

DIKTAT
PEMROGRAMAN
WEB

LEARNING BY SAMPLE

AJAX

CONTROL TOOLKIT



SOETAM RIZKY WICAKSONO

Learning By Sample - AJAX Control Toolkit

Daftar Isi

AJAX	1
Apa itu AJAX ?	2
Persiapan Awal	6
Sekilas ASP .NET	9
AJAX Control Toolkit.....	10
Apa yang dibutuhkan ?.....	11
Bagian Pertama.....	20
Sebelum memulai	21
Numeric UpDown dan Color Picker	22
Rating	35
Collapsible Panel dan Drop Shadow	47
AsyncFile Upload dan Modal Popup	48
Bagian Kedua	49
Sebelum Memulai	50
Accordion dan HTML Editor.....	51
ComboBox dan Confirm Button.....	67
Studi Kasus II : Mini Blog	68
Pengantar	69
Pembuatan Database	71
Master Page.....	75
Halaman Admin	83
Maintenance Data Kategori.....	86
Maintenance Data Blog	91
Menu Halaman Web Admin	101
Halaman Default	103
Bagian Newest Blog.....	106
Bagian Archieve Blog.....	112
Appendix	123
Appendix : Komponen Dasar ASP .NET	124

AJAX

Apa itu AJAX ?

Apakah yang terbayang saat Anda mendengar kata AJAX ? Sebuah klub sepakbola ? Atau sebuah teknik pemrograman baru yang terkesan *high class* dan rumit ? Bagi Anda yang berkecimpung di dalam dunia pemrograman, khususnya di pemrograman web, tentu jawaban kedua yang dipilih.

AJAX atau Asynchronous Javascript and XML merupakan sebuah teknik pemrograman baru tetapi *lama* di dunia pemrograman web. AJAX sendiri pertama kali dicetuskan oleh Jesse James Garret dari Adaptive Path pada bulan Februari 2005 dan lebih mengarah kepada sebuah cara pendekatan para web master ke pihak pelanggan, sehingga pelanggan merasa memiliki *pengalaman baru* dalam berinternet ria.

AJAX sendiri saat ini menjadi booming tersendiri di kalangan para programmer, baik para profesional programmer, kalangan mahasiswa maupun para hobiis web. Implementasi dari AJAX sendiri juga sangat beragam, baik dengan menggunakan PHP, ASP .NET 2.0, JSP, Ruby serta framework independen lainnya yang berjumlah puluhan dan dapat ditemukan dengan mudah di dunia maya.

Penggunaan AJAX secara sederhana lebih mengarah kepada refresh halaman web secara parsial sehingga pengunjung situs tidak merasa meninggalkan situs secara utuh saat terjadi proses *postback* atau submit sebuah data ke web server. Dengan teknik yang sama pula, maka saat ini banyak situs baru yang memanfaatkan teknik AJAX menjadi sebuah situs yang mengarah ke konsep Web 2.0 dengan berbagai fitur yang membuat *wow effect* bagi para pengunjungnya.

Terlepas dari segala jenis teori dan praktek tentang AJAX, masih dapat dimaklumi bahwa masih sedikit sekali programmer web yang berminat ataupun sudah berminat tapi ragu untuk mempelajari teknik baru ini. Beberapa programmer (mungkin juga termasuk Anda) merasa ragu untuk mempelajarinya karena secara konseptual, pembelajaran AJAX membutuhkan kemampuan tingkat intermediate untuk Javascript serta XML itu sendiri. Belum lagi minimnya buku berbahasa Indonesia yang membahas tentang hal ini. Beberapa yang lain

mempelajari AJAX hanya dari sisi visual effect yang dianggap *wow* dengan menggunakan framework yang telah banyak tersedia di internet.

Tapi apakah benar belajar AJAX itu sulit ? Jawabannya mungkin saja ya, mungkin juga tidak. Dan yang pasti, dengan mencoba mempelajari dan (wajib) mempraktekkan apa yang ada dalam buku ini, nantinya Anda akan menjawab tidak. Karena dengan menggunakan bahasa yang *awam* dan teknik sederhana, diharapkan setelah Anda *menghabiskan* buku ini, Anda tidak lagi fobia terhadap kata AJAX.

Dengan konsep *step by step*, teknik AJAX dalam buku ini akan diimplementasikan dengan menggunakan ASP .NET. Mengapa ASP .NET? Dengan banyak keunggulan dan kemudahan (baca buku “Panduan Belajar ASP .NET 2.0” oleh penulis yang sama) serta tool developer gratisnya yaitu Visual Web Developer Express Edition, maka programmer web tidak lagi risau dengan masalah lisensi serta lebih cepat mengimplementasikan sebuah situs.

Tetapi, seperti yang telah jamak diketahui, bahwa kebanyakan para praktisi *malas* jika diharuskan terlebih dulu membaca teori yang sifatnya bertele-tele dan hingga akhir sebuah pembelajaran buku, hanya mendapatkan keterangan tentang sintaks tertentu. Selain itu, kebanyakan buku yang bersifat **syntax oriented** lebih digunakan sebagai bahan referensi dibandingkan sebagai sebuah sarana membuat aplikasi sederhana.

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

Pada dasarnya pengembangan sebuah aplikasi web yang menggunakan teknik AJAX harus memiliki konsep sebagai berikut :

1. Hanya melakukan update terhadap bagian halaman web yang mengalami proses transfer data
2. Bersifat asynchronous sehingga proses dapat dijalankan secara majemuk dalam waktu yang bersamaan
3. Memiliki solusi yang dapat memecahkan masalah yang umumnya dihadapi di aplikasi desktop, misal : drag and drop
4. 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 2.0 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

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 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”.

Persiapan Awal

Buku ini akan sangat cepat diaplikasikan untuk para programmer web yang sebelumnya telah mengenal ASP .NET dengan menggunakan Visual Studio .NET. Tapi juga tidak menutup kemungkinan bagi para programmer web yang sebelumnya telah familiar dengan bahasa pemrograman web lainnya seperti PHP ataupun ASP klasik.

Bagaimana dengan para pemula ? Apa yang harus diketahui sebelum mengaplikasikan buku ini ? Berikut adalah daftar pengetahuan yang seharusnya Anda ketahui terlebih dulu :

1. Logika pemrograman

Jika Anda sama sekali belum pernah melakukan aktifitas pemrograman, disarankan untuk terlebih dulu memahami logika pemrograman dasar, seperti percabangan dan perulangan. Meski dalam studi kasus, hal tersebut tidak disebutkan secara signifikan.

2. Pengetahuan tentang konsep pemrograman web

Jika sebelumnya Anda telah melakukan aktifitas pemrograman, tetapi untuk aplikasi jenis desktop, seperti Visual Basic 6.0, C++ dan yang lain, maka disarankan agar Anda (minimal) mengetahui tentang konsep dari sebuah aplikasi basis web. Didalamnya Anda wajib memahami tentang *sitemap*, konsep navigasi link dan jenis tampilan serta jenis browser dalam sebuah aplikasi basis web.

3. HTML dan CSS

Pengetahuan tentang HTML saat ini seringkali dilupakan para programmer web dikarenakan banyaknya tool desain yang telah mempermudah implementasi dari HTML. Bahkan beberapa tool seperti halnya Dreamweaver, telah mampu menyulap para pemula yang tidak memiliki kemampuan HTML sama sekali, menjadi seorang desainer web yang mumpuni. Tetapi di dalam proses pemrograman web dengan menggunakan ASP .NET , pengetahuan tentang HTML tetap diperlukan karena dalam banyak olah tampilan halaman

web, Anda tetap memerlukan pengetahuan HTML untuk merombak sebuah komponen menjadi lebih *eye catching*.

Sedangkan untuk CSS (Cascading Style Sheet) lebih diperlukan sebagai alat bantu dalam melakukan desain halaman web yang lebih fleksibel dan seragam. Dengan bantuan CSS, seluruh halaman web dapat didesain ataupun dimodifikasi dengan sangat mudah dalam satu langkah saja. Di studi kasus buku ini, desain dilakukan dengan menggunakan CSS secara total. Bagi Anda yang sama sekali buta dengan CSS, tidak perlu khawatir, karena Visual Web Developer Express (VWD Express) telah menyediakan alat bantu, jika Anda ingin melakukan modifikasi desain.

4. Konsep .NET framework

Pengetahuan tentang konsep .NET framework sebenarnya merupakan komponen terpenting sebelum Anda memulai mengerjakan studi kasus di dalam buku ini. Tetapi, secara naluriah, hampir semua praktisi ataupun pemula, seringkali mengabaikan bagian buku yang mengenalkan tentang sebuah konsep. Konsep .NET framework sendiri, secara umum telah dimuat dalam MSDN Library, baik versi online (msdn.microsoft.com) ataupun versi offline. Disarankan untuk tetap membaca (sedikit banyak) tentang konsep .NET framework, karena dengan konsep yang matang, Anda akan mampu mengembangkan studi kasus ini.

5. Bahasa Visual Basic

Dalam studi kasus ini, digunakan ASP .NET dengan menggunakan bahasa pemrograman Visual Basic .NET. ASP .NET sendiri merupakan sebuah CLR yang dapat dibangun dengan beberapa bahasa pemrograman di lingkup .NET framework, seperti Visual Basic .NET, Visual C# dan Visual J#. Jika Anda sebelumnya telah mengenal ASP klasik, ataupun Visual Basic 6.0, terlebih lagi jika Anda telah menggunakan ASP .NET 1.x sebelumnya, Anda tidak akan mengalami kesulitan untuk memahami studi kasus di buku ini. Tetapi jika Anda sebelumnya tidak mengenal Visual Basic, tidak perlu khawatir, karena listing program yang terdapat dalam studi kasus ini, tidak akan mencapai 50 baris listing program, sehingga dipastikan Anda juga tidak akan

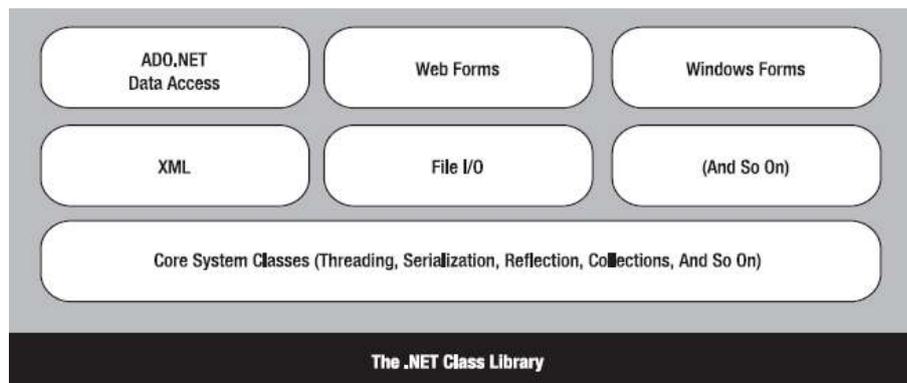
lama dalam beradaptasi dengan Visual Basic .NET. Dan juga tidak menutup kemungkinan, jika Anda ingin melakukan konversi ke dalam bahasa pemrograman lain di lingkup .NET framework, misal Visual C# (dibaca : Visual C Sharp)

6. Javascript

Meski Javascript yang digunakan dalam studi kasus bukanlah termasuk level intermediate, tetapi tetap disarankan untuk mempelajari konsep dari Javascript secara utuh. Dan disarankan pula untuk lebih mempelajari Javascript sebagai sarana manipulasi dari halaman web, bukan hanya memandang Javascript sebagai alat bantu visual effect dalam sebuah situs.

Sekilas ASP .NET

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



Gambar 1. Arsitektur .NET

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.

AJAX Control Toolkit

AJAX Control Toolkit merupakan sekumpulan komponen gratis (dan open source) yang dikeluarkan secara resmi oleh Microsoft untuk membantu para programmer web dalam melakukan pemrograman berbasis web dengan menggunakan teknik AJAX.

Kumpulan komponen ini dapat secara gratis didownload di situs ajax.asp.net. Dan yang perlu diperhatikan bahwa kumpulan komponen ini akan terus berkembang, karena selalu didonasi oleh banyak programmer yang secara sengaja membuat komponen-komponen baru di dalam AJAX Control Toolkit.

Salah satu keunggulan utama dari AJAX Control Toolkit adalah kemudahan penggunaan serta adanya dokumentasi dan contoh yang dapat dilihat di situs ajax.asp.net. Akibatnya, penggunaan AJAX Control Toolkit jauh lebih mudah dilakukan, dibandingkan dengan komponen-komponen lain (baik yang gratis maupun komersil) untuk AJAX.

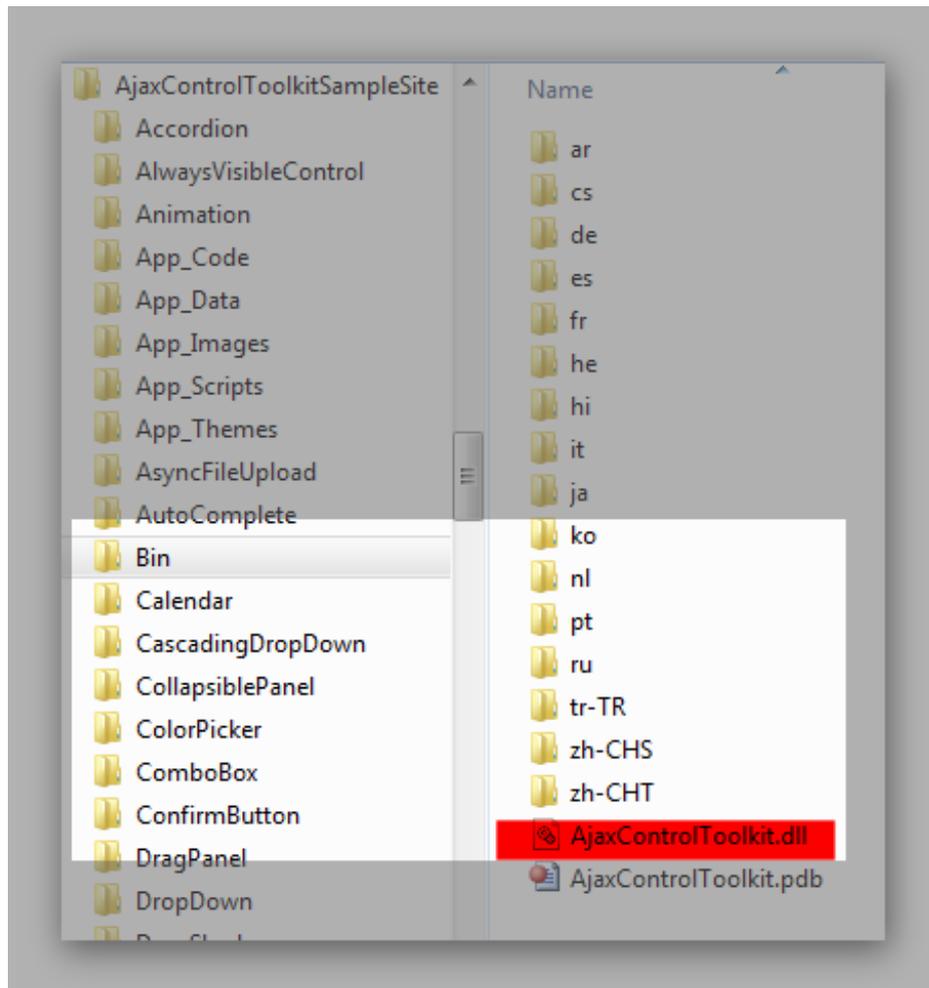
Dalam bab ini, akan dibahas beberapa contoh kecil mengenai penggunaan AJAX Control Toolkit, yang tentu saja contoh-contoh yang akan ditampilkan berbeda dengan contoh yang telah ada di dokumentasi resmi AJAX Control Toolkit. Beberapa perbedaan yang sangat mendasar dalam implementasi contoh di buku ini adalah pemakaian database hampir diseluruh contoh mengenai AJAX Control Toolkit.

Hal tersebut memang sengaja dilakukan, pasalnya di contoh yang resmi mengenai AJAX Control Toolkit kebanyakan hanya membahas mengenai penggunaan komponen dengan menggunakan variabel, sehingga banyak programmer (khususnya pemula) yang merasa kesulitan untuk mengimplementasikan AJAX Control Toolkit dengan menggunakan database.

Apa yang dibutuhkan ?

Beberapa hal yang sangat penting untuk diperhatikan sebelum memulai pengerjaan contoh-contoh yang ada di dalam buku ini adalah :

1. Pastikan terlebih dulu bahwa Anda telah memiliki AJAX Control Toolkit terlebih dulu
 - a. Tentu saja, syarat pertama ini harus didahului dengan melakukan proses download paket AJAX Control Toolkit dari situs <http://www.codeplex.com>
 - b. Dari paket file yang telah didapat, ekstrak ke dalam folder khusus. Kemudian perhatikan folder *bin* yang ada didalam rangkaian paket tersebut, khususnya untuk file *AJAXControlToolkit.dll*. Sebab file inilah yang nanti akan kita copy ke dalam seluruh Solution dari tiap contoh yang ada di dalam buku ini.

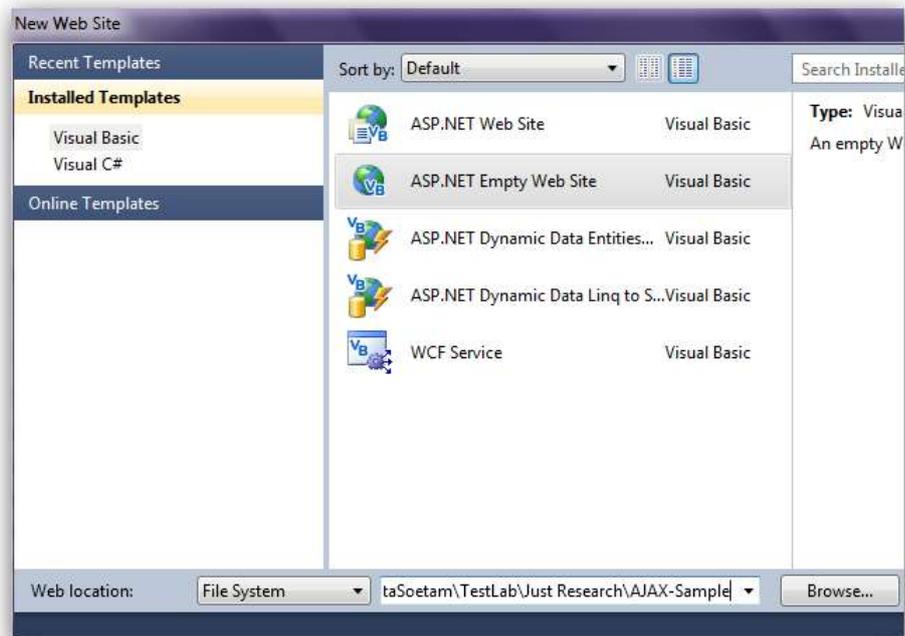


Gambar 2. Instalasi AJAX Control Toolkit

2. Komponen AJAX Control Toolkit telah tersedia di dalam Toolbox.
 - a. Untuk menempatkan AJAX Control Toolkit ke Toolbox, yang pertama kali harus dilakukan adalah membuat sebuah web site di dalam lingkup Visual Web Developer Express atau Visual Studio terlebih dulu.

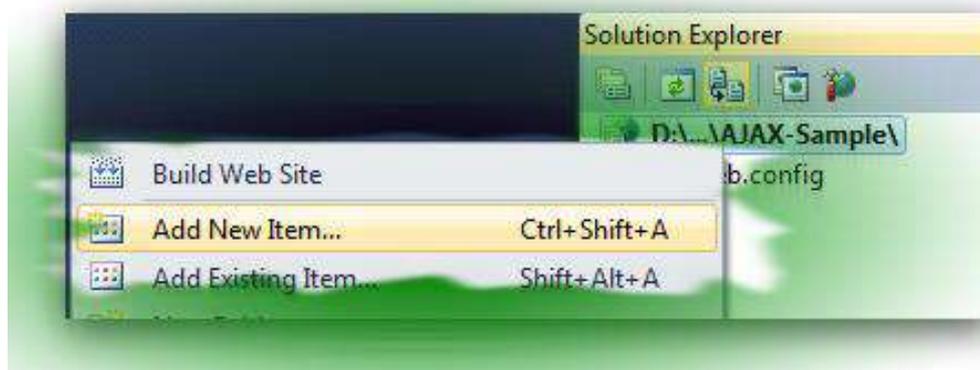


Seluruh contoh yang ada di dalam buku ini menggunakan Visual Web Developer 2010 Express Edition, namun demikian Anda dapat juga menggunakan Visual Studio 2010 versi apapun.

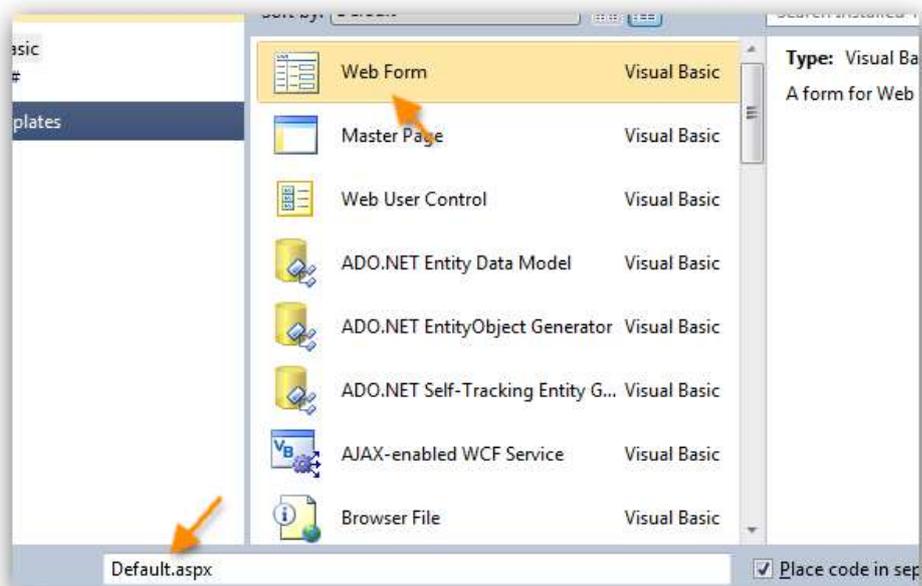


Gambar 3. Kotak Dialog Pembuatan Web Baru

- b. Selanjutnya, buat sebuah web page kosong (langkah ini bisa dilewati jika Anda membuat web site dengan tipe *ASP .NET Website*, sebab didalamnya telah terdapat web pages yang siap untuk diisi).

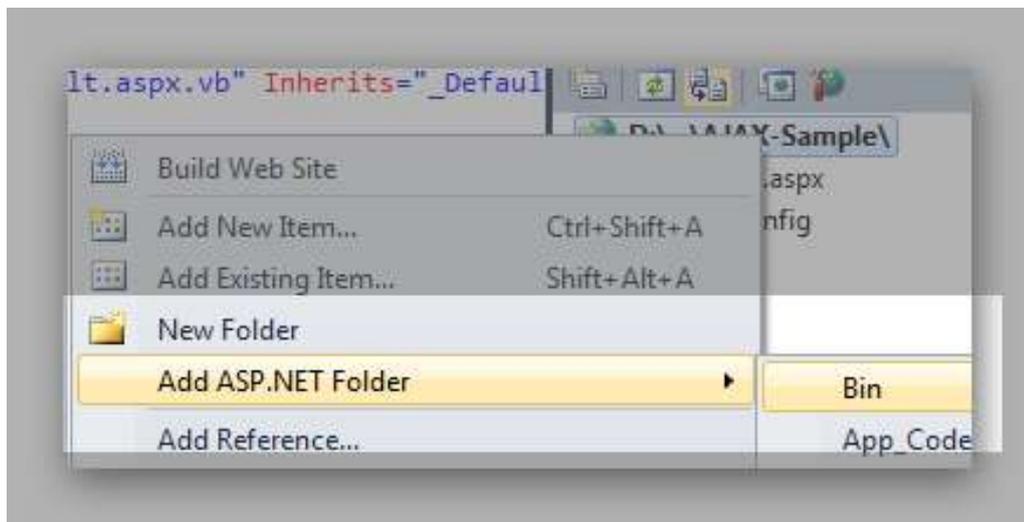


Gambar 4. Penambahan Item Baru



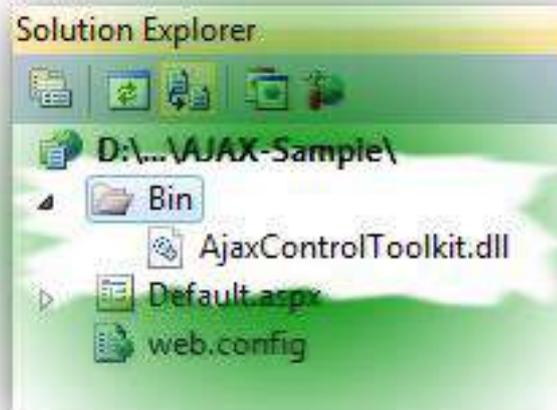
Gambar 5. Kota Dialog Penamaan Halaman Web

- c. Buat folder *bin* di dalam Solution Explorer dengan menggunakan proses klik kanan pada Solution Explorer dan pilih *Add New Folder* lalu buat folder yang dimaksud.



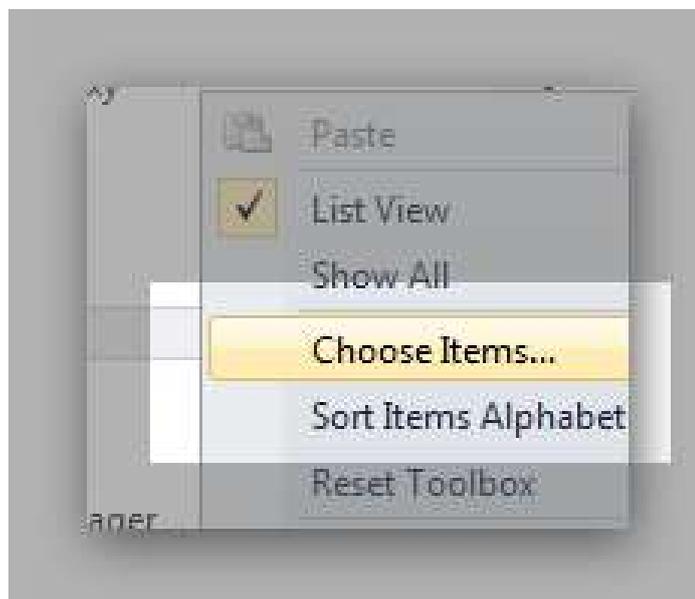
Gambar 6. Penambahan Folder Bin

- d. Kemudian copy file *AJAXControlToolkit.dll* ke dalam folder *bin* pada Solution Explorer.



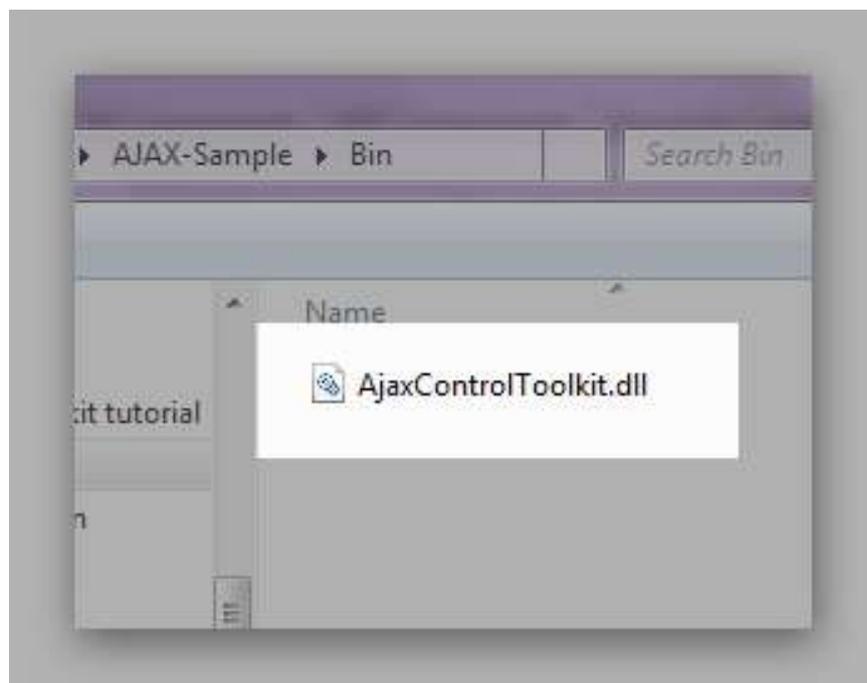
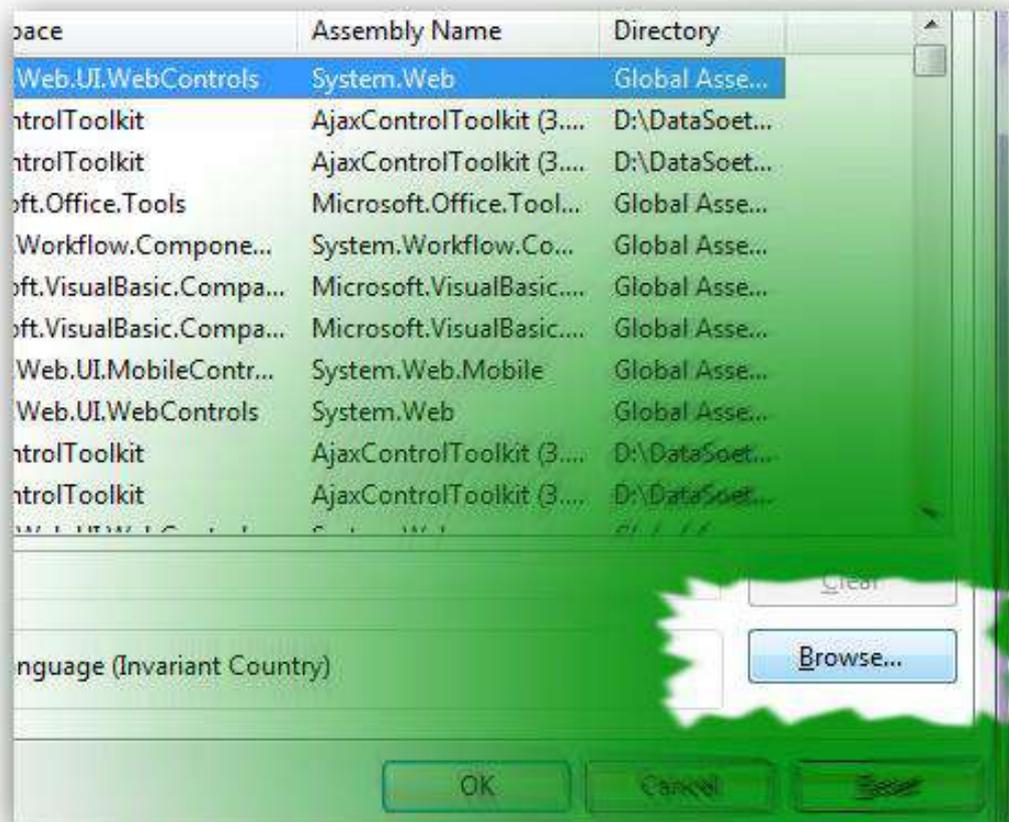
Gambar 7. Hasil Penambahan Folder Bin

- e. Lalu beralihlah ke Toolbox, dan klik kanan pada bagian yang terbawah (umumnya merupakan tab General), dan pilih sub menu *Choose Items*



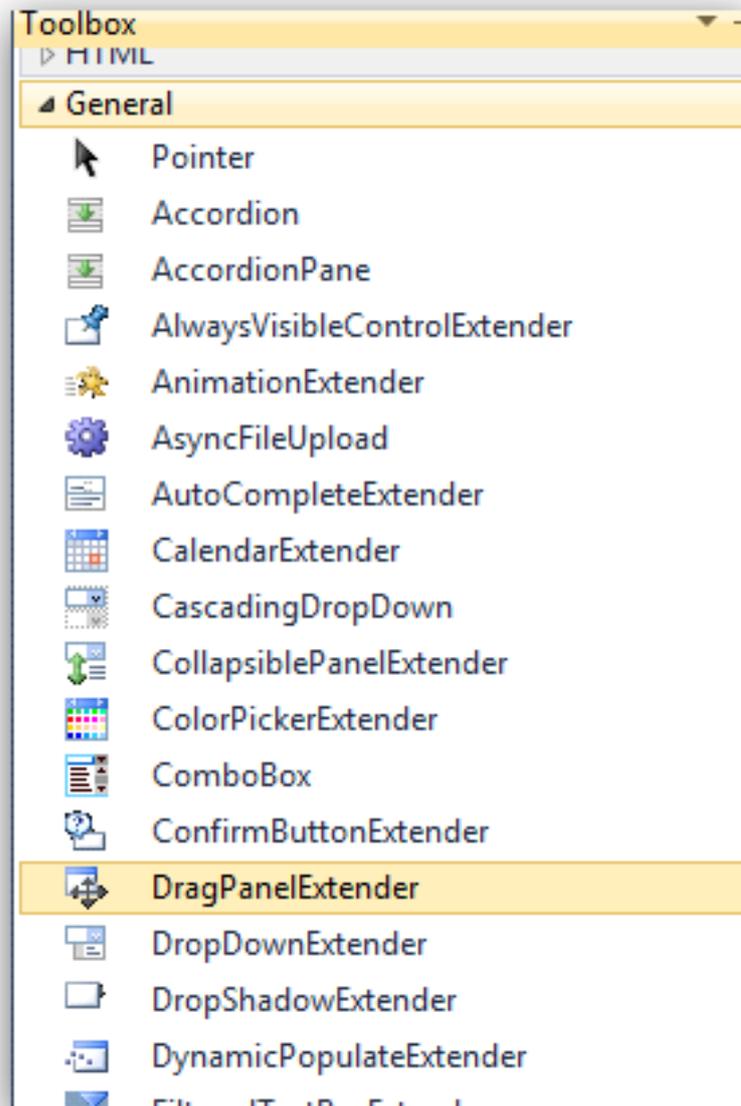
Gambar 8. Pemilihan Item Untuk Toolbox

- f. Pada kotak dialog yang tersedia, klik tombol *Browse* untuk memilih file *.dll*. Pada kasus ini, pilihlah file *AJAXControlToolkit.dll* yang terletak pada folder website yang telah didefinisikan sebelumnya (pastikan bahwa file tersebut telah ada di dalam folder *bin*).



Gambar 9. Hasil Penambahan

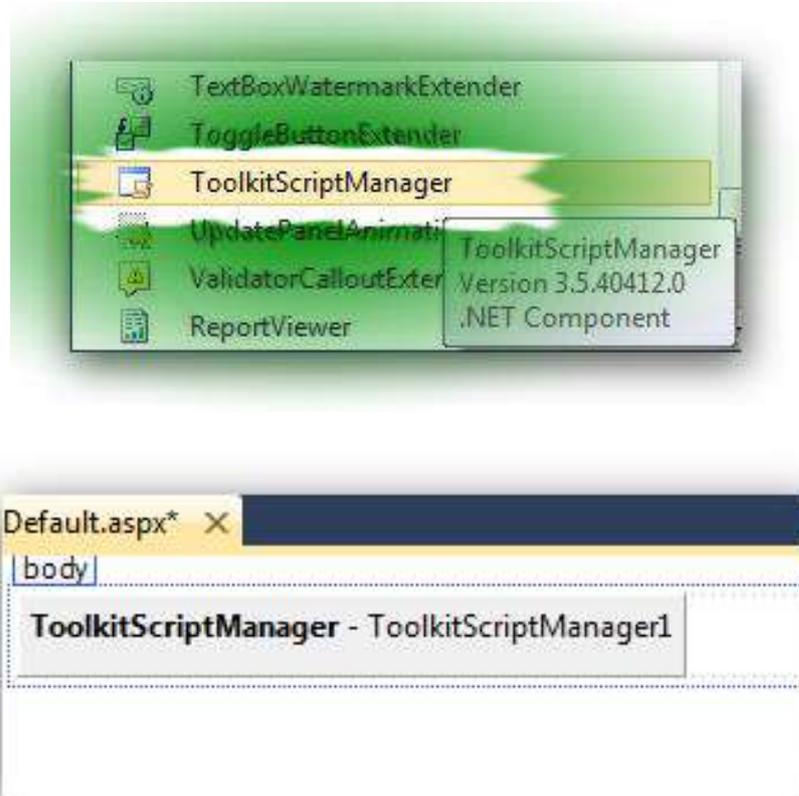
- g. Kini seluruh komponen telah tersedia di dalam Toolbox (perhatikan bahwa proses ini pada saat awal biasanya memakan waktu sedikit lebih lama).



Gambar 10. Hasil Toolbox

3. Tempatkan komponen *Toolkit Script Manager* ke dalam web page yang akan ditempati oleh AJAX Control Toolkit. Pastikan bahwa komponen ini terletak di bagian paling atas dari web page yang akan kita olah. Cara penempatan

dapat dilakukan dengan menggunakan cara *drag and drop* atau lebih praktis dengan melakukan *double click* pada komponen tersebut.



Gambar 11. Komponen ToolkitScriptManager

4. Sekarang Anda telah siap untuk mengerjakan contoh-contoh yang ada di bab berikutnya. Selamat berkarya !!!

Bagian Pertama

Sebelum memulai

Pada bagian pertama ini, seluruh contoh yang ditampilkan hanya melakukan proses dengan menggunakan AJAX Control Toolkit untuk bagian situs yang sederhana dan tanpa melibatkan pengolahan database didalamnya.

Sehingga bagi Anda yang merasa bukan pemula, dapat langsung melewati bagian ini dan langsung menuju ke bagian kedua yang didalamnya melibatkan proses pengolahan database. Sebab pada contoh ataupun tutorial yang banyak beredar di internet, lebih banyak yang tidak melibatkan database didalamnya.

Dan sebelum Anda memulai, diharuskan untuk terlebih dulu mengikuti langkah pada bagian pendahuluan, yakni saat memasukkan file referensi atau *.dll* dari AJAX Control Toolkit ke dalam Solution serta menempatkan komponen AJAX Control Toolkit ke dalam toolbox hingga penempatan Toolscript ke bagian paling atas dari web page. Penting pula diingat bahwa langkah tersebut harus selalu dilakukan setiap akan memulai pembuatan contoh-contoh di bab berikutnya.

Numeric UpDown dan Color Picker

Contoh pertama yang dibuat adalah simulasi sederhana yang didalamnya melibatkan dua komponen dari AJAX Control Toolkit yakni Numeric UpDown dan Color Picker. Numeric UpDown merupakan komponen yang merupakan turunan dari Textbox dan berfungsi sebagai wahana inputan data yang bertipe numerik. Komponen ini juga merupakan kloning dari komponen yang bernama sama untuk pemrograman berbasis desktop.

Sedangkan Color Picker mirip dengan komponen Color Dialog di dalam pemrograman berbasis desktop dan juga memiliki fungsi yang sama yakni melakukan pemilihan warna yang kemudian diaplikasikan ke sebuah proses yang lain. Perlu diingat bahwa dalam .NET Framework, Color merupakan sebuah class yang dapat dimanipulasi dengan berbagai cara, baik dengan mengambil nilai RGB (Red Green Blue) ataupun melalui kode heksadesimal yang didapat dari pemilihan warna.



Untuk lebih jelas mengenai kode warna baik dari mode RGB maupun heksa desimal, dapat dilihat melalui beberapa aplikasi pengolah citra digital seperti GIMP atau Adobe Photoshop.

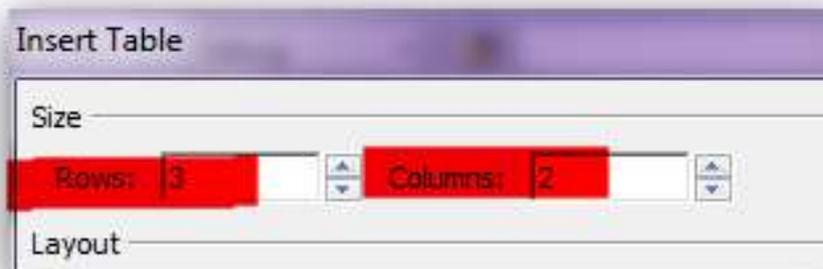
Kini saatnya untuk membuat situs sederhana yang akan mensimulasikan kedua komponen tersebut. Pertama, pastikan bahwa langkah-langkah yang ada di dalam bagian pendahuluan telah dilakukan, hingga komponen ToolkitScriptManager telah tersedia di dalam web page.

Berikutnya, sisipkan sebuah tabel ke dalam web page yang tersedia dengan memilih menu *Insert* → *New Table*. Tabel ini nantinya hanya berfungsi untuk merapikan tampilan yang akan dibuat selanjutnya.



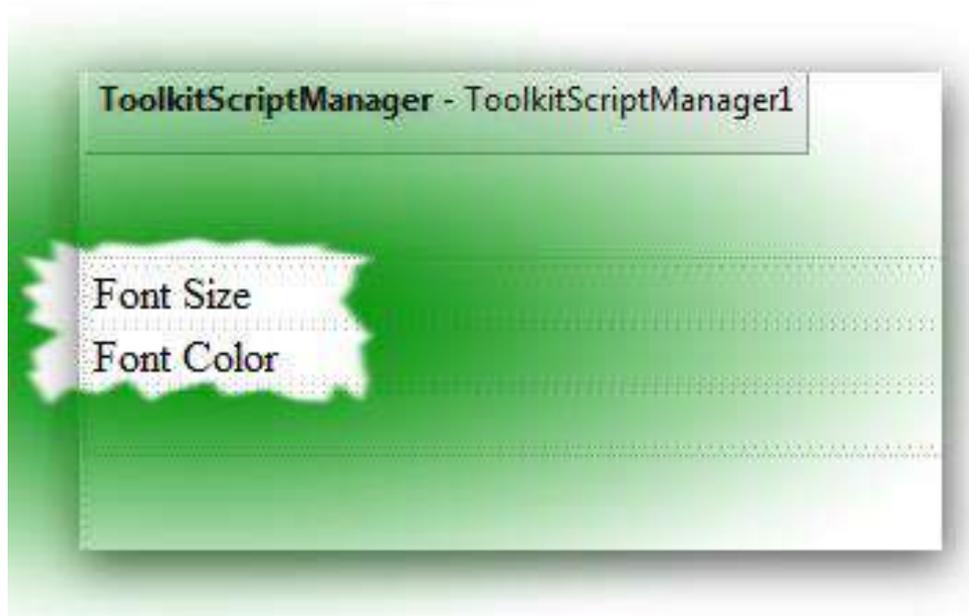
Gambar 12. Menu Insert Table

Pada kotak dialog yang tersedia, buat sebuah tabel dengan jumlah baris sebanyak tiga dan jumlah kolom sebanyak dua. Hal tersebut dilakukan dengan melakukan pengisian pada inputan *Rows* dan *Column*.



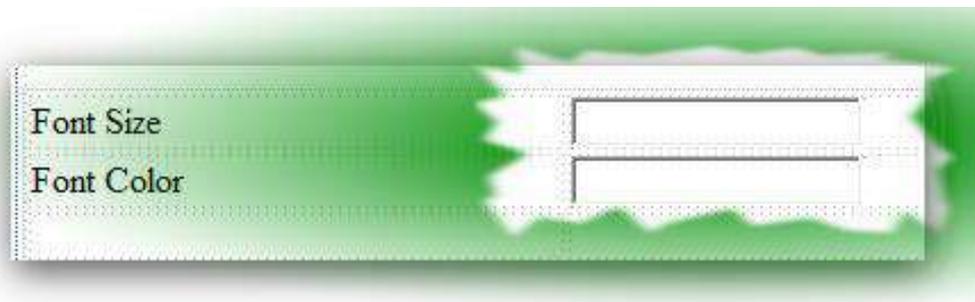
Gambar 13. Dimensi Tabel

Sekarang saatnya untuk mengisi beberapa teks ke dalam tabel yang sudah dibuat. Pada baris pertama kolom pertama, isikan teks *Font Size*, dan pada baris kedua di kolom yang sama isikan teks *Font Color*. Kedua teks tersebut nantinya hanya sebagai keterangan untuk kedua komponen yang akan kita tempatkan di langkah berikutnya.



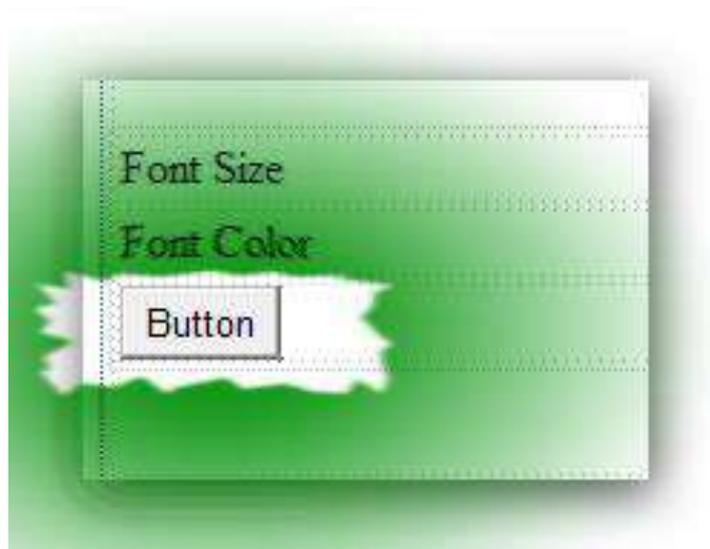
Gambar 14. Keterangan Pertama

Selanjutnya tempatkan dua buah komponen Textbox tepat di kolom kedua di masing-masing baris yang sebelumnya telah berisi teks untuk keterangan.



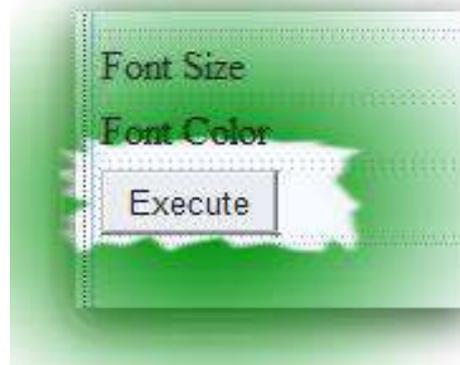
Gambar 15. Penempatan Textbox

Kemudian tempatkan sebuah Button di baris terakhir pada kolom pertama. Button ini nantinya akan berfungsi sebagai media eksekusi hasil pemilihan kedua komponen dari AJAX Control Toolkit.



Gambar 16. Penempatan Button

Setelah selesai, ganti property *Text* dari Button tersebut menjadi *Execute*.

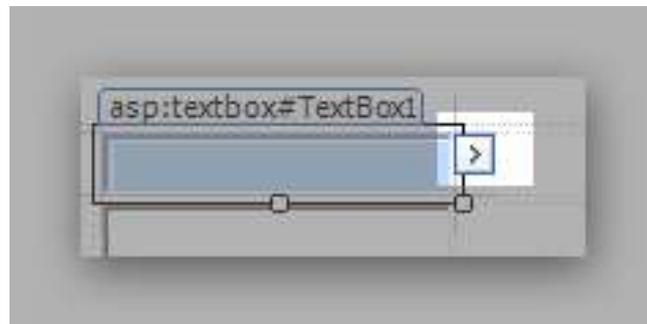


Gambar 17. Property Button

Sekarang saatnya untuk menempatkan AJAX Control Toolkit ke dalam kedua Textbox yang telah ditempatkan sebelumnya. Klik pada *Smart Tag* di Textbox pertama, dan pilih menu *Add Extender*.



Smart Tag merupakan menu tambahan yang terdapat di tiap komponen yang ditempatkan pada web page. Umumnya berupa panah kecil yang mengarah ke kanan.

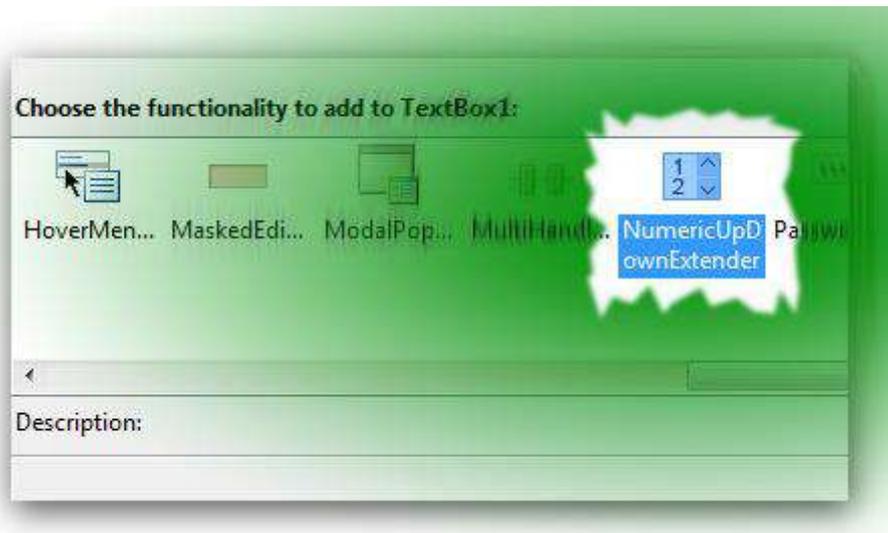


Gambar 18. Penambahan Extender di Textbox

Pada kotak dialog Extender yang tampil, pilih komponen *Numeric UpDown*. Kemudian klik tombol OK untuk kembali ke dalam tampilan mode edit web page.

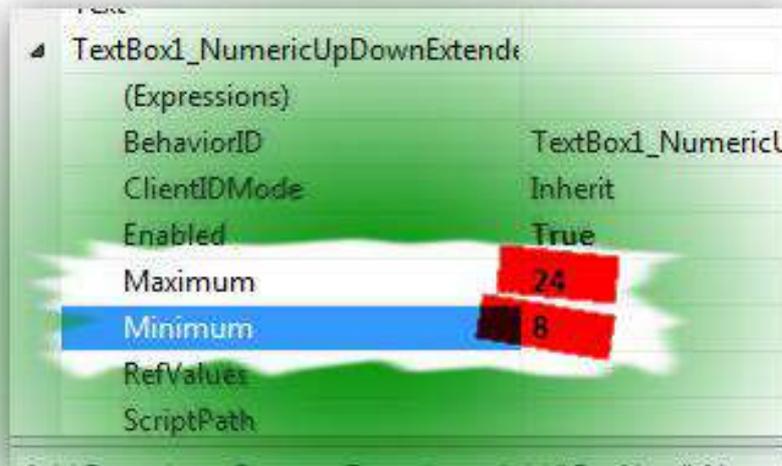


AJAX Control Toolkit telah terintegrasi ke dalam IDE Visual Web Developer Express Edition maupun Visual Studio, sehingga jauh lebih mudah diimplementasikan ke dalam web page yang kita buat.



Gambar 19. Kotak Dialog Extender untuk NumericUpDown

Saat Numeric UpDown telah “ditempelkan” ke dalam Textbox yang pertama, maka di dalam property Textbox tersebut akan ditambahkan property dari Numeric UpDown sebagai bagian dari Textbox tersebut. Kini lihat ke Property Window, dan amati pada bagian *Textbox1_NumericUpDown*. Lalu set pada property *Maximum* dan *Minimum* untuk membatasi inputan pada Textbox tersebut. Pada contoh ini, nilai minimum yang diperbolehkan adalah 8 dan maksimum sebesar 24. Nilai tersebut nantinya menjadi batas dari ukuran font dari teks yang akan ditampilkan pada proses berikutnya. Anda dapat pula berkreasi dengan ukuran nilai minimum dan maksimum yang lain sesuai dengan batasan yang ditentukan sendiri.



Gambar 20. Property NumericUpDown

Kemudian set pula property *Width* dari Numeric UpDown sebesar 100 pixel. Besar dari lebar komponen tersebut dapat juga Anda ganti sesuai dengan kebutuhan.



Gambar 21. Property Width

Sekarang kita beralih ke Textbox yang kedua, dan lakukan cara yang sama dengan Textbox pertama yakni klik pada Smart Tag yang tersedia lalu pilih *Add Extender*. Lalu pada kotak dialog Extender tersebut, pilih komponen Color Picker dan klik OK untuk kembali ke mode edit pada web page.

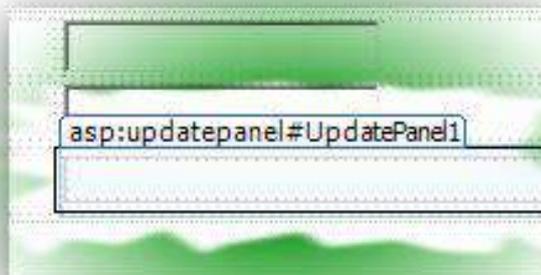
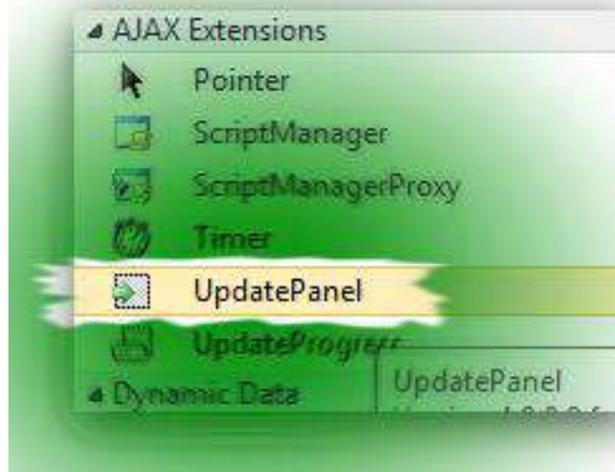


Gambar 22. Kotak Dialog Color Picker Extender

Kini pada tabel di dalam web page, pilih pada baris terakhir di kolom kedua. Pada sel tersebut, tempatkan komponen Update Panel dari tab AJAX Extension di dalam Toolbox.

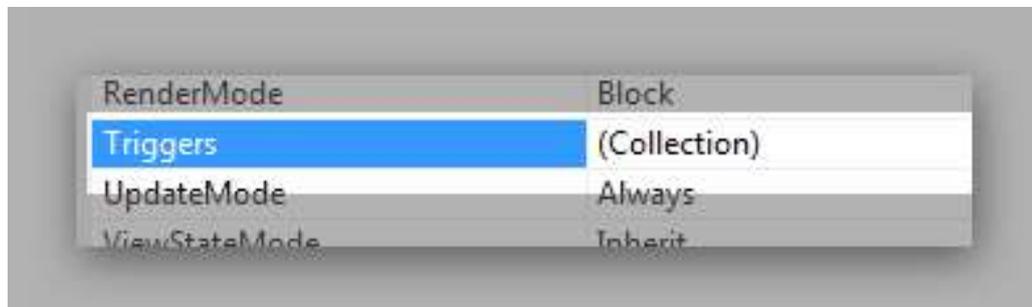


Komponen Update Panel merupakan wahana untuk melakukan proses *partial postback* di dalam penggunaan AJAX pada ASP .NET. Untuk lebih jelas memahami komponen ini dapat membaca buku berjudul *AJAX dengan ASP .NET* dari pengarang yang sama.



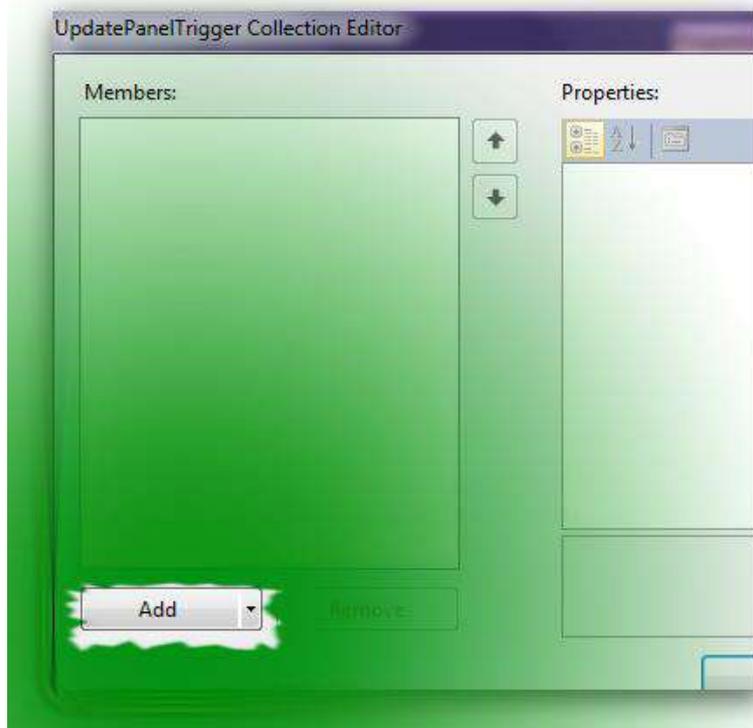
Gambar 23. Penambahan Update Panel

Di dalam Update Panel yang baru saja ditempatkan, beralihlah ke property *Triggers* untuk menambahkan proses yang akan ditempatkan pada komponen tersebut.

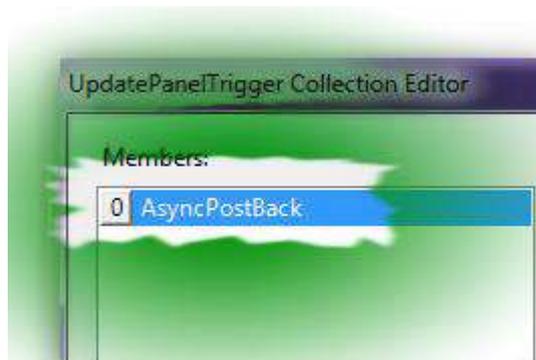


Gambar 24. Property Penambahan Trigger

Kemudian klik pada tombol kecil yang tersedia pada property tersebut, dan perhatikan pada kotak dialog yang muncul sesudahnya.



Di dalam kotak dialog tersebut klik tombol *Add* untuk menambahkan proses *Asynchronous Postback* ke dalam Update Panel tersebut. Proses tersebut nantinya akan ditampilkan tanpa harus menyebabkan web page mengalami *full postback* atau refresh secara keseluruhan, melainkan hanya dilakukan refresh di dalam Update Panel tersebut.

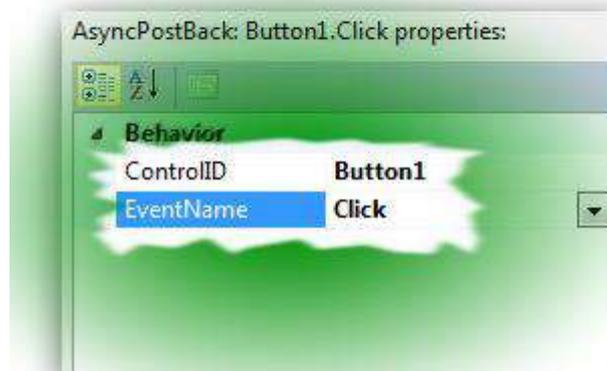


Proses yang akan memicu terjadinya postback tersebut diarahkan dari hasil klik Button yang terdapat di web page. Dalam kasus ini, proses default dari

Button adalah saat pengguna melakukan klik, sehingga diarahkan ke event *click* dari Button.



Sebuah Update Panel dapat memiliki lebih dari satu proses *AsyncPostBack*, dan umumnya *behaviour* yang diarahkan adalah dari komponen yang memiliki proses default postback seperti Button, Linkbutton atau ImageButton.



Setelah melakukan klik tombol OK pada kotak dialog tersebut, langkah terakhir dari setting komponen Update Panel adalah mengganti setting dari property *Update Mode* menjadi *conditional* agar nantinya Update Panel tersebut hanya dikenai proses refresh dari hasil klik Button.



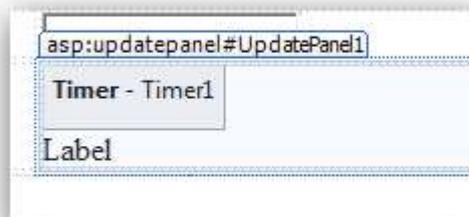
Kini saatnya menempatkan komponen Timer dari tab AJAX Extension ke dalam Update Panel tersebut. Komponen ini nanti akan menjadi tempat dari komponen Label yang diharapkan berfungsi sebagai sebuah jam digital sederhana dan menampung hasil dari inputan Numeric UpDown serta Color Picker. Bagi Anda yang telah familiar dengan pemrograman desktop, maka komponen dengan nama yang sama juga terdapat dalam pemrograman basis desktop.



Lalu set property *Interval* pada Timer menjadi 1000. Komponen Timer nantinya akan melakukan proses refresh berdasarkan nilai yang ada di dalam property *Interval*. Pada property ini, digunakan satuan milidetik sehingga jika kita set nilainya menjadi 1000 maka berarti sama dengan tiap 1 detik akan dilakukan refresh.



Langkah berikutnya adalah menempatkan komponen Label tepat di bawah komponen Timer.



Agar Label yang baru saja ditempatkan nantinya dapat diisi sesuai dengan apa yang kita inginkan, set property *Text* pada Label tersebut menjadi kosong.



Kini saatnya menyetikkan listing program yang akan menjadikan Label tersebut menjadi sebuah jam digital sederhana. Double klik pada komponen Timer, kemudian di dalam mode code Editor, ketikkan listing berikut :

```
Protected Sub Timer1_Tick(ByVal sender As Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Label1.Text = Format(Now, "hh:mm:ss")
End Sub
```

Keterangan Listing :

Pada listing tersebut, fungsi *Format* digunakan untuk memodifikasi tanggal sistem dengan format *jam (hh)*, *menit (mm)* dan *detik (ss)*. Sangat penting diperhatikan bahwa format *mm* dengan menggunakan huruf kecil berarti kita akan menggunakan format menit, sedangkan format *MM* dengan huruf besar berarti format bulan (month).

Beralihlah lagi ke mode edit pada web page dan lakukan double klik pada Button yang tersedia, kemudian ketikkan listing berikut ini :

```
Protected Sub Timer1_Tick(ByVal sender As Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Label1.Text = Format(Now, "hh:mm:ss")
End Sub

Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Label1.Font.Size = CInt(TextBox1.Text)
    Label1.Style.Add("color", "#" & TextBox2.Text)
    Label1.DataBind()
End Sub
```

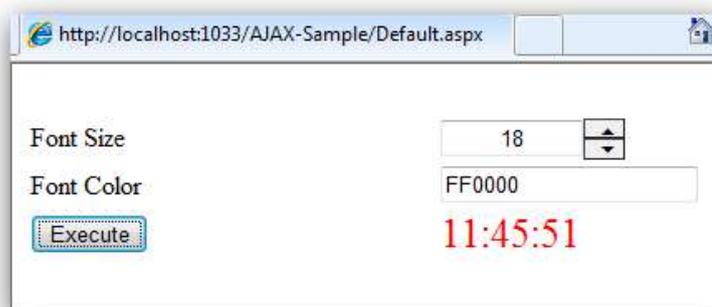
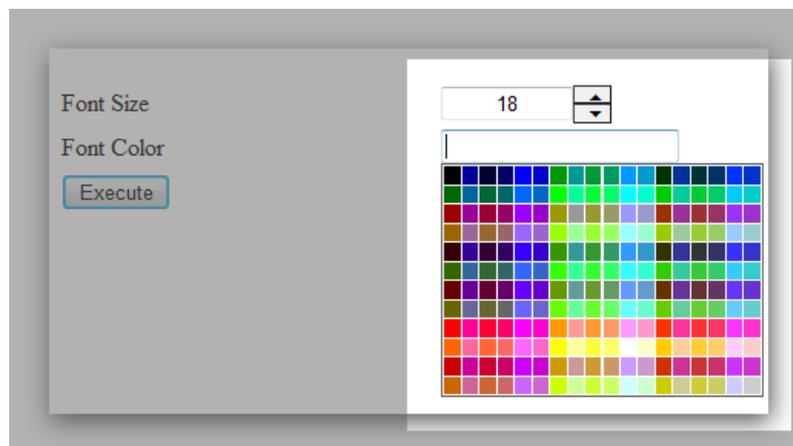
Keterangan Listing :

Perhatikan bahwa untuk menangkap hasil dari Numeric UpDown sama dengan cara menangkap hasil

dari Textbox pada umumnya. Hanya pada contoh ini hasil dari Textbox tersebut dikonversi terlebih dulu menjadi tipe numerik demi meyakinkan hasil yang diperoleh.

Sedangkan cara mengambil hasil dari Color Picker diperoleh dengan mengambil kode heksadesimal yang kemudian ditambahkan ke dalam property style dari Label yang akan diset. Kemudian Label yang dikenai perubahan tersebut direfresh tampilannya dengan menggunakan event *Databind*.

Berikut adalah contoh dari hasil yang diperoleh berupa sebuah jam digital yang dapat diset ukuran font serta warna yang ada didalamnya dengan menggunakan AJAX Control Toolkit :



Cukup mudah dan sangat cepat bukan ? Tarik nafas lalu istirahat sejenak, dan kini saatnya untuk mengerjakan contoh berikutnya...

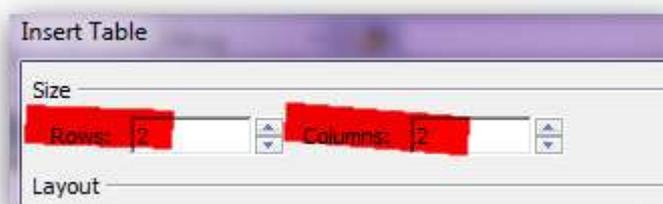
Rating

.....

Pada contoh kasus kali ini hanya akan melibatkan satu komponen dari AJAX Control Toolkit yakni komponen Rating. Komponen ini berfungsi untuk menampilkan rating atau reputasi dari hasil pilihan pengguna di suatu web page atau website. Penggunaan komponen Rating dapat melibatkan pengolahan data hasil rating ke dalam database ataupun ke media penyimpanan lain. Dalam contoh kali ini hanya akan digunakan media penyimpanan berupa file teks.

Maksud dari penggunaan file teks adalah untuk memisahkan hasil dari pemilihan rating oleh pengguna yang umumnya hanya digunakan secara temporer seperti polling atau yang lain. Namun demikian Anda juga dapat menyimpan hasil dari pemilihan rating ke dalam database.

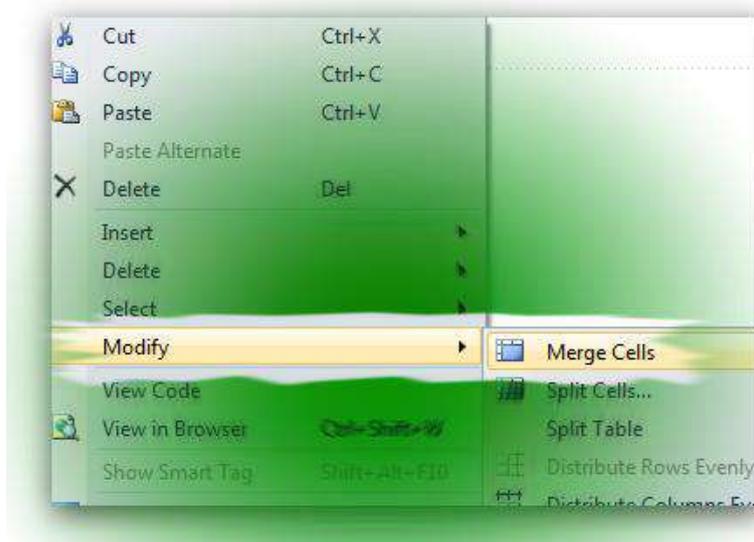
Sebagaimana contoh sebelumnya, pastikan terlebih dulu bahwa Anda telah mengikuti langkah yang ada di dalam bab persiapan hingga penempatan ToolkitScript ke dalam web page terlebih dulu. Selanjutnya di dalam web page yang telah tersedia, tempatkan sebuah tabel dengan jumlah baris dan kolom masing-masing sebanyak dua. Penempatan tabel dengan melakukan pemilihan menu *Insert* → *New Table* dan mengisikan jumlah kolom dan baris pada kotak dialog yang tersedia.



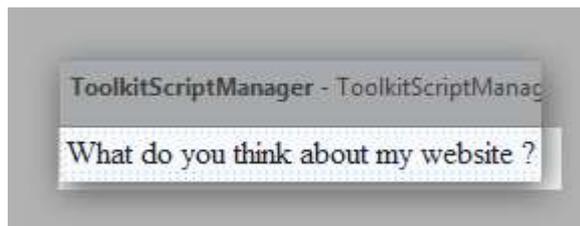
Lalu arahkan mouse ke baris kedua dan blok kedua kolom dari baris tersebut serta lakukan klik kanan pada mouse. Proses tersebut akan memilih kedua kolom yang selanjutnya akan digabung menjadi satu kolom.



Pada menu yang tersedia, pilih *Modify* → *Merge Cells* sehingga kedua kolom tersebut nantinya akan menjadi satu.



Berikutnya isikan teks pengantar pada baris pertama tepat di kolom pertama dengan isian berikut :



Kemudian pada baris kedua yang telah digabung kedua kolomnya tempatkan komponen Literal didalamnya.





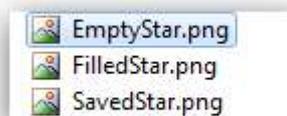
Komponen Literal umumnya digunakan untuk menampung hasil keluaran berupa teks dengan format HTML atau teks yang didalamnya berisi script dan memiliki isi yang cukup panjang. Untuk lebih jelas mengenai penggunaan komponen ini dapat dibaca di buku *Learning by Sample - ASP .NET 3.5* dari pengarang yang sama.

Penggunaan komponen Rating dari AJAX Control Toolkit membutuhkan tiga jenis file gambar yang nantinya berfungsi sebagai representasi dari tampilan Rating yang secara default berupa gambar bintang dalam bentuk kecil. Pada contoh kali ini akan digunakan gambar bintang yang telah tersedia di dalam paket AJAX Control Toolkit.

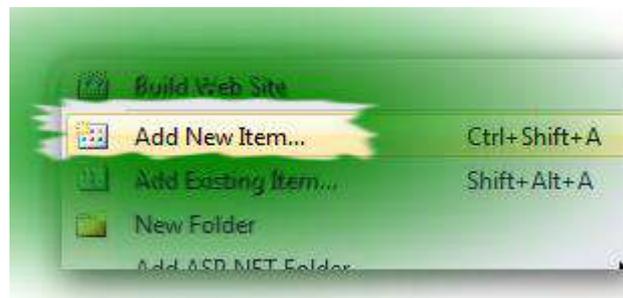
Terdapat tiga buah file gambar di dalam folder hasil ekstrasi AJAX Control Toolkit, yakni di dalam folder *AJAXControlToolkitSample/images*. Ketiga file tersebut adalah :

- a. EmptyStar.png : ☆
- b. FilledStar.png : ★
- c. SavedStar.png : ★

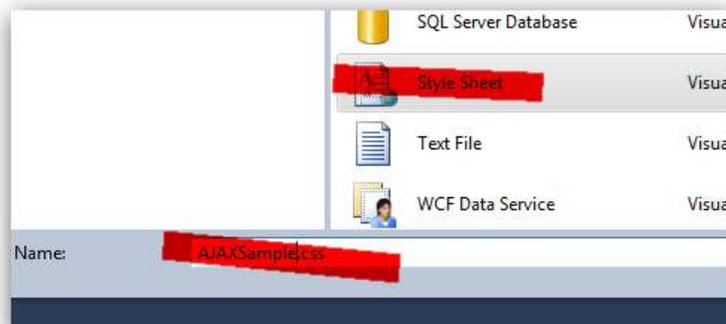
Langkah berikutnya adalah melakukan proses copy dari ketiga file tersebut ke dalam Solution Explorer yang telah kita buat sebelumnya.



Hal lain yang perlu diperhatikan dalam penggunaan komponen Rating adalah pembuatan file style sheet atau CSS (Cascading Style Sheet) yang dibutuhkan agar pada saat proses pemilihan rating, gambar bintang yang dimaksud dapat berganti secara otomatis sesuai mode yang diklik oleh pengguna. Jadi kini saatnya membuat sebuah file style sheet di dalam web site yang kita miliki. Langkah pembuatannya dengan melakukan klik kanan di Solution Explorer dan pilih menu *Add New Item*.



Kemudian pilih tipe *Style sheet* serta beri nama sesuai dengan keinginan Anda. Dalam contoh ini, file style sheet akan diberi nama sebagai *AJAXSample.css*.



Berikutnya ketikkan listing berikut di dalam file style sheet yang baru.

```

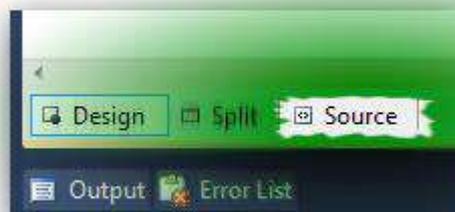
.ratingStar {
    font-size: 0pt;
    width: 13px;
    height: 12px;
    margin: 0px;
    padding: 0px;
    cursor: pointer;
    display: block;
    background-repeat: no-repeat;
}
.filledRatingStar {
    background-image: url(FilledStar.png);
}
.emptyRatingStar {
    background-image: url(EmptyStar.png);}
.savedRatingStar {
    background-image: url(SavedStar.png);}

```



Isi dari file style sheet tersebut merupakan hasil copy dari file style sheet bawaan dari AJAX Control Toolkit. Namun demikian Anda dapat juga memodifikasi isi dari style sheet tersebut sesuai dengan kebutuhan.

Kini saatnya untuk menempatkan file style sheet tersebut ke dalam web page. Beralihlah ke file *Default.aspx* atau web page yang sebelumnya telah kita modifikasi. Lalu pindahkan ke mode edit listing atau *Source*.



Di dalam mode Source tersebut, carilah bagian `<title>...</title>` lalu drag file style sheet yang telah dibuat sebelumnya dari Solution Explorer tepat di bawah bagian tersebut.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
  <link href="AJAXSample.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <form id="form1" runat="server">
```

Kemudian carilah bagian kolom kedua dari tabel yang telah dibuat dan drag komponen Rating didalamnya.

```
<table class="style1">
  <tr>
    <td class="style2">
      What do you think about my website ?</td>
    <td>
    </td>
  </tr>
```



```
<td class="style2">
  What do you think about my website ?</td>
<td>
  <asp:Rating ID="Rating1" runat="server">
  </asp:Rating>
</td>
```



Bagi Anda yang tidak memahami sintaks HTML, notasi *tr* berarti *table row* atau baris, sedangkan *td* merupakan notasi untuk menandakan bagian kolom.

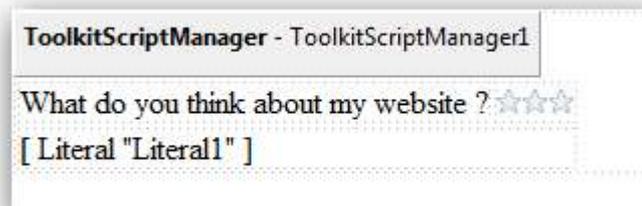
Kini saatnya memodifikasi komponen Rating yang baru saja ditempatkan, dengan menambahkan property khususnya untuk style sheet dari file style sheet yang sebelumnya telah dibuat. Seperti yang telah dijelaskan, bahwa komponen ini membutuhkan tiga buah file gambar yang masing-masing memiliki fungsi saat pengguna menentukan pilihan yang disediakan.

```
<asp:Rating ID="Rating1" runat="server" AutoPostBack="True"
EmptyStarCssClass="emptyRatingStar" FilledStarCssClass="filledRatingStar"
MaxRating="3" StarCssClass="ratingStar" WaitingStarCssClass="savedRatingStar"
OnChanged="SetRating">
```

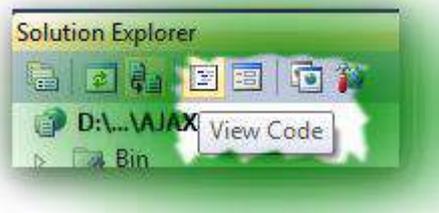
Keterangan Listing :

Pada setting property yang ada, komponen Rating ditentukan memiliki pilihan maksimal sebanyak tiga (property *MaxRating*) dan akan langsung bereaksi saat pengguna melakukan klik didalamnya (ditandai dengan setting *true* pada property *AutoPostBack*). Sedangkan pada saat pengguna melakukan klik, maka akan dieksekusi fungsi dengan nama *SetRating* yang akan dijelaskan di langkah berikutnya.

Untuk melihat hasil dari setting di mode Source, Anda dapat kembali berpindah ke mode Design. Dan berikut adalah contoh hasil penempatan komponen Rating.



Sekarang beralihlah ke mode Code Editor dengan cara melakukan klik pada ikon yang tersedia di Solution Explorer. Hal ini dilakukan untuk memulai pengetikan listing yang menjadi “otak” dari situs contoh yang sedang kita kerjakan.



Pada saat telah memasuki mode pengetikan listing, arahkan cursor ke bagian paling atas untuk melakukan inputan referensi dari AJAX Control Toolkit. Berikut adalah listing yang harus diketikkan :

```
Imports AjaxControlToolkit
```

Kemudian ketikkan prosedur berikut yang nantinya akan dieksekusi saat pengguna melakukan klik di komponen Rating :

```
Sub SetRating(ByVal sender As Object, ByVal e As RatingEventArgs)
    Dim xFileName As String = Server.MapPath("~/rating.txt")
    Dim xRating As String
    If IO.File.Exists(xFileName) Then
        xRating = IO.File.ReadAllText(xFileName)
    Else
        xRating = "0" & vbCrLf & "0" & vbCrLf & "0"
    End If
    Dim xQty() As String = Split(xRating, vbCrLf)
    xQty(Rating1.CurrentRating - 1) += 1
    Literal1.Text = "Rating result of this website is : <br/>" & _
        "Poor : " & xQty(0) & _
        "<br/> Average : " & xQty(1) & _
        "<br/> Great : " & xQty(2)
    xRating = Join(xQty, vbCrLf)
    IO.File.WriteAllText(xFileName, xRating)
    Rating1.CurrentRating = 0
End Sub
```

Sebelum memulai eksekusi dari situs contoh ini, perlu diketahui bahwa nama prosedur *SetRating* harus sama dengan nama prosedur yang telah diberikan pada property komponen Rating sebelumnya.

```
<asp:Rating ID="Rating1" runat="server" AutoPostBack="True"
    EmptyStarCssClass="emptyRatingStar" FilledStarCssClass="filledRatingStar"
    MaxRating="3" StarCssClass="ratingStar" WaitingStarCssClass="savedRatingStar"
    OnChanged="SetRating">
```

Perhatikan pula pada parameter yang ada di prosedur *SetRating*, yakni parameter yang memang khusus ada di dalam komponen Rating demi mengambil nilai yang telah diarahkan oleh pengguna.

```
Sub SetRating(ByVal sender As Object, ByVal e As RatingEventArgs)
    Dim xFileName As String = Server.MapPath("~/rating.txt")
    Dim xRating As String
    If IO.File.Exists(xFileName) Then
        xRating = IO.File.ReadAllText(xFileName)
```

Pada saat komponen Rating telah diklik oleh pengguna, maka hasil dari pemilihan tersebut akan disimpan ke dalam sebuah file teks dengan nama

rating.txt. Nama file maupun lokasi file ini dapat Anda modifikasi sesuai dengan kebutuhan.

```
Sub SetRating(ByVal sender As Object, ByVal e As RatingEventArgs)
    Dim xFileName As String = Server.MapPath("~/rating.txt")
    Dim xRating As String
    If IO.File.Exists(xFileName) Then
        xRating = IO.File.ReadAllText(xFileName)
    End If
End Sub
```

Di bagian berikutnya merupakan bagian untuk membaca file teks (jika sebelumnya sudah ada, atau situs telah dieksekusi lebih dari satu kali), sedangkan jika belum ada file teks yang terbentuk maka akan diisi dengan variabel awal yang berisi nol untuk semua opsi yang tersedia.

```
Sub SetRating(ByVal sender As Object, ByVal e As RatingEventArgs)
    Dim xFileName As String = Server.MapPath("~/rating.txt")
    Dim xRating As String
    If IO.File.Exists(xFileName) Then
        xRating = IO.File.ReadAllText(xFileName)
    Else
        xRating = "0" & vbCrLf & "0" & vbCrLf & "0"
    End If
    Dim xQty() As String = Split(xRating, vbCrLf)
    xQty(Rating1.CurrentRating - 1) += 1
End Sub
```

Bagian selanjutnya adalah penambahan dari hasil pemilihan rating oleh pengguna. Pada bagian ini, opsi yang dipilih akan ditambahkan satu nilai dari nilai awal atau nilai sebelumnya. Fungsi *Split* digunakan sebagai “pemecah” sebuah string ke dalam sebuah variabel bentuk array dengan pemisah tertentu (dalam hal ini pemisah yang dimaksud adalah *vbCrLf* atau *visual basic carriage return line feed* atau penanda ganti baris dalam file teks).

```
End If
Dim xQty() As String = Split(xRating, vbCrLf)
xQty(Rating1.CurrentRating - 1) += 1
Literal1.Text = "Rating result of this website is  

    "Poor : " & xQty(0) & "  

    "<br/> Average : " & xQty(1) & "  

    "Very Poor : " & xQty(2) & "
```

Setelah nilai berhasil ditambahkan, lalu ditampilkan ke dalam komponen Literal dengan menggunakan sedikit tambahan tag HTML yakni tag *
* untuk

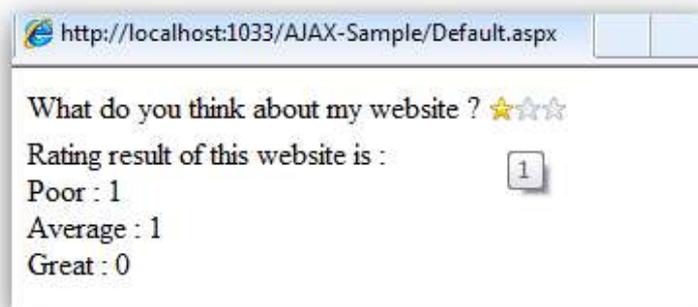
penanda ganti baris. Jika Anda ingin memodifikasi tampilan hasil polling, maka bagian ini yang harus diperhatikan terlebih dulu.

```
xQty(Rating1.CurrentRating - 1) += 1
Literall1.Text = "Rating result of this website is : <br/>" & _
    "Poor : " & xQty(0) & _
    "<br/> Average : " & xQty(1) & _
    "<br/> Great : " & xQty(2)
xRating = Join(xQty, vbCrLf)
IO.File.WriteAllText(xFileName, xRating)
```

Bagian yang terakhir adalah menyatukan kembali variabel array yang sebelumnya didapat dari fungsi *Split* lalu menyimpan kembali di file teks yang sama. Dengan demikian, file teks akan terisi dengan data yang telah diedit sesuai dengan pilihan pengguna pada komponen Rating.

```
"<br/> Great : " & xQty(2)
xRating = Join(xQty, vbCrLf)
IO.File.WriteAllText(xFileName, xRating)
Rating1.CurrentRating = 0
End Sub
```

Berikut adalah contoh hasil dari pilihan dengan menggunakan komponen Rating :



Bagi Anda yang ingin menyisipkan bagian polling ini ke dalam situs yang Anda miliki, maka tinggal memodifikasi ke dalam sebuah user control dan menyisipkan ke dalam bagian situs yang diinginkan. Mudah bukan ?

Collapsible Panel dan Drop Shadow



AsyncFile Upload dan Modal Popup

Bagian Kedua

Sebelum Memulai

Bagi Anda yang telah merasa bukan lagi di level pemula, maka bagian kedua ini sangat disarankan untuk dibaca dibandingkan di bagian yang pertama. Hal ini dikarenakan pada bagian kedua ini, contoh yang ditampilkan melibatkan proses pengolahan database sehingga mungkin sangat berguna untuk diterapkan di dunia nyata.

Dalam pengerjaannya, database yang digunakan pada bagian kedua ini menggunakan SQL Server Express Edition yang menjadi bawaan dari Visual Web Developer Express Edition 2010. Namun demikian bukan berarti Anda tidak bisa mengaplikasikan contoh yang ada dengan menggunakan SQL Server edisi komersial yang mungkin Anda miliki.

Tentu saja hal tersebut juga berarti bahwa Anda telah diasumsikan mengerti konsep dasar dari pengolahan database sehingga di dalam buku ini tidak akan dijelaskan mengenai teori dan pemahaman mengenai *primary key*, *foreign key* ataupun istilah dasar dari database yang lain.



Jika ingin mempelajari mengenai SQL Server secara lebih mendalam, maka Anda dapat membaca buku *Learning By Sample - SQL Server Express Edition* dari pengarang yang sama.

Meski contoh yang ditampilkan terkesan sebagai potongan dari sebuah situs yang besar, hal tersebut semata-mata ditujukan untuk kepentingan edukasi sehingga diharapkan bahwa Anda dapat menemui berbagai variasi dari contoh mengenai AJAX Control Toolkit. Selamat berkarya !!!!

Accordion dan HTML Editor

Asumsi yang digunakan masih tetap seperti pada contoh-contoh sebelumnya, yakni Anda telah melakukan langkah-langkah pendahuluan hingga komponen `ToolkitScriptManager` telah tersedia di sebuah halaman web yang kosong.

Pada contoh kali ini, kita akan sedikit “bermain” dengan komponen `Accordion` yang mampu menampilkan bagian data dengan model “lipatan” sehingga nantinya pengguna hanya akan melihat satu bagian data dari keseluruhan yang ada didalamnya. Sehingga pengguna hanya akan melihat data yang ia butuhkan, tidak secara keseluruhan, meski dalam proses yang dijalankan, seluruh data tetap ditampilkan apa adanya.

Selain `Accordion`, di dalam contoh kali ini juga menggunakan komponen `HTML Editor` dari `AJAX Control Toolkit`. Komponen `HTML Editor` merupakan sejenis `Textbox` yang didalamnya mampu melakukan editing dengan menggunakan style HTML seperti *bold*, *italic* maupun *underline* atau sering juga disebut sebagai *rich text editor*.

Di dalam lingkup pemrograman web, terdapat berbagai varian komponen jenis `Rich Text Editor` baik yang bersifat komersial maupun gratis. Beberapa komponen jenis ini yang gratis dan cukup terkenal antara lain `CKEditor` (dapat diunduh di ckeditor.net) ataupun `tinyMCE` serta `FreeTextBox`. Salah satu sifat default dari semua komponen tersebut adalah keharusan untuk menyertakan sebuah folder yang berisi script dari komponen tersebut yang sering kali merepotkan pada saat proses upload ke web server.

Sedangkan komponen `HTML Editor` yang terdapat di `AJAX Control Toolkit` telah menyatukan seluruh scriptnya dan melakukan proses enkapsulasi secara keseluruhan dalam satu komponen. Sehingga jauh lebih mudah digunakan dan sangat cepat proses pemrogramannya dibanding para pesaingnya.



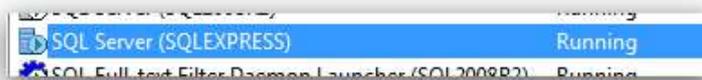
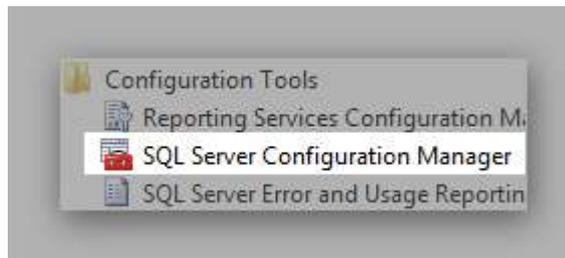
Pastikan bahwa pada saat Anda memilih komponen Rich Text Editor, Anda telah memahami kelengkapan dari tiap komponen yang dipilih.

Sebagai contoh adalah fasilitas untuk melakukan proses “pembersihan” pada saat kita melakukan proses copy-paste dari dokumen Microsoft Word.

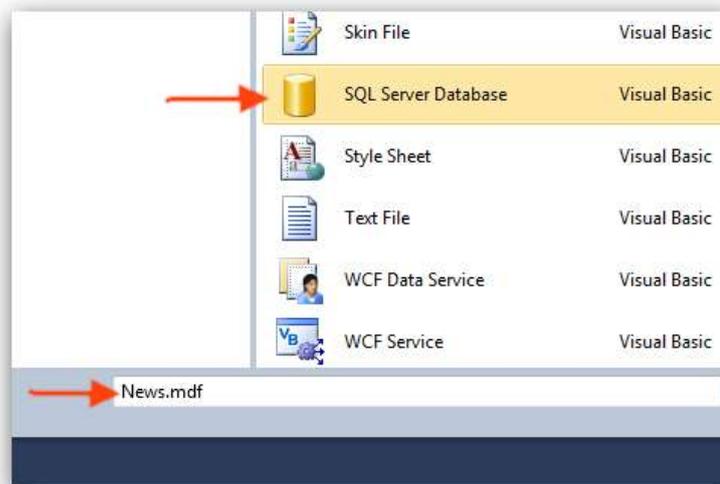
Langkah awal dalam pengerjaan contoh kali ini adalah dengan menambahkan sebuah database di dalam situs yang kita miliki. Di dalam Solution Explorer, tambahkan sebuah database yang akan dijadikan sebagai percobaan dengan melakukan klik kanan pada Solution Explorer dan memilih sub menu *Add New Item*.



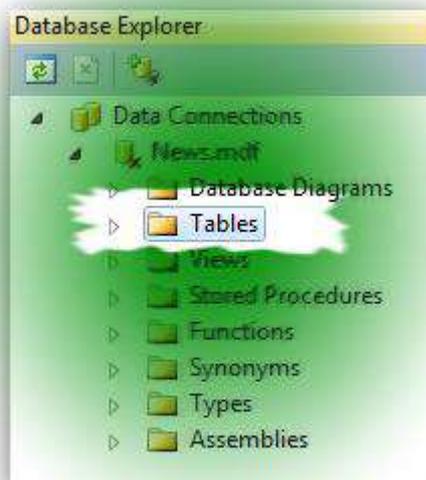
Pastikan bahwa service dari SQL Server Express Edition telah dalam keadaan *running*. Untuk memastikan hal tersebut, dapat dilihat dari shortcut aplikasi di bagian *SQL Server* → *Configuration Tools* → *SQL Server Configuration Manager*. Kemudian di dalam bagian window yang tersedia tepat di node *SQL Server Service* yakinkan bahwa *SQL Server Express* dalam keadaan *running*.



Pada kotak dialog yang tersedia, pilih tipe *SQL Server Database* dan beri nama sesuai keinginan. Dalam kasus ini file database akan diberi nama *News.mdf*.



Setelah database terbentuk, kini saatnya menuju ke bagian window *Database Explorer* (bagi Anda yang menggunakan Visual Studio dapat menuju ke bagian *Server Explorer*). Di dalam tree yang tersedia, klik kanan pada bagian node *Table* dan pilih sub menu *Add New Table*.



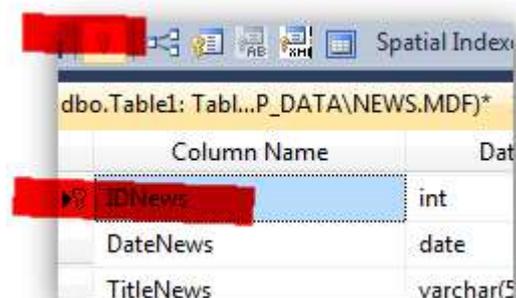


Lalu buat tabel dengan struktur yang sama seperti pada gambar berikut ini

:

Column Name	Data Type	Allow Nulls
IDNews	int	<input type="checkbox"/>
DateNews	date	<input type="checkbox"/>
TitleNews	varchar(50)	<input type="checkbox"/>
ContentNews	varchar(MAX)	<input type="checkbox"/>

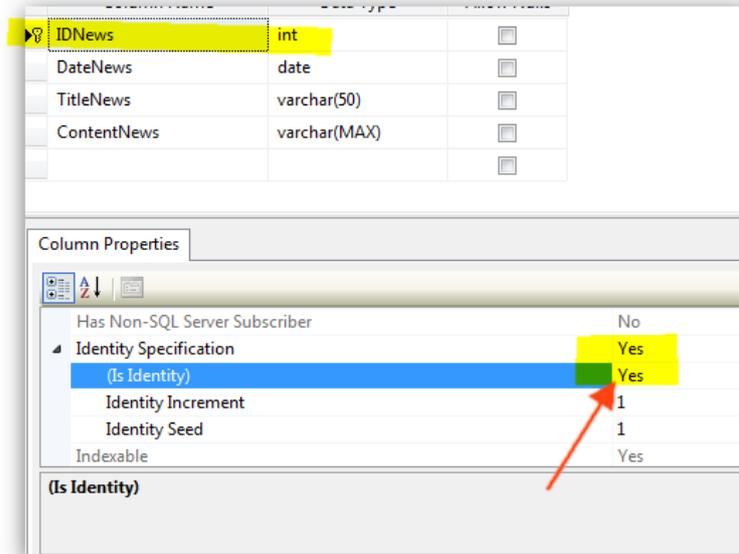
Jangan lupa untuk memberi identitas *primary key* pada field *IDNews*. Pemberian *primary key* dilakukan dengan melakukan klik terlebih dulu pada field yang dimaksud kemudian klik ikon dengan gambar kunci pada bagian toolbar atas.



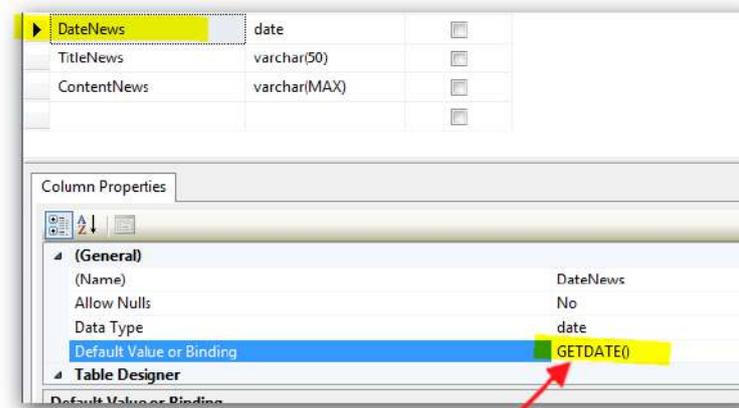
Kemudian set field *IDNews* menjadi field yang memiliki tipe *Identity*. Ini berarti bahwa field tersebut nantinya pada saat proses entri data tidak memerlukan proses entri secara manual namun akan secara otomatis akan terisi dengan nilai numerik secara berurutan. Setting ini dilakukan dengan melakukan klik pada field tersebut dan pilih di property *Identity* yang diganti menjadi nilai *True*.



Secara default setting Identity akan dimulai dari angka 1 dan akan bertambah secara otomatis dengan nilai satu setiap kali terjadi proses entri data.

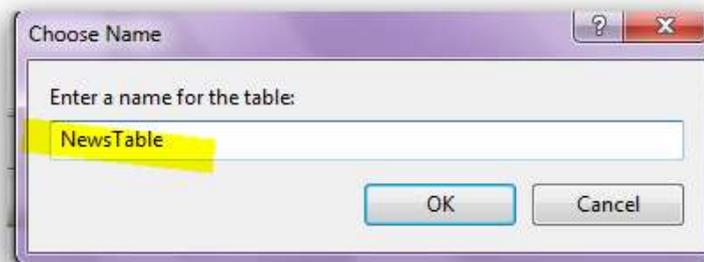
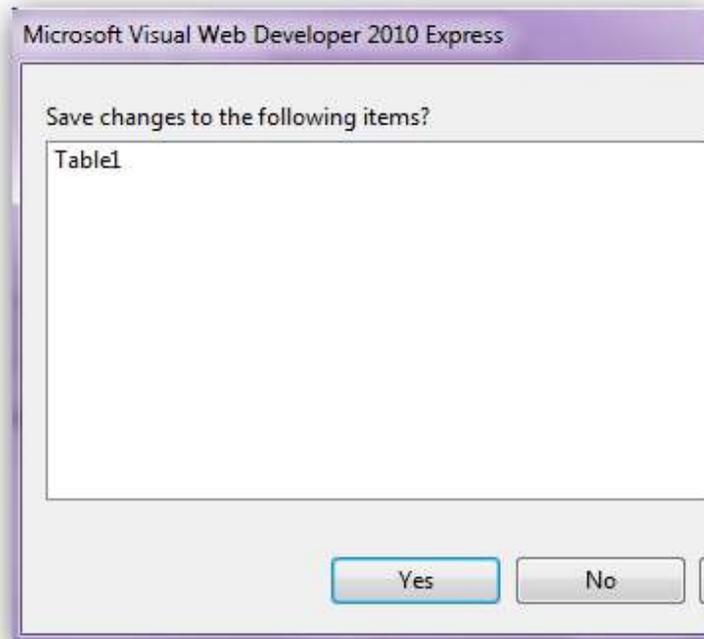


Setting berikutnya dilakukan pada field *DateNews* yang nantinya diharapkan akan secara otomatis terisi dengan tanggal dari sistem (atau web server jika situs telah diupload ke web hosting). Untuk melakukan hal tersebut, klik pada field yang dimaksud dan masukkan fungsi *GETDATE()* pada isian property *Default Value*.

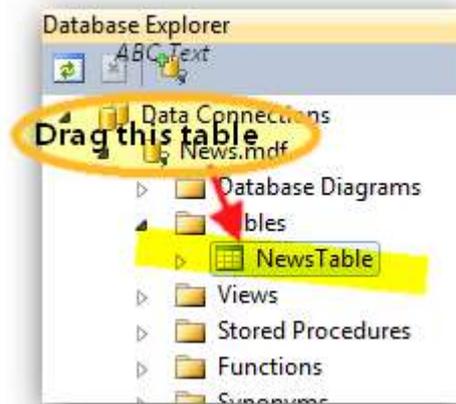


Kini saatnya untuk menyimpan tabel yang telah kita modifikasi strukturnya. Cukup dengan melakukan proses *close* pada window yang sedang

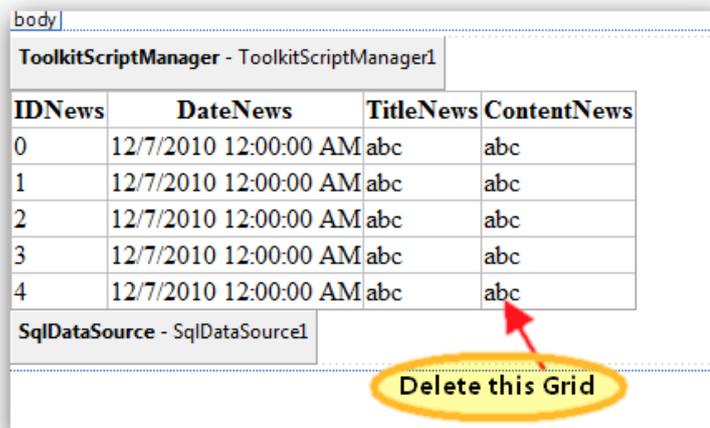
aktif, dan pada saat kotak dialog konfirmasi muncul klik tombol OK dan ketikkan nama tabel. Dalam contoh ini, tabel akan diberi nama *NewsTable*.



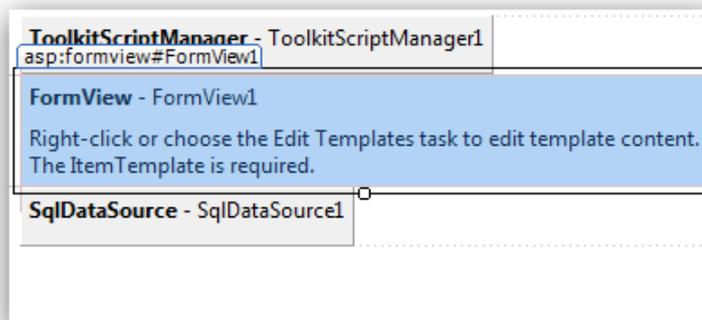
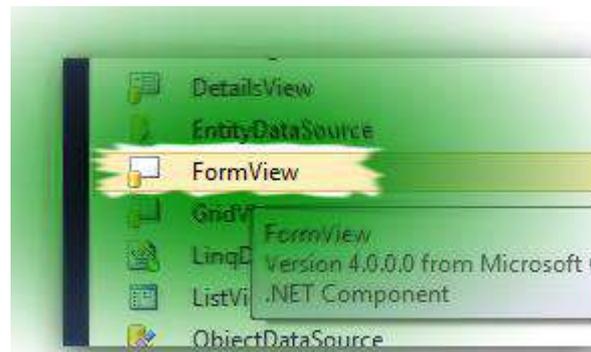
Setelah tabel selesai dibuat, kini drag tabel yang baru saja disimpan ke dalam halaman web yang telah tersedia sebelumnya. Perhatikan bahwa pada saat proses drag tabel dilakukan, maka akan terbentuk sebuah GridView dan sebuah komponen SQLDataSource di dalam halaman web yang kita miliki. Selain itu, jika Anda melakukan eksplorasi pada file *web.config* juga akan terbentuk sebuah *connection string* yang menyatakan koneksi database.



Dari komponen-komponen yang telah terbentuk tersebut, hapuslah komponen Gridview sebab akan kita ganti nanti dengan menggunakan komponen Formview. Penggantian tersebut dilakukan karena kita akan membuat proses inputan data terlebih dulu.



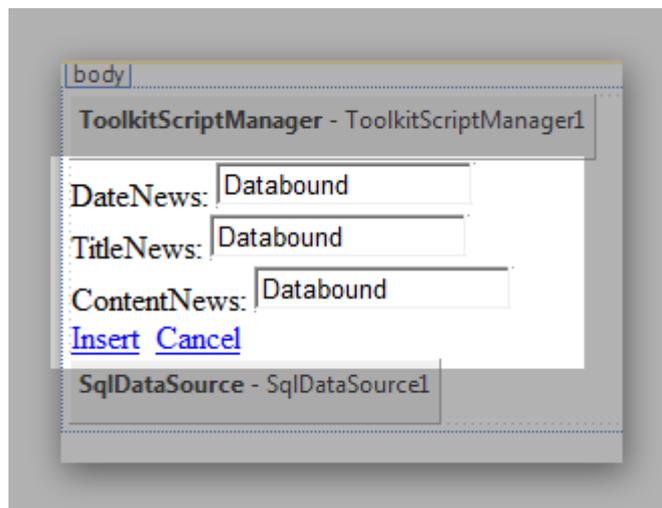
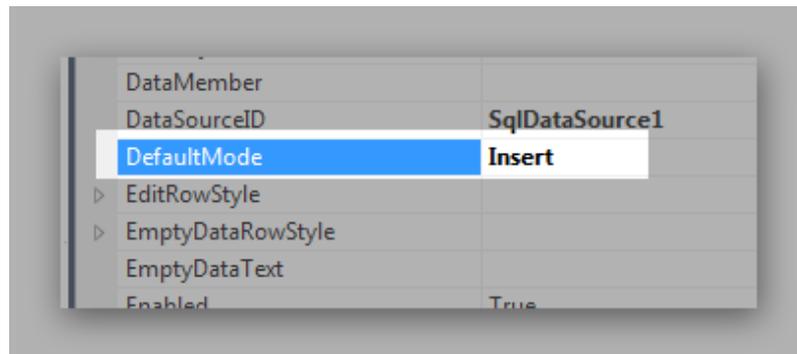
Maka setelah Gridview dihapus, drag atau lakukan double klik komponen Formview dari Toolbox khususnya pada tab Data.



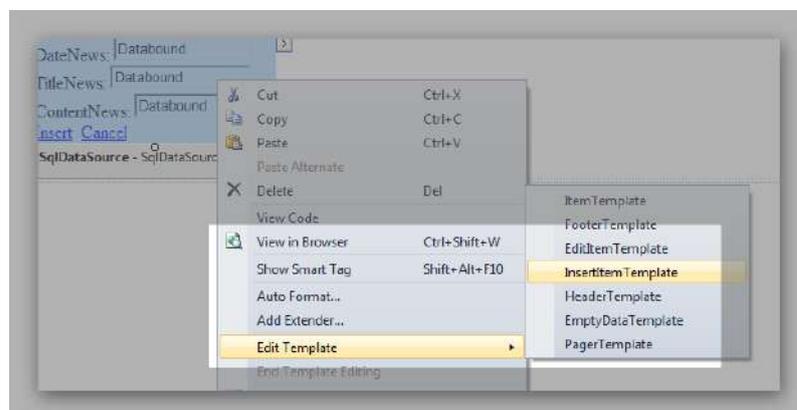
Selanjutnya klik pada Smart Tag yang tersedia di dalam Formview dan arahkan property *Data source* ke komponen *SQLDataSource1* sebagai satu-satunya data source yang tersedia di halaman web tersebut.



Lalu set property *Default mode* dari Formview tersebut menjadi *Insert*. Hal ini nantinya akan mengubah Formview menjadi tampilan khusus untuk proses inputan.

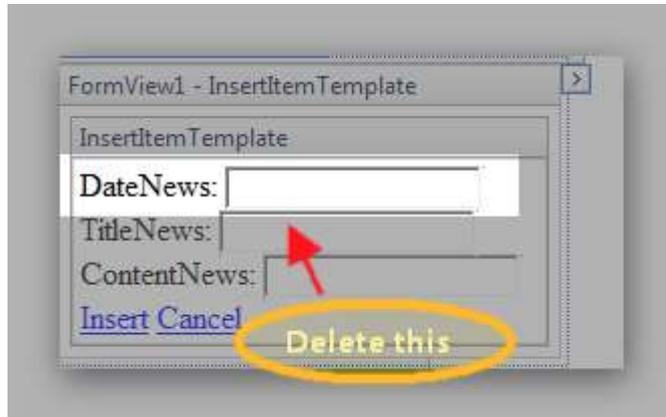


Kini saatnya untuk melakukan sedikit proses edit di dalam template Formview tersebut. Proses ini bertujuan untuk menempatkan komponen HTML Editor ke dalam Formview yang tersedia. Untuk melakukan hal tersebut, lakukan klik kanan pada Formview dan pilih sub menu *Edit Template* → *InsertItemTemplate*.

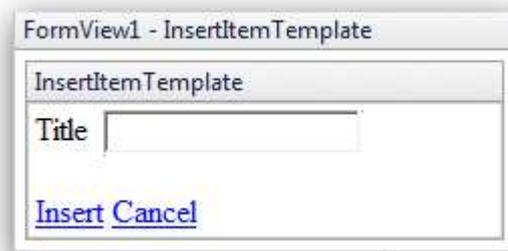


Berikutnya hapuslah isian untuk tanggal atau field *DateNews*. Hal ini dilakukan karena pada saat proses pembuatan tabel sebelumnya field ini telah diisi

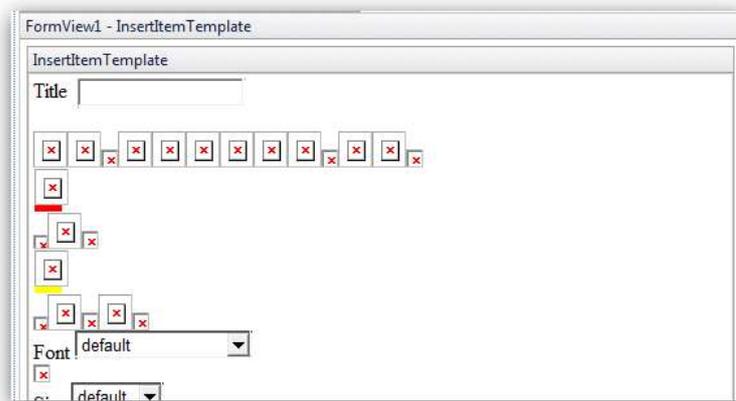
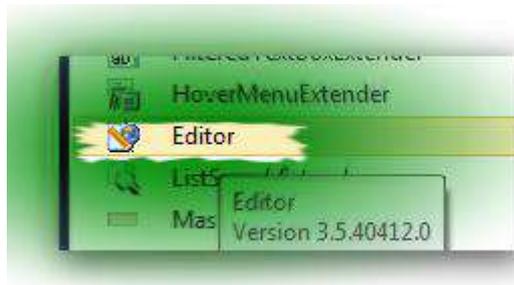
dengan property *Default Value* yang menyatakan akan terisi otomatis dari tanggal sistem, sehingga tidak diperlukan lagi pada saat proses pengisian.



Kini saatnya untuk menempatkan komponen HTML Editor pada bagian field *ContentNews*. Sebelum komponen tersebut ditempatkan, hapus terlebih dulu bagian yang sebelumnya ada untuk field tersebut.



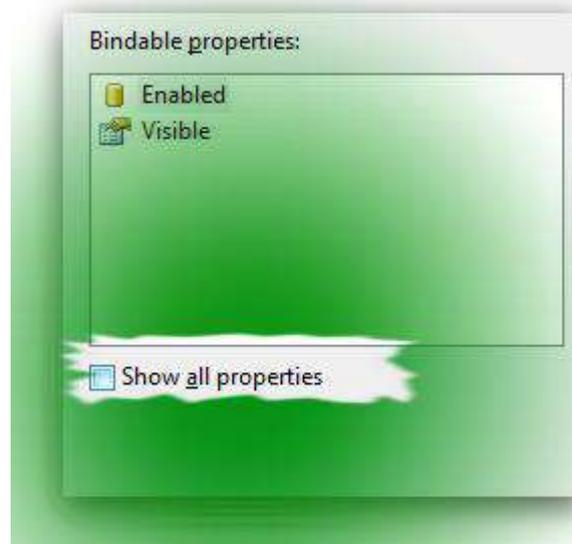
Kemudian tempatkan komponen HTML Editor sebagai penggantinya.



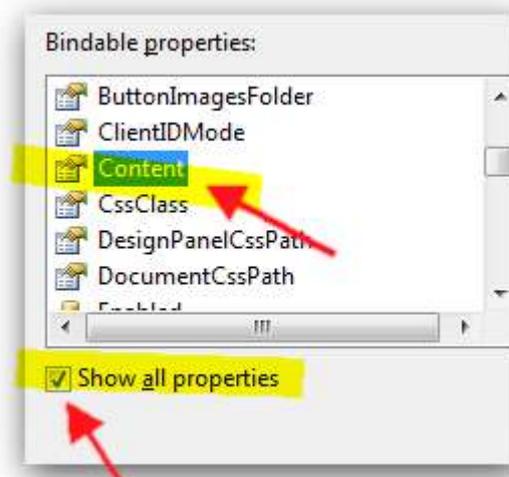
Agar komponen HTML Editor tersebut terkoneksi dengan field yang telah tersedia, klik pada Smart Tag yang ada dan pilih sub menu *Edit Data Bindings*.

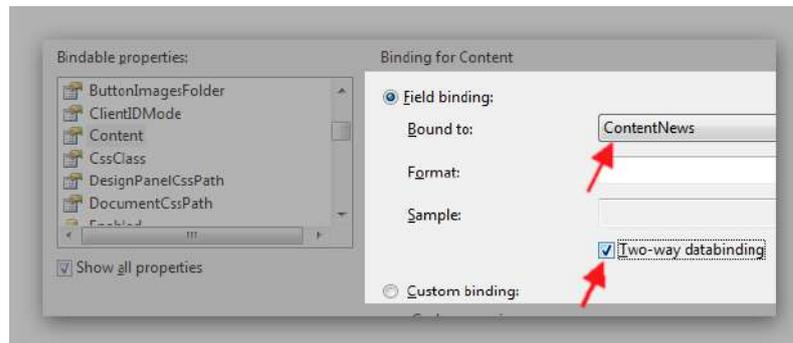


Pada kotak dialog yang tersedia, klik pada checkbox *Show all properties* untuk menampilkan seluruh property yang ada.

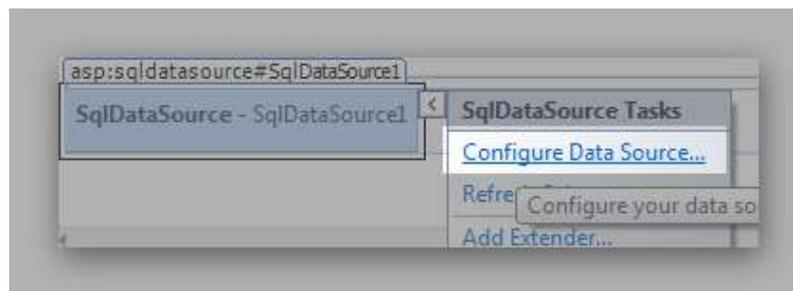


Lalu pilih property *Content* dari komponen tersebut, dan set pada isian *Field binding* agar mengarah ke field *ContentNews*. Sehingga nantinya pada saat pengisian data, seluruh teks yang ada di komponen HTML Editor akan secara otomatis dimasukkan ke field yang dimaksud. Jangan lupa untuk memilih opsi *Two way data binding* agar proses *binding* data tidak bersifat *read only* pada saat eksekusi penyimpanan dilakukan.

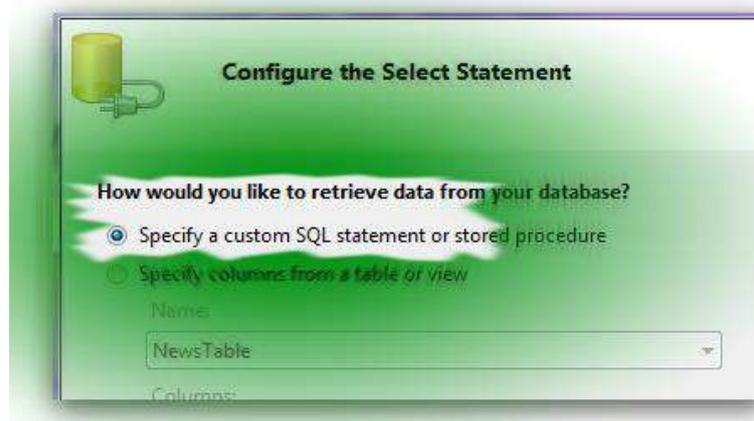




Setelah selesai melakukan setting di Formview yang telah disisipi komponen HTML Editor, maka langkah selanjutnya adalah melakukan proses edit di komponen SQLDataSource yang tersedia. Klik pada Smart Tag di komponen SQLDataSource1 yang ada di halaman web dan pilih sub menu *Configure Data Source*.



Di kotak dialog yang muncul, pilih opsi *Specify.....* untuk memodifikasi perintah SQL yang tersedia.



Kemudian tepat di tab *Insert* modifikasi perintah SQL yang tersedia seperti pada gambar berikut ini :



Perhatikan bahwa pada modifikasi perintah SQL tersebut, kita menghilangkan variabel dan field *DateNews* karena telah diperintahkan untuk secara otomatis terisi dengan tanggal sistem.

Jika sudah selesai melakukan edit pada `SQLDataSource`, kembalilah ke mode Design dari halaman web, dan tempatkan komponen `Accordion` tepat di bawah `Formview` yang ada sebelumnya.



Untuk melakukan edit di komponen `Accordion`, kita harus berpindah terlebih dulu ke mode `Source` dari halaman web. Lalu edit pada bagian `Accordion` seperti pada gambar berikut ini :



```

<asp:Accordion ID="Accordion1" runat="server"
    DataSourceID="SqlDataSource1">
  <HeaderTemplate>
    <div style="background-color:Silver;
      border:solid 1px black;
      padding:2px;cursor:pointer ">
    <h3><%# Eval("TitleNews")%></h3>
    Posted at : <%# Format(Eval("DateNews"), "dd-MM-yyyy")%>
    </div>
  </HeaderTemplate>
  <ContentTemplate >
    <div style="background-color:Aqua ;
      border:solid 1px black;
      padding:2px; border-top:none;">
    <%# Eval("ContentNews")%>
    </div>
  </ContentTemplate>
</asp:Accordion>

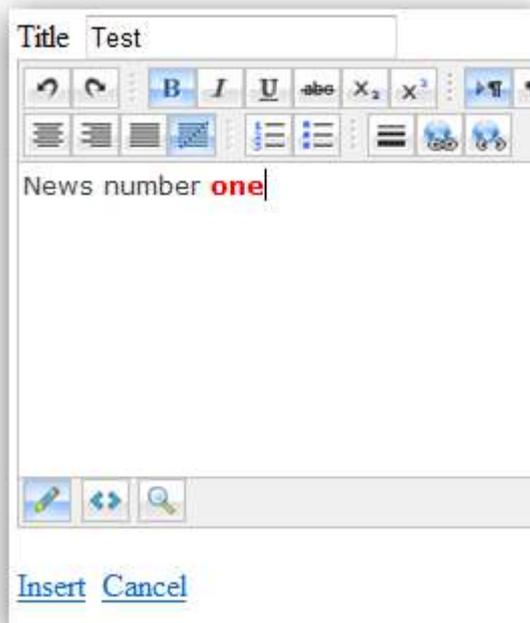
```

Keterangan Listing :

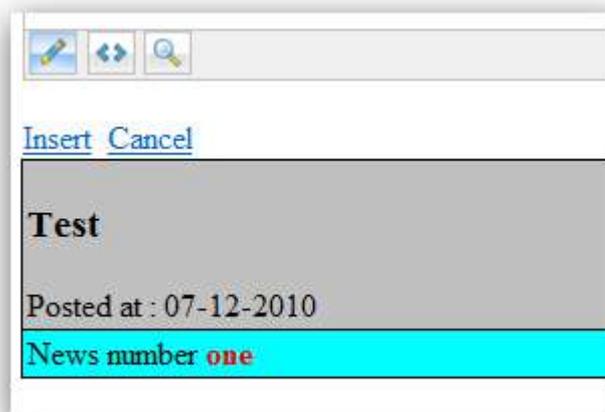
Untuk melakukan setting di komponen Accordion, terdapat dua bagian penting yang harus diperhatikan yakni *Header template* dan *Content template*. Pada bagian pertama, merupakan bagian yang pada saat diklik oleh pengguna akan menampilkan isi dari *Content template* yang saling berhubungan.

Bagian yang dapat Anda modifikasi sesuai selera adalah pada saat Accordion menerima style sheet yang ada di dalam layer (tag *div*) tiap bagian. Pada bagian ini Anda dapat mencurahkan kreatifitas sehingga tampilan Accordion dapat menjadi lebih menarik bagi pengguna.

Jika sudah selesai melakukan edit pada komponen Accordion, kini saatnya untuk mengeksekusi hasil dari halaman web yang kita miliki dengan menekan tombol Ctrl+F5 :



Dan berikut adalah contoh hasil dari inputan di dalam komponen Accordion tersebut :



Sangat menarik dan mudah bukan ? Kini saatnya mengerjakan contoh berikutnya....

ComboBox dan Confirm Button

Pada contoh di sub bab ini didemonstrasikan tentang bagaimana cara mengaplikasikan dua buah komponen dari AJAX Control Toolkit

Studi Kasus II : Mini Blog

Pengantar

Studi kasus berikut ini akan mensimulasikan pembuatan sebuah blog dengan cara yang sangat sederhana. Untuk kepentingan penulisan buku, banyak fitur dan fasilitas dari studi kasus blog ini yang dihilangkan. Penghilangan fitur tersebut tidak akan mengubah esensi dari blog tersebut, tetapi hal yang terpenting dari studi kasus ini adalah penggunaan ASP .NET AJAX serta pemakaian AJAX Control Toolkit itu sendiri.

Blog yang akan dibuat merupakan jenis single blog sederhana, tanpa ada fasilitas untuk mengiputkan komentar di tiap blog yang diisi. Selain itu, fasilitas entri blog hanya berupa teks, tidak menangani entri dengan menggunakan tag HTML.

Karenanya, setelah pembuatan studi kasus ini diharapkan Anda dapat mengembangkan beberapa fitur antara lain :

1. Pemberian fasilitas komentar di tiap berita
2. Pemberian fasilitas archieve bulanan
3. Entri isi blog yang mendukung isi dengan tag HTML dengan menggunakan komponen editor teks yang free seperti FreeTextBox.
4. Pemberian fasilitas untuk mengganti theme dengan menggunakan CSS.

Blog yang akan dibuat sendiri hanya terdiri dari dua halaman web yaitu halaman web utama yang menampilkan lima isi blog terbaru, sekaligus arsip blog yang dibagi berdasarkan kategori. Halaman web utama juga menampilkan form untuk melakukan login bagi administrator. Sedangkan halaman kedua adalah halaman untuk pihak administrator yang melakukan maintenance kategori serta maintenance dari blog itu sendiri.

Dalam studi kasus ini akan digunakan teknik *Master Pages* didalamnya. Penggunaan Master Pages tidak dijelaskan secara rinci dalam buku ini, untuk mengetahui lebih lanjut mengenai pembahasan Master Pages dapat dibaca di buku *Panduan Belajar ASP .NET 2.0* oleh pengarang yang sama.

Langkah pertama dalam pembuatan studi kasus ini sama dengan contoh sebelumnya yaitu membuat sebuah solution dengan tipe ASP .NET AJAX

didalamnya. Kemudian dilanjutkan dengan pembuatan database yang akan dijelaskan dalam sub bab selanjutnya.

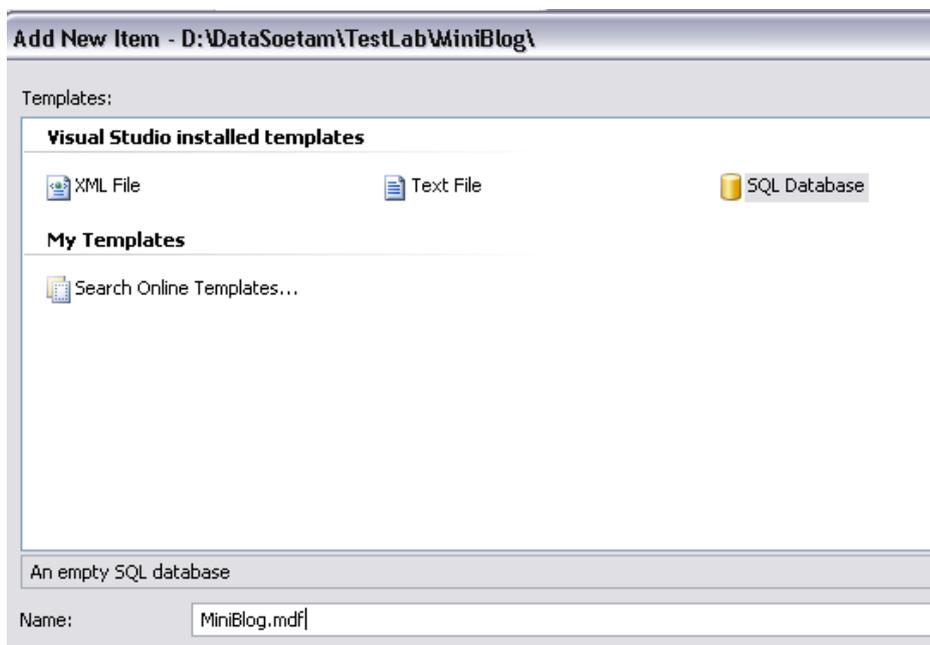


Disarankan untuk mengikuti langkah pembuatan secara berurutan dalam pembuatan studi kasus ini.

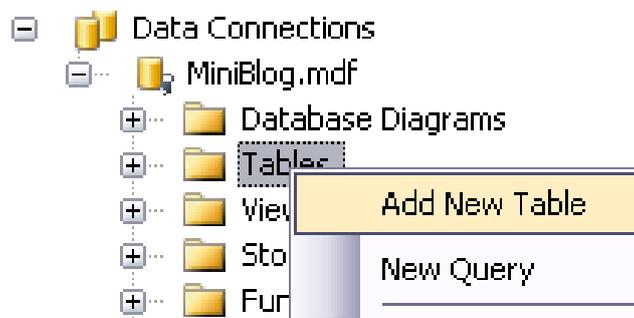
Pembuatan Database

Database dalam studi kasus ini tergolong sederhana, hanya terdiri dari tiga buah tabel yang masing-masing berfungsi untuk menampung data untuk kepentingan login, kategori blog serta isi dari blog itu sendiri.

Dari solution yang telah dibuat sebelumnya, klik kanan pada folder *App_Data* dan pilih sub menu *Add New Item* dan pada kotak dialog yang tersedia, tambahkan sebuah database baru dengan nama *MiniBlog*



Kemudian, klik pada *Database Explorer*, dan pada sub tree *Table*, klik kanan dan pilih sub menu *Add New Table*.



Lalu, buat tiga buah tabel dengan struktur sebagai berikut :

Tabel UserProfile

	Column Name	Data Type	Allow Nulls
	UserID	varchar(10)	<input type="checkbox"/>
	UserName	varchar(50)	<input type="checkbox"/>
	UserPassword	varchar(10)	<input checked="" type="checkbox"/>

Tabel Category

	Column Name	Data Type	Allow Nulls
	Category	varchar(50)	<input type="checkbox"/>
	StatusCategory	bit	<input type="checkbox"/>

Tabel IsiBlog

	Column Name	Data Type	Allow Nulls
	idBlog	int	<input type="checkbox"/>
	TglBlog	datetime	<input type="checkbox"/>
	JudulBlog	varchar(MAX)	<input type="checkbox"/>
	Category	varchar(50)	<input type="checkbox"/>
	IsiBlog	varchar(MAX)	<input type="checkbox"/>
	StatusBlog	bit	<input type="checkbox"/>

Khusus untuk tabel *IsiBlog*, di field *idBlog* diset agar menjadi field yang memiliki karakteristik *autonumber* atau melakukan counter otomatis pada saat sebuah record ditambahkan. Hal ini bisa dilakukan dengan mengubah setting *Identity Specification* menjadi *Yes* pada field tersebut. Untuk melakukannya, klik pada field *idBlog* dan lihat pada kotak property dibawahnya lalu set property *IsIdentity* menjadi *Yes*.

Column Properties	
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1



Penggunaan field tipe *Identity* lebih banyak digunakan untuk field dengan tipe numerik (int, double dan sejenisnya). Perhatikan pula, resiko dari penggunaan field jenis *Identity* yang pada saat sebuah proses penghapusan record yang berada di tengah, maka nomor urut akan meloncat selamanya. Misal : jika terdapat 10 buah record, dan kemudian record ke-5 dihapus, maka nomor 5 akan hilang selamanya, sehingga urutan nomor akan meloncat dari 1 hingga 4 lalu dilanjutkan nomor 6 hingga 10.

Sedangkan untuk field *TglBlog* diset secara otomatis agar pada saat pengisian record baru langsung diisi dengan tanggal dan jam sistem yang bersangkutan, sehingga tidak diperlukan lagi proses entri tanggal dalam pengisian blog nantinya. Untuk melakukan hal tersebut, bisa dengan melakukan setting *Column Properties* pada field *TglBlog* dengan menempatkan fungsi *GetDate()* didalamnya. Sehingga, pada saat record diisikan, field tersebut akan langsung berisikan tanggal pada saat blog disimpan.

Column Properties	
Data Type	datetime
Default Value or Binding	(getdate())
Table Design...	

Berikutnya, isikan data contoh pada tabel *UserProfile* dan tabel *Category* untuk kepentingan testing program. Data contoh dapat diganti sesuai dengan keinginan Anda, tetapi harus tetap diperhatikan pada saat testing dilakukan.

Cara pengisian data contoh dengan melakukan klik kanan pada tabel yang bersangkutan, dan pilih sub menu *Show Table Data*. Kemudian isikan data contoh seperti pada gambar berikut :

Tabel UserProfile

UserID	UserName	UserPassword
admin	Administrator	4dm1n

Tabel Category

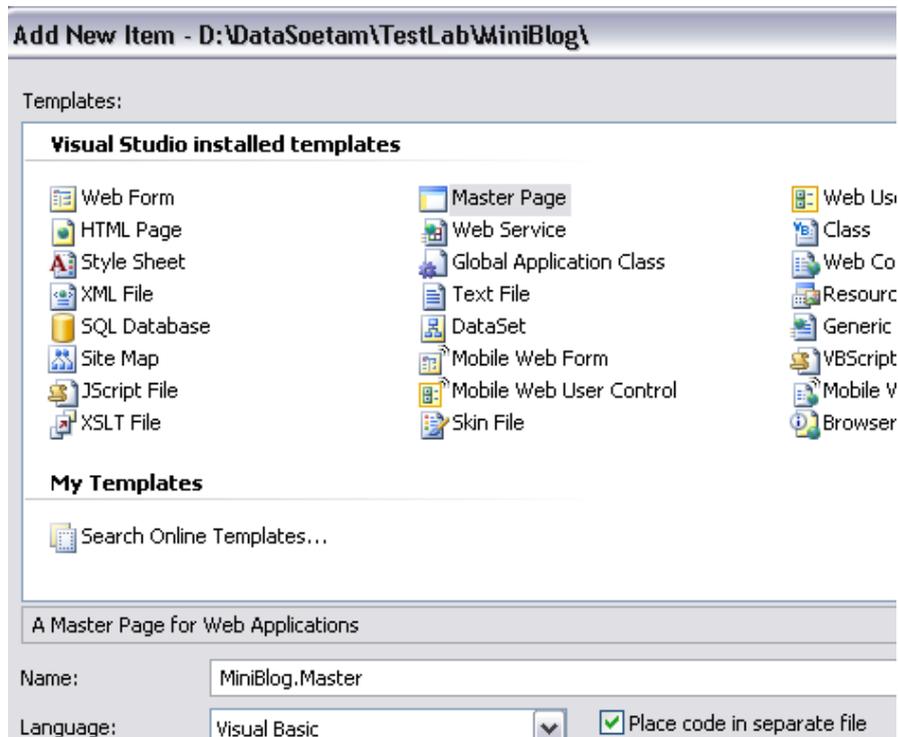
Category	StatusCategory
Campus Life	True
Knowledge	True
Link	True
Personal Life	True

Master Page

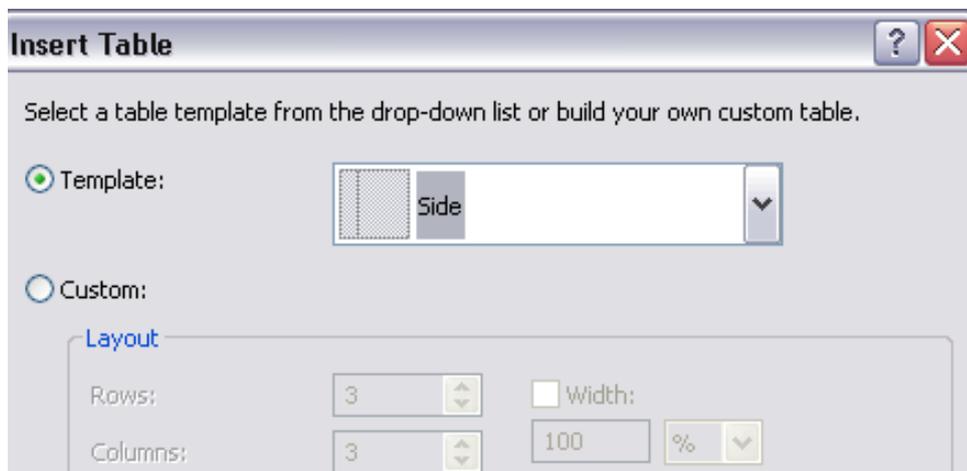
Pembuatan Master Page dilakukan agar terjadi konsistensi desain pada sebuah situs yang memiliki lebih dari satu halaman web. Selain itu juga mempermudah desainer untuk berkolaborasi lebih lanjut dengan programmer dalam menyatukan visi antara desain dan program dalam sebuah situs (untuk mempelajari lebih lanjut mengenai Master Page, bisa dibaca buku dengan judul *Panduan Belajar ASP .NET 2.0* oleh pengarang dan penerbit yang sama).

Pembuatan Master Page dalam sebuah solution dapat dilakukan dengan cara melakukan klik kanan pada solution, kemudian pilih sub menu *Add New Item* lalu dalam kotak dialog yang tersedia pilih tipe *Master Page*. Untuk studi kasus ini, Master Page akan diberi nama *MiniBlog*.

Di dalam Master Page tersebut, secara otomatis terdapat sebuah komponen *ContentPlaceHolder* yang nantinya menjadi tempat dari halaman web yang akan dibuat. Perlu diingat, bahwa sebuah Master Page tidak dapat berdiri sendiri, melainkan harus minimal memiliki sebuah halaman web yang didalamnya mengacu kepada Master Page tersebut.



Berikutnya tempatkan sebuah tabel dalam Master Page tersebut, dengan menggunakan template *side* dari menu *Layout* → *Insert Table* dan set template seperti pada gambar berikut :



Kemudian, set property *valign* tiap kolom dari tabel menjadi *top*. Cara melakukan setting property kolom dari tabel cukup dengan melakukan klik di tiap kolom yang dimaksud dan membuka window property. Hal ini dilakukan agar pada langkah desain berikutnya, komponen yang diletakkan akan secara langsung mengarah ke bagian paling atas dari halaman web.

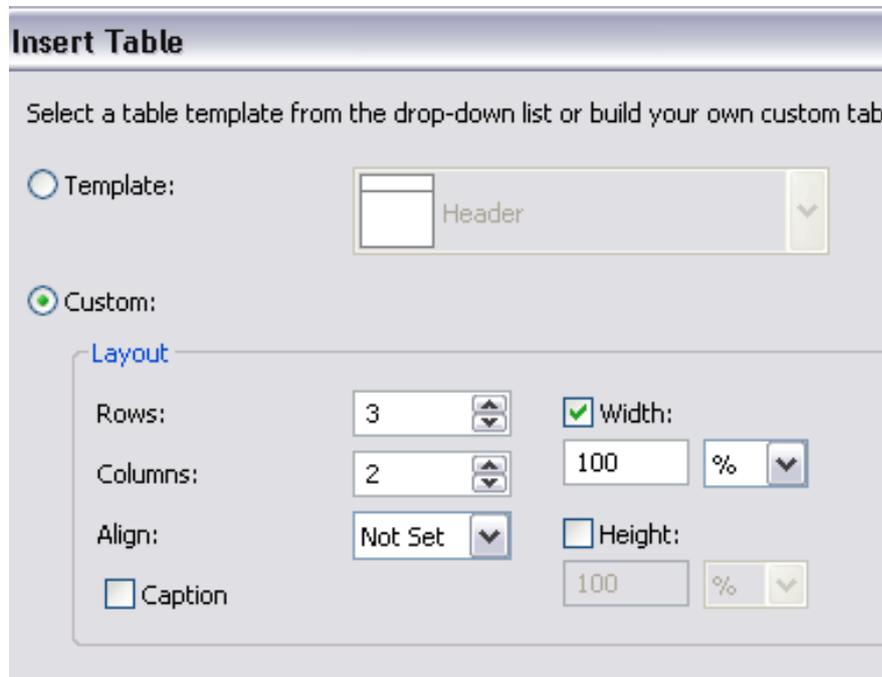
Style	WIDTH: 21
Title	
valign	top
xml:Lang	

Selanjutnya, tempatkan sebuah komponen MultiView  di kolom pertama. Di dalam MultiView tersebut, tempatkan dua buah komponen View  untuk kepentingan halaman utama dan halaman administrator.



Sebuah komponen MultiView tidak bisa berdiri sendiri, didalamnya minimal terdapat sebuah komponen View.

Di View yang pertama, tempatkan judul situs serta sebuah tabel (dengan menggunakan menu *Layout* → *Insert Table*) yang memiliki dua kolom dan tiga baris serta lebar 100% secara relatif dari kolom itu sendiri. Tabel ini nantinya ditempati oleh dua buah Textbox dan sebuah Button untuk kepentingan proses login.



Kemudian masih di View yang pertama, drag tabel *UserProfile* dari Database Explorer di dalam View tersebut. Hapus GridView yang terbentuk dari hasil drag tersebut, karena yang dibutuhkan dalam proses nantinya hanya komponen *SqlDataSource*. Sedangkan di View yang kedua, tempatkan sebuah *LinkButton* untuk kepentingan Logout.

Selanjutnya set property *ActiveViewIndex* menjadi 0 pada *MultiView* agar secara default *MultiView* akan mengarah langsung ke View yang pertama.

(ID)	MultiView1
ActiveViewIndex	0



Pada Textbox yang akan digunakan untuk pengisian Password, set property *TextMode* menjadi *Password* agar Textbox tersebut memiliki fasilitas *masking* pada saat pengisian password.



Lalu set property *DataSourceMode* *SqlDataSource* tersebut menjadi *DataReader*. Hal ini dilakukan karena koneksi yang diperlukan hanya untuk pembacaan data secara sederhana, bukan untuk manipulasi data.



Kini, klik pada Smart Tag di komponen *SqlDataSource* yang tersedia, dan pilih sub menu *Configure Data Source* untuk memodifikasi perintah SQL yang ada didalamnya.



Di dalam kotak dialog yang tersedia, klik pada tombol *Where* untuk mengubah kondisi parameter yang ada di dalam *SqlDataSource* tersebut. Di kotak dialog berikutnya, arahkan field *UserID* sebagai field yang akan dikondisikan dan pilih tipe *Control* untuk tipe parameter dan *Textbox1* sebagai parameter.

Untuk parameter yang kedua, field yang menjadi kondisi adalah *UserPassword* dan sebagai tipe parameternya tetap *Control* sedangkan komponen yang menjadi parameter adalah *Textbox2*.

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: UserPassword

Operator: =

Source: Control

SQL Expression: [UserPassword] = @UserPassword

Value: TextBox2.Text

Parameter properties

Control ID: TextBox2

Default value:

WHERE clause:

SQL Expression	Value
[UserID] = @UserID	TextBox1.Text
[UserPassword] = @UserPassword	TextBox2.Text

Langkah berikutnya adalah melakukan double klik pada Button *Login* kemudian didalamnya ketikkan listing berikut :

```

Dim xreader As Data.IDataReader
With SqlDataSource1
    xreader = .Select _
        (DataSourceSelectArguments.Empty)
    If xreader.Read Then
        Session("Login") = _
            xreader.Item("UserName")
        Response.Redirect("admin.aspx")
    Else
        Page.RegisterStartupScript _
            ("pesan", _
            "<script>alert" & _
            " ('Login Failed !') " & _
            "</script>")
    End If
End With

```



Pembacaan data login, dilakukan dengan memanfaatkan perintah SQL didalam komponen SqlDataSource yang telah dimodifikasi sebelumnya (dengan parameter UserID dan UserPassword yang diambilkan dari dua buah Textbox yang tersedia). Pembacaan data menggunakan mode DataReader yang bersifat *fast forward* dan *ReadOnly* agar pembacaan lebih cepat.

Sedangkan pada saat record yang dimaksud tidak ditemukan (berarti bahwa proses login gagal), akan ditampilkan messagebox yang memanfaatkan Javascript yang diinjeksikan ke dalam listing.

Setelah itu, dobel klik pada LinkButton *Logout* di View yang kedua, kemudian ketikkan listing berikut :

```
Session.Clear()  
Response.Redirect("default.aspx")
```



Penggunaan Session, selain sebagai penanda setelah proses login dilakukan, juga sebagai proteksi nantinya bagi halaman web administrator. Session yang digunakan dalam studi kasus ini sangat sederhana, karena selain terdiri dari satu jenis Session, juga hanya menyimpan field *UserName* dari hasil proses login. Penggunaan Session lebih lanjut dapat dibaca di buku *Panduan Belajar ASP .NET 2.0*.

Halaman Admin

Hingga proses pembuatan Master Page selesai, situs yang dibuat masih belum bisa dieksekusi untuk melihat hasilnya. Karena sebuah Master Page membutuhkan minimal satu halaman web yang mengacu kepadanya agar bisa dieksekusi.

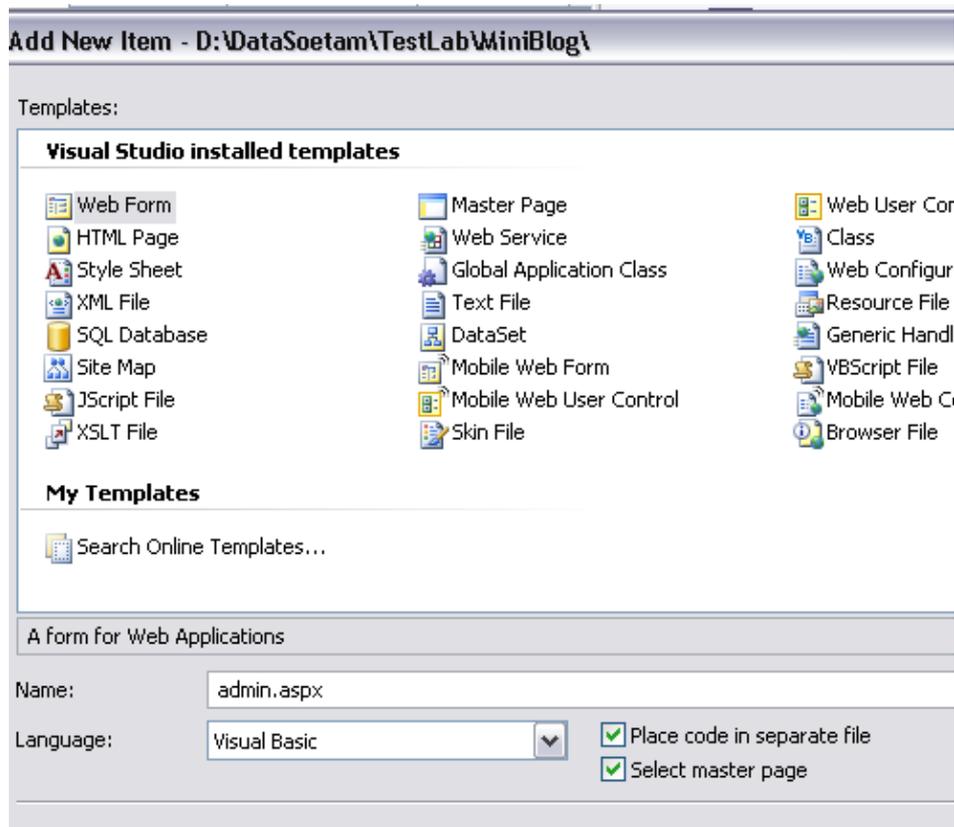
Halaman admin merupakan halaman yang berisi dua fitur yaitu maintenance kategori dan maintenance dari blog itu sendiri. Di dalam halaman ini akan diterapkan teknik AJAX tanpa menggunakan AJAX Control Toolkit didalamnya. Penggunaan AJAX Control Toolkit baru digunakan di halaman utama.

Halaman web untuk admin ini juga akan terproteksi sehingga tidak bisa dieksekusi langsung tanpa harus melalui proses login. Hal ini dimungkinkan dengan adanya proteksi melalui Session yang telah diterapkan sebelumnya di dalam Master Page.

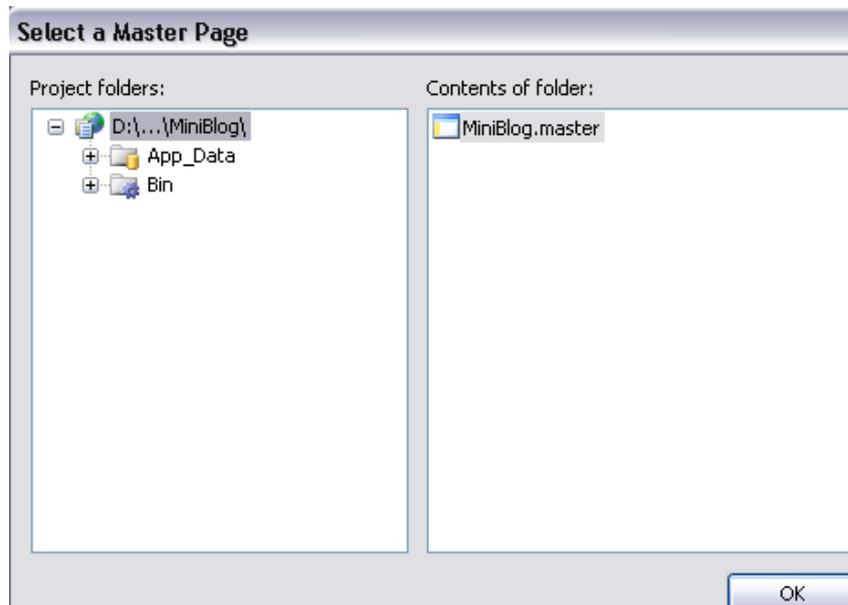


Jenis proteksi lain untuk sebuah halaman web ataupun untuk sebuah folder dapat dilakukan dengan memodifikasi file web.config dan menambahkan tag tertentu didalamnya. Hal tersebut dibahas lebih detail di buku *Panduan Belajar ASP .NET 2.0*

Langkah pertama dari pembuatan halaman admin ini adalah menambahkan halaman web baru yang mengacu ke dalam Master Page yang sudah dibuat. Klik kanan pada solution dan pilih sub menu *Add New Item*. Kemudian pada kotak dialog yang tersedia, pilihlah opsi *Select Master Page* untuk mereferensikan halaman web tersebut ke dalam sebuah Master Page.



Kemudian, di kotak dialog berikutnya, pastikan bahwa pemilihan Master Page telah mengacu ke Master yang sebelumnya telah dibuat. Dalam studi kasus ini, kebetulan hanya terdapat sebuah Master Page, sehingga pemilihan dapat lebih mudah dilakukan. Tetapi jika pada kasus lain terdapat lebih dari satu Master Page, usahakan agar lebih berhati-hati saat melakukan pemilihan Master Page.



Setelah memberi nama halaman web tersebut dengan nama *admin.aspx*, maka didalamnya akan terlihat sebuah komponen *ContentPlaceHolder* yang bisa diedit, dan bagian lain yang berwarna lebih gelap, menandakan bahwa bagian tersebut adalah bagian dari Master Page yang tidak boleh diedit.

Dengan kondisi demikian, maka apabila terjadi perubahan ataupun modifikasi desain awal, perubahan atau modifikasi tersebut hanya perlu dilakukan di dalam Master Page, bukan di tiap halaman yang ada.

Komponen pertama yang ditempatkan di dalam *ContentPlaceHolder* yang tersedia adalah komponen *ScriptManager* yang nantinya dibutuhkan pada saat teknik *AJAX* diimplementasikan. Selanjutnya ditempatkan sebuah komponen *UpdatePanel* untuk mengimplementasikan proses *partial postback* di halaman tersebut.

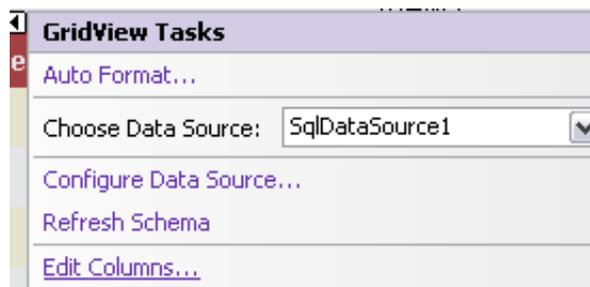
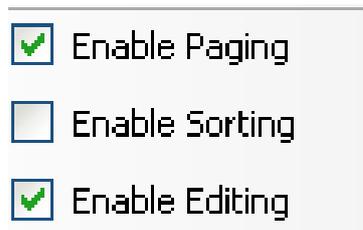


Seperti halnya pada contoh-contoh di bab sebelumnya, pastikan bahwa *ScriptManager* terletak di bagian paling atas dari halaman web yang akan mengimplementasikan teknik *AJAX*.

Di dalam *UpdatePanel* yang tersedia, tempatkan lagi sebuah komponen *MultiView*, lalu didalamnya tempatkan tiga buah komponen *View* yang nantinya akan digunakan untuk proses maintenance data kategori, maintenance isi blog serta menu kecil untuk berpindah antar maintenance tersebut.

Maintenance Data Kategori

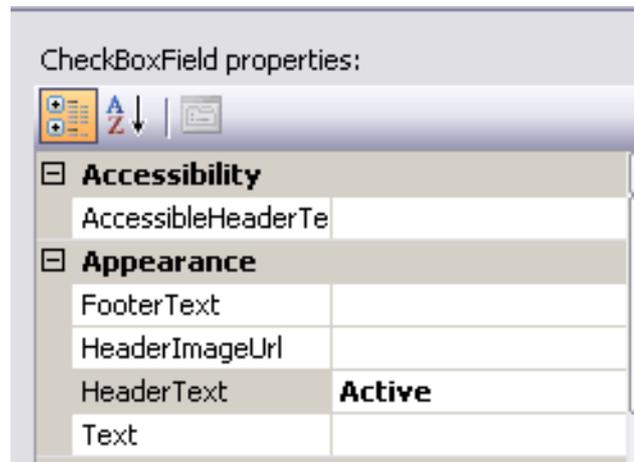
Dimulai dari View yang pertama, drag tabel *Category* dari Database Explorer ke dalamnya. Kemudian, di GridView yang tersedia, klik pada Smart Tag dan pilih sub menu *Enable Editing* agar GridView tersebut mampu melakukan edit untuk field *StatusCategory* dan pilih juga sub menu opsi *Enable Paging*. Lalu pilih sub menu *Edit Column* untuk mengedit tampilan dari GridView tersebut.



Di dalam kotak dialog yang tersedia, edit property *HeaderText* kolom *StatusCategory* menjadi *Active*. Hal ini ditujukan agar pengguna dapat lebih paham mengenai fungsi dari kolom tersebut.



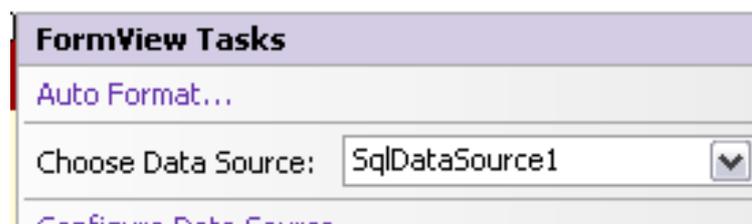
Penggantian nama pada property *HeaderText* sangat berpengaruh dalam sebuah halaman web. Hal ini dikarenakan pengguna pada dasarnya sangat bergantung pada keterangan yang ada dalam halaman web, sedangkan pihak programmer seringkali memiliki kebiasaan untuk menganalogikan pengguna telah memiliki pengertian yang sama dengan dirinya dalam hal pemahaman desain halaman web.



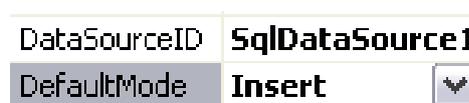
Setting terakhir pada GridView adalah mengubah property *PageSize* menjadi 5. Setting property dilakukan agar pada saat tampilan data kategori lebih dari 5 record, akan menampilkan fitur *paging*, sehingga tampilan GridView tidak terlalu memanjang ke bawah.



Kemudian, tambahkan komponen FormView di di bawah GridView yang sudah diset propertynya. Klik pada Smart Tag di FormView dan arahkan ke komponen SqlDataSource yang tersedia.



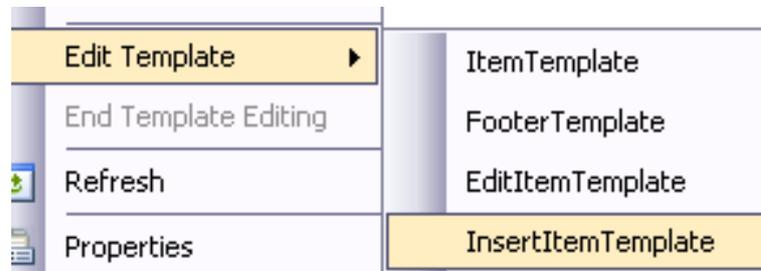
Langkah berikutnya adalah mengubah setting property *DefaultMode* dari FormView tersebut menjadi *Insert*. Hal ini dilakukan karena FormView nantinya hanya menjadi alat bantu dalam pengentrian kategori baru, sedangkan untuk proses edit menggunakan alat bantu GridView.



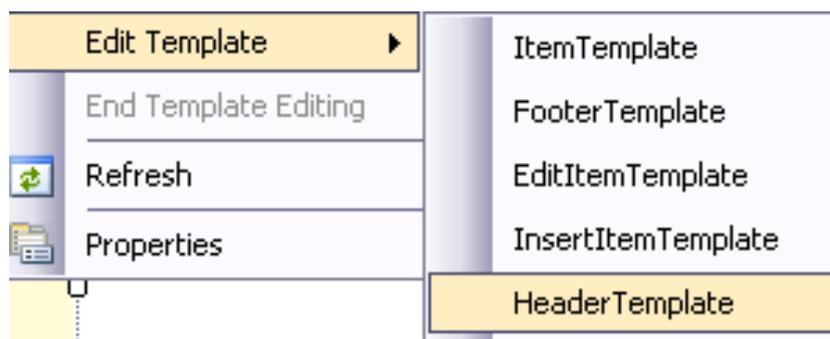


Dalam proses maintenance data kategori tidak terdapat proses penghapusan data kategori. Hal ini dilakukan untuk menghindari anomali data jika pada satu saat terjadi penghapusan data kategori, maka isi blog dalam kategori tersebut tidak memiliki *parent* kategori yang bersesuaian. Untuk “menghapus” sebuah kategori, dilakukan dengan mengedit field *StatusCategory* menjadi *false* dan nantinya akan difilter di dalam halaman utama.

Kemudian, klik kanan pada FormView dan pilih sub menu *Edit Template* → *InsertItemTemplate* untuk mengedit tampilan dalam FormView. Pada mode edit template tersebut, hapus Checkbox untuk field *StatusCategory*.



Setelah itu, klik kanan lagi pada FormView dan pilih sub menu *Edit Template* → *HeaderTemplate* dan beri judul pada FormView tersebut. Setelah selesai, klik pada Smart Tag untuk mengatur property *AutoFormat* sesuai dengan selera.

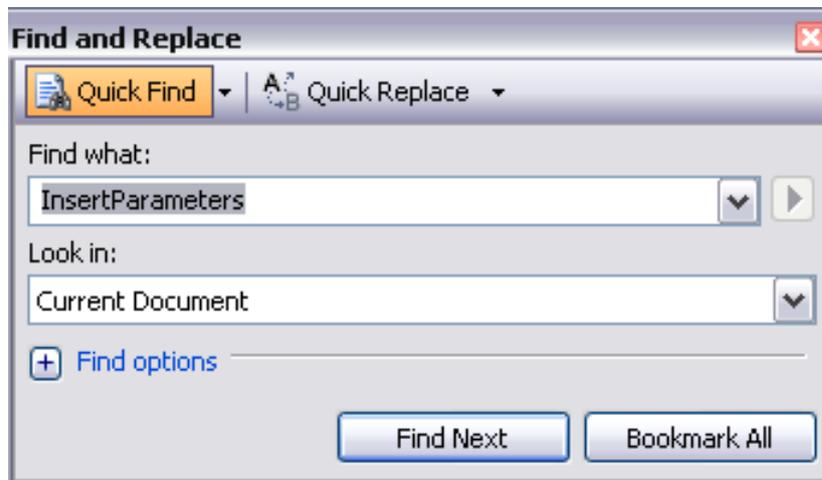


Berikutnya, berpindahlah ke mode Source dari halaman web admin, dan carilah tag *SelectParameter* (bisa dengan menggunakan fasilitas *Find* atau dengan menekan kombinasi tombol Ctrl + F), lalu edit tag dalam *InsertParameters* untuk *SqlDataSource1* menjadi seperti listing berikut ini :

```

<InsertParameters>
  <asp:Parameter Name="Category"
    Type="String">
  </asp:Parameter>
  <asp:Parameter Name="StatusCategory"
    Type="Boolean" DefaultValue="True">
  </asp:Parameter>
</InsertParameters>

```



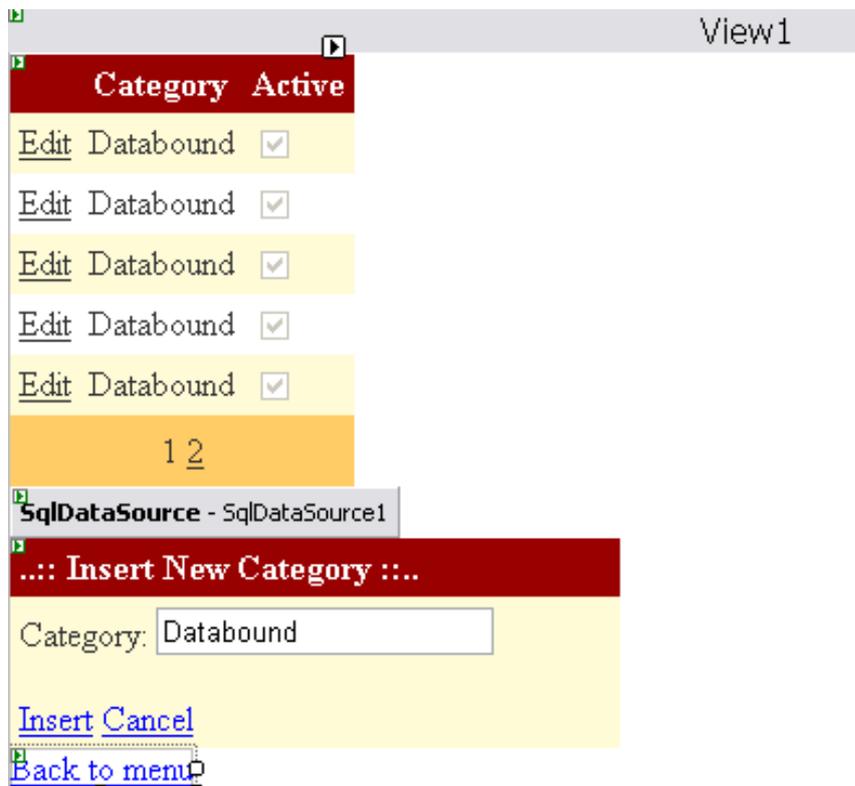
Penggantian parameter pada field *StatusCategory* dengan mengasumsikan bahwa pada saat pertama kali data kategori diinputkan, maka kategori tersebut akan langsung digunakan oleh pengguna. Sehingga status dari kategori tersebut langsung dianggap *true* atau aktif. Hal ini akan mengurangi interaksi pengguna pada saat proses maintenance data kategori, sehingga pengguna merasa lebih cepat dalam melakukan proses.

Langkah terakhir untuk View yang pertama adalah menambahkan sebuah Linkbutton di bawah dari FormView. Ganti property *text* dari Linkbutton menjadi *Back to menu*. Kemudian double klik pada Linkbutton tersebut untuk menyetikkan listing berikut :

```
MultiView1.ActiveViewIndex = 2
```



Linkbutton *Back to menu* berfungsi untuk mengembalikan keadaan halaman web ke kondisi awal halaman web admin. Yaitu menuju ke View yang ketiga atau yang terakhir. Pembuatan View yang terakhir akan dijelaskan pada bagian lanjutan dari studi kasus ini.



Setelah View yang pertama selesai dimodifikasi, maka langkah berikutnya adalah melakukan modifikasi untuk View yang kedua, yaitu View untuk maintenance isi dari blog itu sendiri.

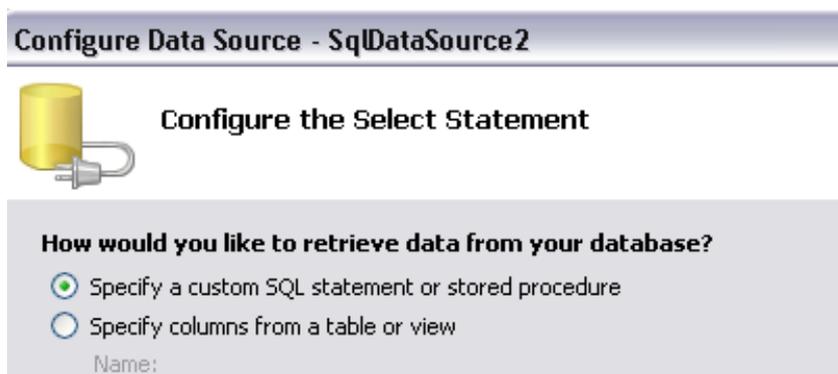
Maintenance Data Blog

Langkah pertama untuk modifikasi View yang kedua adalah melakukan drag tabel *IsiBlog* pada View yang kedua. Dari hasil operasi drag tersebut, maka akan terdapat sebuah komponen `SqlDataSource` yang kedua serta `GridView` baru di atasnya.

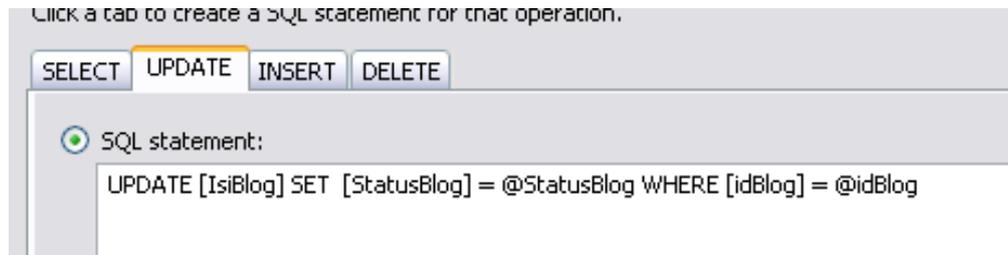
Pada komponen `SqlDataSource` yang kedua, klik pada Smart Tag dan pilih sub menu *Configure Data Source* untuk melakukan edit pada perintah *Update* yang terdapat dalam komponen tersebut. Edit perintah *Update* dilakukan untuk menyesuaikan dengan modifikasi `GridView` yang akan dilakukan pada langkah selanjutnya.



Pada kotak dialog yang tersedia, pilih opsi *Specify a custom SQL Statement* Untuk melakukan editing perintah *Update* secara manual. Opsi ini umumnya dilakukan untuk melakukan editing perintah *SQL non Select* secara lebih interaktif.



Kemudian pada kotak dialog berikutnya, klik pada tab untuk perintah *Update*, dan edit perintah tersebut seperti pada gambar berikut ini.



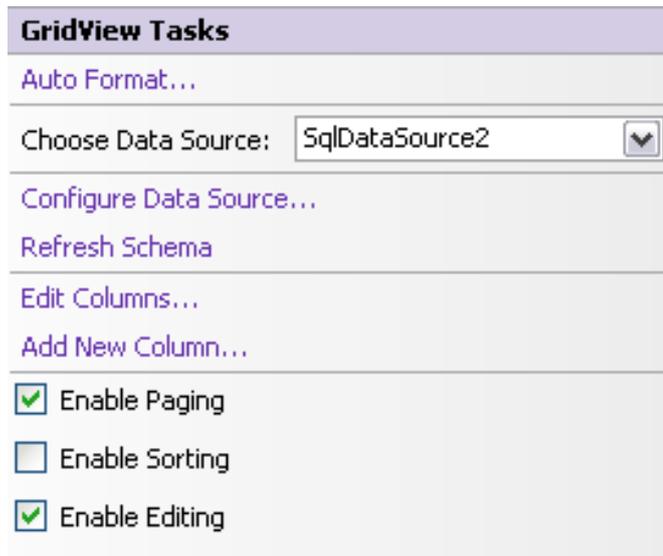
Setelah selesai, berpindahlah ke mode Source, dan cari tag `<SqlDataSource2>...</SqlDataSource2>`. Di dalam lingkup tag tersebut, cari sub tag `<UpdateParameters>..</UpdateParameters>` dan edit seperti pada listing berikut :

```
<UpdateParameters>
<asp:Parameter Name="StatusBlog"
    Type="Boolean"></asp:Parameter>
<asp:Parameter Name="idBlog"
    Type="Int32"></asp:Parameter>
</UpdateParameters>
```

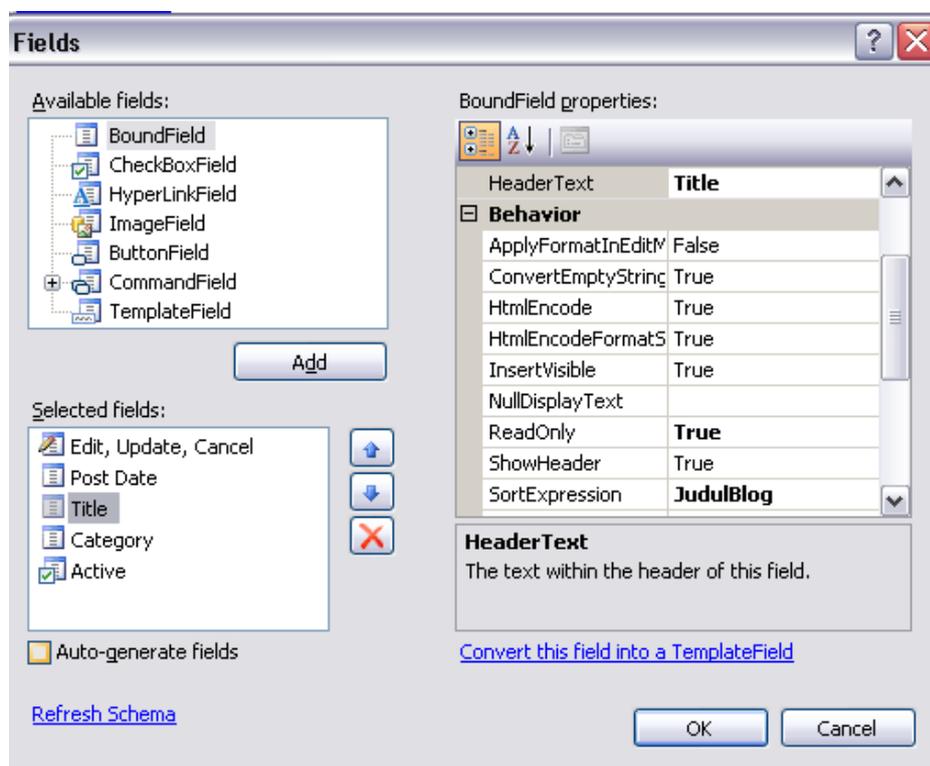


Editing perintah SQL juga bisa langsung dilakukan pada mode Source jika Anda sudah memiliki pemahaman yang lebih baik dalam proses editing tag. Jika tidak, maka lebih baik menggunakan bantuan kotak dialog interaktif yang telah disediakan oleh Visual Web Developer Express. Hal tersebut dikarenakan proses editing langsung di mode Source sangat rentan dan sangat beresiko untuk terjadinya sebuah kesalahan. Dan dalam sebuah halaman web yang kompleks, pencarian kesalahan di dalam mode Source seringkali menjadi sebuah proses yang menjengkelkan sekaligus melelahkan. Disarankan juga untuk tetap memperhatikan pasangan tag dalam mode Source, karena dari pasangan tag tersebut (`<...> ...</...>`) dapat dilacak lebih lanjut, proses modifikasi yang dilakukan oleh kotak dialog interaktif yang telah dilakukan sebelumnya.

Langkah berikutnya adalah melakukan klik pada Smart Tag yang terdapat dalam GridView, kemudian pilih opsi *Enable Paging* dan opsi *Enable Editing*, lalu klik pada sub menu *Edit Column*.



Selanjutnya, di kotak dialog hapus field *idBlog* dan *IsiBlog* dari daftar *Selected Fields*. Lalu lakukan pengeditan kolom dengan mengganti property *HeaderText* untuk kolom *TglBlog* menjadi *Post Date*, *JudulBlog* menjadi *Title* dan kolom atau field *StatusBlog* menjadi *Active*.



Selain itu, set juga property *ReadOnly* dari tiap field tersebut menjadi *True*, kecuali untuk field *StatusBlog* (di bagian *Selected Fields* sudah berubah menjadi *Active*).



Sama halnya dengan proses maintenance data kategori, tidak terdapat proses penghapusan pada proses maintenance isi blog. Karenanya, jika pengguna menginginkan isi blog tidak muncul lagi di dalam halaman utama, cukup dengan melakukan edit field *StatusBlog* menjadi *False* di dalam tabel *IsiBlog*.

Kemudian set property *Width* dari GridView menjadi *100%* agar GridView terlihat *stretch* atau melebar sesuai dengan lebar kolom pada tabel. Ganti juga property *PageSize* menjadi 5 agar GridView tidak terlihat terlalu scroll ke bawah jika isi blog sudah banyak.

Selanjutnya, GridView dapat diatur formatnya dengan melakukan klik pada Smart Tag dan memilih menu *AutoFormat*, lalu atur berdasarkan template sesuai selera.

	Post Date	Title	Category	Active
Edit	Databound	Databound	Databound	<input checked="" type="checkbox"/>
Edit	Databound	Databound	Databound	<input checked="" type="checkbox"/>
Edit	Databound	Databound	Databound	<input checked="" type="checkbox"/>
Edit	Databound	Databound	Databound	<input checked="" type="checkbox"/>
Edit	Databound	Databound	Databound	<input checked="" type="checkbox"/>

1 2

SqlDataSource - SqlDataSource2

Langkah berikutnya adalah menempatkan komponen FormView di bawah GridView. Sama juga dengan proses di maintenance data kategori, FormView hanya akan digunakan untuk melakukan entri data blog.

Pada FormView tersebut, nantinya akan ditempatkan sebuah komponen editor HTML yaitu FCKEditor. Komponen ini bisa didownload secara gratis (dan open source) dari www.FCKEditor.net. Pada saat proses download dilakukan, pilih komponen FCKEditor yang memang dikhususkan untuk ASP .NET (hal ini

dikarenakan komponen FCKEditor memiliki berbagai jenis, misal : FCKEditor yang dikhususkan untuk PHP).



Terdapat beragam jenis komponen editor HTML yang tersedia di internet dan siap untuk didownload, baik yang gratis maupun yang komersial. Selain FCKEditor, juga terdapat komponen lain seperti FreeTextBox, TinyMCE, RTE, WYSWYG open ataupun MCE.

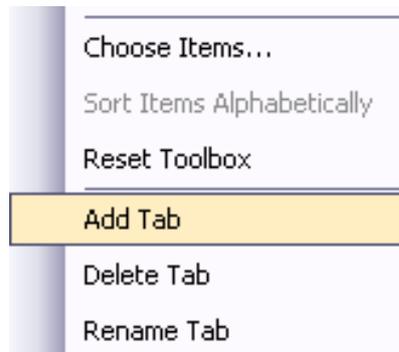
Dari berbagai jenis komponen tersebut, usahakan pilih komponen yang gratis dan memiliki jenis khusus untuk bahasa pemrograman yang kita gunakan. Karena dalam konteks buku ini digunakan ASP .NET, maka lebih baik jika memilih komponen yang memang memiliki kapabilitas khusus untuk digunakan dalam lingkup ASP .NET, seperti FCKEditor, FreeTextBox atau TinyMCE. Hal tersebut akan sangat memudahkan pada saat proses desain maupun proses pemrograman dilakukan.

Dalam studi kasus ini digunakan FCKEditor dengan alasan bahwa komponen ini gratis dan open source dan juga telah terbukti handal digunakan oleh portal berbasis ASP .NET lainnya yaitu DotNetNuke.

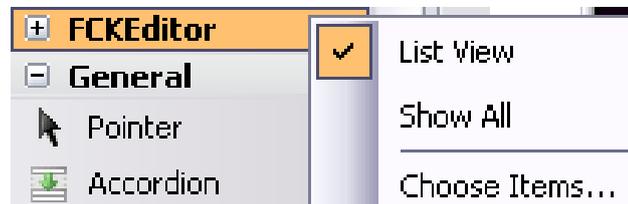
Dari hasil download komponen, maka copy file *FredCK.FCKeditorV2.dll* (umumnya terletak di dalam folder Bin di hasil download komponen) ke dalam folder *bin* di dalam solution yang kita miliki (bisa dengan cara drag & drop dari Windows Explorer, atau dengan melakukan klik kanan pada folder *Bin* dan memilih sub menu *Add Existing Item*).

Kemudian, tambahkan pula satu folder *FCKeditor* ke dalam solution (dengan menggunakan cara yang sama seperti menambahkan file dll). Di dalam folder ini nantinya akan terdapat script serta setting dari FCKEditor yang akan digunakan.

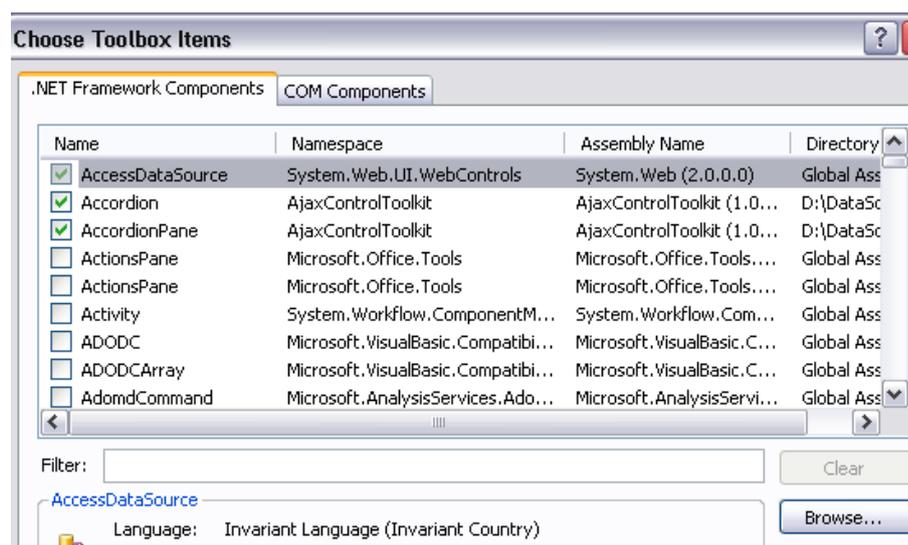
Langkah terakhir sebelum menggunakan FCKEditor adalah melakukan penambahan komponen ke dalam Toolbox yang kita miliki. Pada Toolbox, klik kanan di tab yang paling bawah (dan kosong), lalu pilih sub menu *Add Tab* dan beri nama tab tersebut *FCKEditor*.



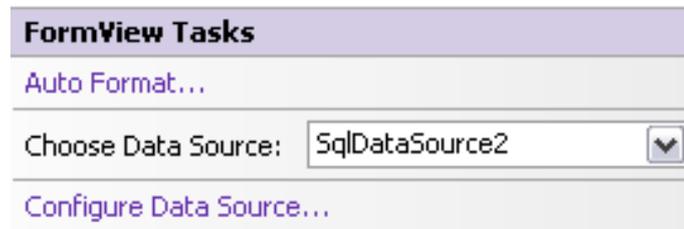
Selanjutnya, klik kanan di tab baru tersebut dan kini pilih sub menu *Choose Item* untuk menambahkan komponen baru di dalam tab tersebut.



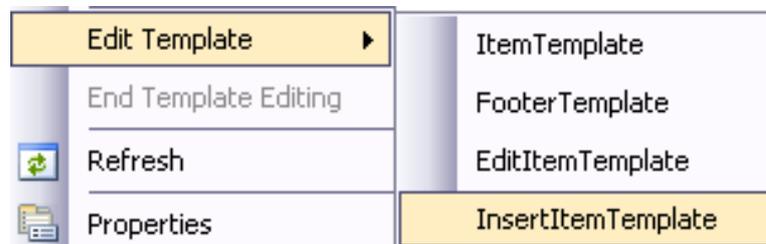
Pada kotak dialog yang tersedia, klik pada button *Browse* dan arahkan pada folder tempat *Bin* di solution lalu pilih file *FredCK.FCKeditorV2.dll*. Hasil dari penambahan ini akan menyebabkan Toolbox terisi komponen baru yaitu *FCKEditor*.



Setelah FormView ditempatkan, klik pada Smart Tag dan set property *DataSource* ke komponen *SqlDataSource* yang kedua. Lalu set property *DefaultMode* dari komponen FormView tersebut menjadi *Insert*.



Kemudian, sama dengan langkah pada maintenance data kategori, klik kanan pada FormView dan pilih sub menu *Edit Template* → *Insert Item Template*. Dan kini saatnya untuk melakukan editing dari template FormView.



Dalam template yang tersedia, edit template, dengan menghilangkan field *TglBlog*, *Category*, *IsiBlog* dan *StatusBlog*, serta menghilangkan keterangan untuk *JudulBlog*. Selanjutnya, drag komponen FCKEditor ke bawah bagian *JudulBlog*.



Lakukan penghapusan field dengan hati-hati, agar komponen yang lain tidak ikut terhapus.



Berikutnya, klik pada komponen FCKEditor dan set property *BasePath* seperti pada gambar berikut ini.

(Expressions)	
(ID)	FCKeditor1
BasePath	~/FCKeditor/



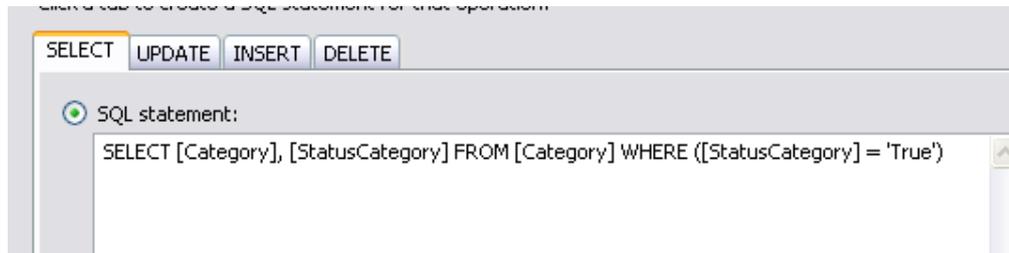
Setting property ini untuk memastikan bahwa komponen FCKEditor akan mengacu pada folder yang telah dicopykan sebelumnya.

Kemudian, di bagian atas dari FormView tersebut, tambahkan keterangan untuk field *Category*, dengan cara melakukan drag tabel *Category* ke dalam FormView. Setela itu, GridView yang terbentuk dari hasil drag tersebut bisa dihapus, sehingga menyisakan komponen *SqlDataSource* yang ketiga dalam halaman web admin.

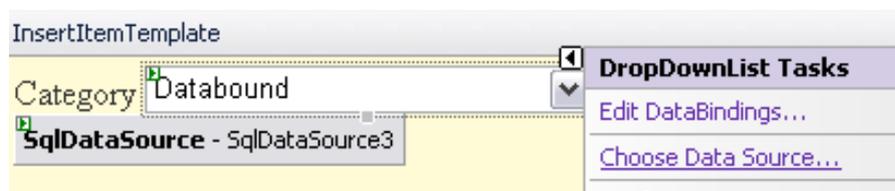


Penempatan *SqlDataSource* hasil drag tabel *Category* bertujuan untuk mengambil data kategori ke dalam FormView dalam bentuk *DropDownList*.

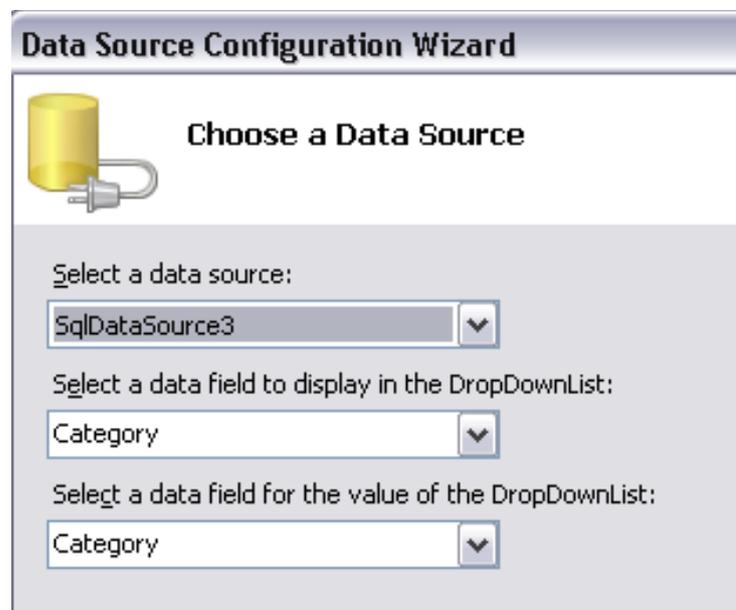
Kini, klik pada Smart Tag dari *SqlDataSource* dan pilih sub menu *Configure Data Source* untuk melakukan edit perintah SQL. Di dalam kotak dialog yang tersedia, klik opsi *Specify a custom SQL....* dan modifikasi perintah *Select* seperti pada rangkaian gambar berikut.



Kemudian, tempatkan sebuah DropDownList di atas SqlDataSource yang telah tersedia untuk menampung data kategori tersebut. Di dalam DropDownList yang ada, klik pada Smart Tag dan pilih sub menu *Choose Data Source* untuk melakukan setting koneksi data dalam DropDownList tersebut.



Di kotak dialog yang tersedia, set property *DataSource* ke komponen SqlDataSource3, dan set property dibawahnya untuk mengacu ke field *Category*.



Setelah proses editing template selesai dilakukan, double klik pada FormView, dan beralihlah ke method *FormView_ItemInserting* lalu ketikkan listing berikut ini :

```
With SqlDataSource2
    .InsertParameters("Category").DefaultValue = _
CType(FormView2.FindControl("DropDownList1"), _
DropDownList).SelectedValue
    .InsertParameters("IsiBlog").DefaultValue = _
CType(FormView2.FindControl("FCKeditor1"), _
FredCK.FCKeditorV2.FCKeditor).Value
End With
```

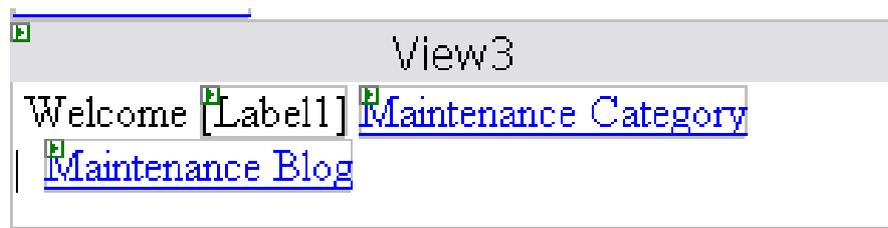


Setting parameter *Insert* di prosedur tersebut ditujukan untuk mengalihkan parameter asli yang sebelumnya telah dihapus.

Jika Anda tidak menempatkan FCKEditor, dan tetap menggunakan Textbox default sebagai isian dari FormView, maka parameter yang kedua yaitu FCKEditor bisa diabaikan.

Menu Halaman Web Admin

View yang terakhir atau View yang ketiga merupakan yang paling sederhana dibandingkan kedua View sebelumnya. Pada View ini hanya terdapat sebuah komponen Label dan dua buah komponen Linkbutton yang berfungsi sebagai navigator dalam halaman web admin.



Komponen Label dalam View berfungsi sebagai penanda session hasil login yang telah dilakukan. Hal ini umumnya dilakukan sebagai salam pembuka dari sebuah halaman web yang baru saja melalui proses login, sekaligus sebagai proses cek dalam halaman tersebut, apakah login telah sesuai dengan yang diharapkan.

Dobel klik pada Linkbutton *Maintenance Category* dan ketikkan listing berikut agar menuju ke dalam View yang pertama :

```
MultiView1.ActiveViewIndex = 0
```

Kemudian klik pada Linkbutton *Maintenance Blog* untuk mengetikkan listing berikut :

```
MultiView1.ActiveViewIndex = 1
```

Langkah terakhir dari View ketiga ini adalah melakukan double klik pada bagian halaman web yang kosong untuk menuju ke prosedur *Page_Load* dan mengetikkan listing berikut kedalamnya :

```
If Session("Login") = "" Then
    Response.Redirect("default.aspx")
Else
    CType(Master.FindControl("MultiView1"), _
        MultiView).ActiveViewIndex = 1
    Label1.Text = Session("Login")
End If
```



Pengecekan login dilakukan pada saat halaman web admin pertama kali dieksekusi, hal ini agar halaman web admin tidak dapat diakses secara langsung tanpa melalui proses login.

Sedangkan saat proses login berhasil dilakukan, maka MultiView yang terdapat di Master Page akan dialihkan ke View yang kedua untuk mengganti menu sebelumnya.

Halaman Default

Halaman web default, merupakan halaman web utama dalam studi kasus mini blog ini. Halaman web default, sudah muncul pertama kali pada saat solution dibuat (lihat lagi di langkah pertama pada studi kasus ini). Tetapi, halaman web utama tersebut belum terhubung ke Master Page yang telah dibuat sebelumnya.

Untuk menghubungkan sebuah halaman web ke dalam Master Page, maka perlu dilakukan modifikasi di dalam mode Source. Modifikasi pertama (setelah berada dalam mode Source) adalah menghapus seluruh bagian di dalam tag `<html> ... </html>`.

Penghapusan seluruh isi bagian dari tag HTML tersebut dilakukan karena, dalam koneksi antara halaman web biasa dengan Master Page, tag HTML akan ditangani oleh Master Page, bukan oleh halaman web.

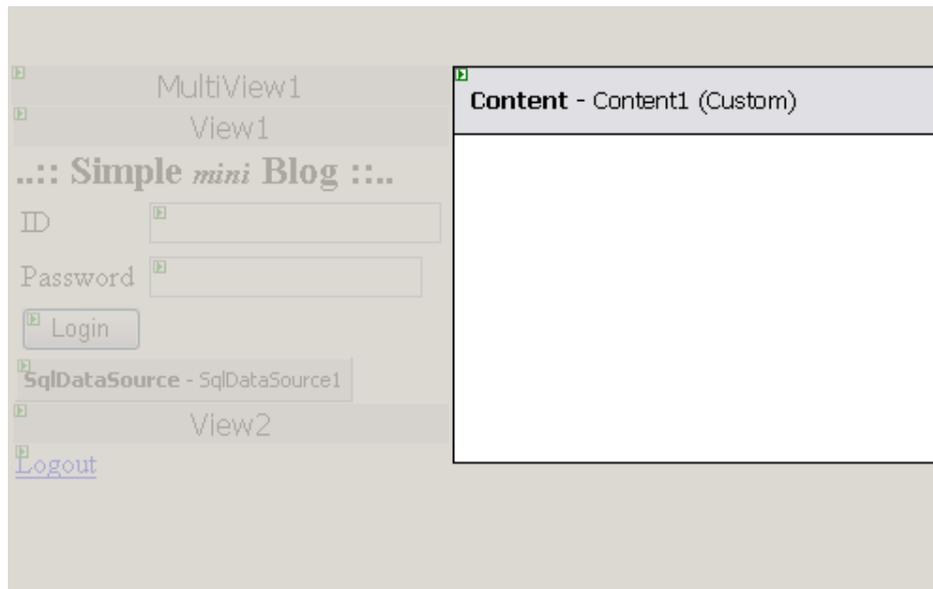
Setelah bagian tag HTML dihapus, modifikasilah bagian tag paling atas dalam mode Source di halaman web tersebut. Modifikasi yang dilakukan adalah menambahkan atribut *MasterPageFile* dan mengisinya dengan nama Master Page yang akan diacu (referensi). Modifikasi dalam halaman web tersebut akan terlihat pada listing berikut :

```
<%@ Page Language="VB" AutoEventWireup="false"
CodeFile="Default.aspx.vb" Inherits="Default"
MasterPageFile="~/MiniBlog.master" %>
```

Berikutnya, tempatkan sebuah komponen *Content* di dalam halaman web tersebut, sebagai penampung dari halaman web tersebut ke dalam Master Page. Penambahan tersebut dilakukan dengan cara menyetikkan listing berikut ini di bagian bawah dari mode Source :

```
<asp:Content ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
</asp:Content>
```

Jika modifikasi sudah dilakukan dengan benar, maka saat berpindah lagi ke mode Design, halaman web akan tampak seperti pada gambar ini.



Di dalam halaman web utama atau halaman default ini nantinya akan diletakkan dua buah komponen *Collapsible Panel* dari AJAX Control Toolkit sebagai navigasi utama yang menangani data blog terbaru serta data arsip dari blog berdasarkan kategori.

Di tiap isi panel dari *Collapsible Panel*, akan ditempatkan lagi komponen *Accordion* (juga dari AJAX Control Toolkit), sebagai ganti dari *GridView* untuk menampilkan detail dari isi blog.

Khusus untuk panel yang kedua, juga akan melibatkan komponen *DropDown Extender* (yang didalamnya terdiri dari sebuah *Textbox* dan *Listbox*) dari AJAX Control Toolkit sebagai ganti dari *DropDownList* biasa, dalam rangka pencarian arsip blog berdasarkan kategori yang ada.

Langkah pertama yang dilakukan dalam halaman default ini adalah melakukan drag komponen *ScriptManager* ke bagian paling atas dari *ContentPlaceHolder* yang tersedia. Seperti contoh sebelumnya, komponen ini wajib ditempatkan di sebuah halaman web yang didalamnya akan menerapkan teknik AJAX.

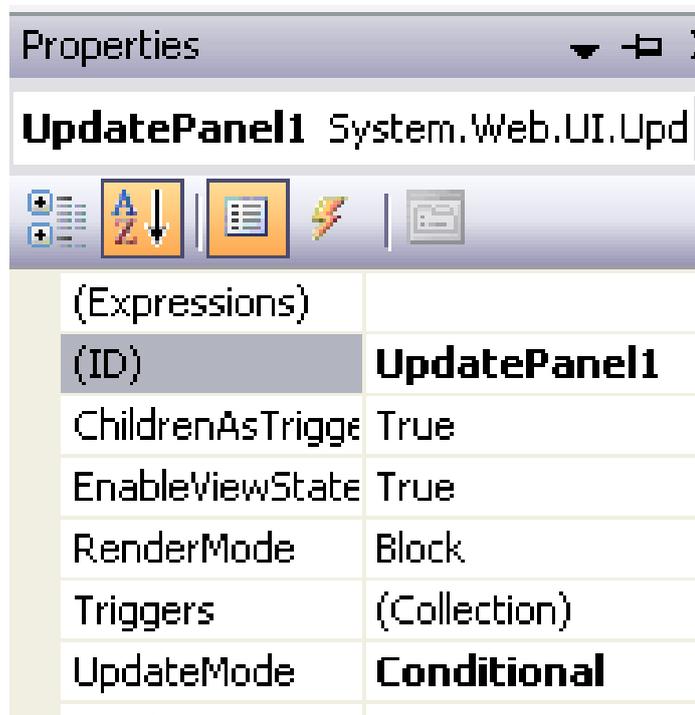
Kemudian, double klik di bagian kosong dari *ContentPlaceHolder* yang tersedia, dan pada prosedur *Page_Load* ketikkan listing berikut ini :

```
CType (Master.FindControl ("MultiView1"), _  
MultiView).ActiveViewIndex = 0
```

Listing tersebut berfungsi sebagai validasi saat pengguna beralih ke halaman utama, terutama jika pengguna baru saja beralih dari halaman web admin. Validasi yang dilakukan adalah mengganti posisi View dari MultiView yang terdapat dalam Master Page agar berpindah ke View yang pertama (posisi awal).

Bagian Newest Blog

Langkah pertama dalam bagian Newest Blog di halaman default adalah menempatkan komponen UpdatePanel dalam ContentPlaceHolder yang tersedia. Lalu set property *UpdateMode* dari UpdatePanel ini menjadi *Conditional* agar tidak selalu melakukan refresh, terutama pada saat bagian archive nanti selesai dimodifikasi.



Berikutnya, drag sebuah komponen Collapsible Panel Extender dari tab AJAX Control Toolkit. Dalam komponen ini nantinya akan diset lebih lanjut propertynya di dalam mode Source.



Tepat, di bawah komponen Collapsible Panel yang baru ditempatkan, drag sebuah komponen Panel. Lalu set property *Width* dari komponen Panel tersebut

menjadi 100% agar komponen Panel nantinya *stretch* (melebar ke samping sesuai dengan resolusi browser) pada saat dieksekusi.

Visible	True
Width	100%

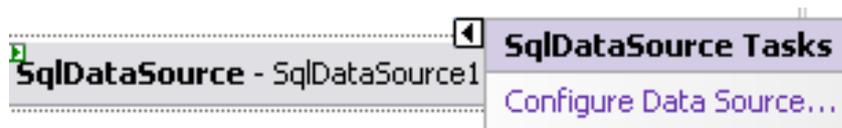
Sangat disarankan untuk melakukan setting property *Width* dalam satuan persen, dibandingkan satuan pixel. Hal ini dimaksudkan agar komponen yang akan *distretch* tidak terpengaruh oleh resolusi layar dari pengguna. Kecuali jika komponen yang dimaksud, memang akan diberikan sebuah ukuran lebar yang statis, maka property *Width* harus diset dengan menggunakan satuan pixel.



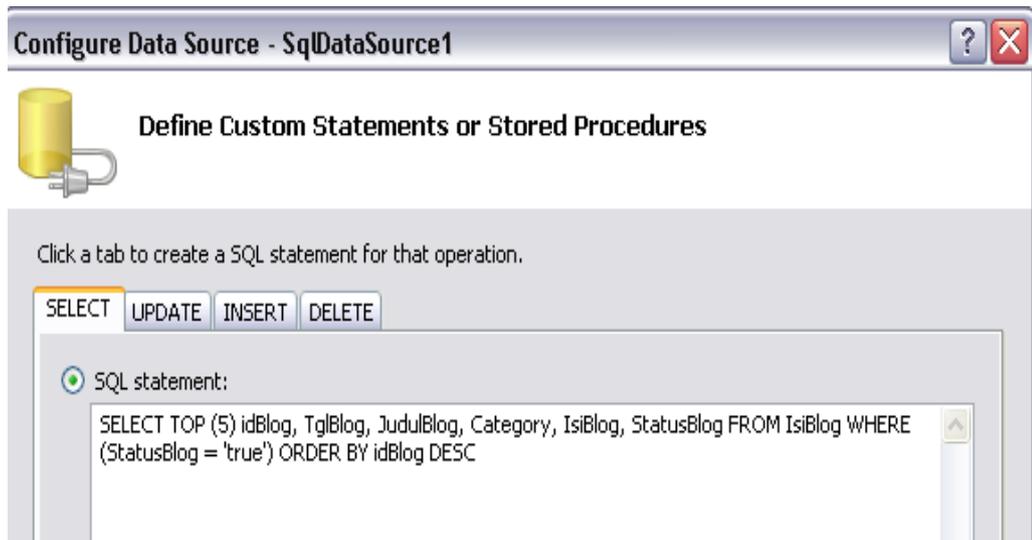
Selain itu, juga harus dipertimbangkan konsep dari desain halaman web itu sendiri, apakah akan menggunakan ukuran relatif (persentase) atau menggunakan ukuran absolut (pixel). Karenanya, jika dalam sebuah pembuatan situs, melibatkan dua orang atau lebih yang masing-masing bertugas sebagai programmer dan desainer, diperlukan komunikasi yang baik agar situs dapat terlihat sesuai rencana awal.

Kemudian, drag tabel *IsiBlog* ke dalam Panel tersebut, dan jangan lupa untuk menghapus GridView hasil dari proses drag dari Panel. Penghapusan ini dilakukan karena hasil koneksi database akan diarahkan ke dalam komponen *Accordion*.

Klik pada Smart Tag dari komponen *SqlDataSource* yang tersedia, dan pilih menu *Configure Data Source* untuk mengganti perintah SQL yang akan menampilkan lima data blog terbaru pada komponen *Accordion*.



Di dalam kotak dialog yang tersedia, pilih opsi *Specify a custom SQL.....*, lalu ketikkan perintah SQL berikut ini :



Parameter *Top 5* dalam perintah SQL tersebut berarti bahwa hanya akan ada lima record teratas yang akan ditampilkan dari hasil perintah Select tersebut. Lima record teratas tersebut adalah record yang memenuhi kriteria field *StatusBlog* -nya true dan telah diurutkan secara terbalik (descending) dari field *idBlog*.

Sehingga perintah Select yang telah dimodifikasi tersebut nantinya akan menampilkan lima isian blog terbaru dari tabel *IsiBlog*, dan dianggap dalam kategori *Newest Blog*.

Setelah kotak dialog selesai prosesnya, maka berpindahlah ke dalam mode Source untuk memodifikasi lebih lanjut bagian dari Update Panel yang pertama. Pada mode Source ini, bagian yang pertama kali adalah tag dari komponen Collapsible Panel. Modifikasi dari tag Collapsible Panel terlihat pada listing berikut :

```
<cc1:CollapsiblePanelExtender
  ID="CollapsiblePanelExtender1"
  runat="server" TargetControlID="Panel1"
  ExpandControlID="LinkButton1"
  CollapseControlID="LinkButton1">
</cc1:CollapsiblePanelExtender>
```



Sebuah Collapsible Panel membutuhkan minimal sebuah Panel sebagai target dan sebuah komponen pemicu (umumnya Linkbutton atau Button) dalam implementasinya.

Di dalam Collapsible Panel tersebut, yang akan menjadi target komponen yang dituju adalah Panel1, sedangkan pemicu dari proses ekspansi Panel adalah komponen Linkbutton1 (yang akan dibuat dan diterangkan selanjutnya). Karena property *ExpandControlID* dan *CollapseControlID* diset ke komponen yang sama, maka Linkbutton1 nantinya akan berfungsi ganda, sebagai pemicu untuk ekspansi Panel sekaligus sebagai pemicu untuk penutupan Panel itu sendiri.

Selanjutnya, masih dalam mode Source, ketikkan listing berikut tepat di bawah bagian Collapsible Panel :

```
<div style="background-color:Yellow;
border:solid 1px red;
font-weight:bold;
font-size:14pt;
cursor:pointer ">
  <asp:LinkButton ID="LinkButton1"
    runat="server">
    Newest Blog
  </asp:LinkButton>
</div>
```

Listing tag tersebut merupakan sebuah layer dalam HTML yang didalamnya terdapat sebuah komponen Linkbutton yang merupakan pemicu untuk proses ekspansi dan penutupan dari Panel di dalam Collapsible Panel.



Di dalam layer tersebut, diset stylenya dengan menggunakan property CSS border (warna dan style bisa diganti sesuai dengan selera). Dan juga diset property cursor menjadi pointer, sehingga pengguna dapat mengetahui, bahwa pada saat kursor mouse melintas di bagian layer tersebut, berarti bagian layer tersebut merupakan bagian *clickable* (dapat diklik).

Proses pembuatan layer tersebut juga dapat menggunakan file CSS (Style Sheet) yang terpisah dan dibuat class CSS tersendiri didalamnya.

Di dalam Collapsible Panel tersebut, yang akan menjadi target komponen yang dituju adalah Panel1, sedangkan pemicu dari proses ekspansi Panel adalah komponen Linkbutton1 (yang akan dibuat dan diterangkan selanjutnya). Karena property *ExpandControlID* dan *CollapseControlID* diset ke komponen yang sama, maka Linkbutton1 nantinya akan berfungsi ganda, sebagai pemicu untuk ekspansi Panel sekaligus sebagai pemicu untuk penutupan Panel itu sendiri.

Berikutnya, adalah pembuatan bagian yang didalamnya menggunakan komponen Accordion. Seperti telah dicontohkan pada bagian sebelumnya mengenai penggunaan Accordion dengan melakukan koneksi ke sebuah database, maka penggunaan Accordion dalam konteks ini akan melibatkan bagian tag *HeaderTemplate* dan *ContentTemplate* untuk menempatkan field-field yang akan ditampilkan di bagian *Newest Blog*.

Listing lengkap berikut ini untuk penggunaan Accordion diketikkan tepat di bawah bagian layer yang telah diketikkan sebelumnya.

```
<ccl:Accordion ID="Accordion1" runat="server"
DataSourceID="SqlDataSource1">
<HeaderTemplate>
<h2><b>
  <div
    style="border-bottom:dashed 1px black;
    cursor:pointer; background-color:Aqua">
    <#eval("JudulBlog") %>
  </div>
</b></h2>
</HeaderTemplate>
<ContentTemplate>
<b>
Posted at :
  <#format(Eval("TglBlog"), "dd-MM-yyyy") %>
  in <#eval("Category") %>
```

```

</b><br />
<%#Eval("IsiBlog")%>
</ContentTemplate>
</cc1:Accordion>

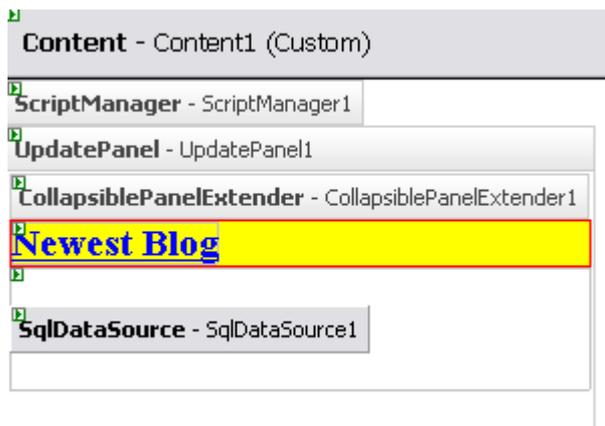
```



Di dalam koneksi database ke dalam komponen Accordion, modifikasi utama terletak pada property *DataSourceID* yang diisi dengan komponen *SqlDataSource* hasil dari drag and drop tabel yang akan dikoneksikan dari Database Explorer.

Sedangkan bagian tag Header Template diisi dengan field yang akan dijadikan judul. Dalam kasus ini, field yang akan dijadikan judul adalah field *JudulBlog*. Di bagian tag ini juga ditempatkan sebuah layer baru yang berfungsi sebagai penanda header. Style dari layer tersebut bisa diganti sesuai dengan selera, dengan asumsi property *cursor* tetap dinyatakan dengan nilai *pointer* agar pada saat pengguna mengarahkan mouse di atas judul dapat langsung mengetahui bahwa bagian tersebut dapat diklik untuk mengetahui isi dari blog.

Untuk bagian Content Template, diisi dengan isi dari blog itu sendiri, yaitu field *TglBlog* yang digabungkan dengan field *Category* dan kemudian dibawahnya (dipisah dengan tag HTML *
* untuk ganti baris) diisi dengan field *IsiBlog*.



Bagian Archieve Blog

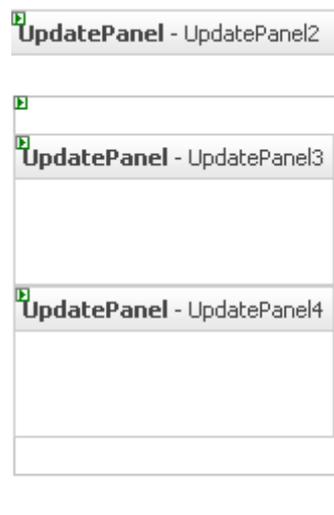
Bagian Archieve Blog merupakan bagian terakhir dari pembuatan situs Mini Blog. Bagian ini masih merupakan satu kesatuan halaman web default, yang salah satu bagiannya telah selesai dibahas yaitu bagian *Newest Blog*.

Bagian archive blog mirip dengan bagian *Newest Blog* yaitu memanfaatkan komponen Collapsible Panel dari AJAX Control Toolkit, tetapi didalamnya terdiri dari sebuah komponen DropDown Extender dan Grid View yang juga memanfaatkan Collapsible Panel didalamnya.



Umumnya penempatan tiap komponen AJAX Control Toolkit membutuhkan sebuah Update Panel sebagai “wadah” dari komponen tersebut untuk melakukan proses partial postback.

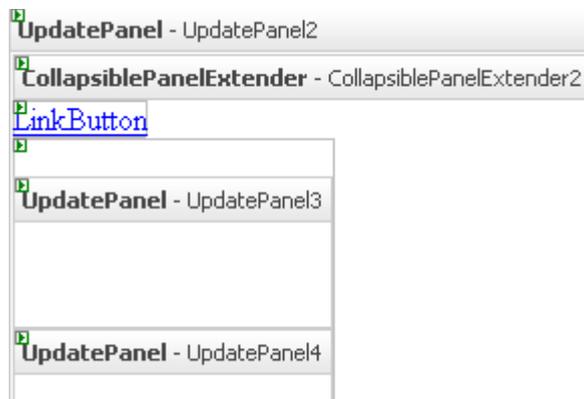
Untuk pembuatan bagian ini, langkah pertama yang harus dilakukan adalah menempatkan Update Panel tambahan yang pertama di bagian bawah dari *Newest Blog*. Di dalam Update Panel tersebut, letakkan sebuah Panel yang didalamnya ditempatkan lagi dua buah Update Panel tambahan, yang masing-masing akan ditempati Drop Down Extender dan Grid View.





Perhatikan penempatan UpdatePanel3 dan UpdatePanel4 yang keduanya berada dalam sebuah Panel baru. Panel ini nantinya akan menjadi pemicu bagi komponen Collapsible Panel berikutnya.

Selanjutnya, tempatkan satu komponen Collapsible Panel di dalam UpdatePanel2 tepat di atas bagian panel yang sudah ditempatkan sebelumnya. Dan drag sebuah Linkbutton di bawah Collapsible Panel tersebut.

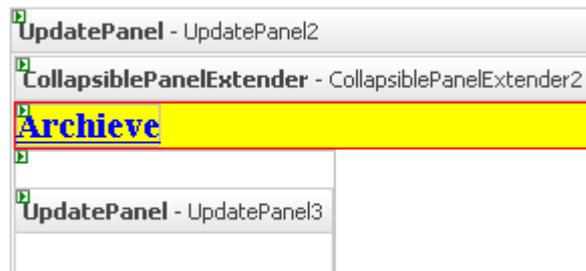


Lalu, berpindahlah ke mode Source dan edit bagian dari Collapsibel Panel dan Linkbutton menjadi seperti pada listing berikut ini :

```
<cc1:CollapsiblePanelExtender
  ID="CollapsiblePanelExtender2"
  runat="server"
  TargetControlID="Panel2"
  ExpandControlID="LinkButton2"
  CollapseControlID="LinkButton2"
  Collapsed="true" SuppressPostBack="true">
</cc1:CollapsiblePanelExtender>
<div style="background-color:Yellow;
  border:solid 1px red; font-weight:bold;
  font-size:14pt; cursor:pointer ">
<asp:LinkButton ID="LinkButton2"
  runat="server" >
  Archieve
</asp:LinkButton>
</div>
```



Modifikasi dari komponen Collapsible Panel untuk mengarahkan Panel target ke Panel2 dan pemicu ke arah Linkbutton yang telah dimodifikasi stylenya dengan menggunakan CSS.



Langkah selanjutnya adalah membuat sebuah tabel di dalam UpdatePanel3 yang terdiri dari dua baris dan dua kolom. Untuk pembuatan tabel tersebut, letakkan cursor di dalam UpdatePanel3 dan pilih menu *Layout* → *Insert Table*.



Kemudian atur jumlah *Rows* dan *Columns* masing-masing menjadi dua.

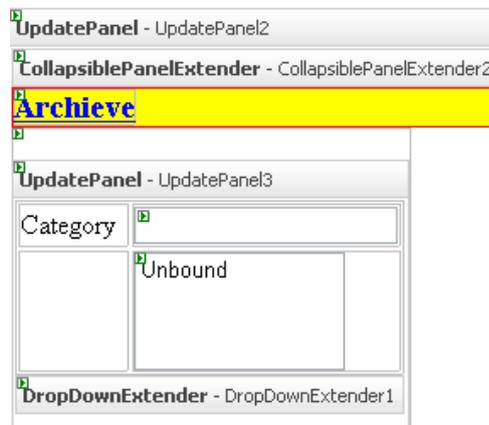


Berikutnya, tempatkan sebuah Textbox di baris pertama kolom kedua. Dan sebuah komponen Panel di baris kedua kolom pertama. Lalu, di dalam Panel tersebut

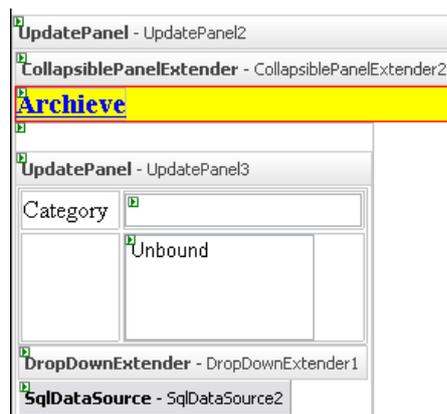
ditempatkan sebuah Listbox. Textbox dan Listbox ini nantinya akan menjadi bagian dari Drop Down Extender yang ditempatkan tepat dibawah bagian tabel tersebut.



Lihat lagi contoh penggunaan Drop Down Extender di sub bab sebelumnya. Sebuah Drop Down Extender membutuhkan sebuah Textbox dan komponen berbentuk list dalam implementasinya.



Lalu, drag tabel *Category* di bawah Drop Down Extender. Dari hasil operasi drag tersebut, hapus Grid View yang terbentuk, sehingga hanya tersisa komponen SqlDataSource yang akan digunakan sebagai data source bagi Listbox yang ada.



Berikutnya adalah melakukan setting terhadap Listbox yang sudah diletakkan. Klik pada Smart Tag di Listbox tersebut dan pilih opsi *Enable Auto Postback* didalamnya. Setelah selesai, klik pada sub menu *Choose Data Source* untuk mengarahkan data ke dalam Listbox tersebut.



Data Source Configuration Wizard



Choose a Data Source

Select a data source:

Select a data field to display in the ListBox:

Select a data field for the value of the ListBox:

Kemudian, berpindahlah ke mode Source lagi dan modifikasi listing dari Drop Down Extender tersebut menjadi seperti pada listing berikut ini :

```
<cc1:DropDownExtender
    ID="DropDownExtender1" runat="server"
    TargetControlID="TextBox1"
    DropDownListID ="Panel5">
</cc1:DropDownExtender>
```



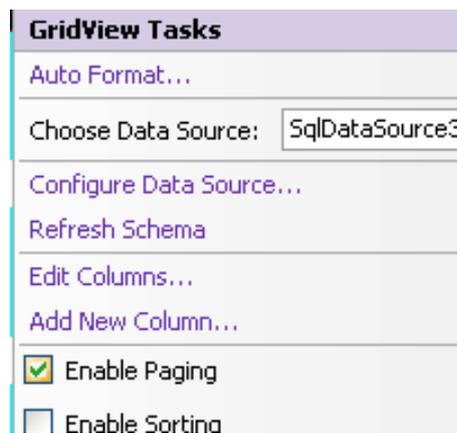
Modifikasi utama dari bagian ini adalah pada di property *TargetControlId* dan property *DropDownControlID* yang masing-masing diarahkan ke dalam Textbox dan Panel.

Asumsi dari Panel tersebut adalah panel yang terdapat tabel baris kedua kolom pertama memiliki property id *Panel5*. Jika ternyata Panel tersebut memiliki id yang berbeda, maka isi property dari *DropDownControlID* juga ikut menyesuaikan.

Kemudian drag tabel *IsiBlog* ke dalam *UpdatePanel4*, dan dari hasil drag tersebut, klik pada Smart Tag di Gridview yang terbentuk, lalu pilih opsi *Enable Paging* untuk Gridview tersebut. Lalu, set property *PageSize* dari Gridview tersebut menjadi 5 dan property *Width* menjadi 100%.

PagerStyle	
PageSize	5

Visible	True
Width	100%



Berikutnya, double klik pada Listbox (di dalam *UpdatePanel3*), dan ketikkan listing berikut ini pada prosedur *ListBox1_SelectedIndexChanged*:

```
TextBox1.Text = ListBox1.SelectedValue  
SqlDataSource3.DataBind()
```



Listing di dalam prosedur *ListBox1_SelectedIndexChanged* merupakan listing yang akan berfungsi sebagai pemicu aksi refresh di dalam Gridview saat Drop Down Extender dieksekusi. Didalamnya memerintahkan agar isi dari Listbox yang diklik oleh pengguna diumpankan ke Textbox sekaligus melakukan refresh terhadap komponen SqlDataSource3 yang nantinya juga akan melakukan refresh ke komponen Gridview secara otomatis.

Selanjutnya, berpindahlah ke mode Source dan cari tag yang mewakili SqlDataSource3, untuk kemudian dimodifikasi seperti pada listing berikut ini :

```
<asp:SqlDataSource
  ID="SqlDataSource3" runat="server"
  ConnectionString=
"<%$ ConnectionStrings:MiniBlogConnectionString1 %>"
  ProviderName=
"<%$ ConnectionStrings:MiniBlogConnectionString1.ProviderName %>"
  SelectCommand=
"SELECT * FROM [IsiBlog] WHERE (([Category] = @Category) AND
([StatusBlog] = @StatusBlog)) ORDER BY [idBlog] DESC">
  <SelectParameters>
    <asp:ControlParameter
      ControlID="TextBox1"
      Name="Category" PropertyName="Text"
      Type="String" />
    <asp:Parameter DefaultValue="True"
      Name="StatusBlog" Type="Boolean" />
  </SelectParameters>
</asp:SqlDataSource>
```



Modifikasi utama dari bagian tag SqlDataSource3 terletak pada perubahan perintah Select. Karena komponen SqlDatassource3 terletak di dalam sebuah Update Panel, maka modifikasi secara visual (dengan menggunakan wizard atau kotak dialog) akan lebih sulit dilakukan, sebab komponen yang berada dalam sebuah Update Panel akan menjadi

“terisolasi”. Karenanya, untuk melakukan modifikasi terhadap komponen SqlDataSource3 tersebut, dilakukan secara manual dari mode Source. Modifikasi lainnya terletak pada pembuatan parameter baru untuk field *Category* yang diisi dari Textbox (hasil refresh dari komponen Drop Down extender, dan field *StatusBlog* yang secara otomatis diisi dengan nilai *True*. Pengisian parameter di field *StatusBlog* dimaksudkan agar hanya isi blog yang diberi status *True* yang akan ditampilkan di bagian ini.

Langkah terakhir (masih tetap dalam mode Source) adalah melakukan modifikasi pada bagian tag GridView menjadi listing berikut ini :

```
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
    AutoGenerateColumns="False"
    BorderStyle="None" BorderWidth="0px"
    DataKeyNames="idBlog"
    DataSourceID="SqlDataSource3"
    EmptyDataText="There are no data records to display."
    ShowFooter="True" ShowHeader="False"
    Width="100%" PageSize="5">
    <Columns>
        <asp:TemplateField
            HeaderText="JudulBlog"
            SortExpression="JudulBlog">
            <ItemTemplate>
                <cc1:CollapsiblePanelExtender
                    ID="CollapsiblePanelExtender3"
                    runat="server"
                    ExpandControlID="Panel3"
                    CollapseControlID="Panel3"
                    TargetControlID="Panel4"
                    Collapsed="true"
                    SuppressPostBack="true"
                    ExpandedText="Hide Detail"
                    CollapsedText="Show Detail"
                    TextLabelID="Label1" >
                </cc1:CollapsiblePanelExtender>
                <asp:Panel
                    ID="Panel3" runat="server" >
                    <div style="background-color:aqua;
                    border-bottom :dashed 1px red;
                    font-weight:bold; font-size:12pt;
                    cursor:pointer; color:White ">
                        <b>Posted at :
```

```

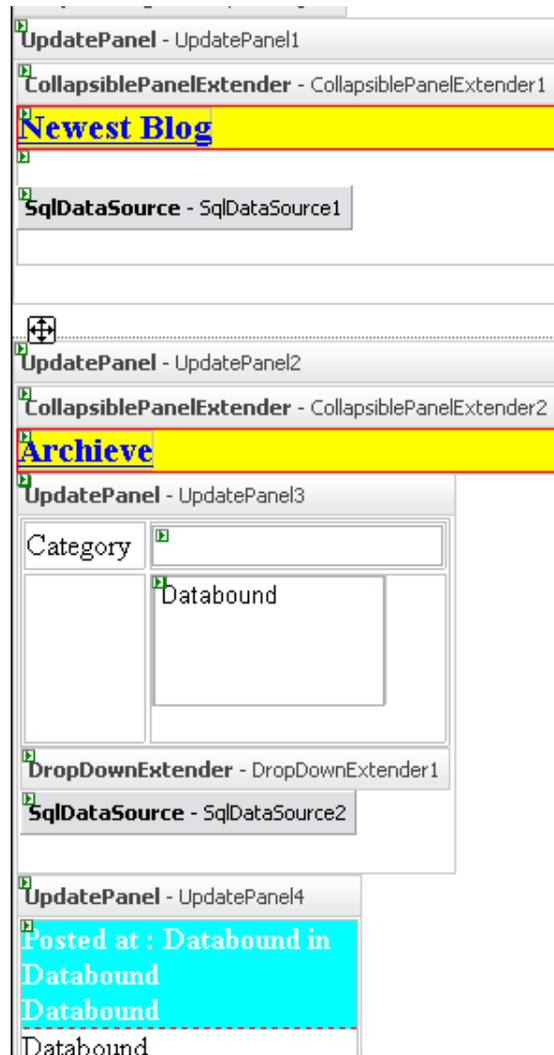
<#format(Eval("TglBlog"), "dd-MM-yyyy") %>
    in <#eval("Category") %>
    </b><br />
        <#eval("JudulBlog") %>
    </div>
</asp:Panel>
<asp:Panel
    ID="Panel4" runat="server"
    Height="0" Width="100%" >
    <#Eval("IsiBlog") %>
</asp:Panel>
</ItemTemplate>
</asp:TemplateField>
</Columns>
</asp:GridView>

```



Modifikasi terakhir dari bagian Archive Blog dilakukan di dalam GridView. Modifikasi ini merupakan modifikasi yang paling panjang di dalam mode Source, sebab di dalam modifikasi ini, selain mengubah jenis kolom dalam GridView menjadi mode template (sehingga dapat ditempati oleh komponen lain), juga menyisipkan sebuah komponen Collapsible Panel serta sebuah Panel di dalam GridView tersebut. Penambahan kedua komponen tersebut, ditujukan agar terjadi efek partial postback saat pengguna melakukan klik pada judul sebuah blog, dan akan langsung menuju ke isi blog itu sendiri secara otomatis.

Pada saat seluruh modifikasi selesai dilakukan, maka dalam mode Design akan terlihat seperti pada gambar berikut ini :



Dan jika situs dijalankan maka contoh hasil terlihat seperti pada gambar ini :

...: Simple *mini*
Blog ...

ID

Password

Login

Newest Blog

Test again

Posted at : 08-04-2008 in Campus Life
Last test ?

Once more

Finally

Test again

ASP .NET 3.x

Archieve

Category

Campus Life

Posted at : 08-04-2008 in Campus Life
Test again

Posted at : 08-04-2008 in Campus Life
Once more

Posted at : 08-04-2008 in Campus Life

Appendix

Appendix : Komponen Dasar ASP .NET

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 *Learning By Sample: ASP .NET 3.5* dari penulis yang sama.

Komponen-komponen tersebut antara lain :

1. 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 

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 passwod) atau bentuk *Multiline* (mampu menampung teks yang sangat banyak isinya).

3. 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 fiturnya. 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 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 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.