
    xconfig.Save()
    lblErrorPass.Text = ""
Else
    lblErrorPass.Text = _
        "Password is blank !"
End If
```

8. Langkah berikutnya adalah melakukan modifikasi pada file `web.config`. Anda dapat melakukan double klik pada file `web.config` di jendela Solution Explorer, kemudian carilah tag *appSetting* dan modifikasi menjadi listing berikut :

```
<appSettings>
  <add key="password"
    value="admin" />
</appSettings>
```

9. Langkah terakhir dalam halaman web ini adalah membuat folder baru yang akan menampung file-file gambar hasil upload dengan nama *images*. Untuk membuat folder baru ini, klik kanan pada nama solution di jendela Solution Explorer dan pilih sub menu *New Folder*, lalu beri nama menjadi *images*.



## **Halaman Utama : *Default.aspx***

---

Kini tiba saatnya untuk tahapan terakhir di dalam pembuatan situs slide show ini. Tahapan terakhir adalah membuat halaman utama yang akan menampilkan galeri gambar dalam mode slide show, baik secara manual maupun secara otomatis berdasarkan jeda waktu tertentu (delay).

Struktur tampilan maupun listing dalam halaman web ini jauh lebih sederhana dibandingkan halaman web sebelumnya yakni halaman untuk administrator. Tetapi di dalam contoh situs ini (seperti telah disebutkan dalam pengantar awal), lebih ditekankan pada penggunaan ASP .NET AJAX di dalam sebuah halaman web.

Sebelum membuat listing program, maka terlebih dulu dibuat struktur tampilan di dalam halaman web utama *default.aspx*. Karena pada saat awal belum ada Master Page, maka untuk mempermudah proses, hapuslah terlebih dulu file *default.aspx* yang terdapat di dalam jendela Solution Explorer. Kemudian buat ulang halaman web tersebut dengan memilih menu *Website* → *Add New Item* lalu pada kotak dialog yang tersedia pilih tipe *Web Form*.



Disarankan untuk mengulangi lagi dari langkah awal agar halaman web yang telah ada dapat mengacu ke Master Page. Meski sebenarnya langkah ini juga dapat dilakukan dengan melakukan modifikasi pada source dari halaman web, tetapi hal tersebut sangat beresiko jika dilakukan oleh programmer tingkat pemula.

Secara otomatis (jika Anda memang telah menghapus file *default.aspx*), maka nama file yang tersedia adalah *default.aspx* di dalam kotak dialog tersebut. Tetapi kali ini, jangan lupa untuk memilih opsi *Select Master Page* agar halaman web mengacu kepada Master Page yang sebelumnya telah dibuat.

Add New Item - D:\DataSoetam\TestLab\BukuASP35\SimpleSlideShow\

Templates:

**Visual Studio installed templates**

Web Form	Master Page
Web User Control	AJAX Client Behavior
AJAX Client Control	AJAX Client Library
AJAX Master Page	AJAX Web Form
AJAX-enabled WCF Service	Browser File

A form for Web Applications

Name:

Language:

Place code in separate file

Select master page

Kemudian ikuti langkah-langkah berikut untuk membuat layout atau tampilan dari halaman utama situs ini :

1. Berpindahlah terlebih dulu ke mode Source dan ketikkan layer HTML berikut berdasarkan style sheet yang sebelumnya telah dibuat.

```
<div class="kiri">
```

```
</div>
```

```
<div class="kanan">
```

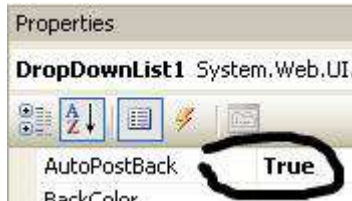
```
</div>
```

2. Lalu berpindahlah lagi ke mode Design setelah pengetikan tersebut. Kini Anda akan melihat dua buah bagian dalam halaman web yaitu bagian kiri dan kanan. Selanjutnya tepat di bagian kiri, sisipkan sebuah tabel dengan jumlah baris sebanyak tiga dan jumlah kolom sebanyak dua. Untuk cara penempatan tabel, dapat dilihat lagi dari sub bab sebelumnya.

Untuk tiap baris, lakukan penempatan berikut :

- a. Di baris pertama kolom pertama ketikkan teks *Album* dan di kolom kedua drag sebuah komponen Dropdownlist. Untuk komponen

Dropdownlist tersebut, set property *AutoPostBack* menjadi *true*.



- b. Di baris kedua, lakukan operasi *Merge Cells* dengan melakukan blok pada kedua kolom yang tersedia lalu klik kanan dan pilih sub menu *Modify Cells* → *Merge Cells*. Di dalam baris yang sudah tergabung jadi satu kolom tersebut, drag sebuah komponen *UpdateProgress*. Kemudian di dalam komponen *UpdateProgress* tersebut, tempatkan sebuah gambar format gif animasi sebagai tanda jika nanti halaman web sedang mengalami proses loading gambar slide show.



---

Gambar atau image bentuk animasi untuk menandai proses loading sangat penting ditempatkan di dalam sebuah halaman web yang mengandung teknik AJAX. Hal ini dikarenakan halaman web yang mengandung teknik AJAX tidak akan menampilkan progress pada browser saat melakukan proses,



sehingga dapat menyebabkan pengguna merasa bahwa proses sedang berhenti.

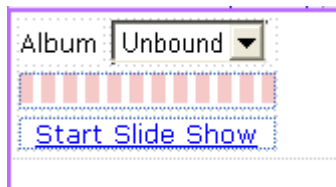
Terlebih lagi jika koneksi internet pengguna tidaklah terlalu cepat, maka gambar animasi tersebut sangatlah berguna. Jika Anda tidak bisa membuat sendiri gambar animasi tersebut, dapat membuat sendiri secara interaktif, gratis dan mudah di situs

[www.ajaxload.info](http://www.ajaxload.info)

---

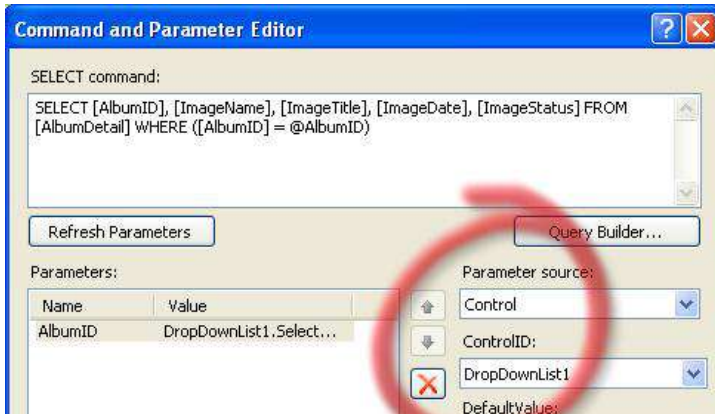
---

- c. Di baris yang terakhir, lakukan operasi *Merge Cells* seperti pada baris kedua dan tempatkan sebuah *UpdatePanel* didalamnya. Di dalam *UpdatePanel* tersebut, drag sebuah *Linkbutton* yang diganti property *Text* menjadi *Start Slide Show*.



3. Kemudian di bagian layer kanan drag sebuah komponen *UpdatePanel*, dan di dalam *UpdatePanel* tersebut drag tabel *AlbumImage* sehingga terbentuk sebuah komponen *GridView* dan *SqlDataSource*. Di komponen *SqlDataSource* yang baru terbentuk, klik tombol kecil yang terletak pada property

*SelectQuery*. Dan di dalam kotak dialog yang tersedia, modifikasi perintah query yang ada menjadi seperti gambar berikut :



Perlu diperhatikan pada isian *Parameter source* dan *Control ID*. Terdapat kemungkinan bahwa komponen *DropDownList1* tidak muncul dalam pilihan karena telah “terbungkus” dalam komponen *UpdatePanel*. Hal ini tidak perlu dirisaukan karena komponen *UpdatePanel* memang bersifat *container* sehingga komponen yang didalamnya seakan-akan sudah tidak bisa dikenali di halaman web.

7. Berikutnya berpindahlah ke mode Source, dan modifikasilah tag GridView yang kedua menjadi listing berikut ini :

```
<asp:GridView ID="GridView2"
    runat="server" AllowPaging="True"
    AutoGenerateColumns="False"
    BorderWidth="0px"
    DataKeyNames="AlbumID, ImageName"
    DataSourceID="SqlDataSource1"
    ShowHeader="False"
    PageSize="1"
    Width="100%"
    EmptyDataText=
    "No Image uploaded yet.....">
<Columns>
    <asp:TemplateField
        HeaderText="AlbumID"
        SortExpression="AlbumID">
        <ItemTemplate>
            <table>
            <tr>
                <img src=
'thumbnail.aspx?name=<%# eval & _
    ("ImageName") & _
    "&mode=slide" %>'
                alt='' />
            </td></tr>
```

```
<tr>
  <td style=
"margin-left:10px; text-align:center ">
  <b>Title : </b>
  <%#Eval("ImageTitle")%><br />
  <b>Uploaded at : </b>
  <%#Format(Eval("ImageDate"), & _
  "dd-MM-yyyy")%><br />
  <%#viewImage()%>
</td>
</tr>
</table>
```

```
</ItemTemplate>
</asp:TemplateField>
</Columns>
<PagerTemplate>
<center>
```

```
<asp:LinkButton
ID="LinkButton1"
CommandName="Page"
CommandArgument="Prev"
Visible="<%#viewLink()%>"
runat="server">Prev -
</asp:LinkButton>
<%= (GridView2.PageIndex) *
```

```
(GridView2.PageSize) + 1%>
/<%=GridView2.PageCount%>
    <asp:LinkButton
ID="LinkButton2"
    CommandName="Page"
    CommandArgument="Next"
    Visible="<%=viewLink()%>"
    runat="server"> - Next
    </asp:LinkButton>
</center>
    </PagerTemplate>
</asp:GridView>
```

Di dalam modifikasi Gridview tersebut, terdapat beberapa poin penting antara lain :

- a. Penggantian fitur paging, yaitu mengubah jumlah total record yang tampil dalam satu page menjadi hanya satu record, dalam hal ini adalah satu buah gambar dalam mode thumbnail. Selain itu, fitur paging yang umumnya menempatkan nomor halaman paging diganti dengan Linkbutton *Prev* dan *Next*, sehingga lebih mengarah ke penggunaan slide show. Kedua Linkbutton tersebut mengeksekusi fungsi *Page* yang merupakan fungsi built ini di dalam Gridview dengan menggunakan parameter atau

argument sesuai dengan teks yang ada didalamnya (previous dan next).

Dan di antara kedua Linkbutton tersebut terdapat penomoran untuk halaman (dan karena satu halaman hanya terdiri dari satu record, maka berarti juga sama dengan nomor record) yang digabungkan dengan total record yang ada dalam tabel.

Untuk mencegah kesalahan pengguna dalam melakukan klik Linkbutton *Prev* dan *Next* (terutama jika sudah berada di record terakhir atau record pertama), maka property *Visible* dari masing-masing Linkbutton tersebut dibatasi dengan fungsi *viewLink* yang isinya akan dijelaskan di bagian pengetikan listing program di sub bab ini pula.

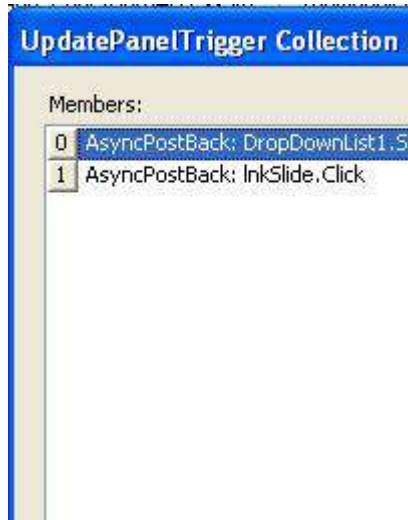


---

Perhatikan bahwa modifikasi untuk bagian bawah paging terdapat di antara tag `<PagerTemplate>` dan dalam tag ini berarti pasti akan ditempatkan di bagian bawah Gridview dan akan muncul di tiap halaman paging.

---

- b. Modifikasi penting lainnya yaitu untuk template field yang menampilkan gambar dengan cara memanggil halaman web *thumbnail.aspx* dengan menggunakan parameter mode *slide* (lihat lagi di dalam sub bab mengenai pembuatan halaman web *thumbnail.aspx*).
  - c. Bagian terakhir terpenting dari modifikasi Gridview ini adalah pada tampilan field *ImageTitle* yang memanggil fungsi *viewImage* yang nanti akan dijelaskan di pengetikan listing program.
8. Setelah selesai, berpindahlah lagi ke mode Design dan klik di UpdatePanel yang menaungi Gridview. Di dalam komponen UpdatePanel tersebut klik pada tombol kecil di jendela property untuk property *Trigger* dan di dalam kotak dialog yang tersedia, arahkan ke *DropDownList1* dan *InkSlide* yang terletak di bagian layer kiri.



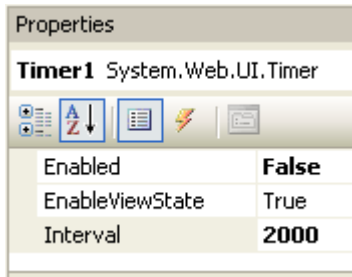
Seperti halnya, pada modifikasi property *SelectQuery* pada komponen *SqlDataSource* yang kedua, terdapat kemungkinan bahwa kedua komponen yang akan direferensi tersebut tidak dikenali karena telah "terbungkus" dalam *UpdatePanel*. Jika memang tidak terdeteksi, maka berpindahlah ke mode *Source* dan modifikasi source *UpdatePanel* tepat di antara tag `</ContentTemplate>` dan tag `</UpdatePanel>` letakkan listing berikut :

```
<Triggers>
  <asp:AsyncPostBackTrigger
    ControlID="DropDownList1"
    EventName="SelectedIndexChanged"
  />
```



```
<asp:AsyncPostBackTrigger  
ControlID="lnkSlide"  
EventName="Click" />  
</Triggers>
```

9. Langkah terakhir adalah melakukan drag sebuah komponen Timer di dalam UpdatePanel yang terakhir dibuat (atau di bagian layer kanan), lalu atur property dari komponen tersebut seperti pada gambar berikut ini :



---

---

Jumlah angka dalam property *Interval* diset dalam satuan milidetik, sehingga dengan setting sebesar 2000 berarti slide show nantinya akan secara otomatis berubah gambar ke record berikutnya dalam waktu dua detik.

---

---

Berikutnya adalah melakukan langkah-langkah untuk pengetikan listing program yaitu :

1. Deklarasikan variabel awal yang diletakkan tepat di bahwa deklarasi *Partial Class*. Deklarasi tersebut adalah untuk obyek dari class *DataAccess* (lihat lagi di sub bab pembuatan file class *DataAccess.vb*) dan sebuah variabel teks untuk Linkbutton yang akan digunakan sebagai pemicu mode slide show.

```
Dim xData As _  
    New DataAccess  
Dim xtext As _  
    String = "Slide Show"
```

2. Kemudian double klik di bagian kosong dari halaman web sehingga menuju ke prosedur *Page\_Load*. Di dalam prosedur ini, yang dilakukan adalah mengambil data dari tabel *Album* dan ditempatkan ke dalam sebuah obyek Dataset yang kemudian di dalam Dropdownlist yang terletak di bagian layer sebelah kiri.

Selain itu, di dalam prosedur ini juga memuat inisialisasi Linkbutton untuk pemicu slide show dengan memberikan nilai awal *Start* dan ditambah dengan variabel awal *xtext* yaitu berisi *Slide Show*. Dan seluruh eksekusi tersebut hanya akan berjalan pada saat halaman web diload pertama kali atau

pada saat pengguna melakukan klik tombol refresh pada browser. Hal ini ditandai dengan adanya percabangan *if not isPostBack* di awal prosedur.

```
If Not IsPostBack Then
    Dim xAlbum As _
        New Data.DataSet
    xAlbum = xData. _
        GetDataSet _
            ("Selectalbum", "")
    With DropDownList1
        .DataSource = xAlbum
        .DataTextField = _
            "AlbumName"
        .DataValueField = _
            "AlbumID"
        .DataBind()
    End With
    lnkSlide.Text = _
        "Start " & xtext
End If
```

3. Berikutnya dobel klik pada Dropdownlist yang terletak pada bagian layer sebelah kiri. Dan karena pada Dropdownlist tersebut sebelumnya telah diset property *Autopostback*-nya menjadi *true*, maka isi dari \_\_\_\_\_ prosedur *DropDownList1\_SelectedIndexChanged* ini akan

langsung dieksekusi pada saat pengguna melakukan pemilihan secara interaktif. Inti dari isi prosedur ini hanyalah melakukan penghentian slide show jika mode slide show sedang aktif, sekaligus melakukan refresh pada Gridview yang terletak di layer sebelah kanan.

```
If lnkSlide.Text = _  
    "Stop " & xtext Then  
    slideShow _  
    (lnkSlide, e)  
End If  
SqlDataSource1. _  
    DataBind()
```

4. Berikutnya double klik pada Linkbutton *Start Slide Show* atau yang sebelumnya telah diganti property *ID*-nya menjadi *InkSlide*, dan ketikkan listing berikut.

```
If GridView2.Rows. _  
    Count > 0 Then _  
    slideShow(sender, e)
```

5. Lalu double klik pada Timer, untuk mengetikkan listing selanjutnya. Di dalam Timer ini merupakan inti dari mode slide show, dengan membuat navigasi dari Gridview berpindah dari paging satu ke paging berikutnya, jika memang terdapat data

didalamnya. Dan karena sebelumnya telah didefinisikan bahwa jumlah record dalam satu kali paging di Gridview hanya satu, maka secara otomatis akan terlihat seakan-akan seperti sebuah slide show otomatis dalam halaman web dengan rentang waktu sesuai yang telah diset di dalam property *Interval* di komponen Timer yaitu dua detik.

```
If GridView2.PageCount _
    > 0 Then
    If GridView2.PageIndex _
        >= GridView2. _
            PageCount - 1 Then
        GridView2. _
            PageIndex = 0
    Else
        GridView2. _
            PageIndex += 1
    End If
End If
```

6. Langkah terakhir dalam pembuatan situs ini adalah mengetikkan prosedur dan fungsi yang membantu dalam eksekusi situs. Terdapat beberapa prosedur dan fungsi yaitu :
  - a. *ViewImage*

Merupakan fungsi yang akan menampilkan gambar dalam ukuran aslinya dengan memanggil file Javascript *showimage.js*. Fungsi yang serupa sesungguhnya juga terdapat di halaman web *admin.aspx* sehingga bagi Anda yang jeli, dapat langsung melakukan operasi copy-paste untuk mendapatkan fungsi ini.

b. *slideShow*

Merupakan prosedur yang dimanfaatkan oleh Linkbutton *Start Slide Show* dengan menggunakan teknik *toggle* yaitu saat diklik pertama kali berarti dalam mode *on* dan diklik berikutnya berarti dalam mode *off*.

c. *ViewLink*

Merupakan fungsi yang menentukan apakah Linkbutton yang terdapat pada *PagerTemplate* di dalam Gridview akan ditampilkan atau tidak. Jika dalam mode slide show, maka Linkbutton tersebut (Prev dan Next) tidak akan ditampilkan dan sebaliknya.

```
Function viewImage()  
If Timer1.Enabled = _  
    False Then  
    Dim xaddress() _
```

```
        As String = _
Split(Request.Url. _
AbsoluteUri, "/" )

    Dim address As String
    For i As Integer = 0 To _
        UBound(xaddress) - 1
        address &= xaddress(i) & "/"
    Next

    Dim xtemp As String = _
        "<a href='#' onclick='" & _
        "showimage(""" "")' >" & _
        "View Original Image </a>"

    Return xtemp

Else

    Return ""

End If

End Function

Sub slideShow _
(ByVal sender As Object, _
ByVal e As EventArgs)

Dim xlen As Integer = _
    Len(sender.text) _
    - Len(xtext)
```

```
Timer1.Enabled = _
    IIf(Left(sender.text, _
        xlen) = "Start ", _
        True, False)
Dim xcaption As String = _
    IIf(Left(sender.text, _
        xlen) = "Start ", _
        "Stop ", "Start ")
sender.text = _
    xcaption & xtext
SqlDataSource1. _
    DataBind()
GridView2.DataBind()
End Sub

Public Function _
    viewLink() As Boolean
    Return Not _
        Timer1.Enabled
End Function
```

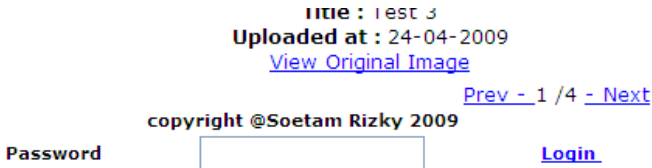


## Testing dan Pengembangan

---

Jika seluruh tahapan pembuatan situs, telah diselesaikan, maka langkah testing yang perlu dilakukan adalah sebagai berikut :

1. Terlebih dulu ingat lagi password yang diset di dalam file web.config, dan cobalah untuk login di halaman utama.



2. Jika login berhasil dilakukan, cobalah untuk membuat sebuah album galeri baru. Untuk nama album dapat diisikan sesuai selera. Pembuatan album sebaiknya lebih dari satu untuk kepentingan testing di halaman utama nantinya.





3. Kemudian lakukan proses upload beberapa image (usahakan dalam format jpg dan dengan ukuran atau resolusi yang tidak terlalu besar agar pada saat testing dapat terlihat dengan jelas di browser) di beberapa album sekaligus. Perlu diingat pula bahwa jika testing dilakukan dalam kondisi online, maka ukuran file image akan sangat mempengaruhi performa dari situs itu sendiri, kecuali Anda memiliki bandwidth internet yang berlebih dan mampu mengakses data dengan sangat cepat. Tetapi dalam testing di stand alone PC hal tersebut tidak akan berpengaruh.



4. Jika sudah selesai, dapat juga dilakukan testing untuk proses editing dalam halaman administrator, dan juga untuk menampilkan gambar yang sudah diupload dalam ukuran yang asli dengan melakukan klik di Linkbutton *View* yang terdapat dalam Gridview.



5. Langkah terakhir di dalam proses testing adalah melakukan proses logout dan kembali ke halaman utama. Di dalam halaman utama tersebut, Anda dapat mengubah album yang telah diupload melalui Dropdownlist, dan mencoba juga mode slide show yang telah dibuat sebelumnya.



Sedangkan dari situs yang telah dibuat dapat dikembangkan beberapa hal antara lain :

1. Interval dari Timer dapat dimasukkan ke dalam file web.config sehingga dapat diatur sendiri oleh pengguna untuk rentang waktu dalam mode slide show.
2. Tampilan slide show dapat diperbaiki dengan menggunakan framework Javascript yang telah memiliki format tampilan lebih bagus seperti *lightbox* atau *jquery*.
3. Dapat diberi tambahan batasan saat proses upload sehingga hanya file image dengan ukuran tertentu yang dapat diupload.

## **Iklan Mini**

---

**Apapun warna kulit seekor sapi, susu yang dihasilkan akan tetap berwarna putih**

**Level :** Mahir

**Tujuan :**

- Membuat situs iklan mini dengan pengguna majemuk dnenga sistem registrasi
- Mengaplikasikan teknik AJAX dalam sebuah situs dengan model *single page solution*.
- Mengaplikasikan koneksi database dengan menggunakan query yang dimodifikasi, tanpa menggunakan wizard.

**Fitur situs :**

- Aplikasi teknik AJAX di dalam koneksi database dan komponen akses data seperti Gridview dan Formview
- Modifikasi penggunaan Treeview untuk tampilan menu dengan mengakses data dari tabel
- Pembuatan content management system secara terintegrasi
- Penggunaan class untuk melakukan enkapsulasi proses akses data

**Estimasi waktu pembuatan :** 3 –4 jam

## Pengantar

---

---

Jika Anda pernah mengakses situs-situs pemasangan iklan gratis ataupun komersil seperti *bekas.com* atau *iklanmini.com* maka dalam contoh terakhir di buku ini akan dibuat situs yang sejenis. Dalam contoh ini situs yang dibuat adalah situs iklan mini dengan bentuk yang lebih sederhana (demi kepentingan penulisan buku) untuk menampilkan iklan-iklan dari berbagai user yang telah teregistrasi sebelumnya.

Meski demikian, contoh situs ini juga dapat dijadikan pola untuk membuat situs yang berbeda tujuan, sebagai contoh, untuk pembuatan blog multi user, situs berita, ataupun situs untuk *social bookmarking*. Sehingga diharapkan kreatifitas pembaca tidak terhenti pada contoh yang sudah diterapkan.

Di satu sisi, telah disebutkan di pengantar bab ini bahwa contoh situs terakhir ini ditujukan bagi Anda yang telah merasa dalam level mahir untuk pemrograman ASP .NET sehingga bisa merasa lebih "nyaman" dalam mengimplementasikan contoh ini. Di sisi lain, bagi Anda yang merasa masih pemula juga tidak ada salahnya untuk mencoba contoh situs ini, terlebih jika Anda telah berhasil menyelesaikan salah



satu contoh situs yang telah dibahas di bab-bab sebelumnya.

Dalam contoh situs ini memang disajikan beberapa teknik pembuatan yang mungkin dirasa tidak lazim untuk sebuah situs dengan menggunakan ASP .NET. Tetapi, mengingat tujuan pembuatan buku ini yang lebih bersifat edukatif, maka berbagai jenis teknik memang sengaja ditampilkan demi keragaman praktek. Diharapkan dengan beragam teknik tersebut, maka pembaca lebih mampu memilih nantinya teknik mana yang dapat diterapkan dalam pembuatan situsnya sendiri.

Sebagai contoh adalah penerapan query yang diambil dari file teks, dan modifikasi total dari Treeview yang mampu menjadi sebuah menu dengan mengakses tabel secara rekursif untuk menampilkan kategori dan jumlah sub kategori yang sesungguhnya bisa tak terbatas. Tentu saja hal tersebut mungkin tidak akan pernah ditemui di buku lain yang membahas mengenai ASP .NET.

Satu hal lain yang sangat penting untuk diperhatikan, adalah jangan pernah menyerah saat terjadi kesalahan kompilasi atau yang lain. Karena kesalahan bukan berarti pemberhentian proses belajar. Karena sebuah proses pemrograman tanpa menemui

kesalahan hampir pasti tidak mungkin dilakukan. Jadi, jangan pernah berhenti untuk berusaha serta belajar, dan selamat berkarya !!!

## Struktur Situs

---

---

Secara umum terdapat tiga bagian besar dari situs ini yaitu :

1. Halaman utama yang akan menampilkan kategori, header iklan, detail iklan dan juga fasilitas untuk pencarian iklan yang ada.
2. Halaman registrasi untuk pengguna baru yang akan mendaftar
3. Folder content management system yang dapat diakses oleh dua jenis level pengguna yakni pengguna biasa atau member dan administrator. Untuk pengguna biasa atau member, dapat melakukan proses pemasangan iklan, edit status dan menghapus iklan serta mengubah profil yang sudah ada. Sedangkan untuk administrator dapat melakukan proses bloking terhadap iklan serta user dan juga mengganti password yang ada. Selain itu, administrator juga dapat melakukan penggantian *announcement* atau pengumuman yang bersifat *sticky* atau selalu muncul di sebuah halaman web.


## Pembuatan Database : MiniAds.mdf

---

Langkah pertama dalam pembuatan situs ini, seperti halnya di contoh yang lain, adalah membuat sebuah solution baru dengan nama *MiniAds*. Selanjutnya, di dalam folder *App\_Data* buat sebuah database baru dengan nama *MiniAds*. Terdapat empat buah tabel di dalam database ini yaitu :

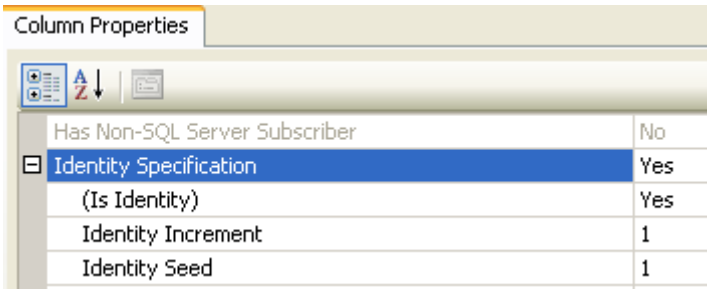
1. Tabel *announcement*

Tabel ini digunakan untuk menampilkan pengumuman dari administrator yang bersifat *sticky* atau selalu muncul di setiap mode tampilan iklan baik detail atau rangkuman. Nantinya hanya satu buah record terbaru dari tabel ini yang akan ditampilkan dalam halaman utama.

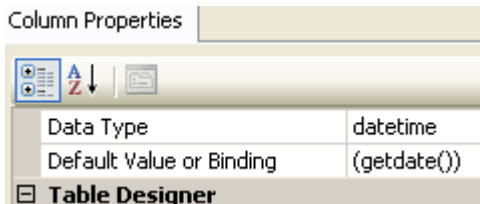
	Column Name	Data Type	Allow Nulls
	AnnouncementID	int	<input type="checkbox"/>
	AnnouncementTitle	varchar(50)	<input type="checkbox"/>
	AnnouncementText	varchar(MAX)	<input type="checkbox"/>
	AnnouncementDate	datetime	<input type="checkbox"/>

Dalam tabel ini, khususnya untuk field *AnnouncementID* memiliki sifat *Identity* yang berarti bahwa field tersebut akan secara otomatis diisi dengan angka secara berurutan mulai dari angka 1 dan seterusnya sesuai urutan entri data. Untuk membuat hal tersebut, klik pada field yang

akan dijadikan *Identity*, lalu di dalam jendela *Column Properties* set property seperti pada gambar berikut :



Selain itu, di field *AnnouncementDate* set property *Default Value* menjadi *GETDATE()* agar pada saat proses entri akan secara otomatis terisi dengan tanggal yang ada di dalam server.

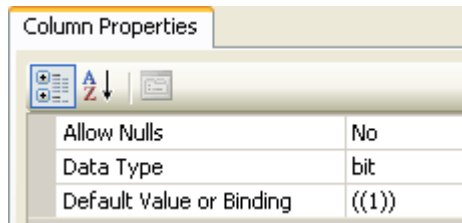


## 2. Tabel *UserProfile*

Tabel ini akan menampung dua jenis user sekaligus yaitu untuk user level administrator dan user level member biasa.

	Column Name	Data Type	Allow Nulls
🔑	UserID	char(10)	<input type="checkbox"/>
	UserName	varchar(50)	<input type="checkbox"/>
	UserPassword	char(10)	<input type="checkbox"/>
	UserLevel	char(10)	<input type="checkbox"/>
	UserCreateDate	datetime	<input type="checkbox"/>
	UserLastLogin	datetime	<input checked="" type="checkbox"/>
	UserEmail	varchar(50)	<input type="checkbox"/>
	UserActive	bit	<input type="checkbox"/>

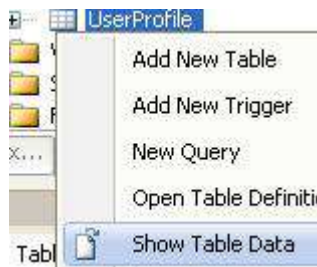
Di field *UserActive* gantilah property *Default Value* menjadi 1 (atau sama dengan *True*), sehingga pada saat proses registrasi member nanti secara default, member tersebut langsung bisa aktif dan melakukan proses login.



Sedangkan untuk field *UserLevel*, property *Default Value* diset menjadi *MEMBER* dengan asumsi dalam proses registrasi bukanlah registrasi sebagai administrator tetapi sebagai member biasa.

Column Properties	
<b>(General)</b>	
(Name)	UserLevel
Allow Nulls	No
Data Type	char
Default Value or Binding	('MEMBER')
Length	10

Dan untuk kepentingan testing nantinya, isilah tabel ini dengan sebuah record baru untuk level administrator. Cara pengisian data dengan melakukan klik kanan pada tabel yang bersangkutan kemudian memilih sub menu *Show Table Data*.



Contoh isi dari data yang diisi adalah :

UserID	UserName	UserPassword	UserLevel
admin	Administrator	admin	ADMIN

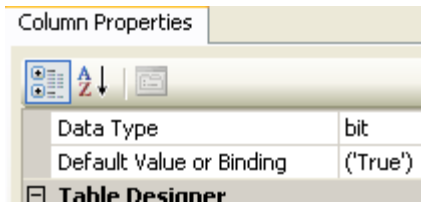
### 3. Tabel *Category*

Tabel berikutnya akan digunakan untuk menyimpan kategori iklan sekaligus menyimpan sub kategori (yang seharusnya nanti mampu memiliki struktur sub kategori dengan jumlah hirarki yang tak

terbatas). Dalam tabel ini (jika digambar dengan menggunakan ER Diagram) memiliki relasi yang rekursif atau relasi ke dirinya sendiri. Dengan cara tersebut, maka tabel ini mampu memiliki hirarki yang tak terbatas, tetapi memerlukan trik tersendiri saat mengaksesnya.

	Column Name	Data Type	Allow Nulls
🔑	CategoryID	int	<input type="checkbox"/>
	CategoryName	varchar(50)	<input type="checkbox"/>
	CategoryActive	bit	<input type="checkbox"/>
	CategoryParent	int	<input checked="" type="checkbox"/>

Dalam tabel ini, set property dari field *CategoryID* agar menjadi *Identity* (lihat lagi pada tabel *Announcement* untuk caranya). Sedangkan di dalam field *CategoryActive* set property *Default Value* menjadi *True*



Dalam tipe field *bit* nilai 0 berarti sama dengan nilai *False* dan nilai 1 sama dengan nilai *True*. Itulah sebabnya



dalam buku ini kedua nilai tersebut diset  
di tabel ini dan tabel sebelumnya.

---

---

#### 4. Tabel *Ads*

Tabel yang terakhir adalah tabel inti dari situs ini yaitu tabel yang menampung data iklan yang dipasang oleh member. Dalam tabel ini juga disimpan berapa kali sebuah iklan diklik oleh pengguna, sekaligus juga nanti akan mampu menangani fitur penjadwalan penampilan tabel pada tanggal tertentu. Sehingga member dapat menjadwalkan sebuah iklan agar tampil pada tanggal tertentu. Fitur ini mungkin sangat akrab bagi Anda para pengguna beberapa template blog seperti Wordpress, yaitu penjadwalan tampilan tulisan blog pada periode tertentu.

	Column Name	Data Type	Allow Nulls
🔑	AdsID	int	<input type="checkbox"/>
	AdsDate	datetime	<input type="checkbox"/>
	CategoryID	int	<input type="checkbox"/>
	AdsTitle	varchar(50)	<input type="checkbox"/>
	AdsText	varchar(MAX)	<input type="checkbox"/>
	UserID	char(10)	<input type="checkbox"/>
	AdsStatus	bit	<input type="checkbox"/>
	AdsClick	int	<input type="checkbox"/>
	AdsDateDisplay	datetime	<input type="checkbox"/>

Beberapa field dalam tabel ini diset ulang propertynya yaitu :

- a. Field *AdsID* diset menjadi field yang bersifat *Identity*
- b. Untuk field *AdsDate* dan field *AdsDateDisplay* property *Default Value* diset menjadi *GETDATE()*
- c. Field *AdsClick* diset property *Default Value*-nya menjadi nilai 0 (nol).

Selain database, di dalam folder *App\_Data* juga dibuat file-file teks yang berisi perintah query untuk diakses di halaman utama. Teknik penggunaan file teks ini memang sedikit tidak lazim, tetapi sengaja ditunjukkan dalam buku ini untuk kepentingan pembelajaran sekaligus dapat menjadi alternatif pembuatan situs yang lain.

Perintah-perintah query yang akan ditampilkan dalam file teks ini sesungguhnya didapat dari fasilitas query builder yang juga terdapat dalam Visual Studio maupun Visual Web Developer Express Edition. Selain itu, juga bisa dibangun melalui SQL Server Management Studio, baik versi Express maupun versi Enterprise.

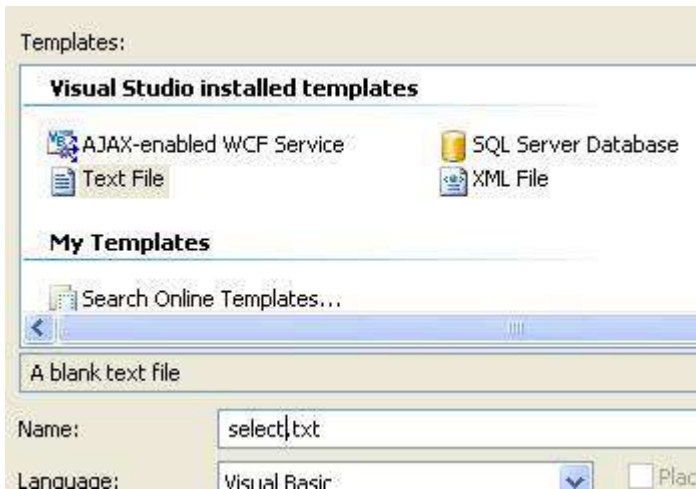


---

Untuk lebih lanjut belajar mengenai perintah query secara lebih lengkap dapat membaca buku *Panduan Belajar SQL Server 2005 Express Edition* dari penulis yang sama.

---

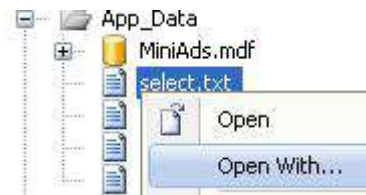
Untuk membuat file teks yang dibutuhkan dapat dilakukan klik kanan di hirarki tree folder *App\_Data* pada jendela Solution Explorer. Lalu pilihlah sub menu *Add New Item* sehingga muncul kotak dialog berikut :



File teks yang pertama diberi nama *select.txt* yang nanti akan diakses untuk tampilan awal di halaman utama *default.aspx*. Di dalam editing ataupun pembuatan file teks ini memang lebih disarankan untuk menggunakan editor teks eksternal seperti notepad

atau edit plus. Hal ini dikarenakan di dalam editor teks eksternal, umumnya terdapat fasilitas untuk melakukan *word wrap* atau tetap melihat seluruh teks meski terdapat dalam satu baris yang sangat panjang.

Agar file teks dapat diedit di editor teks eksternal, maka klik kanan pada file teks yang baru dibuat, kemudian pilih sub menu *Open With*.



Kemudian di dalam kotak dialog yang tersedia, pilih editor teks eksternal yang dimaksud. Dalam buku ini, penulis menggunakan editor teks default yang telah tersedia di dalam sistem operasi Windows yaitu Notepad. Tetapi jika Anda ingin menggunakan editor teks yang lain, seharusnya tidak ada masalah. Beberapa contoh editor teks yang gratis (freeware) dan layak untuk digunakan antara lain adalah metapad dan notepad++.

### Open With - select.txt

Choose the program you want to use to open this

Source Code (Text) Editor (Default)  
 Source Code (Text) Editor With Encoding  
 XML Editor  
 XML Editor with Encoding  
 HTML Editor  
 HTML Editor with Encoding  
**Notepad**

Setelah itu, ketikkan perintah query berikut ini di dalam file teks tersebut. Nantinya file teks tersebut setelah selesai diketikkan, dapat disimpan dan editor teks dapat ditutup kembali.

```
select.txt - Notepad
File Edit Format View Help
SELECT Ads.AdsID,Ads.AdsDate, Ads.CategoryID,
Ads.AdsTitle, Ads.AdsText, Ads.UserID,
Ads.AdsStatus, Ads.AdsClick,
Ads.AdsDateDisplay, Category.CategoryName,
UserProfile.UserName, Ads.CategoryID AS Expr1
FROM Ads left OUTER JOIN Category ON
Ads.CategoryID = Category.CategoryID INNER
JOIN UserProfile ON Ads.UserID =
UserProfile.UserID WHERE (Ads.AdsStatus =
'true') AND (DATEDIFF(day, GETDATE(),
Ads.AdsDateDisplay) <= 0)
```



Dalam query tersebut, akan ditampilkan data dari tabel *Ads* yang mengacu kepada tabel *Category* agar field *CategoryName* juga dapat dimunculkan. Query tersebut

juga mendapat kondisi agar iklan yang muncul adalah iklan yang aktif (dengan isi field *AdsStatus* bernilai *True*) dan tanggal jadwal tampil (dideteksi dengan isi dari field *AdsDateDisplay*) tidak lebih dari tanggal sistem (atau tanggal aktif). Sangat penting diperhatikan bahwa semua teks dalam perintah query tersebut diketikkan dalam satu baris yang sama (tanpa menekan tombol Enter untuk tiap kali perpindahan baris), dan juga jangan lupa untuk pemisahan di tiap kata (seperti spasi) dan juga penggunaan tanda baca.

---

---

Berikutnya, buatlah file teks lagi bernama *selectCategory.txt* yang akan digunakan untuk menampilkan kategori yang dianggap sebagai kategori *Parent* (atau kategori yang memiliki sub kategori lain). Dan dengan langkah yang sama seperti pembuatan file teks sebelumnya, editlah di dalam notepad dan ketikkan teks query berikut :

```

selectCategory.txt - Notepad
File Edit Format View Help
SELECT Category.CategoryName,
Category.CategoryParent, Category.CategoryID,
COUNT(Category.CategoryID) AS Total FROM
Category INNER JOIN Ads ON Ads.CategoryID =
Category.CategoryID LEFT OUTER JOIN Category
AS Category_1 ON Category.CategoryParent =
Category_1.CategoryID WHERE (Ads.AdsStatus =
'true')AND (DATEDIFF(day, GETDATE(),
Ads.AdsDateDisplay) <= 0) AND
(Category.CategoryActive = 'true') AND
(Category.CategoryParent > 0) AND
(Category.CategoryParent =

```

---

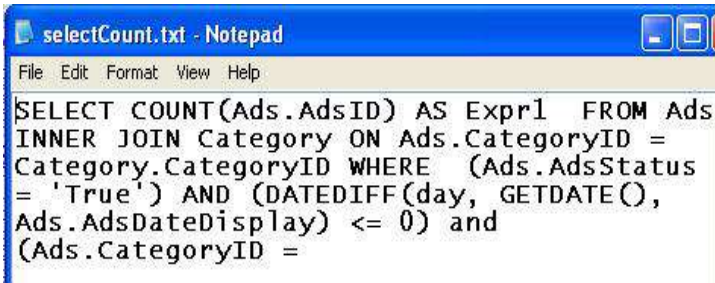
Di dalam query tersebut merupakan query yang *terpotong* karena terdapat parameter yang nanti baru diketikkan pada saat pembuatan halaman utama *default.aspx*. Query tersebut secara umum merelasikan antara tabel *Ads* dengan tabel *Category* dan juga merelasikan secara rekursif antara tabel *Category* dengan tabel *Category* sendiri. Sehingga nantinya dapat dideteksi kategori mana yang menjadi *Parent* dan kategori mana yang menjadi *Child*.

---



File teks untuk query yang ketiga adalah file bernama *selectCount.txt*. Query ini nantinya digunakan untuk menampilkan jumlah posting iklan dalam sebuah kategori. Hampir sama dengan query sebelumnya, query ini juga merupakan query yang terpotong karena

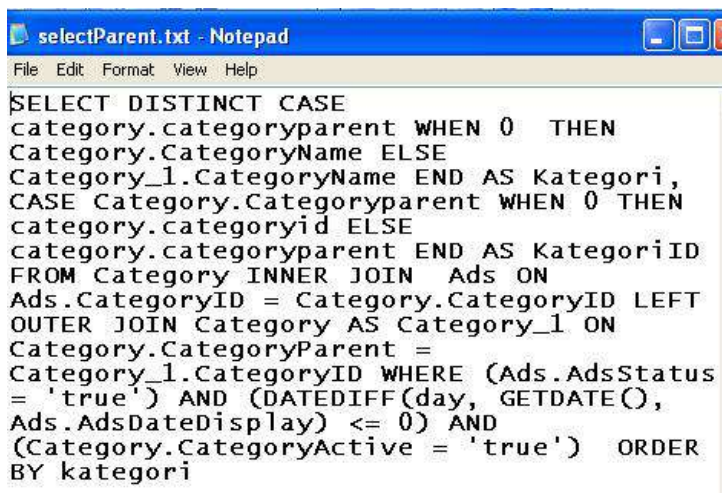
parameter juga baru dientrikan pada pembuatan file halaman utama. Dalam query ini nantinya hanya akan menghasilkan sebuah angka hasil dari jumlah posting iklan yang sesuai dengan kategori yang nanti diumpankan sebagai parameter.



```
selectCount.txt - Notepad
File Edit Format View Help
SELECT COUNT(Ads.AdsID) AS Expr1 FROM Ads
INNER JOIN Category ON Ads.CategoryID =
Category.CategoryID WHERE (Ads.AdsStatus
= 'True') AND (DATEDIFF(day, GETDATE(),
Ads.AdsDateDisplay) <= 0) and
(Ads.CategoryID =
```

File teks yang terakhir adalah untuk menampilkan kategori secara berhirarkis, sehingga nantinya kategori yang disusun dalam sebuah Treeview dapat tersusun sesuai dengan urutan *Parent - Child* yang ada. File teks ini diberi nama *selectParent.txt* dengan isi query sebagai berikut :



A screenshot of a Notepad window titled "selectParent.txt - Notepad". The window has a blue title bar and a menu bar with "File", "Edit", "Format", "View", and "Help". The main text area contains a SQL query. The query uses CASE statements to handle NULL values in the category parent field and joins the Ads table with the Category table. It filters for active categories and ads that are currently displayed. The results are ordered by category name.

```
SELECT DISTINCT CASE
category.categoryparent WHEN 0 THEN
Category.CategoryName ELSE
Category_1.CategoryName END AS Kategori,
CASE Category.Categoryparent WHEN 0 THEN
category.categoryid ELSE
category.categoryparent END AS KategoriID
FROM Category INNER JOIN Ads ON
Ads.CategoryID = Category.CategoryID LEFT
OUTER JOIN Category AS Category_1 ON
Category.CategoryParent =
Category_1.CategoryID WHERE (Ads.AdsStatus
= 'true') AND (DATEDIFF(day, GETDATE(),
Ads.AdsDateDisplay) <= 0) AND
(Category.CategoryActive = 'true') ORDER
BY kategori
```

## **Style Sheet : MiniAds.css**

---

---

Setelah selesai melakukan pembuatan database dan juga file-file teks untuk query, kini saatnya untuk membuat file style sheet yang akan digunakan untuk dasar layout dari situs ini. Style sheet yang dibuat, secara umum terbagi menjadi beberapa bagian yakni :

### 1. Bagian utama

Di dalam bagian utama, halaman web akan berada di tengah browser dengan menyisakan ruang yang sama besar di bagian kiri dan kanan. Sedangkan di bagian tengah tersebut, masih terbagi lagi menjadi dua bagian yaitu bagian kiri untuk tempat menu, serta bagian kanan yang akan menampung isi dari iklan itu sendiri, dan pada saat login akan menampung isi utama dari proses content management system. Selain itu, juga terdapat bagian header dan bagian footer untuk menampung judul situs serta tulisan copyright dalam situs.

### 2. Bagian tampilan iklan

Bagian ini terdiri dari dua bagian yaitu untuk header iklan yang merupakan isi informasi dari iklan itu sendiri, seperti tanggal dan pemasang iklan. Dan bagian kedua adalah untuk judul iklan itu sendiri yang harus tampak beda dengan isi iklan.

Listing lengkap dari style sheet dengan nama *MiniAds.css* yang dibuat adalah sebagai berikut :

```
body
{
    font-family:Verdana;
    font-size:10pt;
}
.halamanUtama
{
    margin-left:auto;
    margin-right:auto;
    width:70%;
}
.header
{
    width:70%;
    margin-left:auto;
    margin-right:auto;
    font-size:14pt;
    font-weight:bold;
    text-align:center;
    margin-bottom:1px;
    padding:1px;
}
.footer
{
    width:70%;
```

```
        margin-left:auto;
margin-right:auto;
font-size:8pt;
font-weight:bold;
text-align:center;
border:solid 1px black;
margin-top:1px;
padding:1px;
}
.kiriUtama
{
    float:left;
    width:30%;
    padding-right:2px;
}
.kananUtama
{
    float:right;
    width:65%;
}
.juduliklan
{
    font-weight:bold;
    font-size:13pt;
    border-bottom:dashed 1px black;
    text-decoration:none;
}
```

```
.headeriklan
{
    font-size:8pt;
    padding:1px;
    text-align:right;
}
```

## Pembuatan Site Map

---

---

Di dalam situs ini didemonstrasikan mengenai cara aplikasi sitemap secara majemuk dalam sebuah situs. Beberapa buku mengenai ASP .NET umumnya hanya membuat implementasi sitemap secara default atau sebuah sitemap. Karenanya contoh situs ini berusaha agar dapat mendemonstrasikan penggunaan sitemap secara majemuk.



---

---

Beberapa programer web mengasumsikan sitemap sebagai sebuah alat bantu untuk membuat menu. Hal ini tidak sepenuhnya salah, karena memang salah satu fungsi sitemap adalah membantu dalam membuat menu dalam sebuah situs. Tetapi sitemap sesungguhnya lebih mempresentasikan sebuah hirarki dari sebuah situs.

---

---

Secara default, penggunaan sitemap yang nantinya akan melibatkan komponen SiteMapProvider akan membaca sebuah file sitemap di root folder dari situs dengan nama *Web.sitemap*. Sehingga pada penamaan file sitemap yang pertama usahakan agar

tetap memiliki nama file yang sama demi kemudahan penggunaan nantinya.

Dalam situs ini terdapat dua buah sitemap yang masing-masing akan digunakan dalam content management system, yakni satu untuk level member dan satu lagi untuk level administrator. Dalam sitemap ini diasumsikan bahwa content management system akan menempati sebuah folder dengan nama *cms*. Sehingga semua link akan mengarah ke folder ini nantinya.

Cara pembuatan sitemap dengan memilih menu *Website à Add New Item* dan di dalam kotak dialog yang tersedia pilih tipe file *Sitemap*. Dua buah sitemap yang dibuat itu adalah :

1. Web.sitemap, dalam template yang tersedia, edit seperti listing sebagai berikut :

```
<siteMapNode url=""
    title="" description="">
<siteMapNode url="~/cms/default.aspx"
    title="Home" description="" />
<siteMapNode url="~/cms/newads.aspx"
    title="New Ads" description="" />
<siteMapNode url="~/cms/editads.aspx"
    title="Edit Ads" description="" />
<siteMapNode url="~/cms/userprofile.aspx"
    title="User Profile"
```

```
        description="" />
<siteMapNode
url="~/cms/default.aspx?id=logout"
    title="Log Out" description="" />
</siteMapNode>
```

## 2. Web2.sitemap

```
<siteMapNode url=""
    title="" description="">
<siteMapNode url="~/cms/default.aspx"
    title="Home" description="" />
<siteMapNode url="~/cms/manageuser.aspx"
    title="Manage Users" description="" />
<siteMapNode url="~/cms/category.aspx"
    title="Manage Category" description=""
/>
<siteMapNode
url="~/cms/announcement.aspx"
    title="Manage          Announcement"
description="" />
<siteMapNode url="~/cms/setting.aspx"
    title="Manage      Site      Setting"
description="" />
<siteMapNode url="~/cms/userprofile.aspx"
    title="User Profile" description="" />
<siteMapNode
url="~/cms/default.aspx?id=logout"
```



```
title="Log Out" description="" />  
</siteMapNode>
```

## Modifikasi web.config

---

Langkah berikutnya adalah melakukan modifikasi web.config yang secara default sebenarnya telah tersedia dalam Solution Explorer. Modifikasi yang dilakukan adalah :

1. Di dalam tag *AppSetting* modifikasi isi yang ada untuk setting situs yaitu teks yang akan muncul di header, footer dan title (tulisan yang muncul di caption browser).

```
<appSettings>
  <add key="header"
    value="...: Mini Ads :..." />
  <add key="footer"
    value=
"copyright @ Soetam Rizky 2009" />
  <add key="title" value="Mini Ads 1" />
</appSettings>
```

2. Sedangkan untuk kepentingan proteksi content management system, terdapat dua macam modifikasi, yaitu :

- a. Carilah bagian tag *Form*

```
<authentication mode="Forms">
  <forms loginUrl="login.aspx"
    defaultUrl="~/cms/default.aspx"/>
</authentication>
```

- b. Kemudian tepat di bawah tag `</system.web>`, tambahkan modifikasi berikut :

```
<location path="cms">
  <system.web>
    <authorization>
      <deny users="?"/>
      <allow users="*" />
    </authorization>
  </system.web>
</location>
```



---

---

Perlu diingat bahwa dalam modifikasi `web.config`, beberapa nilai atribut menerapkan aturan *case sensitive* atau mempedulikan penulisan huruf besar dan kecil. Sehingga perlu ketelitian dalam melakukan modifikasi.

---

---

3. Kemudian di antara tag `</httpmodule>` dan tag `</system.web>` sisipkan modifikasi yang akan berfungsi untuk mengenali Sitemap secara majemuk.

```
<siteMap defaultProvider="MEMBER">
  <providers>
    <add name="MEMBER"
type="System.Web.XmlSiteMapProvider"
siteMapFile="Web.sitemap"/>
    <add name="ADMIN"
type="System.Web.XmlSiteMapProvider"
siteMapFile="Web2.sitemap"/>
  </providers>
</siteMap>
```

## **Pembuatan Class (I) : Misc.vb**

---

Langkah berikutnya adalah melakukan pembuatan class yang akan menampung fungsi-fungsi umum yang akan digunakan di beberapa halaman web sekaligus. Terdapat dua buah file class yang akan dibuat yakni, *Misc.vb* yang menampung fungsi umum dan *DataAccess.vb* yang menampung fungsi-fungsi yang berkaitan dengan akses database.

File class yang pertama dibuat dengan melakukan pemilihan menu *Website à Add New Item* dan di dalam kotak dialog yang tersedia memilih tipe file *Class*. Karena file class secara default disimpan dalam folder *App\_Code*, maka jika terdapat kotak dialog konfirmasi untuk membuat folder tersebut, pilih jawaban ya agar terbentuk folder tersebut.

Di dalam file class *Misc.vb* terdapat beberapa bagian yaitu :

1. Deklarasi Imports sebagai referensi library yang akan diakses dalam class ini. Deklarasi ini diletakkan di bagian paling atas dari class yaitu sebelum deklarasi *Public Class...*

```
Imports Microsoft.VisualBasic  
Imports System.Web.Configuration
```

2. Berikutnya adalah deklarasi variabel umum untuk melakukan akses ke file web.config dan ditempatkan tepat di bawah deklarasi *Public Class*....

```
Dim xconfig As Configuration = _  
    WebConfigurationManager. _  
    OpenWebConfiguration("~/")
```

3. Fungsi pertama yang dibuat adalah fungsi untuk mengakses komponen yang terletak di dalam komponen Formview. Karena nantinya di bagian content management system terdapat beberapa komponen Formview yang harus diakses secara terpisah, maka fungsi *FindInView* akan menjadi sebuah bagian penting dalam situs ini.

```
Public Function _  
    findInView(ByVal _  
    xFormview As FormView, _  
    ByVal xID As String) _  
    As Object  
    Return CType(xFormview. _  
    FindControl(xID), Object)  
End Function
```

4. Bagian berikutnya adalah sebuah fungsi yang berguna untuk mengambil konfigurasi dari file web.config khususnya dari bagian tag *App\_Setting* yang sebelumnya telah dimodifikasi di sub bab

modifikasi web.config. Fungsi yang diberi nama *getSetting* ini memiliki listing sebagai berikut :

```
Public Function _  
    getSetting(ByVal xkey _  
        As String) As String  
Return WebConfigurationManager. _  
    AppSettings(xkey)  
End Function
```

5. Bagian terakhir adalah sebuah prosedur bernama *saveSetting* yang akan dieksekusi saat administrator melakukan penggantian setting dari situs melalui content management system.

```
Public Sub _  
    saveSetting(ByVal xApp _  
        As String, ByVal _  
        xValue As String)  
xconfig.AppSettings. _  
Settings(xApp). _  
Value = xValue  
xconfig.Save()  
End Sub
```

## Pembuatan Class (II) : DataAccess.vb

File class yang kedua adalah *DataAccess.vb* yang bisa dibuat dengan menggunakan langkah yang sama seperti file class yang pertama. Sebagian dari isi file class ini sebenarnya telah dibuat di contoh sebelumnya. Sehingga jika Anda memang mengerjakan contoh-contoh situs sebelumnya, Anda dapat melakukan operasi copy-paste pada beberapa bagian di class ini.

Bagian-bagian yang terdapat dalam class ini adalah :

1. Deklarasi Imports untuk referensi library yang terletak di bagian paling atas dari class atau sebelum definisi *Public Class...*

```
Imports Microsoft.VisualBasic
```

```
Imports System.Data.SqlClient
```



---

Library reference yang ditempatkan di dalam kelas nantinya akan secara otomatis diikutsertakan dalam tiap halaman web yang mengakses class ini.

---



2. Bagian kedua adalah deklarasi variabel koneksi ke database sekaligus untuk Datareader. Deklarasi ini diletakkan di bawah definisi *Public Class*.

```
Dim xConn As New _  
    SqlConnection _  
    (ConfigurationManager. _  
    ConnectionStrings _  
    ("MiniAdsConnectionString1"). _  
    ConnectionString)  
Dim xReader As Data.IDataReader
```

3. Fungsi pertama yang dibuat dalam class ini adalah fungsi untuk menciptakan sebuah obyek Datareader. Fungsi yang bernama *MyReader* ini nantinya membaca sebuah tabel berdasarkan perintah query dan memasukkannya ke dalam obyek Datareader untuk kemudian diakses di dalam halaman web.

Satu hal penting yang perlu diingat bahwa Datareader digunakan untuk mengakses sebuah tabel atau lebih yang hanya bertujuan untuk membaca data, bukan untuk manipulasi data (insert,update atau delete). Hal ini dikarenakan sebuah obyek Datareader memiliki sifat *read only* dan *stream* (hanya bisa dibaca sekali dari record pertama hingga terakhir).

```
Public Function MyReader _
    (ByVal xSql As String) _
    As Data.IDataReader
    If xConn.State = _
    Data.ConnectionState. _
    Closed Then xConn.Open()
    Dim xCommand As _
        New SqlCommand(xSql, xConn)
    xReader = xCommand. _
        ExecuteReader _
        (Data.CommandBehavior. _
        CloseConnection)
    Return xReader
End Function
```

4. Fungsi berikutnya adalah fungsi *CountData* yang nanti akan digunakan untuk menampilkan jumlah posting iklan di halaman utama dan juga di halaman content management system. Karena yang akan dikembalikan berupa nilai angka atau integer sebagai hasil dari jumlah total record yang dimaksud, maka method yang digunakan adalah *ExecuteScalar* yang berfungsi untuk mengambil data bernilai numerik dari sebuah tabel berdasar hasil perintah query *select* dengan parameter jenis agregasi (seperti *count*, *avg* dan *sum*).

```
Public Function _  
    countData(ByVal xtable _  
        As String, ByVal _  
        xfield As String, _  
        ByVal xcondition _  
        As String) As Integer  
    If xConn.State = _  
        Data.ConnectionState._  
        Closed Then xConn.Open()  
    Dim xCommand As _  
        New SqlCommand _  
        ("select count(" & _  
        xfield & ") from " & _  
        xtable & xcondition, _  
        xConn)  
    Return xCommand._  
        ExecuteScalar  
End Function
```



---

---

Untuk lebih jelas memahami method yang ada dalam ADO .NET dapat membaca buku *Pemrograman ADO .NET* hasil tulisan dari *Tan Amelia*.

---

---

5. Selanjutnya, fungsi yang dibuat adalah fungsi bernama *CheckData* yang memiliki kegunaan untuk

melakukan verifikasi, apakah sebuah string terdapat dalam sebuah tabel. Fungsi ini memanggil fungsi yang lain di dalam class yang sama yaitu fungsi *MyReader*.

```
Public Function _  
    checkData(ByVal xSql _  
    As String) As Boolean  
    Dim Reader As _  
        Data.IDataReader = _  
        MyReader(xSql)  
    Return Reader.Read  
End Function
```

6. Kemudian buat sebuah prosedur yang akan dipanggil untuk melakukan pengeditan data dalam sebuah tabel. Prosedur yang diberi nama *UpdateData* ini akan mengambil parameter *xfield* sebagai field yang akan diedit dan kondisi edit akan diambil dari parameter *xCondition*.

Untuk melakukan pengeditan data digunakan method *ExecuteNonQuery* yang memang berfungsi untuk mengeksekusi perintah query non *select* seperti *insert*, *update* dan *delete*.

```
Public Sub _  
    UpdateData(ByVal xTabel _  
    As String, ByVal xField _  
    As String, ByVal _
```

```
xCondition As String)
If xConn.State = _
    Data.ConnectionState. _
    Closed Then xConn.Open()
Dim xCommand1 As _
    New SqlCommand _
    ("update " & xTabel & _
    " set " & xField & _
    " where " & xCondition, _
    xConn)
xCommand1. _
    ExecuteNonQuery()
xConn.Close()

End Sub
```

7. Fungsi terakhir dalam class ini adalah fungsi dengan nama *getDataSet*. Fungsi ini akan mengembalikan sebuah Dataset dari sebuah perintah query yang diumpankan ke dalam parameter *xSql*. Penggunaan Dataset memiliki sifat yang lebih fleksibel dibandingkan Datareader, tetapi memiliki kecepatan akses yang lebih lambat.

```
Public Function _
    GetDataset(ByVal xSql _
    As String) As Data.DataSet
If xConn.State = _
    Data.ConnectionState. _
```

```
        Closed Then xConn.Open()  
Dim xCommand1 As _  
    New SqlCommand _  
        (xSql, xConn)  
Dim xDa As _  
    New SqlDataAdapter  
Dim xDs As _  
    New Data.DataSet  
xDa.SelectCommand = _  
    xCommand1  
xDa.Fill(xDs)  
Return xDs.Copy  
End Function
```

## **Master Page : MiniAds.Master**

---

---

Dalam situs ini terdapat dua buah file Master Page yakni untuk halaman utama dan registrasi serta halaman login, sedangkan yang satu untuk content management system. Dalam sub bab ini akan dibahas mengenai pembuatan Master Page yang pertama dengan nama file *MiniAds.Master*.

Langkah pembuatan untuk Master Page sama halnya dengan pembuatan halaman web baru. Sedangkah tahapan pembuatan Master Page adalah sebagai berikut :

1. Dalam mode Design drag sebuah komponen ScriptManager sehingga komponen tersebut langsung muncul di bagian paling atas dari Master Page. Dengan menempatkan ScriptManager di dalam Master Page, berarti bahwa semua halaman web yang mengacu ke Master Page ini dapat mengimplementasikan teknik AJAX secara langsung.
2. Berikutnya untuk menetapkan layout style sheet ke dalam Master Page, berpindahlah terlebih dulu ke mode Source dan modifikasi di bagian tag `<title>....</title>` menjadi listing berikut ini :

```
<title>...: Mini Ads:...</title>  
<link href="MiniAds.css"  
      rel="stylesheet"
```

```
type="text/css" />
```

3. Masih di dalam mode Source, ketikkan listing untuk memodifikasi tampilan dari Master Page tersebut. Dalam Master Page ini terdapat tiga layer dengan style yang diambil dari style sheet yaitu layer untuk header, halaman utama (yang berisi komponen ContentPlaceHolder dan sekaligus nanti menjadi "wadah" bagi halaman web) serta layer untuk footer yang didalamnya berisi teks copyright dan dua buah Linkbutton untuk menuju ke halaman web Login dan registrasi member baru.

```
<div class="header">
    <asp:Label ID="lblheader"
    runat="server"
    Text="Label"></asp:Label>
</div>
<div class="halamanUtama">
    <asp:ContentPlaceHolder
    id="ContentPlaceHolder1"
    runat="server">
    </asp:ContentPlaceHolder>
</div>
<div class="footer">
    <asp:Label ID="lblfooter"
    runat="server" Text="Label">
    </asp:Label>
```



```
<br />
<asp:LinkButton
    ID="LinkButton1"
    runat="server"
    PostBackUrl="~/Login.aspx">
Login</asp:LinkButton>
  *
<asp:LinkButton
    ID="LinkButton2"
    runat="server"
    PostBackUrl="~/register.aspx">
Register</asp:LinkButton>
</div>
```

## Halaman Registrasi : Register.aspx

---

Berikutnya adalah melakukan pembuatan halaman web yang pertama dalam situs ini yaitu halaman web yang akan digunakan untuk registrasi member baru. Meski selintas terlihat sederhana, dalam halaman web ini terdapat beberapa demonstrasi modifikasi komponen Formview yang nanti juga akan ditemui di bagian content management system.

Tahapan pembuatan dari halaman web ini adalah sebagai berikut :

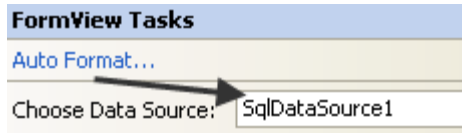
1. Buat sebuah halaman web baru dengan memilih menu *Website* → *Add New Item* dan di dalam kotak dialog yang tersedia, jangan lupa untuk memilih tipe file *Web Form* serta memilih opsi *Select Master Page*. Saat tampil kotak dialog untuk memilih Master Page, pilihlah satu-satunya Master Page yang ada yaitu *MiniAds.Master*.
2. Dalam mode Design di halaman web tersebut, drag tabel *UserProfile* dari jendela Database Explorer (atau Server Explorer jika Anda menggunakan Visual Studio). Sehingga terbentuk sebuah komponen Gridview dan SqlDataSource dalam halaman web tersebut. Selanjutnya, hapuslah komponen Gridview yang sudah terbentuk. Sebagai

gantinya, drag sebuah komponen Formview didalamnya.



Jika Anda kesulitan melakukan drag komponen Formview karena tidak tersedia tempat didalam halaman web, cukup lakukan double klik di komponen Formview yang terdapat di jendela Toolbox.

3. Kemudian klik pada Smart Tag di komponen Formview tersebut dan arahkan property *Data Source* ke komponen *SqlDataSource* yang tersedia.



4. Berikutnya klik kanan pada Formview dan pilih sub menu *Edit Template* à *Insert Item Template*.



Di dalam template yang tersedia tersebut, tempatkan sebuah tabel dengan memilih menu *Table* à *Insert Table*. Jumlah kolom dalam tabel tersebut sebanyak 3 dengan jumlah baris atau row sebanyak 6. Dalam tabel tersebut, atur layout dengan langkah berikut :

- a. Di baris pertama :
- o Pada kolom pertama ketikkan teks *ID*
  - o Pada kolom kedua drag Textbox yang tersedia untuk isian field *UserID*. Lalu drag sebuah komponen *RequiredFieldValidator* di sebelah Textbox tersebut. Set property *ControlToValidate* mengarah ke Textbox yang bersangkutan (jika Anda betul dalam proses drag Textbox maka arahkan ke komponen bernama *UserIDTextbox*), serta set property *ErrorMessage* menjadi tanda asterisk atau tanda bintang (\*). Dan tepat di bawah Textbox tersebut, drag sebuah komponen *UpdateProgress* yang didalamnya diisi dengan file gambar format gif animator sebagai penanda saat terjadi proses pengecekan user ID. Gif animator untuk penanda proses tersebut dapat Anda buat secara gratis dan instan di situs [www.ajaxload.info](http://www.ajaxload.info) .



---

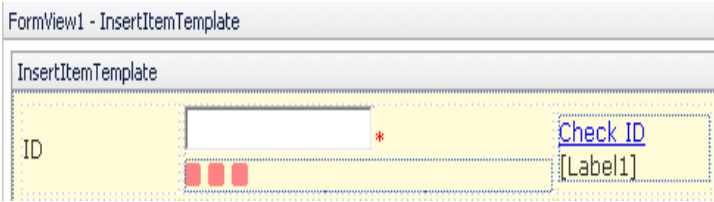
---

Sebenarnya masih banyak situs lain yang menawarkan download gratis untuk file jenis gif animasi atau animator yang biasa digunakan untuk penanda proses dari sebuah situs yang mengaplikasikan teknik AJAX. Anda pun sebenarnya bisa “mencuri” file gif animator tersebut dari situs lain yang mengaplikasikan teknik AJAX dengan jalan melakukan proses *save as* saat browsing ke situs yang dimaksud.

---

---

- o Pada kolom ketiga, drag sebuah komponen `UpdatePanel` didalamnya. Kemudian di dalam `UpdatePanel` tersebut, drag sebuah `Linkbutton` yang property `Text`-nya diganti menjadi `Check ID` serta property `CausesValidation` diisi dengan nilai `False`. Property `CausesValidation` ini diset menjadi `False` agar nantinya tidak terpengaruh dengan `Linkbutton` untuk penyimpanan data. Kemudian drag sebuah `Label` di bawah `Linkbutton` tersebut. `Label` yang baru didrag, set property `Text` menjadi kosong.

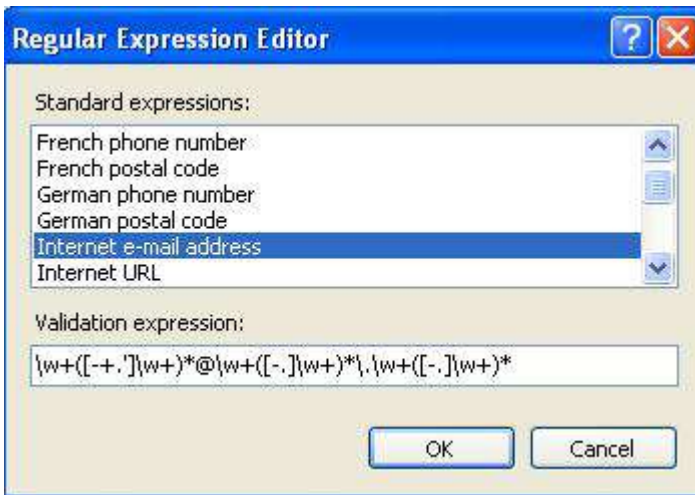


- b. Di baris kedua
- o Pada kolom pertama ketikkan teks *Name*
  - o Pada kolom kedua, drag Textbox untuk field *UserName* yang sebelumnya telah ada di Formview. Di sebelah Textbox tersebut, drag lagi sebuah komponen *RequiredFieldValidator*. Dan seperti halnya di baris pertama, set property *ControlToValidate* mengarah ke Textbox *UserNameTextbox* dan property *ErrorMessage* menjadi tanda asterisk.
- c. Di baris ketiga
- o Di kolom pertama ketikkan teks *Password*
  - o Di kolom kedua, drag Textbox untuk isian *UserPassword*. Kemudian set property *TextMode* menjadi nilai *Password*. Lalu di sebelah kanan dari Textbox tersebut, drag lagi sebuah komponen *RequiredFieldValidator* dengan property *ErrorMessage* bernilai tanda asterisk dan

property *ControlToValidate* mengarah ke komponen *UserPasswordTextBox*.

- d. Di baris keempat
- Pada kolom pertama ketikkan teks *Confirm Password*
  - Pada kolom kedua drag sebuah Textbox baru yang property *TextMode* didalamnya diset menjadi *Password*. Lalu drag sebuah komponen *CompareValidator* di sebelah kanan Textbox tersebut. Set property *ControlToValidate* dari komponen tersebut menjadi *Textbox1* dan property *ControlToCompare* mengarah ke *UserPasswordTextbox* agar jika terjadi kesalahan pengisian password dapat diatasi. Set pula property *ErrorMessage* menjadi teks *Password not match*.
- e. Di baris kelima
- Pada kolom pertama ketikkan teks *Email*.
  - Di kolom kedua, drag Textbox yang berasal dari isian *UserEmail* serta drag disebelahnya sebuah komponen *RegularExpressionValidator*. Set property *ControlToValidate* dari komponen

tersebut mengarah ke *UserEmailTextbox* dan property *ErrorMessage* menjadi teks *Email not valid*. Sedangkan untuk property *ValidationExpression*, klik pada tombol kecil, dan di kotak dialog yang tampil pilih validasi untuk pengisian email.



- f. Di baris terakhir
- o Pada kolom pertama drag sebuah Button baru dan set property *Text* dari Button tersebut menjadi *Submit*.
  - o Pada kolom kedua drag lagi sebuah Button dengan property *Text* yang diset menjadi *Cancel* serta property



*CausesValidation* diset menjadi nilai *False*.

- g. Setelah selesai, hapuslah semua teks serta komponen yang terletak di luar tabel yang sudah ada. Perlu diingat agar berhati-hati saat melakukan penghapusan sehingga hanya komponen yang berada di luar tabel saja yang akan terhapus. Disarankan untuk melakukan penghapusan teks dan komponen tersebut satu per satu agar tidak terjadi kesalahan penghapusan.

The screenshot shows a web form with the following elements:

- ID:** A text input field with a red asterisk (\*) and a blue link labeled "Check ID" with "[Label1]" below it.
- Name:** A text input field with a red asterisk (\*).
- Password:** A text input field with a red asterisk (\*).
- Confirm Password:** A text input field with a red asterisk (\*) and a red message "Password not match" below it.
- Email:** A text input field with a red asterisk (\*) and a red message "Email not valid" below it.
- Buttons:** "Submit" and "Cancel" buttons at the bottom left.

5. Masih dalam mode yang sama, double klik di Linkbutton *Check ID*, kemudian ketikkan listing berikut ini :

```
Dim xLabel As Label = _  
    CType(FormView1. _  
        FindControl("Label1"), _
```

```
Label)
xLabel.Text = _
    IIf(checkUserID, _
        "Not Available", _
        "Available")
```



---

---

Jika pada saat pengetikan listing ini masih muncul tanda error, untuk sementara jangan dihiraukan karena terdapat variabel yang masih belum dideklarasikan di langkah ini.

---

---

6. Kemudian dobel klik pada Button *Submit* untuk mengetikkan listing selanjutnya :

```
If checkUserID() Then
    Page. _
    RegisterStartupScript _
    ("alert", _
        "<script>alert" & _
        "('User ID already " & _
        "exist !'); </script>")
Else
    FormView1. _
        InsertItem(True)
End If
```



---

---

Jika pada saat pengetikan listing ini terdapat tanda "cacing" hijau di beberapa bagian listing, hal tersebut bukan berarti kesalahan pengetikan tetapi merupakan peringatan bahwa sintaks yang diketikkan bersifat *obsolete* atau terdapat sintaks lain untuk versi yang lebih baru.

---

---

7. Langkah berikutnya adalah melakukan double klik di Button *Cancel* untuk mengetikkan listing berikut :

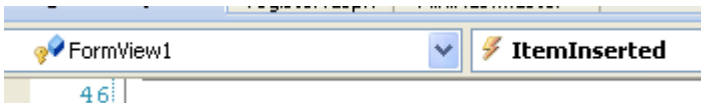
```
Response. _  
    Redirect("default.aspx")
```

8. Dan masih di dalam mode pengetikan listing program, ketikkan sebuah fungsi baru yang menjadi inti dari proses pengecekan user ID baru atau pendaftaran member baru.

```
Function checkUserID() _  
    As Boolean  
    Dim xUserID As String = _  
        CType(FormView1. _  
            FindControl _  
                ("UserIDTextBox"), _  
                TextBox).Text  
    If Trim(xUserID) <> "" Then  
        Dim xData As _
```

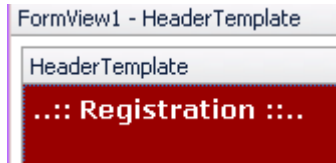
```
New DataAccess
Dim notAvailable _
As Boolean = _
xData.checkData _
("select * from " & _
"userprofile where " & _
"userid='" & _
xUserID & "'")
Return notAvailable
Else
Return False
End If
End Function
```

9. Kemudian listing program yang terakhir adalah untuk *FormView1\_ItemInserted* yang dapat dibuat dengan memilih *FormView1* di combobox *class name* dan *ItemInserted* di combobox *method name*.



```
Response. _
Redirect("default.aspx")
```

10. Berikutnya klik kanan lagi pada Formview yang tersedia dan pilih sub menu *Edit Template à HeaderTemplate*. Lalu ketikkan teks *Registration* dalam kotak yang tersedia.



11. Setelah selesai, Anda dapat kembali ke halaman web di mode Design dan di Formview yang telah dimodifikasi tersebut dapat diklik kanan dan pilih sub menu *End Template Editing*.

12. Langkah terakhir dalam halaman web ini adalah melakukan modifikasi pada property *InsertQuery* di komponen *SqlDataSource*. Klik tombol kecil dari property tersebut, dan pada kotak dialog yang tersedia, edit perintah query yang ada menjadi seperti gambar berikut :



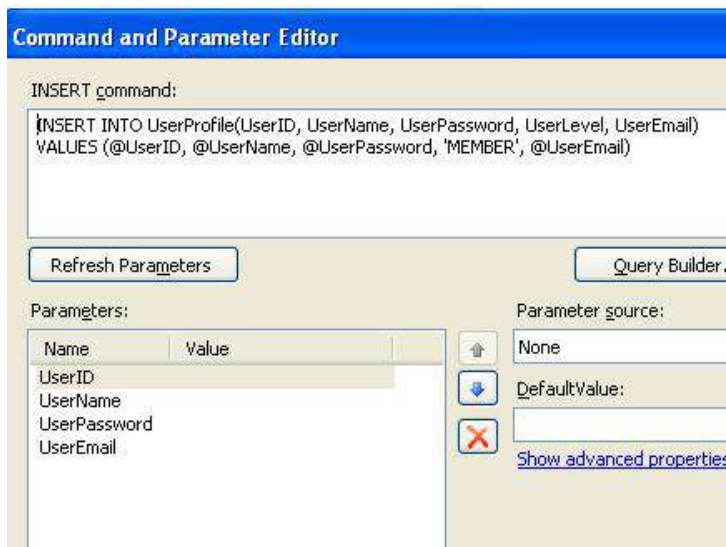
---

Dalam tahapan ini Anda bisa langsung melakukan testing dengan melakukan registrasi beberapa kali untuk user yang berbeda, juga untuk user yang

memiliki ID sama agar pesan kesalahan  
untuk member yang memiliki ID sama  
dapat tampil.

---

---



## Halaman Login : Login.aspx

---

Halaman web berikutnya yang dibuat dalam situs ini adalah halaman web untuk melakukan login, baik sebagai member maupun sebagai administrator. Cara pembuatan halaman web ini sama dengan halaman web sebelumnya, dengan mengacu ke halaman Master Page *MiniAds.Master*. Sedangkan tahapan pembuatannya adalah :

1. Di halaman web yang tersedia, buatlah sebuah tabel dengan jumlah baris sebanyak empat dan jumlah kolom sebanyak dua.
2. Di baris yang pertama lakukan operasi *Merge Cells* dengan cara melakukan blok di kedua kolom dalam satu baris dan melakukan klik kanan kemudian memilih sub menu *Modify Cells* à *Merge Cells*. Lalu di dalam kolom yang sudah menjadi satu ketikkan teks *Login* sebagai header dari proses.



3. Kemudian di dalam baris pertama :
  - a. Di kolom pertama ketikkan teks *User ID*
  - b. Di kolom kedua drag sebuah Textbox dan disebelahnya drag sebuah komponen

RequiredFieldValidator. Untuk komponen yang terakhir, set property *ControlToValidate* mengarah ke *Textbox1* dan property *ErrorMessage* menjadi tanda asterisk.

4. Sedangkan untuk baris kedua :
  - a. Di kolom pertama ketikkan teks *Password*.
  - b. Di kolom kedua drag sebuah Textbox dengan property *TextMode* menjadi *Password*. Kemudian drag lagi sebuah komponen RequiredFieldValidator dengan setting property *ControlToValidate* mengarah ke *Textbox2* dan property *ErrorMessage* juga menjadi tanda asterisk.
5. Di baris yang terakhir :
  - a. Drag sebuah Button di kolom pertama dan ganti property *Text* dari Button tersebut menjadi *Login*.
  - b. Sedangkan di kolom kedua drag sebuah Label dengan property *Text* yang dikosongkan.
6. Langkah terakhir dalam halaman web ini adalah melakukan double klik pada Button *Login* untuk menentukkan listing yang akan melakukan proses login dengan cara memanggil fungsi yang sebelumnya telah dibuat di class *DataAccess* sekaligus juga melakukan update data kapan



terakhir kali member atau administrator tersebut login dengan mencatat waktu login ke dalam tabel *UserProfile*.

```
Dim xData As New DataAccess
With xData
    Dim xQuery As String = _
        "select * from " & _
        "userprofile where " & _
        "userid='" & _
        TextBox1.Text & _
        "' and userpassword='" & _
        TextBox2.Text & _
        "' and useractive='true'"
    Dim xReader As _
        Data.IDataReader
    xReader = _
        .MyReader(xQuery)
    If xReader.Read Then
        Label1.Text = ""
        Session("UserID") = _
            xReader.Item("UserID")
        Session("UserName") = _
            xReader.Item("UserName")
        Session("UserLevel") = _
            xReader.Item("UserLevel")
    xReader.Close()
    .UpdateData("userprofile", _
```

```
"userlastlogin=getdate()", _  
    "userid='" & _  
    TextBox1.Text & "'" )  
FormsAuthentication. _  
    RedirectFromLoginPage _  
    (TextBox1.Text, False)  
Else  
    Label1.Text = _  
        "Login error !"  
End If  
End With
```



---

---

Perlu diingat bahwa halaman web ini belum bisa dieksekusi secara langsung sebab bagian yang akan menjadi tujuan setelah proses login dilakukan belum dibuat. Halaman web ini baru bisa dites jika halaman web yang terdapat dalam folder *cms* telah selesai dibuat.

---

---

## **Folder CMS – Master Page : Admin.Master**

---

---

Sebelum melangkah ke halaman utama, maka seluruh bagian yang terdapat di dalam folder *cms* harus terlebih dulu diselesaikan agar pada saat proses testing dapat berjalan dengan baik. Dalam folder untuk content management system ini, terdapat beberapa halaman web yang dapat diakses oleh level member dan administrator sekaligus dan juga terdapat halaman web yang hanya bisa diakses oleh satu level saja (member saja atau administrator saja).

Yang pertama kali dibuat di dalam folder ini adalah Master Page baru khusus untuk content management system. Pembuatan Master Page yang berbeda dikarenakan nantinya bagian menu situs yang ada di sebelah kiri akan berbeda dengan yang ada di bagian utama. Langkah pembuatan Master Page adalah sebagai berikut :

1. Buat sebuah Master Page di dalam folder *cms* (dengan asumsi bahwa folder tersebut telah dibuat seperti yang telah dijelaskan di sub bab sebelumnya) dengan nama *Admin.Master*.
2. Di dalam mode Design, drag sebuah komponen ScriptManager sehingga terletak di bagian paling atas di Master Page tersebut.

3. Kemudian berpindahlah ke mode Source dan edit tag `<title>...</title>` menjadi listing berikut :

```
<title>...: Mini Ads :...</title>
<link href="../../MiniAds.css"
rel="stylesheet"
type="text/css" />
```

4. Selanjutnya, masih di dalam mode Source, tepat di bawah tag `</ScriptManager1>` modifikasi listing untuk menyisipkan layer dari hasil style sheet.

```
<div class="header"></div>
<div class="halamanUtama">
<div class="kiriUtama">

</div>
<div class="kananUtama">
<asp:ContentPlaceHolder
    id="ContentPlaceHolder1"
    runat="server">

</asp:ContentPlaceHolder>
</div>
</div>
<div class="footer"></div>
```

5. Saat berpindah ke mode Design nantinya telah terbentuk bagian-bagian yang telah terpisah

berdasar layer hasil dari pengetikan listing. Dalam mode design lakukan langkah berikut :

- a. Di bagian header, ketikkan teks *Content Management System*
  - b. Di bagian footer ketikkan teks *copyright*
  - c. Di bagian layer sebelah kiri, ketikkan teks *Welcome* dan drag sebuah komponen Label yang property *Text* di kompone tersebut dikosongkan.
  - d. Masih di layer bagian kiri, drag sebuah komponen *SiteMapDataSource*. Set property *ShowStartingNode* dengan nilai *False* agar menu tidak menampilkan root tag dari file sitemap. Lalu di bawah komponen tersebut, drag sebuah komponen *Menu* dengan setting property *DataSourceId* mengarah ke *SiteMapDataSource1* dan property *Orientation* dijadikan *Vertical*.
6. Kemudian double klik pada bagian kosong dari *Master Page* sehingga menuju ke prosedur *Page\_Load*, lalu ketikkan listing berikut untuk mengatur menu sesuai dengan level pengguna yang melakukan proses login.

```
If Not IsPostBack Then
    SiteMapDataSource1. _
        SiteMapProvider = _
```

```
Trim(Session _  
      ("UserLevel"))  
Label1.Text = _  
      Session("UserName")  
End If
```

## **Folder CMS – Halaman Utama : Default.aspx**

---

---

Setelah selesai membuat Master Page, berikutnya adalah melakukan proses pembuatan halaman web pertama untuk content management system, yaitu halaman default yang akan muncul setelah pengguna melakukan login.

Cara pembuatan halaman web sama dengan proses pembuatan halaman web yang lain, tetapi perlu dipastikan bahwa Anda sebelumnya telah terlebih dulu memilih hirarki folder *cms* di Solution Explorer sebelum memilih menu *Website* → *Add New Item*. Dan juga perlu dipastikan bahwa saat kotak dialog muncul telah dipilih opsi *Select Master Page* yang pada kotak dialog berikutnya mengacu ke Master Page yang baru yaitu *Admin.Master* di dalam folder yang sama.

Setelah muncul tampilan untuk mengatur layout halaman web, lakukan tahapan pembuatan berikut ini :

1. Ketikkan teks *Your Last Login* dan di sebelah teks tersebut drag sebuah Label dengan property *ID* diset menjadi *lblLastLogin*.
2. Setelah itu, berilah jarak ke bawah dengan menekan tombol *Enter* atau *Ctrl+Enter* dan drag sebuah Label dengan property *ID* berisi *lblTotalAds*.
3. Berikutnya, drag lagi sebuah Label dibawahnya dengan property *ID* berisikan teks *lblAmountClick*.

4. Yang terakhir adalah melakukan drag Label lagi dengan property *ID* berisi *lblTotalUser*. Jangan lupa bahwa seluruh Label yang telah didrag tadi seluruh property *Text* dalam Label tersebut dikosongkan.
5. Selanjutnya double klik pada bagian kosong di halaman web tersebut untuk menuju ke prosedur *Page\_Load* (jika double klik terasa sulit, maka berarti Anda perlu lebih cepat melakukan double klik pada mouse). Dalam prosedur ini dibagi menjadi dua bagian yaitu untuk proses logout dengan mengambil parameter dari query string.

Sedangkan bagian kedua adalah untuk menampilkan informasi ke dalam Label yang telah diset di mode Design. Informasi tersebut terbagi lagi menjadi dua bagian yaitu untuk pengguna level member dan pengguna level administrator. Sehingga tidak semua Label nantinya akan tampil saat seorang pengguna berhasil login, tetapi hanya Label yang sesuai dengan level yang dimiliki oleh tiap pengguna. Tampilan informasi ini diambil dari tabel-tabel yang bersesuaian dengan memanggil fungsi-fungsi yang sebelumnya telah dibuat di class *DataAccess*.

```
Dim xData As New DataAccess
If Request.QueryString("id") _
    = "logout" Then
```



```
Session.Clear()
FormsAuthentication. _
    SignOut()
Response.Redirect _
    ("~/default.aspx")
Else
    If Not IsPostBack Then
        Dim xparam As String = _
            IIf(Trim( _
                Session("UserLevel")) _
                <> "ADMIN", _
                " where userid='" & _
                Session("UserID") & _
                "' ", "")
        Dim xSet As Data.DataSet = _
            xData.GetDataset _
            ("select * from " & _
            "userprofile " & xparam)
        lblLastLogin.Text = _
            Format(xSet.Tables(0). _
            Rows(0).Item _
            ("UserLastLogin"), _
            "dd-MM-yyyy hh:mm") & _
            " <br/> &nbsp; &nbsp; &nbsp; " & _
            "&nbsp; &nbsp; &nbsp; since " & _
            Format(xSet.Tables(0). _
            Rows(0).Item _
```

```
        ("UserCreateDate"), _
        "dd-MM-yyyy")
lblTotalAds.Text = _
        "Total Ads : " & _
        xData.countData _
        ("ads", "adsid", xparam)
xSet = xData.GetDataset _
("select * from ads " & _
xparam & " order by " & _
"adsclick desc")
lblAmountClick.Text = _
        "Most clicked ads : " & _
"<br/> &nbsp; &nbsp; &nbsp; &nbsp;" & _
xSet.Tables(0).Rows(0). _
Item("AdsTitle") & " -> " & _
xSet.Tables(0).Rows(0). _
Item("AdsClick") & _
" clicks since " & _
xSet.Tables(0).Rows(0). _
Item("AdsDateDisplay")
lblTotaluser.Text = _
        IIf(Trim(Session _
("UserLevel")) <> _
"ADMIN", "", _
"Total user : " & _
xData.countData _
("UserProfile", _
```

```
"UserID", _  
" where UserLevel" & _  
"<>'ADMIN'"))
```

```
End If
```

```
End If
```

## Folder CMS – User Profile : Userprofile.aspx

Halaman web berikutnya adalah halaman web untuk melakukan proses edit data user profile dari pengguna yang telah berhasil melakukan proses login. Dengan cara yang sama seperti di sub bab sebelumnya, buatlah sebuah halaman web baru dengan nama *UserProfile.aspx* dengan mengacu ke Master Page *Admin.Master*.

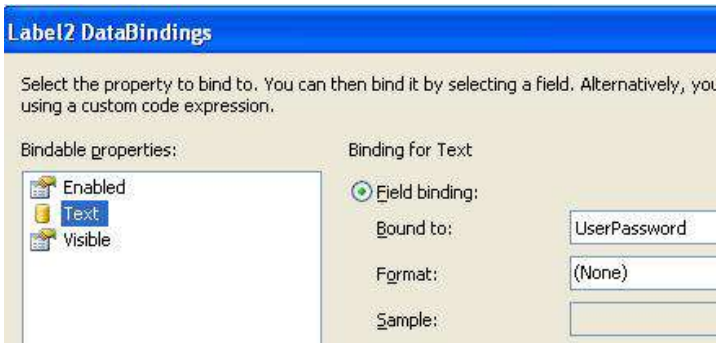
Setelah terbentuk halaman web yang baru, lakukan langkah berikut :

1. Drag tabel *UserProfile* dari Database Explorer, sehingga terbentuk sebuah Gridview dan sebuah *SqlDataSource*. Kemudian hapus komponen Gridview dan gantikan dengan komponen Formview. Pada komponen Formview yang telah terbentuk, set property *DataSourceID* untuk mengarah ke komponen *SqlDataSource1*.
2. Klik kanan pada Formview dan pilih sub menu *Edit Template* → *EditItemTemplate*. Dalam mode edit tersebut, sisipkan sebuah tabel dengan jumlah baris sebanyak lima dan jumlah kolom sebanyak dua. Untuk tiap baris dalam tabel tersebut lakukan langkah berikut :
  - a. Di baris pertama
    - o Di kolom pertama ketikkan teks *Name*

- o Di kolom kedua, drag Textbox yang sebelumnya telah disediakan untuk isian field *UserName*. Kemudian di sebelah Textbox tersebut drag sebuah komponen *RequiredFieldValidator* dengan setting property *ControlToValidate* mengarah ke komponen *UserNameTextBox* dan property *ErrorMessage* berisi tanda asterisk atau tanda bintang (\*).
- b. Di baris kedua
  - o Di kolom pertama ketikkan teks *Old Password*
  - o Di kolom kedua, drag sebuah Label dan dalam Label tersebut, klik pada Smart Tag dan pilih sub menu *Edit Databinding*.



Selanjutnya di kotak dialog yang tersedia set *Field Binding* mengarah ke field *UserPassword*



- c. Di baris ketiga
- o Pada kolom pertama, ketikkan teks *New Password*
  - o Pada kolom kedua, drag Textbox yang sebelumnya telah tersedia untuk field *UserPassword*. Set property *TextMode* dari Textbox tersebut menjadi *Password*. Lalu drag sebuah komponen *RequiredFieldValidator* di sebelah Textbox tersebut. Set property *ControlToValidate* mengarah ke komponen *UserPasswordTextbox* dan property *ErrorMessage* menjadi tanda asterisk.
- d. Di baris keempat
- o Pada kolom pertama ketikkan teks *Confirm Password*
  - o Pada kolom kedua drag sebuah Textbox baru dengan setting property *TextMode*

menjadi *Password*. Di sebelah *Textbox* yang baru didrag, tempatkan sebuah komponen *CompareValidator* dengan setting property *ErrorMessage* menjadi teks *Password not match*. Sedangkan property *ControlToValidate* diarahkan ke komponen *Textbox1* dan property *ControlToCompare* diarahkan ke komponen *UserPasswordTextbox*

- e. Di baris terakhir
  - o Pada kolom pertama pindahkan Linkbutton *Update* didalamnya.
  - o Sedangkan di kolom yang kedua pindahkan Linkbutton *Cancel* yang sudah tersedia.

3. Kemudian hapus semua sisa komponen dan teks yang terdapat di luar tabel yang telah diset. Sekali lagi berhati-hatilah saat melakukan proses penghapusan sehingga hanya komponen yang di luar tabel yang terhapus. Berikutnya klik kanan lagi pada *Formview* dan pilih sub menu *Edit Template à HeaderTemplate* lalu ketikkan teks *User Profile* sebagai judul dari *Formview*.

...: User Profile :..

Name  \*

Old Password

New Password  \*

Confirm Password  Password not match

[Update](#) [Cancel](#)

4. Langkah terakhir adalah melakukan modifikasi pada komponen SqlDataSource, khususnya pada property untuk query. Di property *SelectQuery* klik tombol kecil sehingga menuju ke kotak dialog untuk setting query lalu modifikasi seperti gambar berikut :

SELECT command:

```
SELECT UserID, UserName, UserPassword, UserLevel, UserCreateDate, UserLastLogin,
UserEmail, UserActive FROM UserProfile WHERE (UserID = @UserID)
```

Refresh Parameters Query Builder...

Parameters:

Name	Value
UserID	Session("UserID")

Parameter source:

Session

SessionField:

UserID



Perhatikan bahwa parameter dari perintah Select ini diambil dari Session yang sebelumnya telah dihasilkan dari proses login di halaman web *Login.aspx*,



sehingga nantinya dalam Formview langsung muncul data dari pengguna yang sedang aktif login tanpa mempedulikan dari level member atau administrator.

---

---

Sedangkan untuk property *UpdateQuery*, klik juga tombol kecil dan di dalam kotak dialog yang tampil, modifikasi perintah query dengan perintah berikut :

UPDATE command:

```
UPDATE UserProfile SET UserName = @UserName, UserPassword = @UserPassword  
WHERE (UserID = @UserID)
```

## **Folder CMS – Administrasi User : Manageuser.aspx**

Berikutnya adalah pembuatan halaman web untuk melakukan manajemen user atau member yang telah teregistrasi. Halaman web ini nantinya hanya akan diakses oleh user dengan level administrator. Sehingga saat seorang member atau user dengan level non administrator akan tidak bisa melakukan akses ke halaman web ini.

Salah hal terpenting yang perlu diingat dalam pembuatan halaman web ini adalah berusaha untuk mendemonstrasikan penggunaan akses database dengan sepenuhnya bergantung pada model wizard. Sehingga listing yang ada dalam halaman web ini hanya untuk pengecekan hak akses level user. Hal ini sengaja dilakukan agar Anda dapat mengetahui berbagai jenis teknik akses database dalam buku ini, baik dengan menggunakan wizard, stored procedures, ataupun modifikasi query dengan menggunakan file teks.

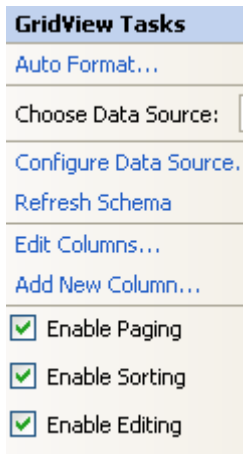
Langkah pembuatan halaman web *manageuser.aspx* selengkapnya adalah sebagai berikut :

1. Buat sebuah halaman web baru di dalam folder *cms* dengan mengacu kepada Master Page *admin.master*.

2. Dobel klik pada halaman web yang masih kosong, lalu ketikkan listing berikut ini di dalam prosedur *Page\_Load*

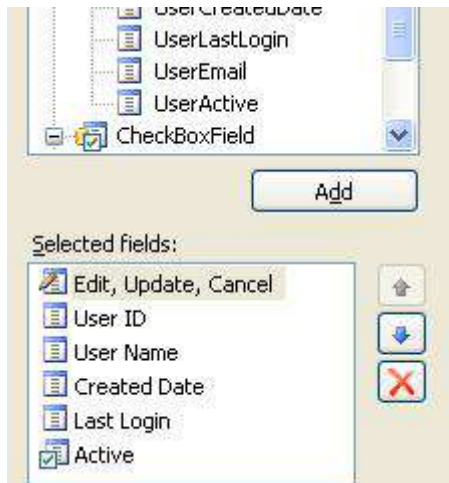
```
If Trim(Session("UserLevel")) _  
    <> "ADMIN" Then  
    Response.Redirect _  
        ("~/cms/default.aspx")  
End If
```

3. Selanjutnya, drag tabel *UserProfile* dari jendela Database Explorer sehingga terbentuk sebuah Gridview dan SqlDataSource dalam halaman web tersebut. Kemudian klik pada Smart Tag untuk Gridview tersebut, lalu pilih opsi menu *Paging*, *Sorting* dan *Editing*.



4. Lalu klik pada sub menu *Edit Columns* untuk menuju ke dalam kotak dialog pengeditan field yang akan ditampilkan. Dalam kotak dialog yang tersedia, hapus field *UserPassword*, *UserLevel* dan field *UserEmail*. Setelah itu, edit property *Header Text* dari field yang tersisa menjadi seperti uraian berikut :

- a. Field *UserID* à *User ID*
- b. Field *UserName* à *User Name*
- c. Field *UserCreateDate* à *Created Date*
- d. Field *UserLastLogin* à *Last Login*
- e. Field *UserActive* à *Active*



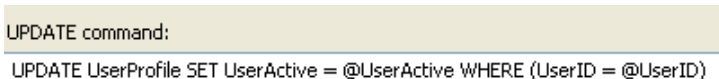
5. Kemudian untuk field *Created Date* ganti property *DataFormatString* menjadi `{0:dd-MM-yyyy}` dan untuk field *Last Login* menjadi `{0:dd-MM-yyyy}`

*hh:mm}*. Lalu set property *ReadOnly* untuk semua field menjadi *True* kecuali untuk field *Active*.

6. Kini di satu-satunya komponen *SqlDataSource* yang tersedia, set property *SelectQuery* menjadi seperti gambar berikut :



7. Dan langkah terakhir adalah melakukan modifikasi untuk property *UpdateQuery*



## Folder CMS – Setting Situs : Setting.aspx

Halaman web berikutnya yang dibuat adalah halaman web yang berfungsi untuk mengganti setting dari situs berdasarkan nilai atribut dari file web.config. Sama halnya dengan halaman web sebelumnya, halaman web ini seharusnya hanya bisa diakses oleh pengguna atau user dengan level administrator.

Salah hal unik dalam halaman web ini adalah tiadanya akses ke dalam database, tetapi digantikan dengan akses ke dalam file web.config dengan memanfaatkan fungsi yang telah dibuat di dalam class *Misc.vb*.

Langkah pembuatan selengkapnya adalah sebagai berikut :

1. Buat sebuah halaman web baru dengan nama *setting.aspx* di dalam folder *cms* dan mengacu ke Master Page *Admin.Master*.
2. Di dalam halaman web yang kosong, double klik sehingga menuju ke prosedur *Page\_Load* lalu ketikkan listing berikut :

```
If Not IsPostBack Then
    With xMisc
        txtheader.Text = _
            .getSetting("header")
        txtfooter.Text = _
```

```
.getSetting("footer")
txttitle.Text = _
.getSetting("title")
End With
End If
If Trim(Session("UserLevel")) _
    <> "ADMIN" Then
    Response.Redirect _
    ("~/cms/default.aspx")
End If
```

3. Di bawah deklarasi *Partial Class...* ketikkan deklarasi variabel obyek berikut :

```
Dim xMisc As New Misc
```

4. Selanjutnya, kembalilah lagi ke mode Design di halaman web dan buat sebuah tabel dengan jumlah baris sebanyak 5 dan dan jumlah kolom sebanyak 2 buah. Untuk masing-masing baris yang tersedia, lakukan tahapan berikut :

- a. Di baris pertama
  - o Pada kolom pertama ketikkan teks *Header*
  - o Pada kolom kedua drag sebuah Textbox dan ganti property *ID* menjadi *txtHeader*
- b. Di baris kedua
  - o Pada kolom pertama ketikkan teks *Footer*

- o Pada kolom kedua drag lagi sebuah Textbox yang property *ID*-nya diganti menjadi *txtFooter*
  - c. Di baris ketiga
    - o Pada kolom pertama ketikkan teks *Title*
    - o Pada kolom kedua drag Textbox yang terakhir dengan setting property *ID* menjadi *txtTitle*.
  - d. Di baris kelima kolom pertama, drag sebuah Button dan ganti property *Text* menjadi *Save Setting*.
5. Langkah yang terakhir dalam halaman web ini adalah melakukan double klik pada Button *Save Setting* untuk melakukan pengetikan listing berikut ini :

```
With xMisc
    .saveSetting _
    ("header", _
    txtheader.Text)
    .saveSetting _
    ("footer", _
    txtfooter.Text)
    .saveSetting _
    ("title", _
    txttitle.Text)
End With
```



## **Folder CMS – Pengumuman : Announcement.aspx**

---

Masih untuk pembuatan halaman web yang hanya bisa diakses oleh pengguna dengan level administrator, halaman web ini dinamakan *announcement.aspx*. Langkah pembuatan pertama ini sama dengan halaman web sebelumnya yaitu dengan membuat halaman web baru di dalam folder *cms* dan mengacu ke Master Page *Admin.Master*.

Di dalam halaman web ini menggunakan sebuah komponen non standard yakni *FreeTextBox*. Seperti telah dijelaskan di contoh situs yang lain, bahwa komponen ini adalah kompone jenis *Rich Text Editor* sehingga mampu untuk melakukan editing teks di dalam browser dengan menggunakan format-format seperti halnya dalam sebuah aplikasi *word processing*. Sebagai contoh adalah kemampuan untuk membuat format tebal (bold) dan cetak miring (italic).

Sesungguhnya terdapat banyak komponen dengan jenis *Rich Text Editor* yang dapat diunduh secara gratis maupun dengan cara membayar. Salah satu yang gratis dan digunakan di dalam contoh situs ini adalah komponen *Freetextbox* yang dapat didownload atau diunduh secara gratis di situs [www.freetextbox.net](http://www.freetextbox.net) .



---

---

Di dalam situs tersebut, pilihlah komponen Freetextbox untuk ASP .NET 2.0. Hingga buku ini selesai ditulis, versi terbaru yang tersedia adalah versi 3. Anda akan mendapatkan sebuah file format zip yang siap untuk diekstrak ke folder situs.

---

---

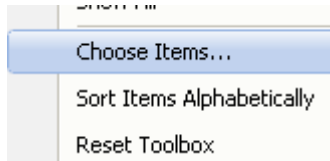
Jika Anda sudah selesai melakukan proses download, maka ekstrak komponen tersebut ke dalam sebuah folder. Untuk file dengan ekstensi *dll*, dapat dipindahkan ke folder *bin* dalam situs. Karena folder *bin* belum ada, maka buatlah terlebih dulu folder *bin* melalui Solution Explorer dengan melakukan klik kanan dan pilih sub menu *Add ASP .NET Folder à Bin*.

Kemudian buatlah lagi sebuah folder baru dengan nama *FreeTextBox* di Solution Explorer, tepat di root folder situs. Dan dari hasil ekstraksi komponen Freetextbox, dapat dicopy folder *Languages* ke dalam folder yang baru dibuat tersebut. Jika Anda ingin menghemat ruang harddisk, maka file-file format *xml* yang terdapat dalam folder tersebut dapat dihapus semua kecuali file *en-US.xml*.

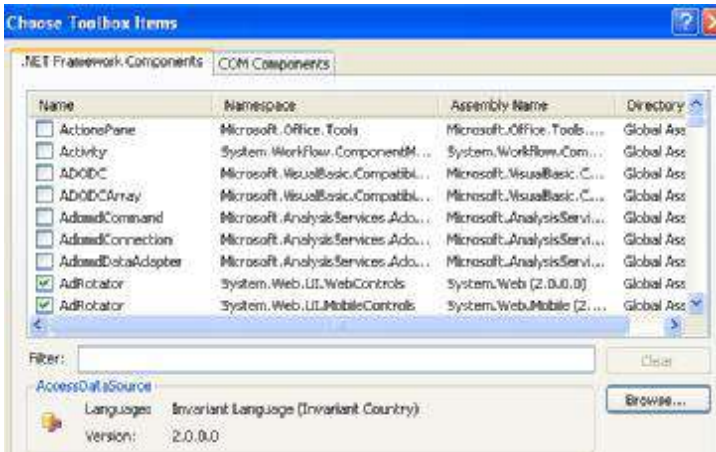


Operasi copy-paste folder atau file ke dalam Solution Explorer dapat dilakukan seperti halnya Anda menggunakan operasi copy-paste dalam Windows Explorer. Sehingga proses tersebut seharusnya bukan proses yang sulit dan lama.

Setelah file *FreeTextBox.dll* tercopy ke dalam folder *bin*, maka lakukan penempatan komponen Freetextbox ke dalam Toolbox. Untuk melakukan hal tersebut, klik kanan pada Toolbox, khususnya di tab *General*, lalu pilih sub menu *Choose Item*.



Kemudian dalam kotak dialog yang tersedia, klik tombol *Browse* dan arahkan ke folder *bin* tempat Anda menyimpan situs yang sedang dibuat. Lalu pilih file *FreeTextBox.dll* dan tutup kotak dialog yang tersedia. Jika proses ini berhasil, maka di dalam tab *General* akan muncul dua buah komponen baru yaitu Freetextbox dan ImageGallery.



Sedangkan untuk langkah berikutnya, ikutilah langkah berikut ini :

1. Di halaman web yang masih kosong lakukan double klik dan ketikkan listing berikut di dalam prosedur *Page\_Load*.

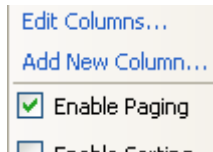
```

If Trim(Session("UserLevel")) _
    <> "ADMIN" Then
    Response.Redirect _
        ("<~/cms/default.aspx")
    End If
    
```

2. Selanjutnya drag tabel *Announcement* dari Database Explorer agar terbentuk sebuah GridView dan komponen SqlDataSource. Dan tepat di bawah komponen SqlDataSource tersebut, drag sebuah komponen Formview. Pada komponen Formview, ganti nilai property *DataSourceID* supaya mengarah

ke *SqlDataSource1*. Ganti juga property *DefaultMode* menjadi *Insert*.

3. Kini klik di Smart Tag pada komponen Gridview dan pilih opsi *Enable Paging*. Lalu masih di menu Smart Tag, pilih sub menu *Edit Columns* sehingga muncul kotak dialog modifikasi field.



Di dalam kotak dialog yang tersedia, hapus field *AnnouncementID* dan field *AnnouncementText*. Sedangkan untuk field *AnnouncementTitle* ganti property *HeaderText* menjadi *Title*. Dan untuk field *AnnouncementDate*, property *HeaderText* diganti menjadi *Date*, serta property *DataFormatString* menjadi *{0:dd-MM-yyyy hh:mm}*



4. Kemudian klik kanan pada Formview, dan pilih sub menu *Edit Template* à *InsertItemTemplate* sehingga menuju ke mode edit. Saat mode edit telah tampil, sisipkan sebuah tabel dengan jumlah baris sebanyak 3 dan jumlah kolom sebanyak dua.

Untuk masing-masing baris, lakukan langkah berikut :

a. Di baris pertama

Untuk kolom pertama ketikkan teks *Title* dan di kolom kedua drag Textbox yang sebelumnya telah ada untuk field *AnnouncementTitle*.

b. Di baris kedua

Blok kedua kolom yang ada dan lakukan operasi *Merge Cells* (lihat lagi sub bab sebelumnya yang membahas mengenai cara menggabungkan dua kolom menjadi satu). Lalu drag komponen Freetextbox didalamnya. Set property *SupportFolder* menjadi *~/FreeTextBox/* sehingga nanti akan mengarah ke folder yang sebelumnya telah dibuat. Set juga property *Width* menjadi *100%* agar komponen tersebut menjadi *stretch* di dalam Formview. Kemudian klik kanan pada Freetextbox dan pilih sub menu *Edit Databinding* lalu di dalam kotak dialog yang tampil arahkan property *Text* ke field *AnnouncementText*. Jangan lupa untuk memilih opsi *Show All Properties* supaya

property *Text* dapat terlihat dalam kotak dialog tersebut.



- c. Di baris terakhir, untuk tiap kolom yang ada drag Linkbutton *Insert* dan *Cancel* yang sudah ada sebelumnya.
5. Lalu hapus semua komponen yang tersisa di luar tabel, dan sekali lagi tetapkan berhati-hati saat melakukan proses penghapusan tersebut.
6. Langkah terakhir adalah melakukan modifikasi property *SelectQuery* dan *InsertQuery* di dalam komponen *SqlDataSource1*. Untuk modifikasi property *SelectQuery* lakukan seperti pada gambar berikut :

SELECT command:

```
SELECT [AnnouncementID], [AnnouncementTitle], [AnnouncementText],  
[AnnouncementDate] FROM [Announcement] ORDER BY [AnnouncementID] DESC
```

Sedangkan untuk modifikasi property *InsertQuery* ikuti tampilan berikut :

INSERT command:

```
INSERT INTO [Announcement] ([AnnouncementTitle], [AnnouncementText],  
[AnnouncementDate]) VALUES (@AnnouncementTitle, @AnnouncementText, getdate())
```



## **Folder CMS – Kategori Iklan : Category.aspx**

---

---

Halaman web berikutnya merupakan halaman web terakhir yang ditujukan khusus untuk administrator. Dalam halaman web ini, meski terlihat sederhana, tetapi didalamnya mendemonstrasikan penggunaan relasi rekursif di dalam sebuah tabel. Bahwa sebuah tabel dapat memiliki relasi dengan dirinya sendiri dan nantinya dapat membentuk sebuah hirarki yang sebenarnya tidak terbatas.

Dalam kasus contoh situs di bab ini, relasi rekursif yang dimiliki oleh tabel *Category* merupakan relasi antara kategori dan sub kategori yang dimiliki. Demi kemudahan penjelasan pembuatan halaman web, maka relasi rekursif yang disediakan sengaja dibatasi hingga dua level saja, yaitu kategori *Parent* dan sub kategori atau kategori *Child*.

Nantinya, di dalam halaman utama relasi rekursif tersebut akan ditempatkan di sebuah komponen Treeview sehingga akan terlihat hirarki yang jelas dalam penyusunan kategori.

Dalam halaman web ini pula terdapat implementasi teknik AJAX sehingga saat pengguna level administrator membuat sebuah kategori baru atau melakukan edit, maka tidak akan merasa meninggalkan halaman web tersebut. Dengan alasan demonstrasi

AJAX itu pula, perlu diingat bahwa pembuatan halaman web ini meski terlihat sederhana tetapi didalamnya mengandung banyak pelajaran yang bisa diaplikasikan ke situs lain. Sedangkan untuk halaman web di area content management system yang lain, jika ingin dikembangkan dengan menggunakan teknik AJAX dapat mengacu pada proses pembuatan halaman web ini.

Langkah awal pembuatan halaman web masih sama dengan sub bab sebelumnya, yaitu membuat sebuah halaman web baru di dalam folder *cms* dan mengacu ke Master Page *Admin.Master*. Sedangkan untuk tahapan berikutnya dapat diikuti langkah berikut ini :

1. Dobel klik pada halaman web yang masih kosong tersebut, sehingga menuju ke prosedur *Page\_Load* dan ketikkan listing ini :

```
If Trim(Session _  
("UserLevel")) <> _  
"ADMIN" Then  
    Response.Redirect _  
    ("~/cms/default.aspx")  
End If
```

2. Tempatkan sebuah komponen UpdatePanel di dalam halaman web yang masih kosong tersebut lalu drag

tabel *Category* dari Database Explorer didalamnya sehingga muncul sebuah Gridview dan sebuah komponen SqlDataSource.

3. Untuk SqlDataSource ganti property *SelectQuery* sehingga menjadi perintah query berikut :

SELECT command:

```
SELECT CategoryID, CategoryName, CategoryActive, CategoryParent FROM Category  
WHERE (CategoryParent = 0)
```

Sedangkan untuk property *InsertQuery* modifikasilah perintah query menjadi :

INSERT command:

```
INSERT INTO Category(CategoryName, CategoryActive, CategoryParent) VALUES  
(@CategoryName, 1, @CategoryParent)
```

4. Kemudian klik pada Smart Tag yang terdapat di dalam Gridview dan pilih opsi untuk fitur *Paging*, *Editing* dan *Selection*.

<input checked="" type="checkbox"/>	Enable Paging
<input type="checkbox"/>	Enable Sorting
<input checked="" type="checkbox"/>	Enable Editing
<input type="checkbox"/>	Enable Deleting
<input checked="" type="checkbox"/>	Enable Selection

5. Lalu pilih sub menu untuk *Edit Columns* sehingga kotak dialog untuk edit field tampil. Dalam kotak dialog tersebut, hapuslah field *CategoryParent* dan field *CategoryID*. Sedangkan untuk field

*CategoryName* ganti property *HeaderText* menjadi *Name*, dan property *ReadOnly* menjadi *true*. Dan terakhir, untuk field *CategoryActive*, ganti nilai property *HeaderText* menjadi *Active*.



6. Kemudian double klik pada Gridview yang baru saja dimodifikasi untuk menyetikkan listing berikut di dalam prosedur *SelectedIndexChanged*

```
Dim xrow As GridViewRow = _  
    GridView1.Rows _  
    (e.NewSelectedIndex)
```

7. Di bawah Gridview yang pertama tersebut, ketikkan sebuah teks *Sub Category* sebagai teks keterangan.
8. Selanjutnya drag lagi komponen *UpdatePanel* yang kedua, dan drag ulang tabel *Category* dari jendela *Database Explorer* sehingga terbentuk Gridview yang kedua beserta *SqlDataSource* yang baru.
9. Di dalam *SqlDataSource*, modifikasi property *SelectQuery* menjadi kotak dialog berikut :

```
SELECT command:
SELECT CategoryID, CategoryName, CategoryActive, CategoryParent FROM Category
WHERE (CategoryParent = @categoryparent)
```

Refresh Parameters Query Builder...

Parameters:

Name	Value
categoryparent	GridView1.SelectedVale

Parameter source:

- Control
- ControlID:
- GridView1



Jika di dalam isian *ControlID* komponen *GridView1* tidak dikenali, maka tidak perlu dikhawatirkan. Hal tersebut terjadi karena komponen *GridView1* telah “terbungkus” di dalam komponen *UpdatePanel* yang pertama.

10. Sedangkan untuk *GridView* yang kedua, klik pada *Smart Tag* yang tersedia, kemudian pilih fitur *Paging* dan *Editing*.

- Enable Paging
- Enable Sorting
- Enable Editing

Berikutnya klik pada sub menu untuk *Edit Columns*. Dalam modifikasi field yang tersedia, sama persis

dengan modifikasi Gridview yang pertama (lihat lagi langkah sebelumnya).

11. Setelah itu, drag lagi sebuah UpdatePanel yang terakhir dalam halaman web ini. Dalam UpdatePanel tersebut, drag sebuah komponen Formview. Ganti nilai property *DefaultMode* untuk Formview tersebut menjadi *Insert* dan property *DataSourceId* mengarah ke komponen *SqlDataSource* yang kedua.
12. Berikutnya, klik kanan pada Formview tersebut, dan pilih menu *Edit Templates à InsertItemTemplate*. Pada template yang tersedia, sisipkan sebuah tabel dengan jumlah baris sebanyak lima dan jumlah kolom sebanyak dua buah. Untuk tiap baris yang ada dalam tabell tersebut, lakukan langkah berikut ini :
  - a. Di baris pertama :

Pada kolom pertama ketikkan teks *Category* dan di kolom kedua drag Textbox yang sebelumnya telah tersedia untuk field *CategoryName*. Tepat di sebelah Textbox yang baru saja dipindah posisinya, drag sebuah komponen *RequiredFieldValidator*. Di komponen tersebut set property *ControlToValidate* mengarah ke

*CategoryNameTextbox* dan property *ErrorMessage* menjadi tanda asterisk (\*).

b. Di baris kedua :

Dari dua kolom yang tersedia jadikan satu dengan menggunakan operasi *Merge Cells* lalu drag *Checkbox* dan set property *AutoPostBack* menjadi *True* serta property *Text* menjadi *Sub Category*. *Checkbox* ini nantinya akan menjadi kunci dari relasi rekursif yang diterapkan di dalam tabel *Category*. Setelah itu, dobel klik di *Checkbox* tersebut, sehingga menuju ke prosedur *CheckedChanged* dan ketikkan listing berikut :

```
CType(FormView1. _  
    FindControl("Panel1"), _  
    Panel).Visible = _  
sender.checked
```



---

Saat *Checkbox* diklik, maka bila *Checkbox* dalam keadaan aktif akan menampilkan sebuah *Panel* berisi *Dropdownlist* yang akan dibuat di baris ketiga.

---

c. Di baris ketiga :

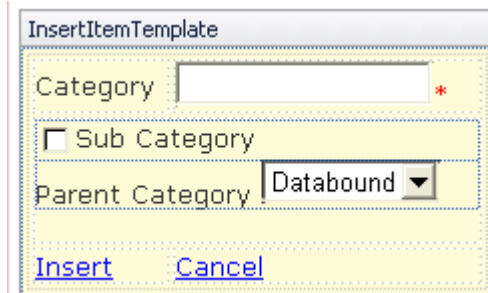
Sama dengan baris sebelumnya, jadikan dua kolom yang tersedia menjadi satu kolom dengan operasi *Merge Cells*. Lalu drag sebuah komponen Panel di dalam baris tersebut. Set property *visible* dari komponen Panel tersebut menjadi *false*. Di Panel yang tersedia, ketikkan teks *Parent Category* dan di samping teks tersebut, drag sebuah Dropdownlist. Lalu set property dari Dropdownlist tersebut seperti pada gambar berikut ini :

DataSourceID	<b>SqlDataSource1</b>
DataTextField	<b>CategoryName</b>
DataTextFormatString	
DataValueField	<b>CategoryID</b>

d. Di baris kelima :

Di masing-masing kolom yang tersedia drag Linkbutton *Insert* dan Linkbutton *Cancel* yang sebelumnya telah tersedia. Jika sudah selesai dilakukan, maka hapus semua komponen dan teks yang berada di luar tabel yang baru saja selesai dimodifikasi, sehingga tampilan Formview akan tampak seperti pada gambar berikut ini :





InsertItemTemplate

Category  \*

Sub Category

Parent Category: Databound ▼

[Insert](#)   [Cancel](#)

13. Kemudian lakukan klik kanan lagi pada Formview dan pilih sub menu *Edit Template à HeaderTemplate*. Ketikkan teks *New Category* sebagai tanda atau judul dari Formview tersebut. Lalu akhiri proses edit Formview dengan cara klik kanan dan pilih sub menu *End Editing Template*.
14. Kini beralihlah ke UpdatePanel yang pertama, dan set property *UpdateMode* menjadi *Conditional*. Kemudian di property *Triggers* tempatkan satu *Asynchronous Trigger* dari Formview saat selesai melakukan proses inputan data (*ItemInserted*).



---

---

Dengan adanya trigger tersebut, berarti bahwa komponen apapun yang terdapat dalam UpdatePanel akan mengalami proses refresh saat prosedur yang ditempatkan dalam trigger tersebut dieksekusi.

---

---

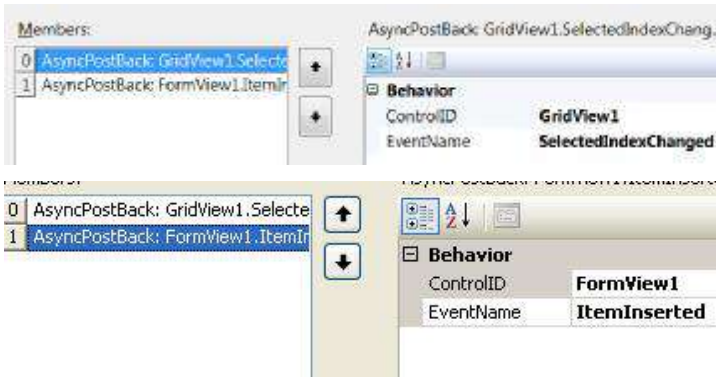


---

Jika Anda kesulitan untuk menemukan letak komponen UpdatePanel saat akan melakukan setting property, Anda dapat langsung menuju ke jendela Properties dan dari combobox yang tersedia pilih UpdatePanel yang akan diset.

---

15. Selanjutnya di UpdatePanel yang kedua (yang didalamnya terdapat GridView untuk Sub Category) set property *Update Mode* menjadi *Conditional*. Dan di UpdatePanel yang terakhir (yang didalamnya ditempati komponen Formview), property *UpdateMode* juga diset menjadi *Conditional* dan property *Triggers* diset seperti pada gambar berikut ini :



Dalam komponen UpdatePanel yang terakhir terdapat dua buah pemicu refresh atau *Trigger* yang ditempatkan yaitu dari Gridview yang pertama (saat pengguna melakukan klik pada Linkbutton *Select* yang terdapat dalam Gridview) dan dari dirinya sendiri (Formview yang ada di dalam UpdatePanel). Saat Formview telah selesai mengalami proses input data, maka UpdatePanel akan langsung menjalani eksekusi refresh didalamnya.

16. Langkah terakhir dalam halaman web ini adalah melakukan double klik pada Formview dan mengetikkan listing untuk prosedur *ItemInserting* dan *ItemInserted* dari Formview tersebut. Perlu diingat untuk kedua prosedur tersebut harus dilakukan dengan mengganti tujuan method terlebih

dulu (secara default, saat dilakukan double klik akan menuju ke prosedur *PageIndexChanged*). Listing untuk kedua prosedur tersebut adalah :

a. ItemInserting

```
Dim xParent As _
    Integer = 0
With FormView1
    If CType(.FindControl _
        ("CheckBox1"), _
        CheckBox).Checked Then
        xParent = CType( _
            .FindControl _
            ("DropDownList1"), _
            DropDownList). _
            SelectedValue
    End If
End With
With SqlDataSource1
    .InsertParameters _
        ("CategoryParent"). _
        DefaultValue = xParent
End With
```

b. ItemInserted

```
GridView1.DataBind()
GridView2.DataBind()
```

## **Folder CMS – Pasang Iklan : NewAds.aspx**

---

Halaman web yang dibuat dalam sub bab ini adalah halaman web inti dalam konteks content management system, yaitu halaman web untuk melakukan input data iklan baru. Halaman web ini nantinya kebalikan dari halaman web sebelumnya yaitu hanya bisa diakses oleh pengguna dengan level member, bukan administrator.

Dalam halaman web ini juga menggunakan komponen Freetextbox, sehingga diasumsikan bahwa Anda telah mengikuti langkah-langkah yang telah ada sebelumnya, atau telah membuat seluruh halaman web yang telah dibuat sebelumnya.

Langkah pertama pembuatan halaman web ini juga masih sama dengan sub bab sebelumnya yaitu dengan membuat halaman web baru yang mengacu ke Master Page *Admin.Master*. Kemudian ikuti langkah berikut ini :

1. Drag tabel *Ads* dari Database Explorer sehingga tampil sebuah Gridview dan SqlDataSource dalam halaman web tersebut. Kemudian hapus Gridview yang terbentuk dan ganti dengan sebuah komponen Formview.

2. Di Formview yang baru didrag tersebut, set property *Width* menjadi *100%* dan property *DefaultMode* menjadi *Insert*.
3. Selanjutnya di komponen *SqlDataSource*, modifikasi property *InsertQuery* seperti pada gambar berikut :

INSERT command:

```
INSERT INTO Ads(CategoryID, AdsTitle, AdsText, UserID, AdsDateDisplay) VALUES (@CategoryID, @AdsTitle, @AdsText, @UserID, @AdsDateDisplay)
```

4. Kemudian klik kanan di Formview dan pilih sub menu *Edit Template* → *InsertItemTemplate* untuk melakukan modifikasi pada template Formview yang ada. Setelah muncul template yang dimaksud, sisipkan sebuah tabel baru dengan jumlah baris sebanyak tujuh dan jumlah kolom sebanyak dua. Untuk tiap baris yang ada, lakukan langkah berikut ini :
  - a. Di baris pertama  
Di kolom pertama ketikkan teks *Category* dan di kolom kedua drag sebuah Dropdownlist dan drag tabel *Category* dari Database Explorer. Saat melakukan drag tabel, maka akan tampil sebuah Gridview, tapi Gridview tersebut kemudian dihapus sehingga hanya tersisa *SqlDataSource*. Selanjutnya di komponen Dropdownlist atur

property yang ada seperti pada gambar berikut ini :

DataSourceID	SqlDataSource2
DataTextField	CategoryName
DataTextFormatString	
DataValueField	CategoryID

Selain setting property tersebut, set juga property *AutoPostBack* dari Dropdownlist tersebut menjadi *True*. Kemudian double klik pada Dropdownlist tersebut, untuk mengetikkan listing di dalam prosedur *SelectedIndexChanged*. Tetapi sebelumnya, deklarasikan variabel umum tepat di bawah deklarasi *Partial Class...* yakni :

```
Dim xMisc As New Misc
```

Dan isi dari prosedur untuk Dropdownlist adalah :

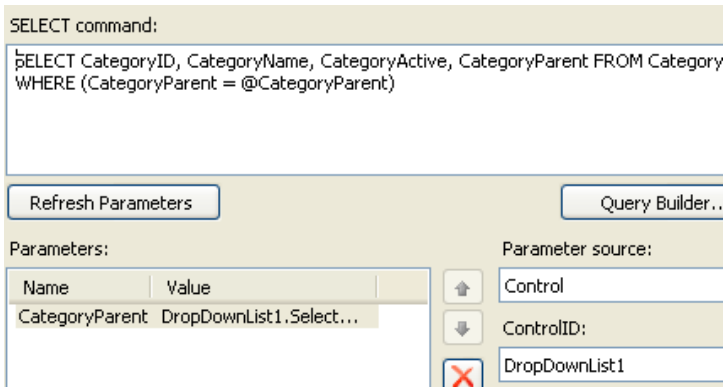
```
With xMisc  
    .findViewById(FormView1, _  
        "SqlDataSource3").databind()  
    .findViewById(FormView1, _  
        "DropDownList2").databind()  
End With
```

b. Di baris kedua

Pada kolom kedua, drag sebuah Checkbox dan ganti property *Text* menjadi *Sub Category* dan property *AutoPostBack* menjadi nilai *true*. Lalu drag sebuah komponen UpdatePanel tepat di bawah Checkbox tersebut. Dalam UpdatePanel yang ada, tempatkan sebuah komponen Panel baru. Jangan lupa untuk mengatur property *Visible* dari Panel tersebut menjadi *False*. Kemudian lakukan langkah yang hampir sama dengan baris sebelumnya yaitu menempatkan sebuah Dropdownlist yang kedua dan drag tabel *Category* dari jendela Database Explorer. Dari hasil drag tabel tersebut, hapus komponen GridView yang terbentuk sehingga hanya tersisa sebuah komponen SqlDataSource.

Di komponen SqlDataSource tersebut, modifikasi property *SelectQuery* seperti pada gambar berikut :





Sedangkan di komponen DropDownList, modifikasi property seperti pada gambar ini :

DataSourceID	SqlDataSource3
DataTextField	CategoryName
DataTextFormatString	
DataValueField	CategoryID

Selanjutnya double klik pada Checkbox, dan di dalam prosedur *CheckedChanged* ketikkan listing berikut ini :

```
ViewPanel _
("Panell1", sender)
```

Agar prosedur tersebut berjalan benar, maka perlu juga dibuat sebuah prosedur baru bernama *ViewPanel*. Sedangkan isi dari prosedur *ViewPanel* adalah :

```
Sub ViewPanel _
    (ByVal xpanel As String, _
    ByVal xcheck As CheckBox)
```

```
xMisc.findInView(FormView1, _  
xpanel).Visible = _  
xcheck.Checked
```

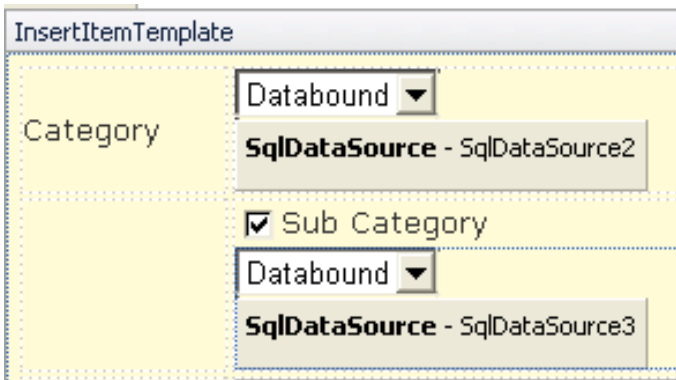
End Sub

Selanjutnya di dalam UpdatePanel yang tersedia, set property *UpdateMode* menjadi *Conditional* dan property *Triggers* seperti pada gambar berikut ini :

Behavior	
ControlID	DropDownList1
EventName	SelectedIndexChanged

Behavior	
ControlID	Checkbox1
EventName	CheckedChanged

Hasil dari modifikasi dua baris pertama tampak seperti pada gambar berikut ini :

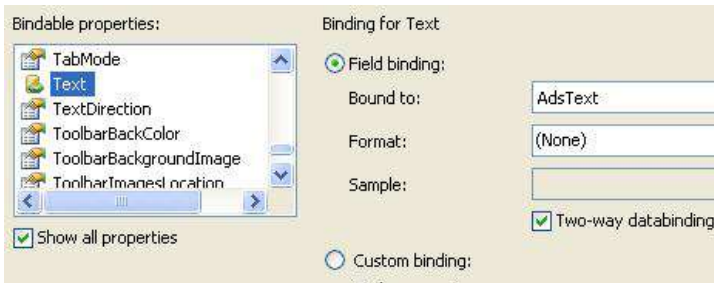


c. Di baris ketiga

Pada kolom pertama ketikkan teks *Title* dan di kolom kedua pindahkan Textbox yang sebelumnya telah tersedia untuk field *AdTitle*.

d. Di baris keempat

Gabungkan dua kolom yang tersedia dengan menggunakan operasi *Merge Cells* lalu drag sebuah komponen Freetextbox didalamnya. Set property *Width* menjadi *100%* dan property *SupportFolders* menjadi *~/FreeTextBox/*. Kemudian klik pada Smart Tag di komponen tersebut, dan pilih menu *Edit Databinding* sehingga muncul kotak dialog setting untuk field yang diatur seperti pada tampilan berikut ini :



e. Di baris kelima

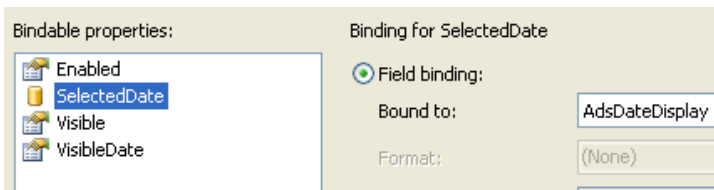
Pada kolom pertama drag sebuah Checkbox baru yang property *ID*-nya diganti menjadi

*chkScheduled* dan property *AutoPostBack* diset menjadi nilai *True*. Sedangkan property *Text* diisi dengan teks *Scheduled Ads*.

Kemudian di kolom kedua drag sebuah komponen *UpdatePanel* baru dengan property *UpdateMode* yang diganti menjadi *Conditional* dan property *Triggers* menjadi seperti pada gambar berikut ini :

Behavior	
ControlID	ChkScheduled
EventName	CheckedChanged

Setelah selesai, drag sebuah komponen *Panel* di dalam *UpdatePanel* tersebut. Dan drag sebuah komponen *Calendar* di dalam *Panel* yang ada. Set property *Visible* dari *Panel* tersebut menjadi *False*, sedangkan untuk komponen *Calendar*, klik pada *Smart Tag* dan pilih sub menu *Edit Databinding*. Saat muncul kotak dialog dari hasil pilihan tersebut, set property *SelectedDate* menjadi seperti pada gambar berikut ini :



Kemudian double klik pada Checkbox di kolom pertama untuk mengetikkan listing berikut di prosedur *CheckedChanged*.

```
ViewPanel _  
("Panel2", sender)
```

f. Di baris yang terakhir, untuk tiap kolom yang ada pindahkan Linkbutton untuk *Insert* dan Linkbutton *Cancel*. Setelah selesai, hapuslah semua field dan teks yang tersedia di luar tabel dan klik kanan pada Formview untuk mengakhiri proses edit dengan memilih sub menu *End Template Editing*. Anda juga bisa menambahkan header untuk Formview dengan melakukan setting pada template *HeaderTemplate* dan mengisinya dengan teks *New Ads*.

5. Langkah terakhir dari pembuatan halaman web ini adalah melakukan double klik pada Formview dan pindahkan method default ke prosedur *ItemInserting* untuk mengetikkan listing berikut :

```
SqlDataSource1. _  
InsertParameters _  
("UserID").DefaultValue = _  
Session("UserID")  
With xMisc
```

```
Dim xCategory As _
Integer = .findInView _
(FormView1, _
"DropDownList1"). _
selectedvalue()
If .findInView _
(FormView1, _
"CheckBox2").checked Then
    If .findInView _
(FormView1, _
"DropDownList2"). _
items.count > 0 Then
        xCategory = _
.findInView _
(FormView1, _
"DropDownList2"). _
selectedvalue()
    End If
End If
SqlDataSource1. _
InsertParameters _
("CategoryID"). _
DefaultValue = _
xCategory
Dim xScheduled As _
CheckBox = _
.findInView _
```

```
(FormView1, _  
    "chkScheduled")  
Dim xDate As DateTime = Now  
If xScheduled.Checked Then  
    xDate = .findInView _  
        (FormView1, _  
            "Calendar1"). _  
            selecteddate  
End If  
SqlDataSource1. _  
InsertParameters _  
("AdsDateDisplay"). _  
DefaultValue = xDate  
End With
```

## Folder CMS – Edit Iklan : EditAds.aspx

Halaman web terakhir yang dibuat di dalam folder *cms* adalah halaman web untuk melakukan edit iklan, tetapi hanya dibatasi untuk menonaktifkan iklan yang sebelumnya telah diinputkan oleh pengguna yang bersangkutan. Halaman web ini relatif sangat mudah dan cepat dikerjakan, karena tidak memiliki listing sama sekali dalam pengerjaannya.



---

Pengerjaan halaman web tanpa adanya listing satu barispun lazim disebut sebagai *zero listing web page*. Teknik ini dulunya selalu dibuat sebagai *marketing gimmick* saat mempromosikan kemudahan penggunaan ASP .NET di awal kemunculannya.

---

Langkah pertama dalam pembuatan halaman web ini masih sama dengan sub bab sebelumnya yaitu di dalam folder *cms* dengan masih mengacu ke Master Page *Admin.Master*. Kemudian ikuti langkah-langkah berikut ini :

1. Tempatkan sebuah komponen UpdatePanel dalam halaman web tersebut lalu drag tabel *Ads* dari jendela Database Explorer ke dalam UpdatePanel



tersebut, dan dari hasil tersebut akan tampil sebuah Gridview dan SqlDataSource.

2. Untuk SqlDataSource, modifikasi property *SelectQuery* menjadi seperti gambar berikut :

SELECT command:

```
SELECT Ads.AdsID, Ads.AdsDate, Ads.CategoryID, Ads.AdsTitle, Ads.AdsText, Ads.UserID, Ads.AdsStatus, Ads.AdsClick, Ads.AdsDateDisplay, Category.CategoryName FROM Ads INNER JOIN Category ON Ads.CategoryID = Category.CategoryID WHERE (Ads.UserID = @userid) and (adsstatus=true) order by ads.adsid desc
```

Refresh Parameters Query Build

Parameters:

Name	Value
userid	Session("UserID")

Parameter source:

- Session
- SessionField:
- UserID



Perhatikan bahwa parameter utama diambil dari *Session UserID* sehingga hanya iklan yang dipasang oleh member yang bersangkutan yang dapat diedit di dalam halaman web ini.

*Session* tersebut diasumsikan telah didapat saat pengguna melakukan proses login (lihat lagi di pembahasan mengenai halaman web *login.aspx*).

Sedangkan untuk property *DeleteQuery* :

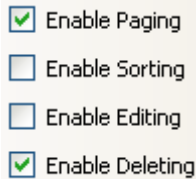
UPDATE command:

```
UPDATE Ads SET AdsStatus = @AdsStatus WHERE (AdsID = @AdsID)
```



Meski property *DeleteQuery* sesungguhnya berfungsi untuk melakukan penghapusan data, tetapi dalam halaman web ini sengaja diganti menjadi perintah untuk mengedit data. Sehingga pada saat pengguna melakukan klik *Delete* yang dilakukan oleh halaman web hanyalah mengedit status aktif dari iklan tersebut.

3. Kemudian klik pada Smart Tag yang terdapat di dalam Gridview dan pilih opsi *Paging* dan *Deleting*.



Setelah itu pilih sub menu *Edit Columns*, dan di kotak dialog yang tersedia hapus field *AdsID*, *CategoryID*, *AdsText*, *UserID* dan *AdsDateDisplay*. Sedangkan untuk field *CategoryName*, pindahkan urutannya di bawah field *AdsDate*. Dan ganti semua property *ReadOnly* untuk semua field menjadi *True*. Lalu untuk field yang ada, ganti property *HeaderText* menjadi :

- a. *AdsDate* à *Date*
- b. *CategoryName* à *Category*

- c. *AdTitle* à *Title*
- d. *AdStatus* à *Active*
- e. *AdClick* à *Click Amount*



---

---

Setelah menyelesaikan halaman web ini, disarankan agar segera melakukan proses testing, dengan menginputkan beberapa data iklan dan beberapa user baru. Selanjutnya cobalah untuk mengedit dan menginput beberapa kategori iklan yang baru, baik dari sub kategori atau kategori langsung. Sehingga pada saat pembuatan halaman web berikutnya dapat langsung terlihat hasilnya.

---

---

## Halaman Utama : Default.aspx

---

Kini sampailah pada langkah yang (benar-benar) terakhir dalam pembuatan situs ini, yaitu pembuatan halaman utama yang ada di root folder. Karena sebelumnya telah tersedia sebuah halaman web *default.aspx*, disarankan untuk menghapus terlebih dulu halaman web tersebut. Kemudian buat ulang sebuah halaman web baru di root folder dengan nama yang sama, tetapi mengacu ke Master Page *MiniAds.Master* (Master Page yang pertama dan terletak di root folder).

Dari keseluruhan halaman web yang ada dalam situs ini, halaman utama ini merupakan halaman yang paling kompleks teknik pembuatannya, karena melakukan proses relasi rekursi sekaligus menerapkan teknik AJAX secara penuh, sehingga pengguna hanya akan berada di dalam satu halaman web saat menjelajah iklan-iklan yang ada.



---

Tentu saja terdapat kerugian dan keuntungan dalam penerapan teknik tersebut yaitu pada proses browsing iklan, maka diharapkan pengguna tidak akan pernah “terlepas”

dari situs yang dibuat. Tetapi di sisi yang lain, sebuah iklan tidak akan bisa dibuat sebuah *bookmark* untuk memudahkan akses didalamnya.

---

---

Untuk menyelesaikan pembuatan halaman web ini, lakukan langkah berikut ini :

1. Setelah pembuatan ulang halaman web *default.aspx*, maka berpindahlah terlebih dulu ke mode Source untuk menempatkan layer dari style sheet yang telah dibuat.

```
<div class="kiriUtama">  
</div>  
<div class="kananUtama">  
</div>
```

2. Kemudian, di dalam mode Design bagian layer sebelah kiri, tempatkan sebuah tabel dengan jumlah baris sebanyak enam dan jumlah kolom sebanyak satu. Untuk tiap baris yang telah tersedia, lakukan langkah berikut :

- a. Di baris pertama  
Drag sebuah Treeview dengan property *ImageSet* diarahkan ke *BulletedList* dan property *ShowExpandCollapse* diberi nilai *False*.
- b. Di baris kedua

Ketikkan teks *Find Ads*

- c. Di baris ketiga  
Tempatkan sebuah Texbox dengan property *Width* diset menjadi *100%*.
- d. Di baris keempat  
Tempatkan sebuah Linkbutton dengan property *Text* menjadi *Search* dan property *ID* berisi nilai *InkSearch*.
- e. Di baris kelima  
Drag sebuah komponen *UpdateProgress* yang didalamnya ditempati sebuah file image format gif animator. Seperti telah dijelaskan di beberapa bab sebelumnya, file-file jenis ini bisa Anda dapatkan secara gratis di situs [www.ajaxload.info](http://www.ajaxload.info)
- f. Di baris terakhir  
Drag sebuah komponen Label yang dikosongkan untuk nilai property *Text*, sedangkan property *ID* diganti menjadi *lblUser*. Dan di bawah Label tersebut, drag sebuah Linkbutton dengan property *Text* menjadi *Logout*. Kedua buah komponen tersebut nantinya akan menjadi tanda bahwa jika seorang pengguna sedang dalam keadaan login sebagai member atau dalam

posisi sebagai *guest*. Sehingga meski dalam keadaan sebagai member masih tetap bisa “melepas” status memberrnya supaya iklan yang telah dipasang dapat segera terlihat (memaksa proses refresh terjadi di halaman web).

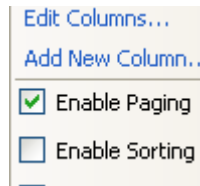


3. Sedangkan di bagian layer sebelah kanan, drag terlebih dulu sebuah komponen UpdatePanel dengan setting property *Triggers* seperti pada gambar berikut (terdiri dari dua buah *Triggers*) :

Behavior	
ControlID	TreeView1
EventName	SelectedNodeChanged

Behavior	
ControlID	LnkSearch
EventName	Click

4. Kemudian di dalam UpdatePanel tersebut, drag tabel *Ads* dari Database Explorer sehingga tampil sebuah Gridview dan sebuah komponen SqlDataSource. Untuk Gridview yang ada, set property *Width* menjadi *100%* dan property *PageSize* menjadi nilai *2*. Lalu klik pada Smart Tag di Gridview untuk memilih opsi *Paging* dan klik di sub menu *Edit Columns* untuk menuju kotak dialog pemilihan field



5. Di dalam kotak dialog untuk pemilihan field, hapus semua field yang tersedia kecuali field *AdsID*. Lalu jadikan field tersebut supaya berjenis *Template Field* dengan mengeksekusi Linkbutton yang terletak di bagian kanan bawah dari kotak dialog tersebut.





6. Setelah selesai, berpindahlah ke mode Source dan di dalam tag bagian GridView, khususnya di bagian tag *ItemTemplate*, modifikasilah seperti pada listing berikut ini :

```
<ItemTemplate>
    <asp:LinkButton ID="LinkButton1"
        runat="server"
        CssClass="juduliklan"
        OnClick="KlikDetail"
        CommandArgument='<#eval("AdsID") %>'
        Text='<#Eval("AdsTitle")%>'>
    </asp:LinkButton><hr />
    <div class="headeriklan">
        <#Eval("UserID")%> -
        <#Format(Eval("AdsDateDisplay"), _
            "dd-MM-yyyy")%>
        - Telah diklik sebanyak
        <#Eval("AdsClick")%> kali
    </div>
    <asp:Panel ID="Panell1"
        runat="server"
        Visible='<#session("Detail") %>'>
        <b>Kategori :
        <#Eval("CategoryName")%>
        </b><br />
        <#Eval("AdsText")%>
    </asp:Panel><hr />
```

```
<asp:LinkButton
ID="LinkButton2"
runat="server"
OnClick="BackToAll"
Visible='<%#session("Detail") %>'>
Back</asp:LinkButton>
<hr style =
"border-bottom:solid 2px red" />
</ItemTemplate>
```

Di dalam modifikasi tersebut, terdapat beberapa hal penting yang harus diperhatikan yaitu :

- a. Adanya komponen Panel di dalam Gridview sebagai tempat teks dan field Kategori dan Linkbutton yang dapat mengembalikan mode data ke mode *summary* atau rangkuman iklan.
  - b. Template awal dari tampilan Gridview yang menampilkan judul iklan sebagai sebuah Linkbutton untuk menuju ke detail dari iklan yang dimaksud.
  - c. Setiap perubahan tampilan akan ditandai dengan Session yang nanti ditempatkan di listing program.
7. Kemudian drag tabel *Announcement* di bawah Gridview. Dari hasil drag tersebut, akan

mendapatkan sebuah GridView yang kedua dan SqlDataSource yang kedua pula. Tetapi kali ini GridView tersebut dihapus, dan sebagai gantinya drag sebuah komponen Repeater di tempat yang sama. Ganti property *DataSourceID* dari Repeater tersebut mengarah ke *SqlDataSource2*. Setelah itu, berpindahlah ke mode Source dan di bagian tag Repeater, khususnya di bagian *ItemTemplate* modifikasi menjadi listing berikut ini :

```
<ItemTemplate>
    <asp:Panel ID="Panel2"
    runat="server"
    Visible='<%=# Announce() %>'>
        <b>News Flash :</b><br />
        <%=#Eval("AnnouncementText")%>
    </asp:Panel>
</ItemTemplate>
```

8. Sedangkan untuk SqlDataSource yang kedua, ganti property *SelectQuery* menjadi seperti pada gambar berikut :

SELECT command:

```
SELECT TOP (1) AnnouncementID, AnnouncementTitle, AnnouncementText,
AnnouncementDate FROM Announcement ORDER BY AnnouncementID DESC
```



---

---

Parameter *Top(1)* berarti bahwa hanya ada satu record teratas yang akan ditampilkan dari hasil perintah *Select* tersebut.

---

---

Kini, untuk pengetikan listing dari halaman web ini, dapat diikuti langkah-langkah berikut :

1. Dobel klik di bagian kosong halaman web sehingga menuju ke prosedur *Page\_Load* dan ketikkan listing berikut :

```
refreshStatus()  
If Not IsPostBack Then  
refreshGrid("")  
With TreeView1  
Dim xRoot As _  
TreeNode = _  
MakeNode("All", "")  
TreeView1. _  
Nodes.Add(xRoot)  
Dim xRow _  
As Data.DataRow  
Dim xParent _  
As Data.DataSet = _  
xData.GetDataset _  
(IO.File.ReadAllText _  
(Server.MapPath _  
("~/app_data/" & _
```

```
        "selectParent.txt"))))
For Each xRow In _
    xParent.Tables(0).Rows
    Dim xTotal As _
    Data.DataSet = _
    xData.GetDataset _
    (IO.File.ReadAllText _
    (Server.MapPath _
    ("~/app_data/" & _
    "selectCategory.txt")) & _
    xRow("KategoriID") & _
    ") GROUP BY " & _
    "Category.CategoryParent," & _
    "Category_1.CategoryName, " & _
    "Category.CategoryID, " & _
    "Category.CategoryName")
Dim xTotalParent As _
    Data.DataSet = _
    xData.GetDataset _
    (IO.File.ReadAllText _
    (Server.MapPath _
    ("~/app_data/" & _
    "selectCount.txt")) & _
    xRow("KategoriID") & _
    ") OR " & _
    "(Category.CategoryParent =" & _
    xRow("KategoriID") & ") ")
```

```
Dim xParentTree As TreeNode = _
    MakeNode(xRow("Kategori") & _
        " - " & _
        xTotalParent.Tables(0). _
        Rows(0).Item(0), _
        xRow("KategoriID"))
xRoot.ChildNodes. _
    Add(xParentTree)
If xTotal.Tables(0). _
    Rows.Count > 0 Then
    For xcount As Integer = _
        0 To xTotal.Tables(0). _
            Rows.Count - 1
        xParentTree.ChildNodes.Add _
            (MakeNode(xTotal. _
                Tables(0).Rows(xcount). _
                Item("CategoryName") & _
                " - " & _
                xTotal.Tables(0). _
                Rows(xcount).Item _
                ("Total"), _
                xTotal.Tables(0). _
                Rows(xcount).Item _
                ("CategoryID")))
    Next
End If
Next
```

End With

End If



---

---

Perlu diingat bahwa beberapa fungsi dan prosedur yang dibutuhkan dalam prosedur ini belum diketikkan (baru akan diketikkan di langkah berikutnya). Sehingga jika pada saat pengetikan muncul pesan error, untuk sementara abaikan terlebih dulu.

---

---

Isi dari prosedur ini, secara umum terbagi menjadi beberapa bagian yakni :

- a. Pengisian Gridview yang pertama sehingga akan ditampilkan iklan secara keseluruhan.
  - b. Pengisian Treeview untuk kategori dengan metode relasi secara rekursif.
2. Deklarasikan variabel utama di bawah deklarasi *Partial Class...*

```
Dim xData As New DataAccess
Dim xsql As String = _
    IO.File.ReadAllText _
    (Server.MapPath _
    ("~/app_data/select.txt"))
```

3. Selanjutnya, ketikkan prosedur dan fungsi yang dibutuhkan dalam prosedur lainnya yaitu :

- a. Prosedur *refreshGrid* yang digunakan untuk mengubah tampilan satu-satunya Gridview saat berpindah dari mode *summary* ke mode detail, dan juga saat sebuah kategori dipilih dari bagian layer sebelah kiri.

```
Sub refreshGrid _
    (ByVal xParameter As String)
    SqlDataSource1.SelectCommand _
        = xsql & xParameter & _
        " order by adsid desc"
    SqlDataSource1.DataBind()
    If Session("Detail") = _
        False Then
        Session("OldParam") _
            = xParameter
    End If
    Session("SQLGrid") _
        = xParameter
End Sub
```

- b. Prosedur *refreshStatus* yang berfungsi untuk mendeteksi apakah pengguna sedang dalam keadaan login sebagai member atau sedang melakukan browsing sebagai *guest*.

```
Sub refreshStatus()
```



```
Dim xVisible _
    As Boolean = IIf _
        (Session("UserID") _
            <> "", True, False)
lblUser.Text = _
    "Welcome " & _
        Session("UserName")
LinkButton4.Visible _
    = xVisible
lblUser.Visible _
    = xVisible
End Sub
```

- c. Fungsi *MakeNode* yang akan digunakan untuk membuat node atau hirarki tree baru di dalam komponen Treeview.

```
Function MakeNode _
    (ByVal xText As String, _
        ByVal xvalue As String) _
    As TreeNode
Dim xNode As New TreeNode
xNode.Text = xText
xNode.Value = xvalue
Return xNode
End Function
```

- d. Prosedur *KlikDetail* yang nanti akan dieksekusi di dalam Gridview saat pengguna melakukan klik di bagian judul iklan.

```
Sub KlikDetail _
  (ByVal sender As Object, _
   ByVal e As EventArgs)
    Session("Detail") = True
    Dim xSet As _
    Data.DataSet = _
    xData.GetDataset _
    ("select top 1 " & _
     "adsclick from ads " & _
     "where adsid=" & _
     sender.commandargument)
    Dim total As Integer = _
    xSet.Tables(0).Rows(0). _
    Item(0) + 1
    xData.UpdateData _
    ("ads", "adsclick=" & _
     total, "adsid=" & _
     sender.commandargument)
    refreshGrid _
    (" and adsid=" & _
     sender.commandargument)
End Sub
```

- e. Prosedur *BackToAll* yang nanti dieksekusi saat pengguna melakukan klik Linkbutton Back yang terdapat di dalam Gridview. Hal ini dilakukan karena dalam sebuah situs yang menggunakan teknik AJAX, untuk memproses histori browsing terakhir pengguna umumnya tidak bisa dilakukan. Dengan adanya prosedur ini (yang menyimpan histori dalam sebuah Session), diharapkan fitur *Back* dapat tetap ada.

```
Sub BackToAll _  
(ByVal sender As Object, _  
  ByVal e As EventArgs)  
    Session("Detail") = False  
    refreshGrid _  
    (Session("OldParam"))  
End Sub
```

- f. Fungsi yang terakhir yaitu fungsi *Announce* yang berguna untuk menyatakan status dari pengumuman yang ditulis oleh administrator, apakah akan ditampilkan atau tidak ke dalam komponen Repeater.

```
Public Function _  
    Announce() As Boolean
```

```
With xData
    Dim xtemp As _
    Data.DataSet = _
    .GetDataset _
    ("select * " & _
    "from announcement")
    If xtemp.Tables(0). _
    Rows.Count > 0 Then
        Return True
    Else
        Return False
    End If
End With
End Function
```

4. Setelah selesai, kembalilah lagi ke mode Design lalu dobel klik pada komponen Treeview sehingga menuju ke prosedur *SelectedNodeChanged*.

```
Dim xparam As String = _
    " and (Ads.CategoryID = " & _
    TreeView1.SelectedNode.Value & _
    " or Category.CategoryParent = " & _
    TreeView1.SelectedNode.Value & ")"
refreshGrid _
(IIf(TreeView1.SelectedValue = _
    "", "", xparam))
Session("Detail") = False
```

5. Kemudian double klik pada Linkbutton untuk *Search* dan ketikkan listing berikut ini :

```
refreshGrid(" and ads.adstitle " & _  
"like '%" & TextBox1.Text & _  
"%' or ads.adstext like '%" & _  
TextBox1.Text & "%'")
```

6. Selanjutnya double klik pada Linkbutton untuk *Logout* :

```
Session.Clear()  
FormsAuthentication. _  
    SignOut()  
Response.Redirect _  
    ("default.aspx")
```

7. Dan langkah (yang benar-benar) terakhir adalah melakukan double klik pada GridView dan ketikkan listing berikut di prosedur *PageIndexChanged* (arahkan method default ke method tersebut terlebih dulu).

```
refreshGrid _  
(Session("SQLGrid"))
```

ScriptManager - ScriptManager1

ContentPlaceHolder1:Output

- ▶ Root
  - ▶ Parent: 0
    - ◀ Leaf: 1
    - ◀ Leaf: 2
  - ▶ Parent: 1
    - ◀ Leaf: 1
    - ◀ Leaf: 2

Find Ads

[Search](#)

[! Use!]

[Logout](#)

Label

abc

---

abc - testbound - telah didi casberak U ke i  
Kntngnri : mhr.  
abc

abc

---

abc

---

abc - testbound - telah didi casberak U ke i  
Kntngnri : mhr.  
abc

abc

---

abc

---

asp:SqlDataSource#SqlDataSource1

SqlDataSource1: SqlDataSource1

News-Flash :  
abc

News-Flash :  
abc

News-Flash :  
abc

News-Flash :  
abc

News-Flash :  
abc

News-Flash :  
abc

SqlDataSource: SqlDataSource2

Label  
[Login](#) \* [Register](#)

## **Testing dan Pengembangan**

---

---

Setelah selesai melakukan pembuatan situs secara keseluruhan, sebelum melakukan testing terdapat beberapa hal penting yang harus diperhatikan antara lain :

1. Periksalah terlebih dulu apakah semua file-file yang telah dibutuhkan telah selesai dibuat, seperti file-file teks yang berisi query di folder *App\_Data* ataupun file pendukung untuk komponen *Freetextbox*.
2. Periksa juga apakah semua halaman web yang telah dijelaskan telah selesai dibuat.
3. Siapkan rancangan kategori iklan yang akan dibuat sebagai data testing sehingga saat situs dijalankan tidak terlihat kosong
4. Pastikan bahwa saat dilakukan proses *Build* sudah tidak ada error satu pun di setiap halaman web yang sudah dibuat.

Apabila seluruh hal yang sudah disebut telah dipenuhi untuk kepentingan testing, dapat dilakukan langkah berikut ini :

1. Lakukan pengisian data terlebih dulu sebelum menuju ke halaman web utama (*default.aspx*). Terlebih dulu lakukan registrasi member dan cobalah lakukan login ke area member.

**...: Mini Ads ...:**

**...: Registration ...:**

ID	<input type="text"/>	<a href="#">Check ID</a>
Name	<input type="text"/>	
Password	<input type="text"/>	
Confirm Password	<input type="text"/>	
Email	<input type="text"/>	
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>		

copyright © Soetam Rizky 2009  
[Login](#) \* [Register](#)

2. Kemudian lakukan login ke area administrator untuk melakukan input data kategori iklan dan juga sub kategori sesuai selera. Setelah selesai, cobalah pula untuk mengganti setting situs baik dari sisi header, footer maupun title.

**...: Mini Ads ...:**

**Login**

User ID	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="Login"/>	

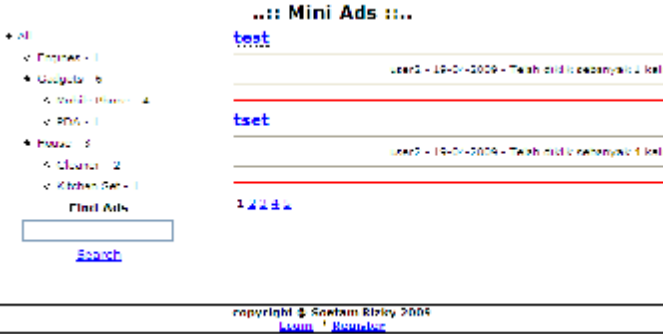
copyright © Soetam Rizky 2009  
[Login](#) \* [Register](#)

3. Sedangkan di area member, dapat dilakukan entri iklan baru. Saat melakukan proses entri ini, usahakan melakukan entri iklan dengan berbagai variasi, baik dari sisi tampilan (dengan memanfaatkan Freetextbox) maupun dari variasi



kategori yang dipilih. Usahakan agar entri iklan ini dilakukan dengan menggunakan member yang berbeda-beda.

4. Selanjutnya dapat dilihat di halaman web utama yang sudah berisi data-data testing



Sedangkan pengembangan yang dapat dilakukan dari contoh situs ini adalah :

1. Teknik AJAX dapat diperluas dengan menerapkannya di halaman web yang belum mengimplementasikan.
2. Teknik penempatan query dalam file teks dapat diganti dengan menempatkan ke dalam stored procedures sehingga dapat lebih terintegrasi ke dalam database.
3. Tampilan dapat diperbaiki dengan menggunakan teknik penempatan Themes sehingga dapat diganti sesuai selera pengguna.

4. Dapat ditambahkan fasilitas RSS sehingga dapat terjadi *newsfeed* bagi para pengguna.
5. Dapat pula ditambahkan fasilitas *Tag Cloud* untuk tiap iklan yang dipasang sehingga bisa didapat iklan mana yang terbanyak dalam situs.

## Appendix : Pemrograman Web

---

Karena buku ini ditujukan untuk semua level, baik pemula maupun mahir. Maka tidaklah adil jika bagi Anda para pemula yang sama sekali belum pernah mengenal pemrograman web tidak diberi sedikit pengantar. Dalam tambahan bab ini, dikutipkan selintas mengenai dasar teori pemrograman web dari buku *Panduan Belajar ASP .NET 2.0* dari penulis yang sama.

Pemrograman aplikasi berbasis web atau yang lazim disebut sebagai pemrograman web memiliki konsep yang berbeda dengan pemrograman aplikasi berbasis desktop. Salah satu hal terpenting yang perlu diperhatikan adalah perbedaan – perbedaan mendasar yang membedakan antara pemrograman web dengan pemrograman desktop, yaitu :

1. Stateless

Dengan arsitektur yang cenderung *bebas*, maka sebuah aplikasi web hampir tidak memiliki batasan bagi para penggunanya. Sebagai contoh adalah penggunaan browser yang mampu melakukan proses multi tab ataupun multi window, akibatnya, tidak ada batasan bagi seorang pengguna melakukan proses yang sama dalam window atau tab browser yang berbeda.

2. Multi platform

Sebuah aplikasi web harus mampu dijalankan dalam sebuah browser di lintas platform atau sistem operasi. Sebuah aplikasi web ASP .NET yang dibangun di sistem operasi Windows, sebagai contoh, juga harus mampu dijalankan di browser sistem operasi lain, misalnya di sebuah distro Linux.

3. Navigation sentris

Dibandingkan dengan aplikasi desktop yang lebih menekankan kepada *step by step sentris*, misalnya penggunaan menu yang telah diatur sedemikian rupa urutannya, sebuah aplikasi web lebih ke arah navigation sentris. Hal ini secara umum akan membebaskan pengguna untuk mengerjakan sebuah proses tanpa harus melalui prosedur semestinya. Sebagai contoh adalah proses logout yang juga dapat dilakukan dengan menutup aplikasi browser.

4. Based on page

Dengan konsep ini, maka sebuah aplikasi web juga mengandalkan nilai estetika selain nilai dari proses yang dilakukan itu sendiri. Berbeda dengan aplikasi desktop yang lebih banyak menafikan desain dari aplikasi itu sendiri, maka aplikasi web juga

bergantung pada keindahan desain demi mendukung proses yang ada didalamnya.

5. Just browser

Secara umum, sebuah aplikasi web hanya membutuhkan sebuah browser tanpa perlu instalasi *runtime* ataupun aplikasi lain. Meski banyak pula aplikasi web yang membutuhkan *plug-ins* tertentu dalam browser, seperti Flash player ataupun media player, tetapi hampir tiap browser yang ada telah memiliki fasilitas yang sama.

6. Less secure

Salah satu hal yang menjadi isu klasik dalam pemrograman web adalah masalah keamanan yang jauh lebih rentan dibanding aplikasi desktop. Terlebih lagi jika aplikasi web tersebut digunakan oleh pengguna secara jamak di internet. Pihak pengguna yang bisa jadi sangat banyak dengan berbagai tujuan, baik ataupun buruk, lebih sulit dideteksi dibandingkan dengan pengguna di aplikasi desktop.

Dari perbedaan mendasar yang telah disebutkan, maka bisa disebutkan beberapa hal yang harus diperhatikan sebelum memutuskan apakah aplikasi yang akan dibuat harus berbasis web atau berbasis desktop, antara lain :

1. Diperlukan aplikasi web jika :
  - a. Informasi yang akan ditampilkan harus bisa diakses semua orang di seluruh dunia
  - b. Infrastruktur pengguna yang tidak terbatas dan memiliki berbagai variasi
  - c. Pengaturan aplikasi seminimal mungkin
  - d. Tidak terlalu banyak memanfaatkan fasilitas yang terdapat dalam sistem operasi
  - e. Keamanan minimal
2. Diperlukan aplikasi desktop jika :
  - a. Pemanfaatan sistem operasi lebih optimal
  - b. Keamanan lebih tinggi tingkatannya
  - c. Informasi yang ditampilkan bersifat eksklusif
  - d. Pengaturan aplikasi jauh lebih rumit
  - e. Infrastruktur beragam

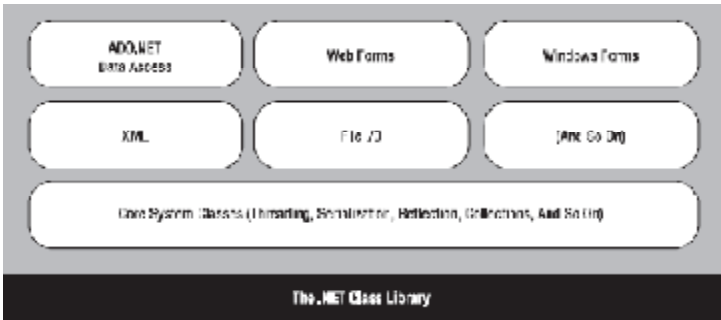
## Appendix : ASP .NET ?

---

Bagi Anda yang benar-benar pemula, tentu saja masih banyak yang bertanya tentang pengantar dari ASP .NET. Seperti selalu dijelaskan di beberapa bab dalam buku ini bahwa untuk lebih memahami konsep dasar dari ASP .NET Anda disarankan untuk membaca buku *Panduan Belajar ASP .NET 2.0* dari penulis yang sama. Karena dalam buku tersebut lebih berorientasi ke pemahaman pembelajaran awal dengan contoh-contoh kecil untuk tiap penjelasan teori.

Meski demikian, demi kemudahan bagi Anda para pembaca yang masih dalam level pemula, maka di bagian ini akan dikutipkan sekilas pengantar teori dari ASP .NET. Berikut ini adalah kutipan pemahaman dasar dari buku *Panduan Belajar ASP .NET 2.0*.

Hingga kini masih banyak developer yang bertanya, apakah ASP .NET merupakan sebuah bahasa pemrograman ? Jawabannya tentu saja bukan ! ASP .NET merupakan bagian dari .NET Framework yang menyediakan class dan tools untuk membuat aplikasi web serta dapat diimplementasikan dengan menggunakan bahasa pemrograman yang mendukung .NET Framework seperti Visual Basic .NET, Visual C#, Visual J# ataupun IronPhyton



Karenanya sangatlah tidak obyektif jika harus membandingkan antara ASP .NET dengan bahasa pemrograman web lainnya seperti PHP, Cold fusion ataupun JSP. Sebab ASP .NET bukanlah sebuah bahasa pemrograman yang berdiri sendiri. Meski demikian implementasi pemrograman web dengan menggunakan ASP .NET memiliki beberapa kelebihan dan (pastinya) kekurangan dibanding bahasa pemrograman web lainnya.



## Appendix : VWD Express vs VS Pro

---

Seluruh contoh yang ada di dalam buku ini telah dites baik dengan menggunakan Visual Web Developer Express Edition 2008 (VWD Express) dan juga dengan menggunakan Visual Studio 2008 Profesional Edition (VS Pro). Umumnya para pemula selalu bertanya apakah perbedaan dari kedua utilitas tersebut, apa untung ruginya dan juga sebaiknya menggunakan yang mana.

Secara teknis Anda bisa melihat perbedaan dari kedua utilitas tersebut di situs resmi ASP .NET yaitu [www.asp.net](http://www.asp.net) atau di situs Microsoft. Tetapi secara umum, perbedaan utama dari kedua utilitas tersebut adalah harganya. Seperti jamak diketahui, bahwa VWD Express merupakan utiliti yang benar-benar gratis sedangkan VS Pro merupakan utiliti yang berbayar.

Tentu saja, utilitas yang gratis selalu memiliki banyak kelemahan dan kekurangan didalamnya. Tetapi bukan berarti bahwa yang gratis tidak bisa dimanfaatkan secara optimal dalam pembuatan situs.

Salah satu kelemahan mencolok dari VWD Express adalah tiadanya fasilitas untuk melakukan *Publish Web Site* yang mampu melakukan kompilas

situs ke dalam bentuk `.dll` sehingga lebih aman dari pencurian listing program. Karenanya dalam buku ini sengaja ditiadakan bab mengenai proses deploy agar para pembaca yang menggunakan VWD Express tidak mengalami kebingungan mencari fitur tersebut.

Bagi Anda para pemula, pelajar ataupun mahasiswa memang sangat disarankan untuk memilih VWD Express yang gratis, berukuran lebih kecil dan juga lebih ringan proses eksekusinya. Tetapi bagi Anda yang memang telah profesional dalam mengerjakan situs, memang sangat disarankan untuk menggunakan VS Pro.

## Appendix : Konsep Dasar

---

Bagi Anda para pemula, memang sangat disarankan untuk mempelajari konsep dasar lebih lanjut di buku *Panduan Belajar ASP .NET 2.0* dari penulis yang sama. Meski demikian di bab ini akan dibahas secara sekilas beberapa konsep dasar, terutama yang berkaitan dengan contoh-contoh di dalam buku ini.

Beberapa konsep dasar tersebut adalah :

### 1. Folder khusus

Dalam ASP .NET terdapat beberapa folder khusus yang secara default tidak akan terlihat dalam browser oleh pengguna. Folder-folder ini memiliki fungsi yang masing-masing telah dibagi sendiri berdasarkan namanya. Beberapa folder khusus yang digunakan dalam contoh situs di buku ini antara lain :

#### a. Folder App\_Data

Merupakan folder yang secara default menampung file database baik dari Microsoft Access, SQL Server ataupun file teks. Tetapi perlu diingat, bahwa pada saat Anda melakukan hosting ke webhosting, database bisa jadi tidak disimpan dalam folder ini,

tetapi dalam tempat khusus yang telah disediakan oleh webhosting.

b. Folder App\_Code

Merupakan folder default untuk menyimpan file-file class ataupun fungsi Web Service. Tetapi jika situs telah melalui proses *publish* dan kompilasi, maka isi dari folder ini akan menjadi file-file berekstensi *.dll* di dalam folder *bin*.

c. Folder Bin

Merupakan folder yang didalamnya menyimpan file referensi berekstensi *.dll* atau file hasil kompilasi. File referensi umumnya berasal dari komponen non standard, seperti Freetextbox atau FCKEditor.

2. File web.config

Merupakan nyawa dari sebuah situs, karena didalamnya terdapat konfigurasi penting yang dibaca saat pertama kali situs ditampilkan di browser. Dalam contoh-contoh di buku ini, file web.config sangat banyak dimanfaatkan sekaligus dimanipulasi terutama untuk setting yang bersifat singular atau tunggal sehingga akan mubazir jika disimpan dalam tabel di sebuah database.

Secara default, sebuah situs akan memiliki sebuah file `web.config`. Tetapi beberapa situs yang terbilang kompleks dapat memiliki lebih dari satu file `web.config`.

### 3. Master Page

Banyak yang berpendapat bahwa keberadaan Master Page sama halnya dengan tag HTML *Iframe* yang mampu membagi tampilan situs secara konsisten. Memang pendapat ini tidak sepenuhnya salah, karena salah satu fungsi dari Master Page memang hampir sama dengan tag HTML *Iframe*. Tetapi Master Page sendiri sesungguhnya bukan hanya untuk sekedar menyeragamkan tampilan sebuah situs, tetapi didalamnya juga dapat berfungsi untuk memodifikasi situs secara keseluruhan.

### 4. Sitemap

Merupakan file format XML yang dapat digunakan untuk menentukan navigasi situs. Secara sederhana, sitemap merupakan file yang akan menjadi menu dalam sebuah situs. Tetapi sebenarnya juga dapat dimanfaatkan untuk kepentingan *levelling* atau membatasi hak akses pengguna dalam sebuah situs. Seperti telah dijelaskan di bab sebelumnya dalam buku ini,

bahwa sebuah situs secara default hanya memiliki sebuah sitemap, tetapi tetap dapat dimodifikasi sehingga satu situs dapat memiliki lebih dari satu sitemap.

## Appendix : Komponen Penting

---

Bagi Anda para pemula, terdapat beberapa pengetahuan dasar yang sangat penting diketahui saat atau sebelum memulai pembuatan contoh situs yang ada di buku ini. Salah satu pengetahuan penting tersebut adalah memahami sekilas tentang komponen-komponen penting yang dalam ASP .NET, terlebih untuk yang digunakan dalam contoh di buku ini.



---

Untuk penjelasan lebih dalam mengenai contoh dan konsep dari komponen-komponen yang ada dalam ASP .NET, Anda dapat membaca buku *Panduan Belajar ASP .NET 2.0* dari penulis yang sama.

---

Komponen-komponen tersebut antara lain :

1. Label **A** Label

Komponen ini umumnya hanya digunakan sebagai keterangan atau teks penanda biasa dari sebuah situs. Jika Anda masih rancu untuk memilih menggunakan teks biasa (dari HTML) atau menggunakan Label, maka dapat disimpulkan bahwa penggunaan Label umumnya digunakan

untuk teks yang isinya membutuhkan manipulasi tertentu pada saat situs ditampilkan.

2. Textbox  TextBox

Merupakan komponen yang secara default digunakan untuk menerima inputan dari pengguna. Beberapa property penting dari komponen ini adalah property *Text* untuk hasil inputan dan property *TextMode* yang digunakan untuk menentukan apakah Textbox menerima inputan dalam bentuk biasa, bentuk password (dengan masking karakter atau seperti halnya isian password) atau bentuk *Multiline* (mampu menampung teks yang sangat banyak isinya).

3. Button  Button

Merupakan komponen yang nantinya akan dapat diklik oleh pengguna dalam sebuah situs. Salah perbedaan utama dari Button dan Linkbutton adalah di dalam Button dapat menerima penekanan tombol Enter dari pengguna jika property *UseSubmitBehavior* diberi nilai *True*, meski kedua komponen tersebut memiliki perilaku yang hampir sama.

4. Linkbutton  LinkButton

Memiliki perilaku yang sama dengan Button tetapi memiliki property *PostBackURL* yang dapat diisi



langsung tujuan halaman web saat pengguna mengeksekusi Linkbutton.

5. Dropdownlist  DropDownList

Jika sebelumnya Anda akrab dengan pemrograman visual berbasis desktop, maka komponen ini sama dengan komponen Combobox di pemrograman berbasis desktop. Fungsi utamanya untuk memilih sebuah item dari kumpulan item, baik dari sebuah array, item yang dientrikan manual pada property *Items* ataupun dari tabel di sebuah database.

6. Checkbox  CheckBox

Merupakan komponen yang hanya menerima dua jenis nilai yaitu *True* (yaitu saat tanda centang aktif) dan *False*. Umumnya digunakan untuk menerima manipulasi data dengan tipe *Boolean*.

7. Multiview  MultiView dan View  View

Merupakan dua komponen yang saling terkait. Sebuah View tidak dapat berdiri sendiri tanpa Multiview dan sebaliknya. Sebuah View merupakan *anak* dari Multiview, sehingga sebuah Multiview dapat terdiri dari banyak View sekaligus. Penempatan View sebenarnya mirip dengan pembagian halaman web ke dalam layer yang terpisah, tetapi pada saat eksekusi halaman web

tersebut pengguna tidak akan merasakan bahwa bagian yang terpisah sesungguhnya berada dalam satu halaman web yang sama.

Property terpenting dalam Multiview adalah *ActiveViewIndex* yang menyatakan View nomor berapa yang akan ditampilkan pada saat tertentu.

8. Calendar  Calendar

Merupakan komponen dasar dalam ASP .NET untuk menerima inputan berupa tanggal atau bertipe *Date*. Selain menerima inputan, juga dapat menampilkan tanggal dari sebuah variabel atau record di dalam tabel. Property terpenting dari komponen ini adalah *SelectedDate* yang akan mengembalikan data tanggal yang telah diklik atau dipilih oleh pengguna. Satu hal terpenting yang harus diingat dari komponen ini adalah bahwa di tiap kali pengguna melakukan pemilihan tanggal maka akan menyebabkan halaman web mengalami proses *PostBack* ke server (dengan kata lain halaman web akan mengalami proses refresh). Sehingga akan sangat merugikan jika situs memang telah dihost online sedangkan koneksi dari pengguna tergolong *pas-pasan*. Karenanya disarankan untuk menerapkan teknik AJAX saat menggunakan komponen ini.

9. Panel  Panel

Merupakan komponen yang biasanya digunakan sebagai *tempat penampungan sementara* dari kumpulan komponen lain yang akan dimanipulasi secara serentak. Misalnya sebuah kumpulan komponen akan diset property *Visible*-nya menjadi *False* secara serentak setelah pengguna mengeksekusi sebuah Button, maka kumpulan komponen tersebut dapat dimasukkan ke dalam sebuah Panel.

Dalam penerapan teknik AJAX, khususnya dengan menggunakan komponen dari AJAX Control Toolki, penggunaan Panel sangat dibutuhkan di beberapa fituranya. Sebagai contoh adalah penggunaan komponen ModalPopUp dan CollapsiblePanel yang sangat bergantung pada penggunaan Panel.

10. Literal  Literal

Memiliki fungsi yang hampir sama dengan Panel, tetapi isi dari Literal bukanlah komponen melainkan teks atau tag HTML yang diumpanbalik dari listing program.

11. Menu  Menu

Merupakan menu navigasi yang secara default mengambil struktur menu situs dari file *web.sitemap*. Secara default pula pada saat

dieksekusi komponen ini akan menghasilkan script javascript sehingga dapat langsung menampilkan efek visual (seperti sliding) pada saat diklik oleh pengguna di dalam browser.

#### 12. Treeview TreeView

Di dalam ASP .NET komponen ini seringkali dijadikan alternatif dari komponen Menu. Tetapi satu kelebihan dari komponen Treeview adalah mampu menampilkan menu dalam struktur hirarkis, bukan hanya dari file Sitemap tetapi juga dari tabel maupun dari file XML biasa. Tetapi kelemahan dari komponen ini tidak mampu tampil dalam bentuk Vertical seperti halnya komponen Menu.

#### 13. FileUpload

Komponen ini memiliki fungsi penting dalam sebuah situs yaitu untuk menerima inputan file dari pengguna agar bisa ditempatkan di folder yang telah ditentukan di dalam web server. Secara default, batas maksimal ukuran file yang bisa diupload adalah 4 Mb, meski nantinya dapat dimanipulasi ulang di dalam file web.config.

#### 14. Gridview GridView

Merupakan *nyawa* terpenting dari sebuah situs yang menggunakan akses database. Komponen yang dulunya bernama Datagrid di ASP .NET 1.x ini

dikenal sangat *sakti* karena fleksibilitas pengaturan serta fitur yang dimiliki. Tetapi di sisi lain komponen ini merupakan komponen yang paling kompleks pengaturannya dalam sebuah situs. Satu hal penting yang penting untuk diperhatikan saat modifikasi adalah mengganti tipe field menjadi *Template Field* akan menyebabkan field tersebut dapat secara bebas disisipi oleh tag HTML maupun script (baik server script maupun client script) sehingga programmer dapat lebih leluasa mengatur tampilan. Dan juga jangan terpaku dengan tampilan wizard awal yang dihasilkan oleh GridView, karena komponen yang satu ini sesungguhnya memiliki kekuatan yang menjadi kelebihan utama ASP .NET dibanding bahasa pemrograman web lainnya.

Dengan melakukan modifikasi pada template yang telah tersedia, sebenarnya komponen GridView mampu melakukan koneksi dengan model tabel majemuk atau lebih dari satu tabel dalam satu GridView. Hal ini bisa dilakukan dengan berbagai macam variasi teknik implementasi, seperti memodifikasi perintah query dari komponen SqlDataSource yang dijadikan koneksi dasar dari GridView atau juga dengan menempatkan komponen lain yang menjadi *binding* ke tabel lain.

Teknik-teknik tersebut juga dijelaskan di contoh-contoh situs yang ada dalam buku ini.

15. Repeater  Repeater

Merupakan komponen untuk akses database yang memiliki template paling sederhana. Tetapi dari kesederhanaan tersebut, malah membuat banyak programmer pemula menghindari penggunaan komponen ini. Karena untuk menggunakan komponen ini, seorang programmer harus mampu menguasai tag HTML dengan baik dan juga mengetahui secara pasti pengaturan field mana yang akan ditampilkan melalui server script. Satu ciri pokok dari komponen ini adalah pada saat sebuah kumpulan record ditampilkan dalam Repeater, maka di tiap pemisah record tidak akan melakukan proses ganti baris.

16. Formview  FormView

Komponen ini sangat banyak digunakan dalam buku ini karena fungsi uniknya yang sangat cocok dipakai sebagai form manipulasi data, khususnya di proses inputan dan edit data. Banyak programmer pemula yang terpaku bahwa sebuah Formview hanya dapat melakukan koneksi dengan satu tabel. Sehingga saat dibutuhkan entri sebuah tabel yang mengacu ke tabel yang lain, mereka menghindari dari

penggunaan Formview karena dianggap komponen ini tidak mampu melakukannya. Tetapi di dalam buku ini ditunjukkan bahwa sebenarnya Formview dapat mengacu ke banyak tabel, dan bahkan dalam sebuah Formview dapat terdapat source dari tabel yang lain. Dan semua fitur tersebut dapat dilakukan dengan menggunakan teknik drag-drop yang sederhana.

Hal penting yang perlu diingat bahwa dalam form terdapat beberapa template yang digunakan sesuai fungsinya yaitu *InsertItemTemplate* (untuk inputan data baru), *EditItemTemplate* ( untuk modifikasi data) dan *ReadOnly* (yang sesungguhnya memiliki fungsi lebih lengkap yaitu untuk inputan data baru, proses edit dan juga proses hapus data). Dalam contoh yang terdapat dalam buku ini, untuk sebuah situs tidak seluruh template tersebut akan digunakan. Sehingga Anda tidak perlu terpancang pada fungsi yang ditawarkan oleh Formview.

Hal penting lain adalah bahwa di tiap template tersebut, terdapat dua jenis method yang akan dieksekusi yaitu yang berakhiran *ing* (dalam bahasa Inggris, akhiran *ing* berarti *present tense* atau sedang dilakukan) dan yang berakhiran *ed* (dalam bahasa Inggris hal ini menunjukkan sudah

dilakukan atau *past tense*). Untuk yang berakhiran *ing* seperti *ItemInserting* atau *ItemUpdating* berarti apapun yang ada didalamnya akan dieksekusi saat data sedang dimanipulasi. Akibatnya bisa disisipkan parameter-parameter yang dibutuhkan untuk perintah query yang sedang dieksekusi. Sedangkan dalam method berakhiran *ed* seperti *ItemUpdated* atau *ItemInserted* akan dijalankan saat proses query telah dianggap selesai. Sehingga didalamnya lebih layak disisipkan perintah-perintah yang mengembalikan kondisi halaman web ke posisi sebelum proses modifikasi data dilakukan.

#### 17. `SqlDataSource`

Merupakan komponen yang berfungsi sebagai jembatan antara koneksi database dengan halaman web. Secara default, dengan menggunakan fasilitas wizard drag-drop tabel dari jendela Database Explorer akan mampu menempatkan komponen ini lengkap dengan query dan juga koneksi yang bersesuaian di dalam file web.config. Sesuai dengan namanya, komponen ini hanya berlaku untuk database dari SQL Server, sedangkan untuk database dari vendor lain (seperti MySql atau Oracle), Anda harus mendapatkan provider yang sesuai dari vendor yang bersangkutan. Meski



demikian, dalam buku ini juga ditunjukkan cara koneksi database tanpa harus bergantung kepada komponen yang satu ini.

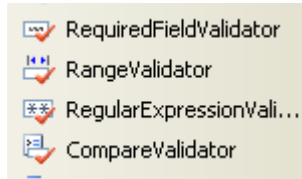
Property penting yang biasanya dimodifikasi dalam komponen ini adalah property yang berhubungan dengan perintah query seperti *SelectQuery*, *InsertQuery*, *UpdateQuery* dan *DeleteQuery*. Modifikasi dari property yang berisi perintah query dapat dilakukan dengan menggunakan kotak dialog yang telah disediakan, baik dalam Visual Web Developer Express maupun dalam Visual Studio. Satu hal menarik dari kotak dialog tersebut adalah fasilitas *query builder* yang sangat membantu bagi para pemula dalam membangun query yang bebas dari kesalahan ketik. Bahkan bagi para programmer mahir, fitur tersebut sangat membantu untuk membangun query yang terbilang cukup kompleks, karena perintah query yang dibangun dapat langsung dites hasilnya.

#### 18. SiteMapDataSource SiteMapDataSource

Fungsi utama dari komponen ini adalah sebagai jembatan antara file sitemap dengan komponen yang akan dijadikan navigasi situs seperti Menu atau Treeview. Tidak banyak property yang harus diset dalam komponen ini, umumnya hanya

property *ShowStartingNode* yang diset menjadi *False* jika memang file sitemap yang menjadi referensi tidak memiliki *root map* didalamnya.

## 19. Validator



Komponen jenis validator merupakan komponen yang secara otomatis menghasilkan javascript saat situs ditampilkan di browser. Javascript yang ada merupakan fungsi untuk validasi inputan data dari komponen tertentu (yang paling banyak digunakan adalah Textbox), sehingga programmer tidak perlu lagi membuat fungsi validasi secara terpisah untuk melakukan pengecekan input dari pengguna.

## Appendix : Akses Data

---

Buku ini memang secara keseluruhan hanya menampilkan contoh situs dengan menggunakan akses data file teks dan SQL Server 2005 Express Edition. Meski begitu, tidak banyak konsep database yang ditawarkan dalam buku ini, baik dari segi konsep database dan tabel, relasi tabel maupun perintah query yang digunakan.

Untuk mempelajari mengenai konsep database, disarankan agar Anda membaca buku-buku yang membahas mengenai basis data. Sedangkan untuk perintah query secara mendalam dapat membaca buku *Panduan Belajar SQL Server 2005 Express Edition* dari penulis yang sama.

Tetapi jika dalam pengerjaan sebuah situs Anda diharuskan menggunakan database selain SQL Server seperti MySQL atau Oracle, maka disarankan agar Anda terlebih dulu melakukan download provider yang bersesuaian langsung dari vendor database yang digunakan.

Sedangkan untuk perintah query, secara mendasar tidak terlalu banyak perbedaan antara database dari satu vendor dengan vendor yang lain. Hal

ini dikarenakan perintah SQL secara umum adalah standard baku internasional.

## Appendix : ASP .NET AJAX

---

Dalam buku ini, beberapa contoh situs benar-benar memanfaatkan teknik AJAX secara optimal. Meski demikian tidak semua pemanfaatan AJAX, khususnya dengan menggunakan ASP .NET AJAX. Sebenarnya pembahasan mengenai ASP .NET AJAX secara lebih mendalam dan konseptual terdapat di dalam buku dengan judul *AJAX Dengan ASP .NET 2.0* dari penulis yang sama. Tetapi (sekali lagi) demi kemudahan para pembaca, maka di sub bab ini akan dicuplikkan sekilas mengenai dasar konsep dari ASP .NET AJAX dari buku tersebut.

AJAX (Asynchronous Javascript and XML) pertama kali dicetuskan pada 18 Februari 2005 oleh Jesse James Garret dari Adaptive Path dalam artikel berjudul "AJAX, A new approach to web applications". Artikel tersebut menyebutkan bahwa untuk lebih menarik perhatian pengunjung situs dibutuhkan teknik baru yang dapat mengatasi keterbatasan aplikasi web, sehingga kemampuan aplikasi desktop dapat dimasukkan ke dalam aplikasi web. Meski tergolong baru, teknologi AJAX saat ini mengalami booming yang sangat menakjubkan di dunia internet, sehingga seakan

menjadi standar baru bagi semua situs web baik komersial ataupun non komersial.

AJAX sendiri lebih dikategorikan sebagai sebuah teknik pemrograman tanpa memandang bahasa pemrograman yang mengimplementasikannya. Banyak teori dan pendapat tentang definisi sesungguhnya dari AJAX, tetapi dari sekian banyak definisi tersebut, dapat disimpulkan bahwa AJAX dapat terdiri dari beberapa komponen penting antara lain :

1. Presentasi web yang menggunakan HTML dan CSS
2. Data yang ditransportasikan dalam bentuk XML dan adanya penggunaan JSON (Javascript Object Notation)
3. Adanya proses XMLHttpRequest
4. Penggunaan Javascript
5. Pada dasarnya pengembangan sebuah aplikasi web yang menggunakan teknik AJAX harus memiliki konsep sebagai berikut :
6. Hanya melakukan update terhadap bagian halaman web yang mengalami proses transfer data
7. Bersifat asynchronous sehingga proses dapat dijalankan secara majemuk dalam waktu yang bersamaan

8. Memiliki solusi yang dapat memecahkan masalah yang umumnya dihadapi di aplikasi desktop, misal : drag and drop
9. Lebih sedikit terjadi proses postback

Dari teknologi yang digunakan tersebut, AJAX dapat diimplementasikan dengan menggunakan berbagai macam bahasa pemrograman web seperti PHP, JSP, Coldfusion ataupun seluruh bahasa pemrograman yang mendukung ASP .NET seperti Visual Basic .NET atau Visual C#. Dengan adanya beragam AJAX Framework (baik yang gratis ataupun komersil), maka programmer web tidak lagi harus menguasai secara expert tentang XML, Javascript dan bahasa pemrograman yang digunakan untuk dapat mengaplikasi teknik AJAX.

AJAX sendiri telah mengalami perkembangan yang sangat pesat, sehingga kian banyak situs besar yang telah mengaplikasikannya seperti Google, Yahoo dan Microsoft sendiri. Tentu saja dengan teknik baru tersebut, banyak hal yang makin bisa diimplementasikan dalam sebuah situs, bahkan mulai banyak aplikasi desktop yang diusung sebagai aplikasi web, misal : aplikasi perkantoran (word processing, spreadsheet, presentation) sejenis Microsoft Office, aplikasi pengolah image sejenis Photoshop dan masih

banyak lagi. Meski memiliki berbagai kelebihan yang membuka sudut pandang baru terhadap pemrograman web, tetapi AJAX sendiri memiliki berbagai kelemahan, diantaranya :

1. Kesulitan dalam mengimplementasikan Javascript terutama dalam hal debugging
2. Tidak seluruh browser support AJAX
3. Masih sulitnya diimplementasikan di multi device
4. Sulitnya menangani implementasi jika pengunjung situs melakukan penekanan tombol Back di dalam browser
5. Proses loading awal yang membutuhkan waktu lebih lama dibanding situs biasa.

Karenanya, tidak semua solusi situs sesuai dengan penerapan teknik AJAX. Meski demikian, saat ini hampir semua situs "modern" berlomba - lomba mengaplikasikan teknik AJAX dengan berbagai macam tema baru. Situs - situs tersebut umumnya dimasukkan ke kategori Web 2.0 yang kini menjadi jargon marketing baru bagi situs yang mengimplementasikan teknik AJAX.

Di lingkup AJAX Framework yang semakin lama semakin menjamur di dunia maya, perlu diperhatikan beberapa hal dalam pemilihannya :



1. Lebih baik memilih framework yang dikhususkan untuk bahasa pemrograman yang digunakan, misal: untuk ASP .NET lebih baik menggunakan framework yang memang dikhususkan untuk ASP .NET 2.0 seperti ASP .NET AJAX, Anthem .NET, Magic AJAX atau AJAX pro.
2. Pilih framework sesuai kebutuhan, terutama jika framework tersebut berbasis Javascript murni seperti scriptaculous atau Dojo. Karena terkadang efek visual atau teknik lain yang ditawarkan tidak dibutuhkan semuanya dalam situs yang akan dibuat.
3. Pilih framework (jika bisa) yang terkecil ukurannya, atau dalam ukuran yang bisa ditoleransi, sehingga akses terhadap situs tersebut tidak berat.
4. Jika memang harus menggabungkan antara satu framework dengan framework yang lain, perhatikan terlebih dulu, apakah kedua framework tersebut tidak menimbulkan masalah di kemudian hari.
5. Dan jika memang bisa mendapatkan framework yang gratis dengan kualitas tinggi, tidak usah berusaha mencari framework yang komersial !!!

Secara umum, meski AJAX saat ini menjadi sebuah tren yang tidak mungkin terelakkan di dunia pemrograman web, perlu diperhatikan apakah memang

AJAX dibutuhkan dalam situs tersebut atau tidak. Karena penggunaan AJAX sebenarnya lebih ditekankan terhadap penghematan traffic internet serta kemudahan dan pengalaman baru bagi pengunjung situs, bukan sekedar mengikuti tren atau agar situs terlihat lebih "modern".

ASP .NET AJAX merupakan nama resmi dari framework AJAX dari Microsoft yang sebelumnya memiliki nama ATLAS. Pemilihan framework ini dalam buku dengan alasan utama bahwa framework ini memang dikhususkan untuk ASP .NET 2.0 dan juga telah dilengkapi berbagai komponen tambahan yang bersifat open source dan terpaket dalam AJAX toolkit (dapat didownload di [www.codeplex.com](http://www.codeplex.com) dan akan dibahas pada bab tersendiri). Selain itu, ASP .NET AJAX tidak menyertakan file runtime, melainkan terinstalasi dalam GAC (Global Assembly Cache) sehingga tidak terjadi pembengkakan ukuran file hasil pemrograman. Dan juga, ASP .NET AJAX juga tersedia untuk bahasa pemrograman web lain seperti PHP dan JSP di Microsoft AJAX Library.

Meski demikian, tidak berarti bahwa ASP .NET AJAX merupakan framework AJAX yang terbaik dan satu-satunya untuk ASP .NET 2.0. Karena masih banyak framework lain yang dapat dipilih sesuai

kebutuhan dan juga sama – sama gratis seperti Anthem .NET, Magic AJAX atau AJAX Pro.

Terdapat tiga komponen terpenting dalam ASP .NET AJAX yaitu :

1. Script Manager

Tiap halaman web yang akan ditempati oleh teknik AJAX (perlu diingat, bahwa dalam satu situs, tidak semua halaman web ditempati oleh teknik AJAX), harus ditempati oleh komponen ini. Karena dengan adanya komponen ini, proses update parsial dalam sebuah halaman web dapat dilakukan. Dalam satu halaman web, cukup terdapat satu komponen Script Manager.

2. Update Panel

Jika komponen Script Manager merupakan otak dari implementasi teknik AJAX, maka dapat diibaratkan bahwa Update Panel adalah “badan” yang berfungsi untuk mengeksekusi dari teknik AJAX itu sendiri. Tiap komponen yang akan diberi sifat AJAX dan masuk ke proses update parsial harus “dibalut” dengan komponen Update Panel. Dalam sebuah halaman web dapat terdapat lebih dari satu Update Panel. Tiap Update Panel juga dapat mempengaruhi Update Panel yang lain dengan melakukan setting di sub tag Trigger. Secara default, Update Panel hanya

akan melakukan update dengan kondisi tertentu (conditional), tetapi dapat juga diberikan setting agar melakukan update tanpa kondisi, jika komponen yang terletak dalam Update Panel tersebut secara otomatis terpengaruh dengan proses yang terjadi dalam satu halaman web.

### 3. Update Progress

Tiap gambar atau tulisan yang terjadi dalam komponen ini akan ditampilkan jika terjadi proses update parsial dalam halaman web yang mengimplementasikan teknik AJAX. Komponen ini umumnya dibutuhkan untuk penanda bagi pengunjung situs agar tahu bahwa sedang terjadi proses dalam halaman web. Karena dalam sebuah halaman web yang mengimplementasikan AJAX, progress bar browser umumnya tidak menunjukkan aktifitas, sehingga seringkali pengunjung situs berasumsi bahwa koneksi internet tidak berjalan.